A Design Framework for Product Families Implemented with Additive Manufactured Variable Platforms Yao Xiling 2017

NANYANG TECHNOLOGICAL UNIVERSITY

# A Design Framework for Product Families Implemented with Additive Manufactured Variable Platforms


**Yao Xiling**

School of Mechanical and Aerospace Engineering


2017

# Abstract

Challenges exist in developing new product family design methodologies that can utilize the additive manufacturing (AM) advantages to improve the product family competitiveness in a cost efficient way. In this research, the author attempts to compensate the performance compromise in conventional product families by proposing the novel concept of an additive manufactured variable platform (VP); and the major objective of this thesis is to develop a design framework that helps designers implement additive manufactured VP modules within product families. During the development of the design framework, the following methodologies have been proposed: 1) a representative mathematical model to formulated the VP-based product family configuration; 2) a hybrid machine learning approach to select appropriate AM design features for addtive manufactured VP modules; 3) a fuzzy time-driven activity-based costing method to predict the AM production cost variation as a function of VP design adjustments, 4) a novel evaluation metric for VP-based product family designs, and 5) a multiobjective design optimization technique to identify the modular configuration and engineering design parameters of a product family. A case study of an R/C racing car family design is used throughout this thesis to demonstrate the usefulness of the proposed methodology. The implementation of additive manufactured VP modules shows a significant improvement in a product family's performance at the price of a slight decrease in design commonality. As the major contribution of this research, the proposed design framework provides designers and enterprises with a guideline in developing high-performance and cost-efficient product families by utilizing AM-enabled design flexibility. The two previously separate

research domains, i.e. AM and product family design, are integrated using the proposed framework and the constituent methodologies.

# Table of Contents

## List of Figures

## List of Tables

## List of Acronyms

| | |
|---|---|
| 3D | Three Dimensional |
| AM | Additive Manufacturing |
| ANFIS | Adaptive Neuro-Fuzzy Inference System |
| CCI | Composite Commonality Index |
| CP | Consistent Platform |
| DFAM | Design for Additive Manufacturing |
| DoV | Degree of Variation |
| FDM | Fused Deposition Modeling |
| FTDABC | Fuzzy Time-Driven Activity-Based Costing |
| GA | Genetic Algorithm |
| MS | Market Share |
| PcP | Process Setting Platform |
| PcV | Process Setting Variant |
| SLM | Selective Laser Melting |
| VP | Variable Platform |

# Acknowledgements

I would like to thank my supervisor, Professor Moon Seung Ki, for his guidance and support throughout my Ph.D. study. It is from him that I first acquired the concept and knowledge in product platform and product family design. I would like to thank my co-supervisor, Dr. Bi Guijun from Singapore Institute of Manufacturing Technology (SIMTech), who provided me with opportunities in learning and practicing additive manufacturing technologies. I am also grateful to Professor Chua Chee Kai, the Former Chair of School of Mechanical & Aerospace Engineering, who guided me in my undergraduate research and eventually brought me into the Ph.D. program.

I would like to thank my colleagues at Singapore Centre for 3D Printing, SIMTech, and Design Sciences Lab, for their help and friendship. I would also like to thank my family for their support throughout my study.

Finally, I would like to acknowledge the financial support from SERC, A*STAR Industrial Additive Manufacturing Programme, SIMTech-NTU Joint Lab, and an AcRF Tier 1 grant (RG94/13) from Ministry of Education, Singapore.

# Chapter 1 Introduction

## 1.1 Background

Additive manufacturing (AM) is a collection of manufacturing processes that fabricate parts using layer-by-layer fusion of materials. Compared to conventional subtractive and formative manufacturing processes, AM processes have the unique capabilities to create parts with complex geometries, multi-functionalities, and, in some cases, superior material properties. Other than prototypes, functional components can now be produced using various types of AM processes, which have found applications in different areas such as aerospace, automotive, and biomedical industries [1]. AM processes may be different in many aspects such as the stock material, energy source, material bonding mechanism, material delivery method, and build environment etc. However, despite of the above mentioned varieties, all AM processes follow the same general processing chain that consists of the Computer-Aided Design (CAD) modeling, printable file format generation, model slicing, AM tool path generation, machine setup, layer-by-layer material fusion, and post-process.

With the advantage in fabricating products with superior performances that are difficult to be achieved by conventional manufacturing processes, AM technologies have greatly increased design freedoms in product development [2]. As summarized by Seepersad [3], design freedoms provided by AM have brought challenges in multiple design for additive manufacturing (DFAM) research areas, including complex structures optimization, DFAM knowledge management, empirical database construction, and Computer-Aided Engineering (CAE) software customization for AM.

The above mentioned topics have been researched extensively, and a detailed review on DFAM studies is presented in Chapter 2.

Platform-based product family design is a strategy to provide product variety for satisfying diversified customer needs in multiple niche markets, while at the same time limiting the cost by implementing commonality in component designs and manufacturing processes [4]. A product family consists of multiple product variants covering different market segments, and a product platform is a common part shared by more than one product variant in the family [5]. Two generic approaches have been applied in product family design, i.e. the *top-down* approach and *bottom-up* approach. In general, product families can categorized into two major types: i.e. the *module-based* product family and *scale-based* product family [6]. In the real industry, platform strategies have been applied by many enterprises, from aircraft manufacturers to household appliance start-up companies [7-9], to improve the competitiveness of their product families.

In conventional product families, the same platform module is shared among all product variants in the product family to save the development cost and shorten the lead time [10, 11]. However, the consistently shared platform usually lacks design freedom due to the fixed manufacturing process configuration such as the prefabricated mold for casted components. Since AM provides significantly higher design flexibility than traditional manufacturing techniques at similar or even lower costs, implementing an additive manufactured platform module can be a reasonable design solution to improve the competitiveness of a product family. Therefore, in this research, the author aims to integrate the advantages of AM with product family design strategies and propose a systematic design framework to support designers and manufacturers in various stages of product family development.

## 1.2 Research Motivation

The emerging AM technologies provide new opportunities in product design by introducing new performance-enhancing design features and enabling high design flexibility/customizability. New design methodologies and frameworks need to be developed as guidance for enterprises to apply AM technologies in their product design or re-design processes. Optimizing additive manufactured products with both performance and cost considerations remains a challenge that needs to be tackled.

The strategy of platform-based product family design has been practiced in various industries to achieve cost efficiency in mass-customized product development and manufacturing. However, product families usually suffer from the performance compromise in individual product variants as the trade-off of design commonality and platform sharing. Therefore, in this research, the author attempts to implement additive manufactured platform modules in a product family, aiming to use the AM-enabled design flexibility to resolve or reduce the performance compromise in conventional product family designs. Several questions are raised when AM techniques are applied in the product family development process:

1) How the attributes of an additive manufactured platform be different from the attributes of a conventional consistent platform used in previous literature, and how can we model the product family configuration with the additive manufactured platform?

2) Design space has been enlarged by AM with an increasing number of new AM design features being developed and made available to designers. How can we improve the efficiency of exploring the AM-enabled design space and make the AM design feature selection easy even for novice designers? Is it possible to develop an

automated AM design feature selection algorithm based on a collection of coded databases?

3) How will the AM production cost be changed when AM process settings and platform designs are adjusted? The solution will be useful when the cost of a product family needs to be managed at the design stage.

4) How can we develop metrics to evaluate the design of a product family implemented with additive manufacture platforms?

5) How can we optimize the modular configuration and engineering design parameters of a product family simultaneously?

Based on the motivations discussed above, the author develops a design framework to support product family design where additive manufactured variable platform modules are implemented. The design framework consists of product family configuration modeling, AM design feature selection, AM production costing, product family design evaluation, and multiobjective design optimization. The proposed design framework aims to guide designers to make decisions in different stages during the product family development.

## 1.3 Research Objectives

In the present work, the principle objective is to develop a design framework for product families implemented with additive manufactured platform modules. To achieve the above principal objective, sub-objectives are proposed and described as below.

**Sub-objective 1: Propose the overall structure of the design framework**

Steps and sequences in the product family design framework are to be proposed in order to provide designers with an overview of methodologies in different design stages.

**Sub-objective 2: Define platform concepts and the product family configuration**

When AM is applied in product family design and production, new concepts are to be proposed to describe the characteristics of additive manufactured platform modules. Additive manufactured platforms allow higher design freedoms and different cost structures compared with conventional platforms. The product family configuration containing different types of modules and design parameters is to be defined and modeled mathematically.

**Sub-objective 3: Explore AM-enabled design space for platform modules**

The author aims to develop a computational approach for design knowledge management, which supports automatic and intelligent AM design feature selections at the conceptual design stage. Furthermore, platform design constraints are to be identified by correlating them to AM process settings for managing production costs.

**Sub-objective 4: Evaluate product family designs**

Product family design evaluation metrics are to be formulated to measure the performance and commonality of a product family. The evaluation metrics can also be used as objective functions in the product family design optimization process.

**Sub-objective 5: Optimize product family designs**

The product family design optimization problem is to be formulated. The modular configuration and engineering design parameters are the input variables that need to be optimized. The optimization result aims to generate optimal product family design solutions that achieve both performance enhancements and cost savings. Through the optimization, the proposed new platform concepts and design strategies aim to overcome the major drawback of conventional product families, i.e., the performance compromise due to platform sharing.

## 1.4 Research Scopes

The focus of this research is to develop methodologies for product platform and product family design, including the concept modeling, a computational AM design feature selection algorithm, an AM production costing approach, product family design evaluation metrics, and a product family design optimization method.

Commercial AM machines and materials available at Singapore Centre for 3D Printing, Nanyang Technological University are used in this research, while novel developments in AM processes and materials themselves are out of the scope of this research.

A case study of R/C racing car family design is used throughout the thesis to illustrate the proposed methodologies. The demonstrated methodologies can also be extended to other applications that may be explored in future work but not in this thesis.

## 1.5 Organization of the Thesis

**Chapter 1**: An overview of this research is presented.

**Chapter 2**: Relevant previous studies are reviewed, mainly in the fields of AM process developments and investigations, design for additive manufacturing (DFAM) methodologies, and platform-based product family design strategies. Research gaps are identified.

**Chapter 3**: An overview of the proposed product family design framework is presented. Methodologies embodied in the design framework are briefly explained. The fundamental concept of additive manufactured variable platform is proposed. The novel product family configuration consisting different categories of modules and design parameters is defined. A generalized mathematical model is formulated to represent a product family configuration, in which modules and design parameters are expressed by matrices and vectors.

**Chapter 4**: Design knowledge of AM design features and target components are coded numerically and achieved in databases. A novel hybrid machine learning approach is developed for automatic AM design feature selection. The case study on R/C racing car design is used to illustrate the proposed AM design feature selection method.

**Chapter 5**: The concepts of AM process setting platforms and variants are defined. AM production costs are estimated by the proposed Fuzzy Time-Driven Activity-Based Costing (FTDABC) approach, with an adaptive neuro-fuzzy inference system (ANFIS) adopted for time equation modeling. AM process setting adjustments' feasible space boundary is determined by solving an optimization problem. Variable platforms' design parameter constraints are then identified in order to limit cost increments due to design variations.

**Chapter 6**: The Complex Commonality Index (CCI) is formulated to measure product family commonality as an indicator of the cost saving potential. The market share (MS) is used to measure a product family's performance in the market that consists of competing alternative products. The multiobjective product family design optimization problem is formulated to optimize a product family's modular configuration and engineering design parameters simultaneously. The case study of designing an R/C racing car family is used to illustrate the product family design evaluation and optimization method.

**Chapter 7**: The methodologies and findings of this research are summarized. The contributions, limitations and future research directions are also suggested.

# Chapter 2 Literature Review

In this chapter, previous research on additive manufacturing (AM) technologies, design for additive manufacturing (DFAM) principles, and platform-based product family design methodologies are reviewed. Research gaps are identified and discussed at the end of this chapter.

## 2.1 Additive Manufacturing Technologies

### 2.1.1 Additive Manufacturing Processes

*2.1.1.1 An Overview of AM Processes:* In the work of Gibson et al. [1], working principles of different categories of AM processes are introduced. In particular, powder-bed fusion is a type of AM process which scans an energy source (e.g. a laser or electron beam) or sprays adhesive droplets selectively over the raw material in form of a powder bed. The scanned powders are then fused to form a unified solid part, while the rest of the loose powders can be recycled. In the direct energy deposition processes, raw materials in powder or wire form are delivered through a nozzle, while the laser or electron beam projects on the powder stream and simultaneously melt the materials on-the-fly. Melted materials re-solidify and bonded to the previous layers once the heat source has moved away. Among various powder-bed fusion AM processes, selective laser melting (SLM) is one of the most promising AM processes for fabricating functional metal components with high speed and accuracy [12]. As one type of powder-bed fusion AM processes, SLM uses laser to selectively melt the metal powders which re-solidify and form metallurgical bonding in a layer-by-layer format [12]. SLM has found numerous applications in aerospace, biomedical, automotive, and energy industries [13]. Recent technical advances in SLM machines and processes are summarized in [14].

*2.1.1.2: AM Process Settings:* AM process settings have significant impact on the quality and cost of additive manufactured products. Cheng et al. [15], Canellidis et al. [16], and Strano et al. [17] investigated the effect of build orientations on part accuracy, surface roughness, build time, post-process time, energy consumption, and the amount of required support materials. In the work of Verma [18], the amount of material waste in SLM was found related to the build orientation. Due to the anisotropy of the as-built material in SLM, the build orientation affects the material's tensile properties, as observed by Kempen et al. [19]. Support structures are used in many AM processes to support the overhang portion of the part and transfer heat from the upper layers to the base plate. Strano et al. [20] studied the impact of the support pattern to the material waste, energy, and time consumption in SLM. Key process parameters in SLM include scan speed, layer thickness, laser power, hatch spacing, and scan pattern, which influence the densification, microstructure, and hence mechanical properties of the as-built metal components. Effects of laser power and scan speed on surface roughness, density, microstructure, and hardness were investigated by Dadbakhsh et al. [21] and Alrbaey et al. [22]. Laser energy density (LED) was calculated from the laser power, scan speed, and hatch spacing, and its effects on material's wear resistance were studied by Gu et al. [23]. Different scan patterns can generate different microstructures in SLM [24-26]. AM process parameters also influence the environmental impact of AM fabrication. For instance, in the work of Verma and Rai [27], the laser intensity was correlated to the energy consumption in SLM. Other than AM processes themselves, the heat treatment as a post-process technique can also remarkably affect the materials properties of additive manufactured parts [25, 28].

### 2.1.3 Costing in Additive Manufacturing

Hopkinson and Dickens [29] analyzed the unit cost of an additive manufactured part and compared it with the cost in a traditional manufacturing process (injection molding). Ruffo et al. [30] developed an AM cost model which consisted of direct costs and indirect costs. Both aforementioned studies suggested that a traditional manufacturing process is more economical when the mass production volume is beyond thousands, while additive manufactured parts have lower unit costs when the production volume is small. Life-cycle cost (LCC) analysis on additive manufactured products was done by Lindemann et al. in [31, 32]. Cost drivers in the intrinsic lifecycle of AM were identified as: preparation, machine, heat treatment, post-process, and material. The variation of these cost drivers were then measured as response to the changing build rate, machine utilization rate, unit material costs, and machine investment costs. Time Driven Activity Based Costing (TDABC) was used as the cost calculation method by Lindemann et al. [31, 32]. The LCC model consisted of six phases, i.e. the (1) conception and definition, (2) design and development, (3) production, (4) installation, (4) usage and maintenance, and (5) disposal [32]. It was found from the LCC analysis that a significant cost saving effect could be achieved if designers were more familiar with design rules. In the work of Williams et al. [33], a cost model for the fused deposition modeling (FDM) AM process was constructed using the build time, material cost, labor cost, machine operation cost, machine maintenance cost, machine purchasing cost, and total number of produced parts. Atzeni et al. [34] proposed a cost model for metal parts fabrication by SLM. The result suggested that AM techniques could have an economic advantage in small to medium batch production compared with traditional manufacturing techniques. A general AM implementation framework was proposed by Mellor et al. [35] that aimed to guide company managers in adopting AM for greater competitiveness. The proposed framework was comprised of five factors: AM technology, supply chain, organization, operation, and strategy. Sustainability issues in AM processes were investigated by Bourhis et al. [36]. A design for sustainable additive manufacturing (DFSAM) methodology was proposed, which combined DFAM principles and environmental impact assessment methods. In the work of Bourhis et al. [37], environmental impacts of AM were modeled using the Lifecycle Analysis (LCA) approach based on the fluid consumption, material consumption, and electric consumption.

The literature review shows that AM costing approaches have focused on estimating absolute cost values; however, they lack the capability to predict cost changes as an effect of relative AM process setting adjustments and part design variations. In Chapter 6 of this thesis, the author adapts a time-driven costing method that relates AM cost changes with time consumption changes which are further mapped to part designs variations. Furthermore, the proposed costing method is capable to estimate uncertainties and fluctuations in cost terms through the adaption of fuzzy inference systems.

## 2.2 Design for Additive Manufacturing

Studies in AM-specific design capabilities and constraints are reviewed in Sections 2.2.1 and 2.2.2 respectively.

### 2.2.1 AM Capabilities and Design Features

As summarized by Rosen [38], unique design capabilities enabled by AM can be classified into shape complexity, material complexity, hierarchical complexity, and functional complexity. Many special AM design features such as cellular structures, topology optimized structures, integrated joints, and multi-material components have been introduced as new design freedoms for products' performance enhancement [39]. Maidin et al. [40] constructed an AM design feature database which allowed designers to view the information about design feature geometries and applications in the conceptual design stage. Su et al. [41] suggested a set of design guidelines for non-assembly mechanisms fabricated by selective laser melting (SLM). The relationship between a mechanism's manufacturability and SLM process strategies was also studied. The application of topology optimization (TO) in additive manufactured structures was presented by Brackett et al. [42] who analyzed the manufacturing constraints and established a workflow for redesigning TO structures in AM. In the same research, three novel methods (i.e. lattice structure, multi-material, and processing parameter variation) were proposed in handling intermediate density region in TO structures. Namasivayam and Seepersad [43] used topology optimization in designing lattice skins of deployable structures applied in unmanned aerial vehicle (UAV) inflatable wings. The lattice structure is one of the most popular design features in additive manufactured parts due to its high strength and light weight characters [2, 3, 38]. In general, a lattice structure is constructed by periodically arranging unit cells in the 3D space. Diamond-shape and gyroid-shape unit cells were fabricated by SLM in the work of Yan et al. [44] and Yan et al. [45] respectively. The relationship between the volume fraction and the compressive properties was established based on the Gibson-Ashby model. Manufacturability of lattice structures was studied by Wang et al. [46], in which the critical inclined angle and minimum strut size were found experimentally. The dimensional accuracy of unit cells fabricated by SLM was also investigated by Wang et al. [46]. In the work of Nguyen et al. [47], the Conformal Lattice Structure™ (CLS) was proposed and optimized by the augmented size matching and scaling (SMS) method with finite element analysis (FEA) for stress simulation. Multi-material and multi-functional design is another remarkable design capability provided by AM. Oxman [48] proposed the concept of "variable property rapid prototyping (VPRP)" to represent bio-inspired structures that had heterogeneous material compositions to achieve multi-functionality. In the work of Brackett et al. [49], electronic elements and wires were placed within a component. A voxel-based approach was developed for CAD modeling of the electronics-embedded design, and a multi-material binder-jetting AM process was used for fabrication. The design method, manufacturing constraints, and composition optimization of metallic multi-material parts with discrete compositional gradients were studied by Hu et al. [50], where the ultrasonic consolidation (UC) AM process was used for fabricating the multi-material parts.

However, the above studies focused on the construction of specific geometries without providing designers with a guideline to select appropriate AM design features. Moreover, the literatures mainly discuss about the design methods in the detailed design phase for additive manufactured components, while AM design feature recommendation at conceptual design phase needed further research.

### 2.2.2 AM Design Constraints and Rules

Although design freedoms have been significantly improved by AM technologies, constraints and rules still exist in various aspects of DFAM practices, such as material selections and dimensional limitations etc. [51]. DFAM rules should be followed to ensure the part's manufacturability and performance requirement to be met [52]. In the work of Hague et al. [53], mechanical properties of materials in stereolithography (SLA) and selective laser sintering (SLS) were used to help designers select the right material for the intended service environment. Design rules were investigated based on existing design for manufacturing (DFM) guidelines for injection moulding. It was found that although some existing DFM rules also apply to AM, many are no longer valid in AM since many design limitations in conventional processes could be removed. In the work of Adam and Zimmer [54], AM design rules were established to determine the allowable part position, wall thickness, length/orientation of non-supported overhang structure, and hole/column radius. Rezaie et al. [55] applied the AM design rules in the development of topology optimized structures, by limiting the geometric parameters within the process-specific constraints. The development of powder-based AM processes has enabled direct fabrication of functional metallic parts, and the design rules for metal AM have been researched intensively. Samperi et al. [56] used the Benchmarking Features Comparison Table to record the minimum and maximum values of three types of geometrical features in parts fabricated by SLM, and a DFAM Process Workflow was proposed for SLM. Vayre et al. [57] investigated the design constraints in the direct metal deposition (DMD) AM technique, and proposed a generic four-step design process that which included initial shape generation, geometric parameters definition, parameter optimization, and manufacturability validation. In the work of Vayre et al. [58], the manufacturability of parts fabricated by electron beam melting (EBM) was studied, and the effects of geometric design parameters on the powder removal efficiency and support structure density were investigated. Kranz et al. [59] established the design rules for basic shapes in lightweight lattice structures fabricated by laser additive manufacturing (LAM), and compiled the design rules using tables for easy retrieval and visualization. Zimmer and Adam [60] proposed a process independent method to define DFAM rules using standard elements. In the work of Liu and Rosen

[61], product design requirements (e.g. accuracy and surface finish) were mapped to manufacturing rules (feasible space of process variables) that were further mapped to specific process variables (e.g. part orientation and layer thickness) values. The feasible attribute value ranges of the standard elements were determined to ensure AM manufacturability. An on-going research effort in DFAM rules is the current development of the standard ASTM WK 38342 "New Guide for Design for Additive Manufacturing [62]", which aims to compile a collection of design rules for different AM processes and applications.

## 2.3 A Product Family and A Product Platform

### 2.3.1 Platform Selection and Optimization

The design process of a platform-based product family consists of two levels of decision making, i.e. (1) identifying the platforms to be shared and (2) optimizing the design variables for each product variant in the family [5]. Maximizing the performance and design commonality are frequently used as objective functions in product family design optimization. However, there is usually a trade-off between these two objectives, which has been observed in most literature [11, 63-65].

Specific platform selection and product family optimization techniques in the literature are introduced as follow. In the work of Chowdhury et al. [63], a product family was designed using the Comprehensive Product Platform Planning ($CP^3$) Framework. The $CP^3$ framework was composed of the $CP^3$ Model (for product family representation) and the $CP^3$ Optimization (for platform and design parameter identification). In the same research, it was suggested that population-based evolutionary optimization algorithms were ideal for solving product family design problems. It was also discovered that the tendency of platform formation decreased with increasing production volume [63]. de Weck et al. [11] proposed a two-level optimization method, in which the family level (upper level) and the product variant level (lower level) optimization iterated to maximize the overall profit, while the sales volume of a product variant was a function of its performance and price. Sawai et al. [66] formulated a platform design optimization problem for an industrial robot family to minimize the total robot weight, while the performance targets were treated as constraint functions in the problem. Moon et al. [64] applied the multiobjective particle swarm optimization (PSO) to select platforms and identify design variables simultaneously. A product family penalty function (PFPF) measuring the design commonality and a deviation function measuring the product performance were used as the two conflicting objective functions. A similar research was done by Tong et al. [67], who formulated a multiobjective optimization problem for a product family's modular configuration with the objectives to minimize the cost and maximize the product performance. Pareto-optimal solutions were found by a Pareto genetic algorithm (PGA) with the customized penalty function, fitness function, genetic coding, and genetic operation. Genetic algorithm (GA) was used in the work of Jiao et al. [68] to select platform modules and the level of features in all modules. The customer-perceived benefit of each product variant was constructed from part-worth utility functions, which were determined using conjoint analysis. Maximizing the ratio of the total utility over the total cost was used as the objective function. A similar method was presented in the work of Fujita and Yoshida [69], where both the design commonality and each product variant's attribute levels were optimized simultaneously. The expected final profit was used as the objective function. In the research of Luo et al. [65], a linear programming embedded genetic algorithm was applied to select product variants to be placed in the market, choose attribute levels, and to decide prices. Fuzzy numbers were used to model the imprecise information in market size and part-worth utilities. Genetic algorithm was applied in product family re-design by Thevenot and Simpson [70] to maximize the product family design commonality. In the work of Ji et al. [71], product variants were selected to be included in the family, while all product variant's design parameters were predetermined and treated as constants. Multiobjective genetic algorithm (MOGA) was used to maximize both the product family competitive advantage (PFCA) (a measure of the family's profitability) and the commonality index (CI) (a measure of the degree of design parameter sharing). A standard logit model for customer choice probability estimation was used to formulate the PFCA. The concept of a flexible platform was proposed by Suh et al. [72, 73]. The flexible platform was allowed to change over time in response to future uncertainties of the dynamic market conditions. In the work of Kashkoush and ElMaraphy [74], given the complete profile of a product line, hierarchical clustering was applied to create various product families in a reconfigurable assembly system (RMS). Based on three similarity coefficients for the assembly sequence, product commonality, and product demand respectively, three different sets of product families were formed. And finally a single consensus clustering tree was formed to represent all three families. Bryan et al. [75] introduced a mathematical model for the concurrent design of a product family and a RAS, which were jointly optimized using GA.

Existing product family design methods rely on the assumption that identical modules are shared as platforms in a family, which has caused the drawback of performance compromise. Motivated by the enhanced design freedom in AM, the author hypothesizes that AM enables a new platform strategy that can improve product family performance at lower or similar costs compared to conventional strategies. In this thesis, a novel concept of a variable platform is proposed.

### 2.3.2 Cost and Commonality Metrics for Product Families

Reducing cost is a major purpose of implementing platform strategies in product family design, and therefore a family-level cost model is usually included in the formulation of a product family design problem. Agyapong-Kodua et al. [76] provided a general review of cost modeling techniques for manufacturing systems. More specifically, a product family cost estimation system based on activity-based costing (ABC) method was proposed by Park and Simpson [77, 78]. Wang et al. [79] performed the life-cycle cost (LCC) modeling on product families and investigated the impact of a part or order change on the LCC. It was concluded that the increase of product variety in a product family will increase the LCC of the family. Chowdhury et al. [63] used a cost decay function to describe the phenomenon that the unit production cost decreased when higher design commonality was implemented in product families. Fujita and Yoshida [69] constructed three independent models for the design cost, facility cost, and production cost respectively. In these three models, any change in module design based on the platform module would lead to an increase all three types of cost. Luo et al. [65] used a linear additive cost model to calculate the product family cost as the summation of a fixed cost and a variable cost. The variable cost was the total of unit cost in all attribute levels of all selected products placed in the market.

Instead of directly calculating the product family’s total cost which requires lots of company or market information that is mostly uncertain and inaccessible at the product design stage, commonality metrics that indicate the degree of platform sharing are sometimes used as indirect measures of cost saving potentials of a product family.

A review and comparison of existing commonality metrics can be found in [5, 70]. Most existing commonality metrics were formulated using only the quantity of shared components or design parameters [71]. However, the Comprehensive Metrics for Commonality (CMC) proposed by Thevenot and Simpson [70] integrated both cost and component information. In the CMC formulation, the total component cost of a family was calculated as the summation of unit cost of all component variants in the family; and the unit cost of each component variant was the summation of the material cost, process cost, and set-up cost. The commonality metric proposed by Johnson and Kirchain [80] included a flexible weighting parameter whose values represented the relative importance of the corresponding components. The weighting parameter could be mass, cost, and investment. The relationship between the commonality metric and the resultant cost savings was modeled by a simple linear regression.

Although some of the metrics, such as the Comprehensive Metric for Commonality (*CMC*) proposed by Thevenot and Simpson [70], contain cost information, they consider only the absolute component costs, while relative values of cost savings due to platform sharing are not involved in the commonality evaluation. In Chapter 7 of this thesis, the author proposed a novel commonality metric that can reflect the cost saving potentials of platform sharing on different levels (i.e. the parametric level, modular level, and manufacturing process level).

### 2.3.3 Performance and Competitiveness Metrics for Product Families

Other than reducing costs, another major objective in product family design is to improve or at least preserve the performance of individual product variants. Various metrics have been developed to measure product performance and competitiveness in the market that contains competing alternative products from different manufacturers.

In the work of Moon et al. [64], the deviation function calculated as the difference between a performance indicator and its target value was used as the product variant's performance measure. The part-worth utility is one of the most popular techniques in modelling customer-perceived value of a product, which is approximated by a linear function of utility weights and product attributes [65]. In the work of Jiao et al. [68], the summation of all product variants' customer utility in their respective market segments were used as the performance measure of the product family. Bryan et al. [75] assumed that customers would prefer products with the highest utility. In contrast, Luo et al. [65] assumed that customers preference was determined by the "utility surplus" calculated as the difference between the utility and selling price. Ji et al. [71] applied the standard logit model to calculate the probability of customers choosing a product variant among multiple competing alternatives. And then the customer choice probability was used to formulate the product family competitive advantage (PFCA) as a family's performance metric. The more complex mixed logit model was used by Chen et al. [81] to estimate market demand of product lines, where utility weights were modelled by probability density functions.

### 2.3.4 Design Knowledge Representation and Management

In order to reduce designers' learning time and to their working efficiency, design knowledge including available design features, design requirements, and customer preferences can be represented and managed using computational approaches that can assist designers in their decision making processes [82]. A general review of existing methods and tools in product design knowledge representation was provided by Chandrasegaran et al. [83]. In the work of Nanda et al. [84], a knowledge management framework was developed for product family design. The framework combined the methods of network bill of material (NBOM), formal concept analysis

(FCA), ontology (OWL), and an object-relational database management system (ORDBMS). More specifically, linguistic and parametric design information/knowledge was represented by the FCA, while the object-relational database served as a central design repository of product design knowledge. The proposed framework was applied to product family re-design in order to increase the design commonality. Customer preference and functional requirements could be mapped to individual module's engineering design targets using data mining methods, such as classification [85, 86] and clustering [87] techniques. Moon et al. [88] proposed an ontology-based design knowledge representation approach for product family design. Chen and Wei [89] used object-oriented techniques in modeling product design feature and process knowledge, and created a framework of design feature evaluated against process-defined design rules. In the work of Xue and Dong [90], product design and manufacturing features in injection moulding were coded based on functional aspects, and fuzzy C-means (FCM) clustering was carried out to identify similar features that could be implemented in the concurrent design. Myung and Han [91] developed an expert system shell to integrate the knowledge base (design constraints and rules) with the design libraries (features library, parts library, and design units library), for the purpose of product feature re-design. Liu and Rosen [61] proposed a knowledge modeling and reuse strategy for additive manufacturing process planning. More specifically, in the proposed strategy, an old product's design features were modeled by ontology, and then a knowledge base composed of (IF-THEN) manufacturing rules was used to map new product design requirements to new AM process variables. In the research of Moon [92], association rule mining was applied in design knowledge acquisition. And then, a fuzzy C-means clustering based method was applied to partition various product functions into subsets, in which a platform module could be selected based on functional similarities. Fuzzy membership functions could be used to represent uncertainties in design knowledge. Hsiao and Tsai [82] applied a fuzzy neural network to estimate the relationship between a product form design and adjectival feelings of customers. Chi et al [93] introduced a knowledge-based expert system that performed failure modes and effect analysis (FMEF) to assist inexperienced designers in quality improvement, material selection, and cost assessment. Huang et al. [94] developed an integrated computational intelligence approach to generate and evaluate product concepts, based on a genetic algorithm (GA) for design optimization and a neural network for design evaluation. A similar method was presented by Sun et al. [95], who identified the best design concept candidate using a neural network-based fuzzy reasoning approach. In the work of Zhang and Xue [96] and Xue and Dong [90], a database containing primitive design features and a knowledge-base containing (IF-THEN) manufacturing process rules were used jointly to represent product lifecycles in concurrent design.

However, the above decision-making methods did not achieve the capability of recommending performance-enhancing design features for products' conceptual designs. Moreover, since AM is a relatively new technology, a computational method is desirable for inexperienced designers to explore AM-enabled design freedoms.

### 2.3.5 A Process Platform and A Process Family

A process family was a set of manufacturing processes variants, while the shared sub-process structure was defined as a process platform [97-99]. Implementation of commonality in both product and process designs is a strategy to provide variety to the market while reducing design and production costs. Due to Wang et al. [100], a generalized platform-based process family could be constructed by three constituents: (1) generic process features, which were composed of product features and workshop resources; (2) process knowledge, which was a set of IF-THEN rules describing manufacturing process constraints and limitations; and (3) generic process structures, which contained operations and the relationships between operations. The Generic Product-Process Structure (GPPS) was proposed in the work of Zhang et al. [97] and Jiao et al. [99] to model all common structures in product and process families. The GPPS was constructed from a Generic Product Structure (GPdS, i.e. product platform), a Generic Process Structure (GPcS, i.e. process platform), and the relationship between the GPdS and GPcS which represented the mapping from products to processes. Product and process variants were derived from the GPPS. All elements in the GPPS were modeled by object-oriented modeling methods using the Unified Modeling Language (UML). However, the above literatures focused on representing the overall structure (or sequence) of a manufacturing line that consisted of multiple processes (e.g. forming, assembly, etc.), whereas the detailed parameters of any particular process were out of the research scope. Williams et al. [33] implemented the concept of process platform in additive manufactured products. Variety in production capacity is managed by manipulating batch sizes, process parameters, and machine types. In the above article, AM process parameters for all customized product variants in a family were standardized, and product variant designs were assumed given. However, the process parameters were not able to adapt to the change in product design, which was a limitation of the above mentioned work. Furthermore, the above article did not consider the role of uncertainties in manufacturing time and cost prediction, which could be another drawback if the proposed method was applied in real industrial scenarios.

## 2.4 Research Gaps

Novel product design methodologies need to be developed for companies to explore the opportunities brought by new AM technologies. AM processes are expected to be applied more broadly by product developers in different industries in the near future, following the standardization of technologies and the fall of machine and material prices [101]. As an approach to develop mass-customized products while reducing cost and lead-time, platform strategies can be applied to the design of product families implemented with additive manufactured modules.

The literature review shows that extensive research has been conducted in the DFAM and product family design independently. However, few studies have been reported to link these two research fields together. This limitation is two-fold: 1) AM design features has rarely been implemented in the platform design of a product family, and 2) product family design methodologies have not been developed or re-invented in the situation where additive manufactured modules are included in the product family. To fill the research gap, in this thesis, the author proposes the concept of an additive manufactured variable platform, and a novel design framework is proposed to implement additive manufactured variable platforms in product families.

Performance compromise caused by design commonality is one of the major drawbacks in conventional platform-based product family design, which is the resultant trade-off of cost savings achieved by platform sharing [6, 9, 64]. In this research, the author aims to compensate, at least partially, the performance compromise by apply additive manufactured variable platforms which allow high design flexibility at managed production costs. The author creates novel commonality and performance metrics to evaluate product families implemented with the additive manufactured variable platforms. With the objectives to improve performance and save costs, a design optimization method is developed to support decision-makings in product family design, including module selection, platform selection, and design parameter identification.

The research gaps stated above are quite general. In the following chapters that explain the details of the proposed methodologies, more specific and in-depth discussions of research gaps in previous research will be made.

# Chapter 3 A Product Family Design Framework and Definitions of Concepts

In this chapter, a product family design framework and its constituent steps and methodologies are summarized as an overview.

## 3.1 An Overview of the Design Framework

The proposed design framework for product families implemented with additive manufactured modules is illustrated in Figure 3.1.

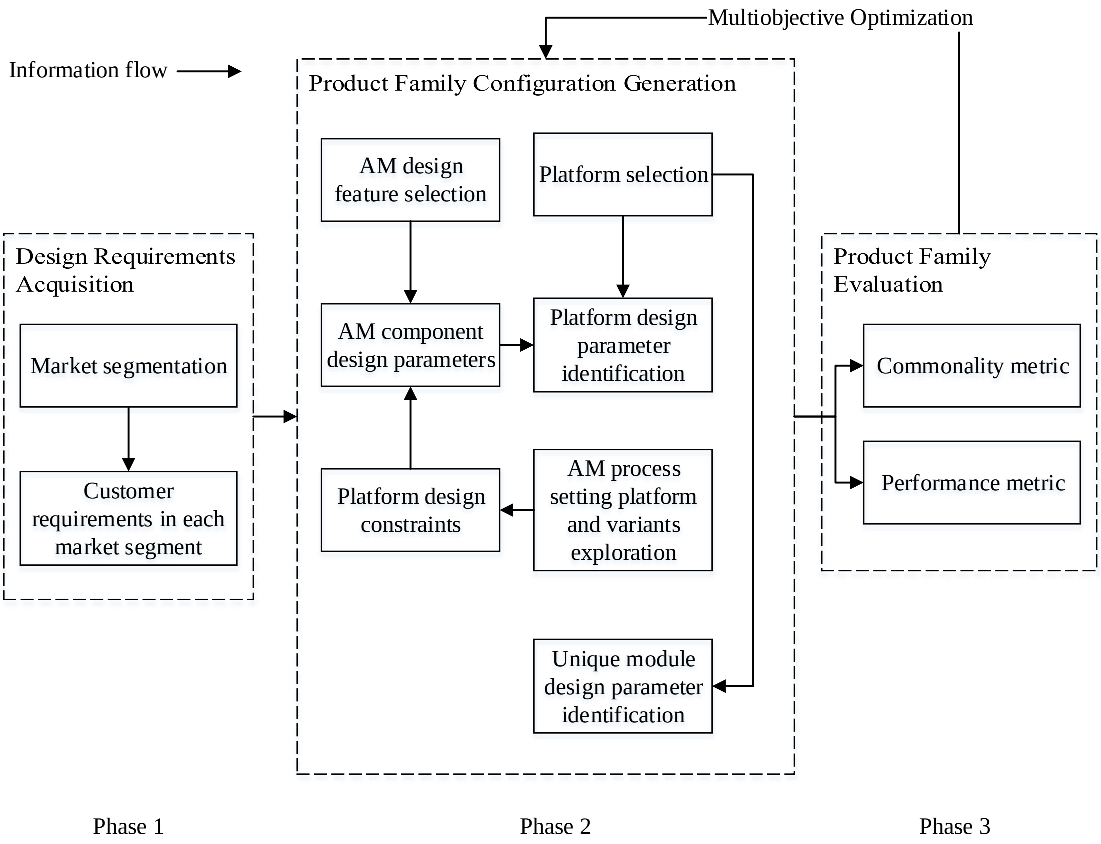



Figure 3.1: The product family design framework overview

The design framework consists of three major phases described as follows.

Phase 1. Design requirement acquisition: including market segmentation and customer requirement identification in each market segment.

Phase 2. Product family configuration generation: including platform identification, platform design parameter identification, process setting platform identification, and unique module design parameter identification. To utilize AM-enabled design space, appropriate performance-enhancing AM design features are selected for additive manufactured platform modules. AM process setting platforms and variants are explored, and the process setting adjustments' feasible space boundary is identified. Platform design constraints are derived in order to ensure manufacturability and control product family production costs.

Phase 3. Product family evaluation. Quantitative evaluation metrics are formulated in Phase 3. The evaluation metrics include a commonality metric and a performance metric, which are specific to the product family. The formulation of the metrics is presented in Chapter 6. The input variables of the product family evaluation models include the modular structure and component design parameters of the product family. The feedback arrow directed from Phase 3 to Phase 2 represents the iterations needed in product family design optimization. In the optimization process, the evaluation results calculated in Phase 3 are used as the objective functions.

The product family design aims to improve the performance of different product variants through design diversification and save costs through design commonality. However, both objectives may not be satisfied at the same time, while tradeoffs may exist between cost savings and performance enhancements [64].

Therefore, a multiobjective optimization approach is applied in the product family design.

In this thesis, the author focuses on Phase 2 and Phase 3 of constructing, evaluating, and optimizing a product family. The design requirement acquisition in Phase 1 can be carried out using marketing approaches [68], which is beyond the scope of this research. Therefore, design requirements are assumed known a priori during methodology development and case studies in this research.

## 3.2 Methodologies Preview

Methodologies within each step of the design framework are briefly previewed in Sections 3.2.1 to 3.2.3, as summarized in Figure 3.2. The details of each proposed methodology and the applications will be described in Chapters 4 to 6.

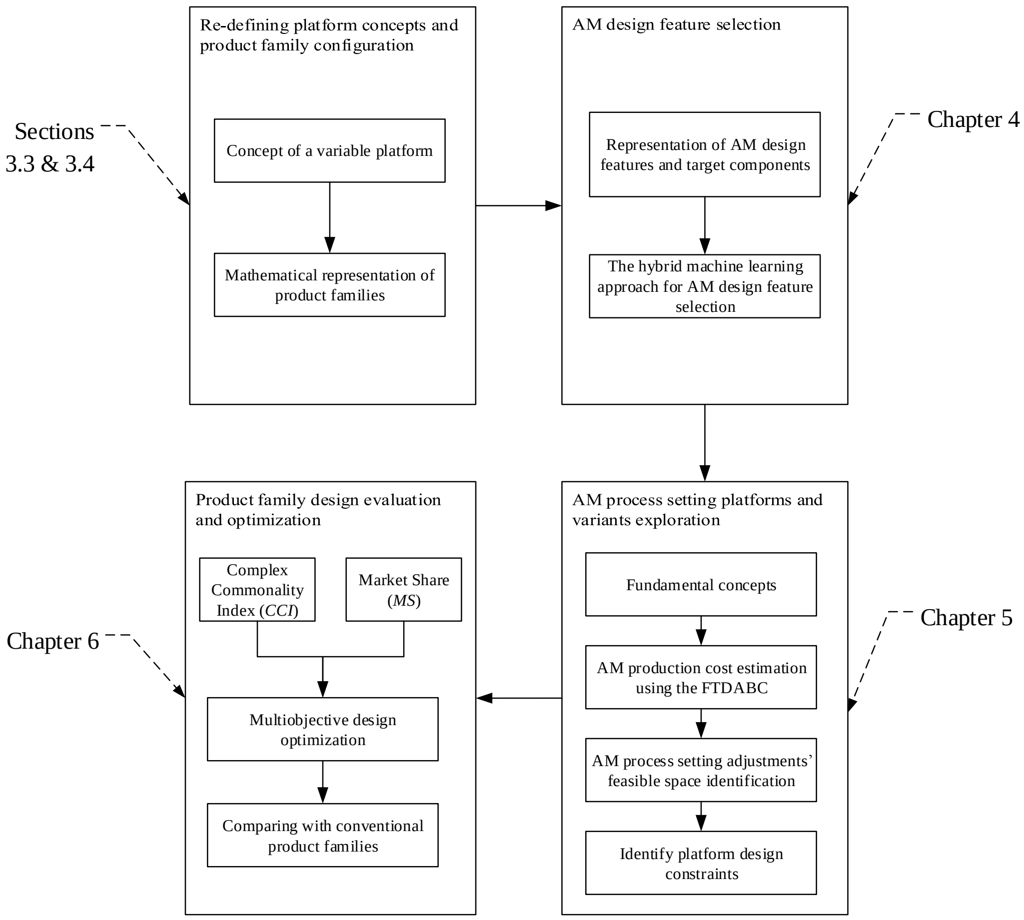



Figure 3.2: The preview of methodologies

Novel concepts in platform-based product family design are defined with AM technologies deployed in the production. The concept of an additive manufactured variable platform (VP) is proposed to describe the platform design variability due to the flexibility of AM techniques. A product family configuration is proposed to contain variable platforms, consistent platforms, and unique modules. A generalized mathematical model is formulated to represent the proposed product family configuration on both modular and parametric levels. In the formulated model, vectors and matrices are used to represent a product family's modules and different types of engineering design parameters. Detailed explanation of the aforementioned concepts and terms will be presented in Sections 3.3 and 3.4.

### 3.2.1 AM Design Feature Selection

To implement additive manufactured modules in product families, appropriate performance-enhancing AM design features are to be selected. With the purpose of organizing AM design knowledge for easy retrieval, usage, and updating, a computational approach is proposed to represent AM design features and target components. Three categories of design knowledge, i.e. "loadings" (i.e. the loading conditions of a structure such as forces and heat fluxes etc.), "objectives", and "properties", are coded and archived in databases. The hybrid machine learning approach is developed to select AM design features automatically and intelligently. In the proposed hybrid machine learning approach, hierarchical clustering is used to identify the relationship between the target component and AM design feature candidates in the database. A binary classifier trained by existing industrial application examples is used to determine the final sub-clusters resulted from the hierarchical clustering process. The AM design features contained in the resultant final sub-clusters are recommended to designers.

### 3.2.2 AM Process Setting Exploration and Costing

After AM design features are selected, the design constraints of additive manufactured VP modules are identified in order to control and limit cost increment due to VP design variations in the product family. AM production cost increment is estimated based on process setting adjustments using the proposed Fuzzy Time-Driven Activity-Based Costing (FTDABC) procedure. In the FTDABC method, an adaptive neuro-fuzzy inference system (ANFIS) is adapted to predict time equations. The process setting adjustments' feasible space is searched by solving a multiobjective optimization problem. The identified feasible space boundary and then mapped to a VP module's design constraints using a Mamdani model.

### 3.2.3 Product Family Design Evaluation and Optimization

Product family design is evaluated in terms of family-level commonality and performance. In this research, the Complex Commonality Index (*CCI*) is formulated to measure three levels of commonality, i.e. the manufacturing process level, modular level, and parametric level. The *CCI* value is used as an indicator of cost saving potentials brought by a platform strategy. The Market Share (*MS*) is formulated based on the customer choice probability to measure a product family's performance in the market that contains multiple competing alternative products. A multiobjective design optimization problem is formulated for a product family. Both the *CCI* and *MS* are used in the objective functions, while the module selection and engineering design parameters are treated as input variables to the optimization problem. A population-based evolutionary algorithm is used to solve the optimization, and a chromosome encoding pattern is proposed so that both the modular configuration and engineering design parameters can be optimized simultaneously.

In this chapter, the concepts of product platforms in product families are re-defined in the design process of additive manufactured products. Compared with conventional manufacturing techniques, AM processes are different in terms of the design space and cost sources. Therefore, some concepts or criteria traditionally considered in product family design need to be modified or re-invented to fit in the context of AM.

## 3.3 Fundamental Concepts of Variable Product Platforms

As summarized by Pirmoradi et al. [5], a platform is "a set of components, technologies, subsystems, processes, and interfaces that form a structure" which is shared by all product variants in the family, in the aim of maximizing commonality, minimizing development and production cost, and minimizing individual performance losses. There are two types of platforms: scalable platforms and modular platforms. Based on these two platforms, variants in scale-based (or parametric) product families can be developed by shrinking/stretching the platform's scalable variables; and variants in module-based (or configurable) product families can be developed by adding, removing, and substituting functional modules [6].

Conventionally, a platform module fabricated by traditional manufacturing processes is identical across the product family, with the same features and design parameter values shared by multiple product variants [6, 102]. In this research, such a conventional platform module is defined as a *consistent platform* (CP). Any change in a CP design will require modifications in traditional manufacturing processes such as fabricating a new mold or planning a new welding path. Such process modifications usually result in an increase in production cost. However, such cost increment due to platform design changes, however, is not always occurred in AM. The layer-by-layer manner of material fusion in AM makes it possible to fabricate components with different geometry or topology by applying a similar manufacturing process strategy, i.e. sharing the same process platform. In addition, tooling costs in conventional casting and machining manufacturing are no longer incurred in AM. Therefore, the production cost may not be largely increased by changing the platform design. However, other cost increment drivers, such as the manpower and resource consumptions, cannot be eliminated from the process of changing the platform design in a product family.

With AM technologies deployed for production, an additive manufactured product platform module can be subjected to design variations across different product variants in the family. In other words, product variants may share similar but not exactly the same platform modules. Such an additive manufactured platform module is defined as a *variable platform* (VP) in this research in contrast to the conventional CP. Furthermore, an instance of a VP in a particular product variant is defined as a *platform variant* (PV), whose design aims to satisfy the performance requirement of the individually customized product variant. Therefore, the application of additive manufactured variable platforms is expected to compensate the product variant's performance lost compared with the individually optimized design, while such performance compromise is one of the major disadvantages in conventional CP-based product families [6, 9, 64].

Comparing with the flexible platform concept proposed by Suh et al. [72, 73], the variable platform proposed in this research differs in several aspects. A flexible platform allows changes in the future based on market change, while a variable platform allows changes at the present moment based on individual product variant's performance requirement or customer needs. In a flexible platform based product family, all products variants at any time instance share a common platform, while in a variable platform based product family, product variants may use different platform variants but at the same time maintaining the low lifecycle cost.

## 3.4 Product Family Configuration and Representation

In this section, a generalized mathematical model is formulated to represent the configuration and design parameters of a platform-based product family. Mathematical terms, variables, and notations defined in the model will be used in other steps of the proposed product family design framework.

A novel product family configuration is proposed to implement unique modules, consistent platform modules, and additive manufactured variable platform modules. Figure 3.3 illustrates the configuration of a representative family of two product variants positioned in two market segments respectively. In this research, the term "*module*" represents either an individual functional component or a group of components that perform a basic task.

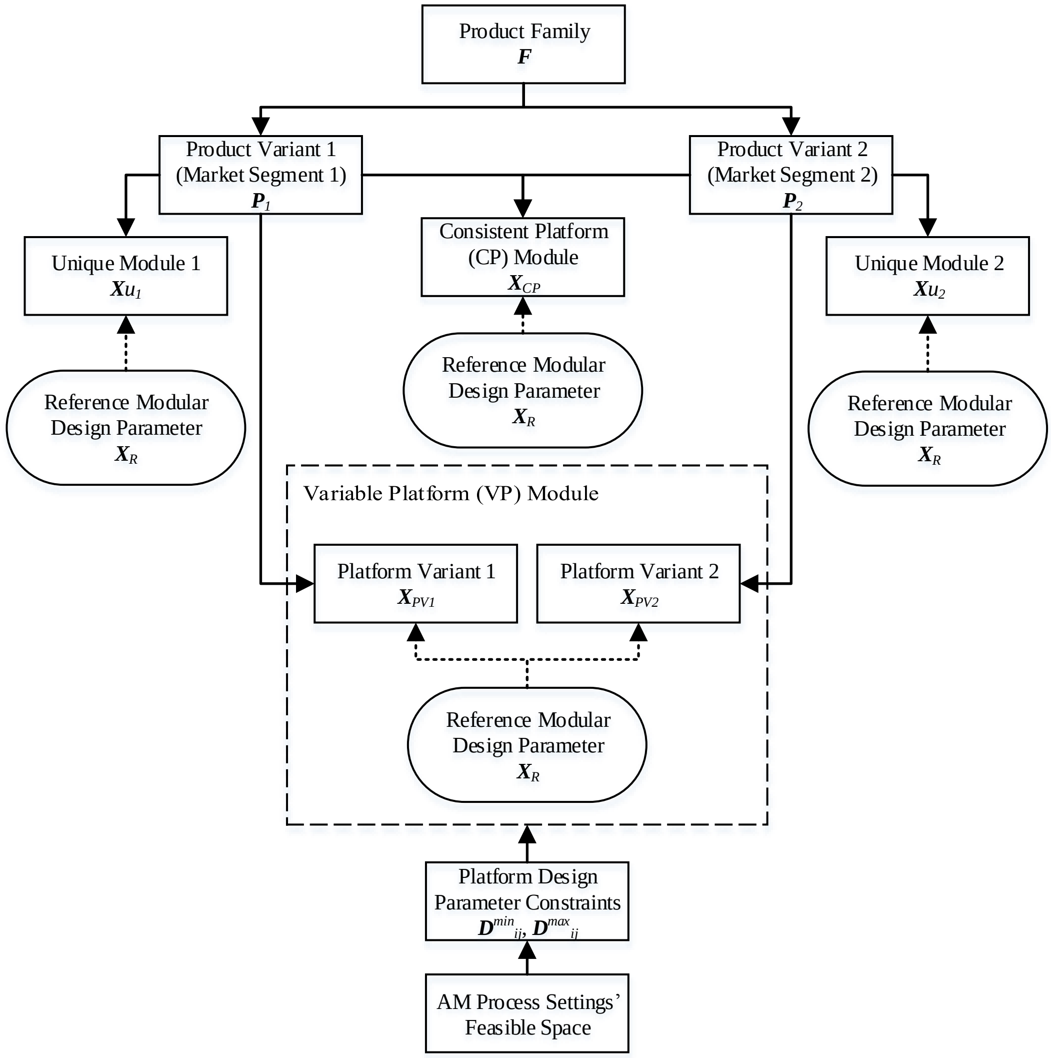



Figure 3.3: A representative family of two product variants

A product family $\boldsymbol{F}$ is a vector as formulated in Eq. (3.1), which contains $N$ product variants positioned in $N$ different market segments.

$$\boldsymbol{F} = \left[\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_i, \ldots, \boldsymbol{P}_N\right]^T \tag{3.1}$$

The configuration of the $i^{th}$ product variant $\boldsymbol{P}_i$ is represented by a vector of $M$ modules, with the $j^{th}$ module defined by a design parameter vector $\boldsymbol{X}_{ij}$. $M$ is the total number of all types of module candidates in the product family. $\boldsymbol{P}_i$ can also be expressed as the product of the *Module Selection Matrix* $\boldsymbol{B}_i$ and the *Module Candidate List* $\boldsymbol{X}_L$. $\boldsymbol{X}_L$ is a vector of all available modules; and $\boldsymbol{B}_i$ is a $M \times M$ diagonal matrix whose diagonal elements $b_{jj}$ are either 1 or 0, depending on whether the module candidate $\boldsymbol{X}_j$ is included in the $i^{th}$ product variant. The definition and relationships of $\boldsymbol{P}_i$, $\boldsymbol{X}_L$, and $\boldsymbol{B}_i$ are formulated in Eq. (3.2).

$$
\begin{aligned}
&\boldsymbol{P}_i = \begin{Bmatrix} \boldsymbol{X}_{i1} \\ \boldsymbol{X}_{i2} \\ \vdots \\ \boldsymbol{X}_{ij} \\ \vdots \\ \boldsymbol{X}_{iM} \end{Bmatrix} = \begin{bmatrix} b_{11} & 0 & \dots & 0 & \dots & 0 \\ 0 & b_{22} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \cdots \\ 0 & 0 & \cdots & b_{jj} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & b_{MM} \end{bmatrix}_i \begin{Bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_j \\ \vdots \\ \boldsymbol{X}_M \end{Bmatrix} = \boldsymbol{B}_i \boldsymbol{X}_L \\
&b_{jj} = \begin{cases} 1, \ if \ Module \ \boldsymbol{X}_j \ is \ used \ in \ product \ variant \ i \\ 0, \ otherwise \end{cases} \\
&\boldsymbol{X}_{ij} = \begin{cases} \boldsymbol{X}_j, \ if \ b_{jj} = 1, Module \ \boldsymbol{X}_j \ is \ used \ in \ product \ variant \ i \\ 0, \ otherwise \end{cases} \\
&where \ i = 1, 2, \dots, N \ and \ j = 1, 2, \dots, M
\end{aligned}
\tag{3.2}
$$

The *Modular Commonality Matrix* $\boldsymbol{C}$ of a product family is defined as the element-wise summation of all matrices $\boldsymbol{B}_i$ corresponding to all $N$ product variants, as formulated in Eq. (3.3). $\boldsymbol{C}$ is a $M \times M$ diagonal matrix with its diagonal element $c_{jj}$ be an integer ranging from 0 to $N$. When the value of $c_{jj}$ is larger than 1, the corresponding module $\boldsymbol{X}_j$ is a platform module shared by multiple product variants in the family; when $c_{jj}$ is 1, the corresponding module $\boldsymbol{X}_j$ is a unique module used in only one product variant but not shared as a platform; and when $c_{jj}$ is 0, the corresponding module $\boldsymbol{X}_j$ is not used in any product variant in the family.

$$\boldsymbol{C} = \sum_{i=1}^{N} \boldsymbol{B}_i = \begin{bmatrix} c_{11} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & c_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{jj} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & c_{MM} \end{bmatrix}$$

$$\begin{cases} 1 < c_{jj} \leq N, \ \mathit{Module}\ \boldsymbol{X}_j\ \mathit{is\ a\ platform\ module} \\ c_{jj} = 1, \ \mathit{Module}\ \boldsymbol{X}_j\ \mathit{is\ a\ unique\ module} \\ c_{jj} = 0, \ \mathit{Module}\ \boldsymbol{X}_j\ \mathit{is\ not\ used\ in\ the\ family} \end{cases} \tag{3.3}$$
$$\mathit{where}\ i = 1, 2, \ldots, N\ \mathit{and}\ j = 1, 2, \ldots, M$$

Design parameters in a platform-based product family are classified into three major types:

(1) *Unique module design parameters* $\boldsymbol{X}_U$: design parameters of unique modules that are not shared by more than one product variant in the family.

(2) *Consistent platform design parameters* $\boldsymbol{X}_{CP}$: design parameters of consistent platform modules that are identically shared by multiple product variants.

(3) *Platform variant design parameter* $\boldsymbol{X}_{PV}$: design parameters of variable platform modules, which can take different values in different product variants.

In practice, designers usually refine or optimize their design based on a set of given references. The vector $\boldsymbol{X}_R$ represents a module candidate's reference design parameters. In this research, $\boldsymbol{X}_R$ can be either the parameters of existing/original component design or pre-determined by designers in an initial design phase. $\boldsymbol{X}_R$ will be treated as constants during design optimization, from which new and optimal design parameters $\boldsymbol{X}_U$, $\boldsymbol{X}_{CP}$, and $\boldsymbol{X}_{PV}$ are derived for individual products variants.

Mathematically written in Eq. (3.4), $\boldsymbol{X}_{U,ij}$, $\boldsymbol{X}_{CP,ij}$, and $\boldsymbol{X}_{PV,ij}$ are the products of $\boldsymbol{X}_{R,i}$ and the *Design Parameter Variation Matrix* $\boldsymbol{D}_{ij}$ which determines design parameter values of the $j^{th}$ module in the $i^{th}$ product variant during the optimization process. $\boldsymbol{X}_{R,j}$ is a vector of $n$ design parameters. $\boldsymbol{D}_{ij}$ is a $n \times n$ diagonal matrix whose diagonal elements $d_{kk}$ are dimensionless positive scale factors for controlling the variation of each design parameter across different products in the family.

$$
\begin{aligned}
&\boldsymbol{X}_{\Omega,ij} = \boldsymbol{D}_{ij}\boldsymbol{X}_{R,j} \\
&\Rightarrow \begin{Bmatrix} x_j^1 \\ x_j^2 \\ \vdots \\ x_j^k \\ \vdots \\ x_j^n \end{Bmatrix}_i = \begin{bmatrix} d_{11} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{kk} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & d_{nn} \end{bmatrix}_{ij} \begin{Bmatrix} x_j^1 \\ x_j^2 \\ \vdots \\ x_j^k \\ \vdots \\ x_j^n \end{Bmatrix}_R \\
&\forall\ \boldsymbol{X}_{\Omega,ij} \in \left\{ \boldsymbol{X}_{U,ij}, \boldsymbol{X}_{CP,ij}, \boldsymbol{X}_{PV,ij} \right\} \\
&where\ i = 1,2,\ldots,N\ and\ j = 1,2,\ldots,M
\end{aligned}
\tag{3.4}
$$

Platform modules' design parameters are subject to constraints imposed by the production process. Such platform design constraints are expressed in an inequality equation shown in Eq. (3.5), where $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ contain the minimum and maximum allowable values of diagonal elements $d_{kk}$ in $\boldsymbol{D}_{ij}$, respectively.

$$
\boldsymbol{D}_{ij}^{min} \le \boldsymbol{D}_{ij} \le \boldsymbol{D}_{ij}^{max} \tag{3.5}
$$

Using the above modeling method, the proposed product family configuration can be fully described by $\boldsymbol{B}_i$, $\boldsymbol{D}_{ij}$, and $\boldsymbol{X}_R$ for all $i$ and $j$ values in the family. The designers can design a product family by either selecting the values in $\boldsymbol{B}_i$, $\boldsymbol{D}_{ij}$, and $\boldsymbol{X}_R$ manually, or solving a product family design optimization problems formulated based on the model.

The idea of variable commonality levels of platform design was studied in literatures. For example, Khajavirad and Michalek [103, 104] identified the limitation of "all-or-none" sharing of design parameters that resulted in suboptimal solutions. To overcome the limitation, the authors extended the commonality index to the continuous space and allowed multiple levels of design parameter sharing. In other words, a design parameter could be shared by any number (larger than one) of product variants rather than all of them in the family. Case studies in bathroom scale and electric motor family design showed improved optimal solutions using the relaxed commonality index. Similarly in the work of Hernandez et al. [105], multiple levels of commonality for different platform design parameters were enabled for the purpose of product customization. In the case study of pressure vessel design, a large set of combinations of design parameters were generated, from which customers could select the preferred one based on their specific requirements. Another relevant study was done by Khire and Messac [106], who proposed the Selection-Integrated Optimization method to design adaptive systems (such as a HVAC system) whose parameters were meant to change under changing working conditions. The key difference between the proposed product family configuration in this work and that in the aforementioned literatures lies in the definition of decision variables. In this research, product variants are composed of various interchangeable modules, and a module is a "variable platform (VP)" if it is shared by more than one product variants although all of its design parameters are adjustable in the family. Both the modular configuration and design parameters are decision variables in this work. However, in the aforementioned literatures, the commonality was on the parametric level. In other words, a component was a "platform" only when at least one of its design parameters was kept the same in multiple product variants sharing this component. The modular configuration was given and hence not part of the decision variables. The above difference comes from the different relationship between design parameters and manufacturing cost in traditional and additive manufacturing processes. In the above references [103-105], unshared design parameters led to extra tooling costs (e.g. in die casting and forging), which may not be valid for AM.

### 3.5 Summary

In Chapter 3, the proposed product family design framework was briefly explained. The concepts of the consistent platform (CP) and variable platform (VP) were introduced in product families implemented with additive manufactured modules. VP modules made by AM techniques can be subject to design variations across the product family. A novel product family configuration was proposed, and the mathematical model was introduced to represent the product family configuration on both modular level and parametric level.

# Chapter 4 AM Design Feature Selection

As introduced in the product family design framework described in Chapter 3, in order to implement additive manufactured modules in product families, appropriate AM design features should be selected for product performance enhancement. Similar to [39, 40, 107, 108], "design feature" in this work denotes the special performance enhancing features enabled by AM, which is different from the general term in computer-aided design.

Compared to conventional subtractive and formative manufacturing processes, AM processes are more capable of creating parts with complex geometries, multi-functionalities, and other performance enhancing features [1]. Design space has been enlarged significantly by the availability of numerous AM design features [3]. However, in practice, many designers tend to rely on their knowledge and personal experience in selecting appropriate design features at the conceptual design stage [82], while there is a lack of systematic and intelligent techniques to explore AM-enabled design space. Such an empirical design process is time-consuming and inefficient, especially when an increasing number of new AM design features are developed and made available to designers in the future. Therefore, an automated AM design feature selection method is desired. Furthermore, working efficiency of novice designers can be improved if knowledge and experience accumulated in past design projects done by sophisticated designers can be utilized in new design projects. A limitation of existing design knowledge representation and management methods in literature is that they are not able to combine knowledge discovered from both sources (i.e. current design feature databases and existing industrial application examples) and use it to generate new design proposals.

The work described in this chapter aims to develop an automated and intelligent AM design feature selection method by developing computational design knowledge representation and management methodologies that are able to obtain information from current AM design feature databases and existing industrial application examples. The proposed design method is expected to help designers explore AM-enabled design freedom, in order to utilize AM benefits with higher efficiency. In this research, the author proposes a novel AM design feature selection method based on a hybrid machine learning approach, which combines techniques of clustering (unsupervised learning) and classification (supervised learning). Three dimensions of design knowledge of AM design features and target components are coded in a binary format. Hierarchical clustering is performed on all coded AM design features and target components, resulting in a dendrogram. The cut-off threshold of the dendrogram is iteratively updated to define AM design feature selection, by applying a support vector machine (SVM) classifier trained using existing industrial application examples implemented with different AM design features. The final output of the proposed hybrid machine learning approach is a sub-cluster that is used as a conceptual design proposal with the selected AM design features. The hierarchical structure of the resultant sub-cluster can also indicate the possibility of combining different AM design features based on their similarities. A case study of designing a product family of R/C racing cars is carried out, in which the proposed hybrid machine learning approach demonstrates the capability of identifying appropriate AM design features.

## 4.1 A Computational AM Design Feature Selection Method

In the proposed AM design feature selection method, appropriate AM design features are mapped to target components through a hybrid machine learning approach, in which design knowledge is represented and managed computationally. As illustrated in Figure 4.1, the proposed hybrid machine learning approach consists of the following four procedures: 1) design knowledge of AM design features and target components are coded in a binary format and then stored in a database using an object-oriented approach; 2) hierarchical clustering is applied on all archived AM design features and target components, resulting in a dendrogram with a hierarchical-tree structure; 3) a support vector machine (SVM) classifier is trained by existing industrial application examples implemented with different AM design features; and 4) the hierarchical clustering and the SVM classifier are combined to formulate an SVM-based progressive dendrogram cutting algorithm that finds the final cluster containing the recommended AM design features.

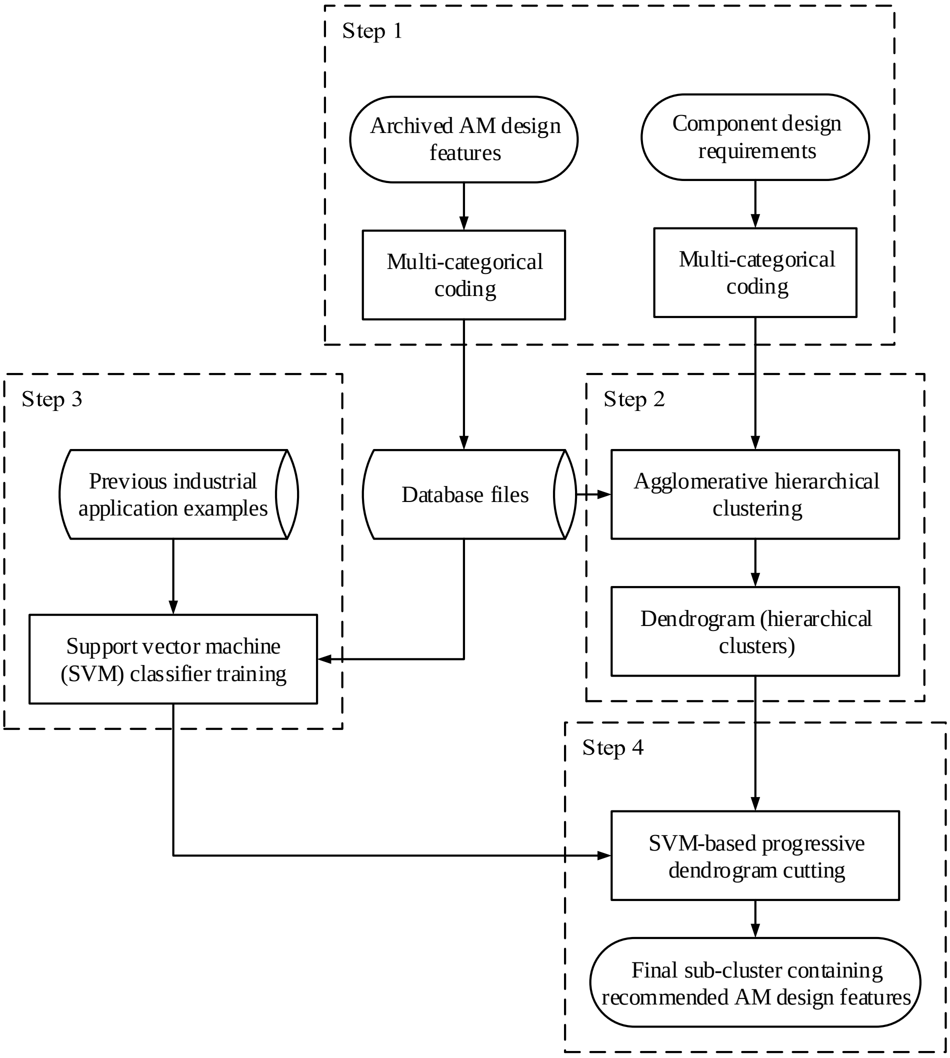



Figure 4.1: The general procedure of the proposed computational AM design feature selection method

### 4.1.1 Representation of AM Design Features and Target Components

In this research, functionality-centric design knowledge inherent in AM design features and target components can be classified into three categories: 1) loadings, 2) objectives, and 3) properties as shown in Figure 4.2. In order to store and extract the AM design features in database files, their relevant design knowledge is coded with numerical digits that will be used in mathematical operations of the proposed AM design feature recommendation algorithm. The detailed explanation and coding method for each category of AM design knowledge are presented below.

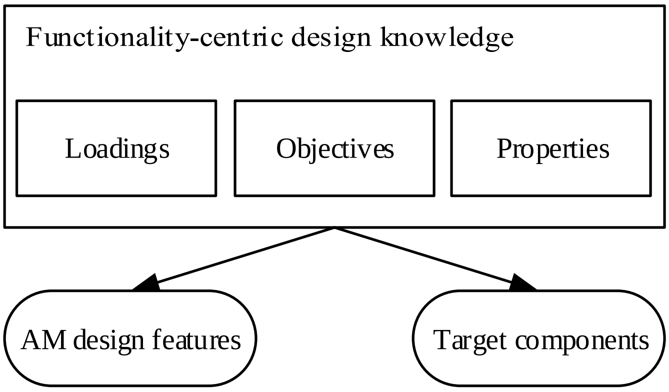



Figure 4.2: Categories of functionality-centric design knowledge

The "loadings" category contains a list of physical inputs, including mechanical loadings, fluid and heat flux, and electromagnetic loadings. The "loadings" category is coded in a binary pattern as shown in Table 4.1. The digit "1" indicates that the corresponding loading will be applied to the design feature or the target component; while the digit "0" indicates that the corresponding loading is absent. The loadings of an AM design feature or a target component are stored in databases as a vector $\boldsymbol{L} = [l_1, l_2, \ldots l_i \ldots]^T \left(l_i \in \{0,1\}\right)$.

Table 4.1: Coding of the "loadings" category

| Types | Loadings | Code | |
|---|---|---|---|
| | | ***Present*** | ***Absent*** |
| Static mechanical loadings | Inward pressure | 1 | 0 |
| | Outward pressure | 1 | 0 |
| | Static bending | 1 | 0 |
| | Static compression | 1 | 0 |
| | Static friction | 1 | 0 |
| | Static tension | 1 | 0 |
| | Static torque | 1 | 0 |
| Dynamic mechanical loadings | Impact | 1 | 0 |
| | Dynamic bending | 1 | 0 |
| | Dynamic compression | 1 | 0 |
| | Dynamic tension | 1 | 0 |
| | Dynamic torque | 1 | 0 |
| | Kinetic friction | 1 | 0 |
| Fluid and heat flux | Air friction | 1 | 0 |
| | Air resistance | 1 | 0 |
| | Liquid friction | 1 | 0 |
| | Liquid resistance | 1 | 0 |
| | Heat conduction | 1 | 0 |
| | Heat convection | 1 | 0 |
| | Heat radiation | 1 | 0 |
| Electro-magnetic loadings | Electric current | 1 | 0 |
| | Magnetic flux | 1 | 0 |

The “objectives” category contains a list of design purposes of the AM design features and target components. As shown in Table 4.2, the “objectives” category is coded with numerical ratings from 1 to 3 indicating levels of relevance, and it is also stored as a vector $\boldsymbol{O} = [o_1, o_2, \ldots o_i \ldots]^T \left(o_i \in \{1,2,3\}\right)$.

Table 4.2: Coding of the "objectives" category

| **Applications** | **Relevance level** | | |
|---|---|---|---|
| | ***Irrelevant*** | ***Marginally relevant*** | ***Very relevant*** |
| Aesthetics | 1 | 2 | 3 |
| Cable routing | 1 | 2 | 3 |
| Damping | 1 | 2 | 3 |
| Decreasing friction | 1 | 2 | 3 |
| Dissipating heat | 1 | 2 | 3 |
| Ease of mounting | 1 | 2 | 3 |
| Fastener removal | 1 | 2 | 3 |
| Fastening | 1 | 2 | 3 |
| Housing | 1 | 2 | 3 |
| Human body-shape compliance | 1 | 2 | 3 |
| Increase durability | 1 | 2 | 3 |
| Increasing compression | 1 | 2 | 3 |
| Increasing expansion | 1 | 2 | 3 |
| Increasing fluid resistance | 1 | 2 | 3 |
| Increasing friction | 1 | 2 | 3 |
| Increasing linear flexibility | 1 | 2 | 3 |
| Increasing torsional flexibility | 1 | 2 | 3 |
| Instant assembly | 1 | 2 | 3 |
| Reducing compression | 1 | 2 | 3 |
| Reducing expansion | 1 | 2 | 3 |
| Reducing fluid resistance | 1 | 2 | 3 |
| Reducing weight | 1 | 2 | 3 |
| Resisting linear distortion | 1 | 2 | 3 |
| Resisting torsional distortion | 1 | 2 | 3 |
| Ventilation | 1 | 2 | 3 |

The “properties” category contains a list of key properties, including mechanical, chemical, thermal, and electromagnetic properties that have crucial effects on the intended performance of the AM design features and target components. As shown in Table 4.3, the "properties" category is coded with numerical ratings of importance and stored as a vector $\boldsymbol{P} = \left[p_1, p_2, \ldots p_i \ldots\right]^T \left(p_i \in \{1, 2, 3\}\right)$.

Table 4.3: Coding of the "properties" category

| **Types** | **Properties** | **Importance level** | | |
|---|---|---|---|---|
| | | ***Unimportant*** | ***Marginally important*** | ***Very important*** |
| Mechanical | Coefficient of friction | 1 | 2 | 3 |
| | Creep rate | 1 | 2 | 3 |
| | Density | 1 | 2 | 3 |
| | Fatigue life | 1 | 2 | 3 |
| | Hardness | 1 | 2 | 3 |
| | High-T strength | 1 | 2 | 3 |
| | Impact toughness | 1 | 2 | 3 |
| | Shear modulus | 1 | 2 | 3 |
| | Tensile strength | 1 | 2 | 3 |
| | Wear rate | 1 | 2 | 3 |
| | Yield strength | 1 | 2 | 3 |
| | Young's modulus | 1 | 2 | 3 |
| Chemical | Bio-compatibility | 1 | 2 | 3 |
| | Oxidation rate | 1 | 2 | 3 |
| Thermal | Specific heat capacity | 1 | 2 | 3 |
| | Thermal conductivity | 1 | 2 | 3 |
| | Thermal absorptivity | 1 | 2 | 3 |
| Electro-magnetic | Electric conductivity | 1 | 2 | 3 |
| | Magnetic permeability | 1 | 2 | 3 |

The lists in Tables 4.1 – 4.3 were compiled by surveying research papers and reports (e.g. [3, 38, 39, 107, 108]), online resources (e.g. case studies [109-111] and news [112] etc.), and other AM application projects conducted in-house at Singapore Centre for 3D Printing. The lists are intended to be comprehensive at the author's best knowledge. A simple 3-point scale is used in Tables 4.2 and 4.3, since it is easy for designers to comprehend, fast to read and rate, and less susceptible to the differences in individual designers' subjective opinions compared with other more complicated scales.

The "loadings", "objectives", and "properties" categories in the functionality-centric design knowledge are usually interrelated. For example, an AM design feature or a target component that carries a "static tensile loading" may has "resisting linear distortion" as the design objectives, and the "Young's modulus" and "tensile strength" are the key properties. In this research, the expert knowledge of experienced designers has been stored by coding the archived AM design features, and hence it can be extracted by inexperienced designers by reading the coded vectors. An example of the coding pattern of an AM design feature (***F***) and a target component (***R***) is shown in Eq. (4.1).

$$\begin{aligned}
\boldsymbol{F} &= \begin{Bmatrix} [Loadings] \\ [Objectives] \\ [Properties] \end{Bmatrix} = \begin{Bmatrix} [110 \ \cdots \ 010 \ \cdots] \\ [132 \ \cdots \ 112 \ \cdots] \\ [212 \ \cdots \ 111 \ \cdots] \end{Bmatrix} \\
\boldsymbol{R} &= \begin{Bmatrix} [Loadings] \\ [Objectives] \\ [Properties] \end{Bmatrix} = \begin{Bmatrix} [100 \ \cdots \ 000 \ \cdots] \\ [122 \ \cdots \ 311 \ \cdots] \\ [010 \ \cdots \ 101 \ \cdots] \end{Bmatrix}
\end{aligned} \tag{4.1}$$

Since design knowledge is represented numerically, it can be used in computational techniques. The proposed hybrid machine learning approach is introduced in Section 4.1.2.

### 4.1.2 The Hybrid Machine Learning Approach for AM Design Feature Selection

To satisfy customer needs, AM design features need to be mapped to each target component's design requirements. Such a mapping provides a design solution at the early conceptual design stage. To automate the AM design feature selection process, the author proposes a hybrid learning approach that consists of two integrated techniques: 1) hierarchical clustering using an unsupervised learning procedure that groups AM design feature candidates and target components based on their similarity, and 2) the SVM classification based on a supervised learning procedure that estimates AM design feature selection criteria.

The hierarchical clustering results in a dendrogram which can be cut into multiple hierarchical sub-clusters. Existing industrial application examples from literature and online archives are used as training data sets in the SVM. The trained SVM classifier is then applied to determine the dendrogram cut-off threshold that defines the final AM design feature selection.

#### *4.1.2.1 Hierarchical Clustering of AM Design Features and Target Components*

Clustering is the process of dividing unlabelled data into groups (or clusters) in such way that data within the same cluster are similar, while those in different clusters are dissimilar [113]. Hierarchical clustering creates a hierarchical structure in which sub-clusters merge in the order determined by inter-cluster distances [114]. Compared to other clustering algorithms, hierarchical clustering does not require a predetermined cluster number, and different cluster sets can be created simply by specifying different granularity. Another advantage of the hierarchical clustering is that the hierarchical relationship among data is intuitively comprehensible to human observers, and more informative than results from flat clustering algorithms. Therefore, in this research, the author implements the hierarchical clustering on AM design features and design requirements to identify the relationships among them.

Cosine distance has been used in measuring dissimilarity between vectors in high-dimensional spaces, such as in text searching problems [115]. In functionality-centric AM design knowledge (i.e. "loadings", "objectives", and "properties") as discussed in Section 4.1.1, the cosine distance between any two vectors in the same category is calculated as:

$$
\begin{aligned}
& d_{\phi}(\boldsymbol{A},\boldsymbol{B}) = 1 - \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} = 1 - \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \\
& \phi = L(\,for\,\text{"}loadings\text{"}), O(\,for\,\text{"}Objectives\text{"}), and\ P(\,for\,\text{"}Properties\text{"}) \\
& d_{\phi}(\boldsymbol{A},\boldsymbol{B}) \in [0,1]
\end{aligned}
\tag{4.2}
$$

where vectors ***A*** and ***B*** can be from two different entities (an "entity" can be either an AM design features or the target component). $A_i$ and $B_i$ are the $i^{th}$ elements of ***A*** and ***B*** respectively. Since all vectors contain only non-negative real elements, the cosine similarity $d_{\varphi}(\boldsymbol{A},\boldsymbol{B})$ is in the range of [0,1]. A smaller $d_{\varphi}(\boldsymbol{A},\boldsymbol{B})$ value indicates a shorter cosine distance; and $d_{\varphi}(\boldsymbol{A},\boldsymbol{B})$ is 0 when ***A*** and ***B*** are exactly the same. In the hierarchical clustering algorithm, the overall distance metric $D$ between any two entities ***M*** and ***N*** is the Euclidean distance derived from different categories of cosine distances, calculated as:

$$
D(\boldsymbol{M},\boldsymbol{N}) = \sqrt{d_L(\boldsymbol{L}_M,\boldsymbol{L}_N)^2 + d_O(\boldsymbol{O}_M,\boldsymbol{O}_N)^2 + d_P(\boldsymbol{P}_M,\boldsymbol{P}_N)^2} \tag{4.3}
$$

where ***L***, ***O***, and ***P*** represent vectors in the "loadings", "objectives", and "properties" categories respectively. A symmetric proximity matrix (***PM***) can be constructed to store $D$ values between all AM design features and the target component, as shown in Eq. (4.4):

$$\begin{array}{c c} & \begin{array}{cccc} \boldsymbol{F}1 & \boldsymbol{F}2 & \cdots & \boldsymbol{R} \end{array} \\ \boldsymbol{PM} = \begin{array}{c} \boldsymbol{F}1 \\ \boldsymbol{F}2 \\ \vdots \\ \boldsymbol{R} \end{array} & \begin{bmatrix} 0 & D(\boldsymbol{F}1,\boldsymbol{F}2) & \cdots & D(\boldsymbol{F}1,\boldsymbol{R}) \\ & 0 & & D(\boldsymbol{F}2,\boldsymbol{R}) \\ & & \ddots & \vdots \\ & & & 0 \end{bmatrix} \end{array} \tag{4.4}$$

where $\boldsymbol{F}$ and $\boldsymbol{R}$ represent AM design features and the target component respectively

The Complete-Linkage $S(C1,C2)$ is defined as the distance between the most dissimilar members (with the largest $D$ value) from any two clusters $C1$ and $C2$ [116]. The mathematical expression of $S(C1,C2)$ is shown in Eq. (4.5). In the proposed hierarchical clustering process, the Complete-Linkage is used to represent inter-cluster distances.

$$S(C1,C2) = \max\left\{D\left(\boldsymbol{M}^*,\boldsymbol{N}^*\right):\ \boldsymbol{M}^* \in C1,\ \boldsymbol{N}^* \in C2\right\} \tag{4.5}$$

Using the distance metrics, proximity matrix, and Complete-Linkage discussed above, the hierarchical clustering algorithm is formulated. The pseudo-code is shown in Figure 4.3. Through iteratively merging the closest sub-clusters and updating the proximity matrix, hierarchical clustering results in a tree-like structure called a dendrogram as shown in Figure 4.4, with all archived AM design features and the target component located at its nodes. On the horizontal axis, the label "$\boldsymbol{R}\times\times$" represents the target component to be designed, and "$\boldsymbol{F}\times\times$" represents an AM design feature. Multiple levels of sub-clusters are linked by inverted U-shape lines, whose heights indicate inter-cluster distances calculated by the Complete-Linkage.

1. *Initialize: total number of points = N; cluster level L = 0.*

2. *Compute* $D(\boldsymbol{M},\boldsymbol{N})$ *between all* $(\boldsymbol{M},\boldsymbol{N})$ *and construct the initial proximity matrix* $\boldsymbol{PM}_0$.

3. *Merge the most similar clusters with* $S(C1,C2)_L = \max\ D(\boldsymbol{M}^*,\boldsymbol{N}^*)_L$.

4. *Update the proximity matrix* ($\boldsymbol{PM}_L$) *using* $S(C1,C2)_L$ *values*

5. *Update L = L+1.*

6. *Go to Step 2 until all points are contained in one cluster (i.e. L = N - 1).*

Figure 4.3: The pseudo-code of the hierarchical clustering algorithm

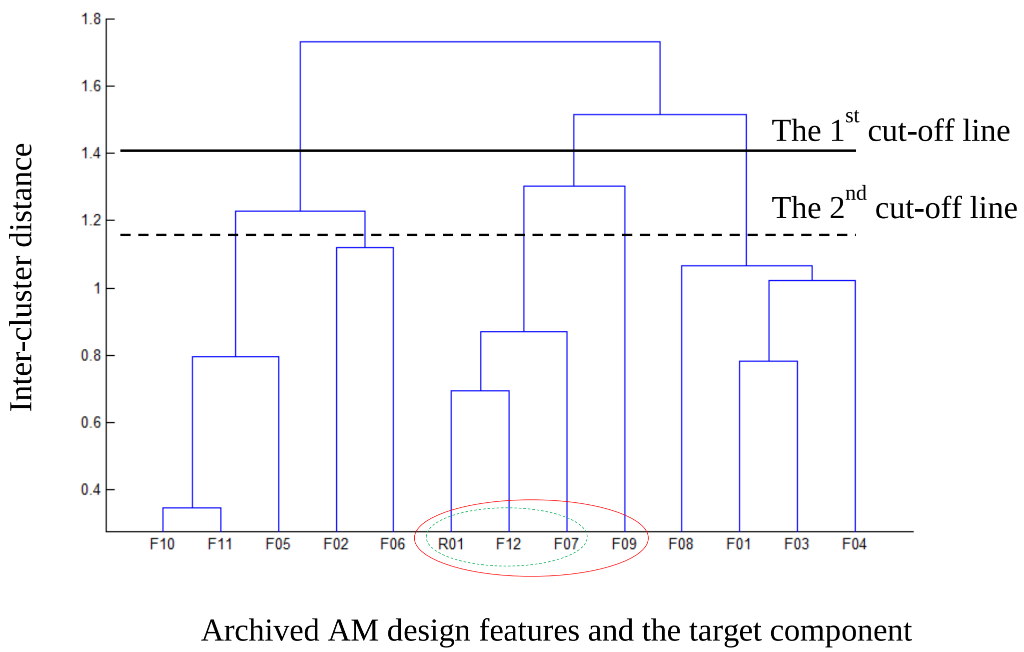



Figure 4.4: A dendrogram example showing the hierarchical clustering result

The clustering result can be interpreted as follows. If the dendrogram in Figure 4.4 is trimmed at the 1$^{st}$ cut-off line, the target component ***R***01 and the AM design feature ***F***12, ***F***07, and ***F***09 are grouped in the same sub-cluster circled by the dashed line, which indicates that the component ***R***01 is possible to be designed by implementing the AM design feature ***F***12, ***F***07, and ***F***09 expected performance enhancement. However, this result is not deterministic. The shift of the cut-off threshold may create different sub-clusters and hence different AM design feature selection result. For example, if the dendrogram in Figure 4.4 is trimmed at the 2$^{nd}$ cut-off line, ***R***01, ***F***12, and ***F***07 are now grouped in the sub-cluster circled by the dashed line, while the feature ***F***09 is excluded from the sub-cluster and hence it will not be considered to be implemented in ***R***01. Therefore, the decision of the cut-off threshold is crucial in determining the final clustering result. Cluster evaluation criteria have been developed to find the cut-off threshold purely based on statistical characteristics of the data set, such as the pooled within-cluster dispersion measurement [117]. However, conceptual design problems usually involve human designers' knowledge and judgments in decision making. Therefore, existing cut-off threshold identification methods are not satisfactory in the application of AM design feature selection, and a novel method implemented with supervised learning capability is proposed, as described in Sections 4.1.2.2 and 4.1.2.3.

#### *4.1.2.2 The SVM Classifier Trained by existing industrial application examples*

The dendrogram introduced in Section 4.1.2.1 displays the hierarchical relationship among different AM design features and target components based on the calculated distance metric. However, the dendrogram alone does not directly indicate which AM design features are to be selected.

The decision-making process of selecting AM design features can be considered as a binary classification problem. In the existing industrial application examples, the selected design features belong to the "*positive*" class (denoted as +1) while the unselected ones belong to the "negative" class (denoted as -1). Supervised machine learning techniques can be applied to train the binary classifier using the given training data that consist of input vectors and corresponding target values [118]. Support vector machine (SVM) has been reported in literatures as a promising classification algorithm due to its superior accuracy and resistance to over-fitting [119-121]. According to Delgado et al. [122] who compared 17 types of classifier on 121 datasets, SVM was one of the two most accurate classifiers (along with random decision tree). Based on the recommendation by the aforementioned study, in this research, several different classifiers (i.e. SVM, decision tree, k-nearest neighbors, and neural network) were tested in classifying the industrial application examples, and SVM outperformed the others by showing the highest accuracy (90.2%) on the test data. Therefore, in this research, support vector machine (SVM) is used for AM design feature classification. Details of the SVM was described by Vapnik [123]. Given a set of $N$ training data pairs $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \in R^n$ and $y_i \in \{+1, -1\}$, the optimal separating hyperplane (OSH) defined by $(\boldsymbol{W}, b)$ with the maximum margin is found by solving the following optimization problem:

$$
\begin{aligned}
\underset{\boldsymbol{W},b,\xi}{min}: \quad & \frac{1}{2}\|\boldsymbol{W}\|^2 + C\sum_{i=1}^{N}\xi_i \\
s.t. \quad & y_i\left(\boldsymbol{W}^T\phi(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i \\
& \xi_i \geq 0 \\
& j = 1, 2, \ldots, N
\end{aligned}
\tag{4.6}
$$

where $C$ is the penalty parameter, and $\xi_i$ is the slack variable. Input vectors $\boldsymbol{x}_i$ are mapped to a higher dimensional space by the function $\phi$ with the corresponding kernel function defined as the dot product: $K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \phi\left(\boldsymbol{x}_i\right)^T \phi\left(\boldsymbol{x}_j\right)$. A linear OSH can then be found in this higher dimensional space. The radial basis function (RBF) kernel $K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\gamma \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right)$ is used in this research. After the hyperplane $(\boldsymbol{W}, b)$ has been found by training, any new input vector $\boldsymbol{x}$ can be classified by the function $sign\left(\boldsymbol{W}^T \phi\left(\boldsymbol{x}\right) + b\right)$ with the output being either +1 or -1. Training data pairs closest to the OSH are called support vectors, which are the most critical points that define the OSH and the margin.

In this research, training data pairs are obtained from existing industrial application examples, including those presented in literature and retrieved from online archives. For each existing industrial application example, the component's design requirements are coded, and cosine distances in "loadings" ($d_L$), "objectives" ($d_O$), and "properties" ($d_P$) categories from all AM design feature candidates to the target component are calculated in the process discussed in Section 4.1.2.1. Therefore, for each AM design feature candidate in the industrial application example, a training data pair is expressed in the following format:

$$\begin{aligned} \left(\boldsymbol{x}_i, y_i\right) &= \left(\left[d_L, d_O, d_P\right]_i^T, y_i\right) \\ y_i &= \begin{cases} +1 \ (\text{"}positive\text{"}) \\ -1 \ (\text{"}negative\text{"}) \end{cases} \end{aligned} \tag{4.7}$$

Assuming that $N$ industrial application examples are used for the SVM classifier training, the total number of training data pairs is $N \times n$, where $n$ is the number of AM design feature candidates in the database. Some of the training examples with their implemented AM design features and corresponding values of $d_L$, $d_O$, and $d_P$ are shown in Table 4.4. The implemented AM design features belong to the "*positive*" class, while the rest in the feature candidate database belong to the "*negative*" class. The industrial application examples in Table 4.4 were retrieved from literatures [39, 107] and online archives [109, 110]. The distance values ($d_L$, $d_O$, and $d_P$) were calculated using the formula in Eq. (4.2).

#### *4.1.2.3 The Combination of Hierarchical Clustering and the SVM Classification in the Hybrid Machine Learning Approach*

The AM design feature selection is carried out through the proposed hybrid machine learning approach, which combines hierarchical clustering and the SVM classification. The general steps of the proposed hybrid machine learning approach are described as follows, with the flowchart illustrated in Figure 4.5.

Firstly, a *starting cut-off line* for the hierarchical clustering dendrogram is specified by setting the initial number of sub-clusters at only two. The resultant sub-cluster containing the target component is defined as the "*active sub-cluster*" (denoted as $C_0$), while the other sub-cluster is defined as the "*inactive sub-cluster*" (denoted as $C_1$). Secondly, an SVM-based progressive dendrogram cutting process is carried out, in which the trained SVM classifier is used to update the cut-off line iteratively. And thirdly, the final cut-off line is found after its updating iterations end; and the corresponding final sub-clusters are isolated from the dendrogram. The AM design features contained by the final sub-clusters are recommended to designers.

Table 4.4: Example training data from existing industrial application examples

| **Component** | **Implemented AM design features** | $d_L$ | $d_O$ | $d_P$ |
|---|---|---|---|---|
| The sole of a soccer boot | Hollow studs | 0.7 | 0.59 | 0.39 |
| | Spikes | 0.68 | 0.57 | 0.22 |
| | Surface texture | 0.87 | 0.62 | 0.58 |
| Stool | Ergonomics – freeform surface | 0.45 | 0.58 | 0.28 |
| | Surface texture | 0.86 | 0.50 | 0.41 |
| | Internal ribbing | 0.67 | 0.65 | 0.13 |
| Headband of a headphone | Integrated socket | 0.80 | 0.95 | 0.25 |
| | Weave structure | 0.43 | 0.52 | 0.10 |
| Car grill | Honeycomb feature | 0.59 | 0.69 | 0.31 |
| Automotive engine heat sink | Internal cooling fins | 0.28 | 0.20 | 0.05 |
| | Internal flow path | 0.47 | 0.50 | 0.10 |
| Aircraft engine racket | Thin-wall beam | 0.43 | 0.39 | 0.11 |
| | Topology optimized feature | 0.45 | 0.52 | 0.26 |
| Fuel injection swirler | Internal blade array | 0.33 | 0.25 | 0.07 |
| | Internal shelving | 0.38 | 0.47 | 0.12 |
| Hip implant | Perforated surface | 0.70 | 0.58 | 0.31 |
| | Porous structure | 0.37 | 0.62 | 0.23 |
| Exhaust gas probe | Internal flow path | 0.55 | 0.70 | 0.15 |
| | Thin-wall shell | 0.53 | 0.63 | 0.22 |
| Satellite antenna support | Topology optimized structure | 0.56 | 0.35 | 0.12 |
| plenum chamber in vehicle engine | External ribbing | 0.62 | 0.41 | 0.34 |
| | Integrated manifold | 0.36 | 0.30 | 0.13 |
| Suspension swingarm | Lattice structure | 0.32 | 0.45 | 0.24 |
| UAV body frame | Spiral feature | 0.45 | 0.48 | 0.26 |
| | Wireframe structure | 0.48 | 0.54 | 0.17 |

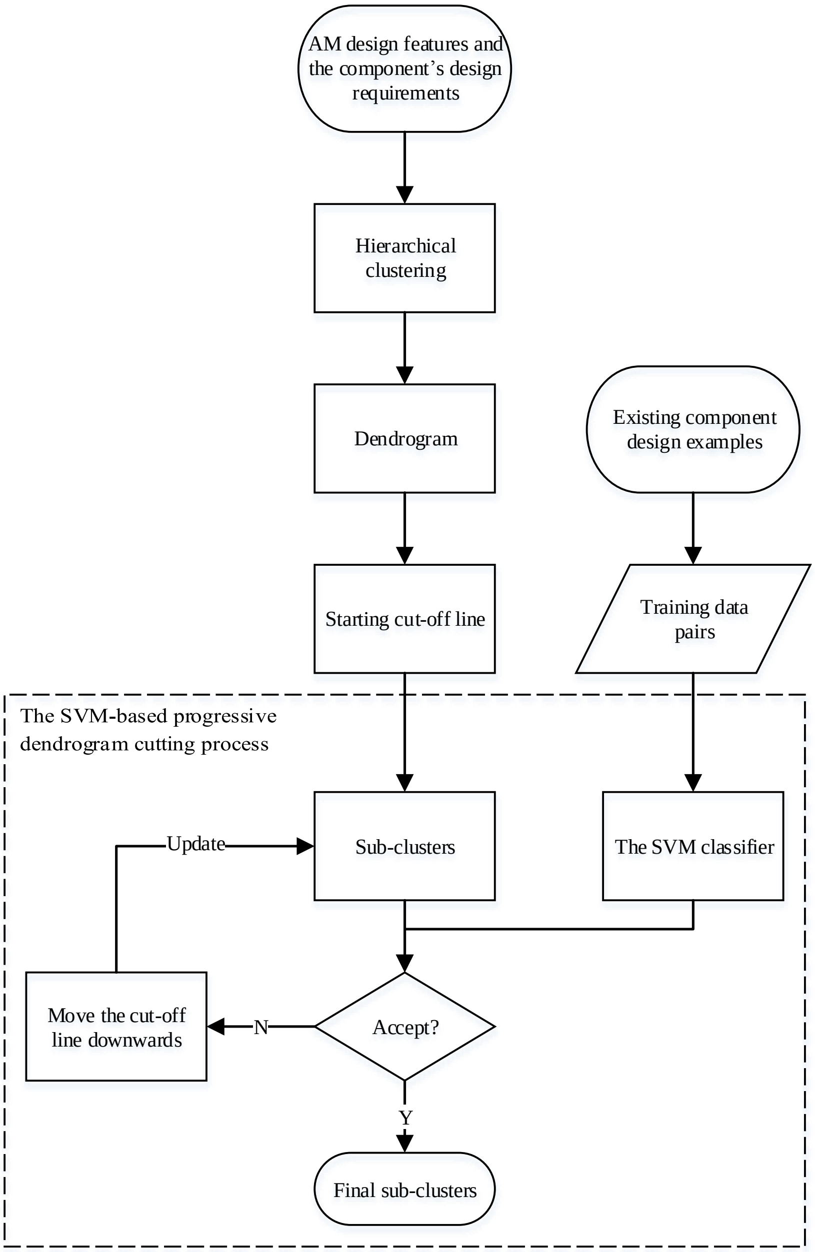



Figure 4.5: The hybrid machine learning approach for AM design feature selection

The SVM-based progressive dendrogram cutting algorithm is the core of the above procedure. By shifting the cut-off line downwards, the current active sub-cluster $C_0$ is further trimmed into two new sub-clusters, among which the one containing the target component is updated as the new $C_0$, while the other one is updated as the new $C_1$. The distance $D(\boldsymbol{F}_i, \boldsymbol{R})$ from each AM design features ($\boldsymbol{F}_i$) in $C_1$ to the target component ($\boldsymbol{R}$) in $C_1$ is computed using Eqs. (4.2) and (4.3). The smallest $D$ value is found and its corresponding design feature is denoted as $\boldsymbol{F_S}$. The trained SVM classifier is then applied to classify $\boldsymbol{F_S}$. If the output label is negative ("*-1*"), the other AM design features in the same sub-cluster $C_1$ also have high possibility to be rejected since the Complete-Linkage is used for measuring inter-cluster distances in hierarchical clustering. In this case, $C_1$ is discarded from the feature candidate list, while the active cluster $C_0$ is trimmed in the next iteration when the cut-off line is shifted further downwards. The progressive dendrogram cutting and sub-cluster updating process is iterated until the stop criterion is met, i.e. when the classification output label of $\boldsymbol{F_S}$ is positive ("*+1*"). The final sub-clusters $C_0$ and $C_1$ are isolated from the dendrogram. The AM design features contained by the final $C_0$ and $C_1$ are recommended as appropriate AM design features for the target component. The pseudo-code of the SVM-based progressive dendrogram cutting algorithm is shown in Figure 4.6.

```
WHILE ("Trimming the current active sub-clusters C0" == TRUE)
{
  Decrement the cut-off line in C0;
  Form two sub-clusters C[1] and C[2];
  IF (the target component is contained by C[1])
    C0 = C[1]; C1 = C[2];
  ELSE
    C0 = C[2]; C1 = C[1];
  FOR (all AM design features Fi in C1)
    Compute D(Fi, R);
  Identify FS in C1 with the smallest D value;
  SVM_Label = SVM_Classify(FS);
  IF (SVM_Label == "-1")
    "Trimming the current active sub-clusters C0" = TRUE;
    Start the next iteration of the WHILE loop;
  ELSE
    "Trimming the current active sub-clusters C0" = FALSE;
    Break from the WHILE loop;
}
Isolate the final sub-clusters C0 and C1;
Retrieve AM design features contained in the final sub-clusters C0 and C1;
```

Figure 4.6: The pseudo-code of the SVM-based progressive dendrogram cutting algorithm

As an illustration, the shifting of the dendrogram cut-off line in three consecutive iterations is shown in Figure 4.7. $CL^1$ is the starting cut-off line resulting in only two sub-clusters, while $CL^2$ and $CL^3$ represent two new line positions as $CL^1$ is shifted downwards in order to trim the active sub-cluster into new $C_0$ and $C_1$. Sub-clusters corresponding to $CL^1$, $CL^2$, and $CL^3$ are listed in Table 4.5. Assuming that the dendrogram cutting process stops at $CL^3$ when the classification output label of $\boldsymbol{F_S}$ is positive, the AM design features ***F12 F07***, and ***F09*** contained by the final $C_0$ and $C_1$ are recommended for the target component ***R01***.

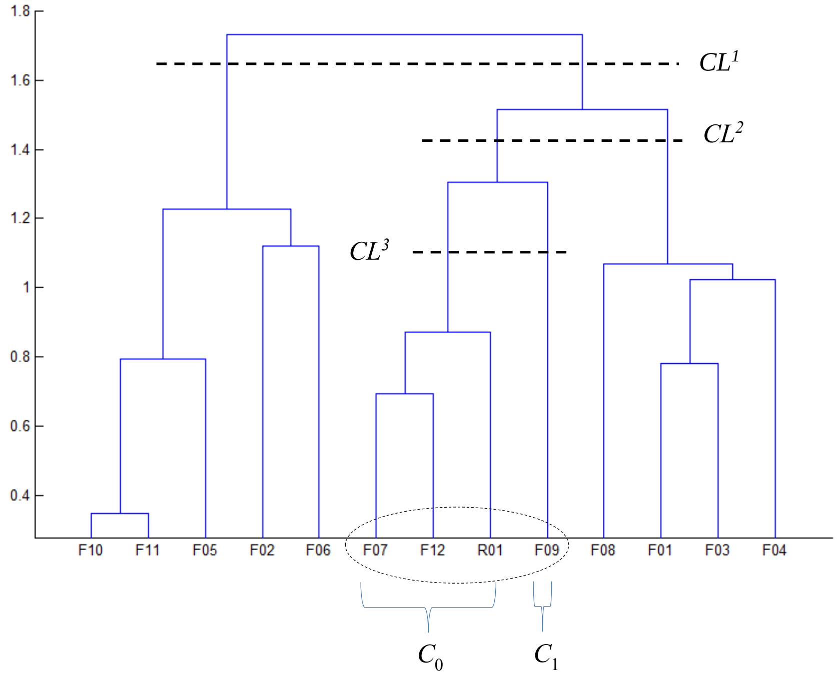



Figure 4.7: Update of the cut-off line in the SVM-based progressive dendrogram cutting process

Table 4.5: Sub-clusters in different iterations of the cut-off line update

| Iteration | Cut-off line | $C_0$ | $C_1$ |
|---|---|---|---|
| 1 | $CL^1$ | {***R01***,***F12***,***F07***,***F09***,***F08***,***F01***,***F03***, ***F04***} | {***F10***,***F11***,***F05***,***F02***,***F06***} |
| 2 | $CL^2$ | {***R01***,***F12***,***F07***,***F09***} | {***F08***,***F01***,***F03***, ***F04***} |
| 3 | $CL^3$ | {***R01***,***F12***,***F07***} | {***F09***} |

By using the hybrid machine learning approach, appropriate AM design feature selection is suggested and treated as a design solution. However, human designers' judgments cannot be avoided in most design processes. Therefore, in practice, the design solution generated by the proposed computational AM design feature selection method needs to be evaluated by human designers with their experience and preference, which leads to the final decision. When new design knowledge, including new AM design features and new industrial application examples, are available in the future, they can be used to update the AM design feature selection process by updating relevant databases.

## 4.2 Case Study

In the case study, R/C racing car components were designed and fabricated by AM techniques. To illustrate the proposed method, four different components were chosen as the target components for AM design feature implementation: the bumper (***R***01), wheel (***R***02), suspension arm (***R***03), and driveshaft (***R***04). The existing designs of the four target components are shown in Table 4.6. Despite the fact that there are over hundreds of AM design features [107], only twenty-one of them, as listed in Table 4.7, are demonstrated in the case study for easy visualization of resultant dendrograms. These AM design features were coded and organized using the object-oriented representation method in the database implemented in MATLAB®.

The proposed hybrid learning algorithm was applied to select appropriate AM design features for each target component. Resultant dendrograms are shown in Figures 4.8 to 4.11, where the final cut-off lines ($CL^F$) obtained from the SVM-based progressive dendrogram cutting process are indicated by horizontal dash lines. Final sub-clusters $C_0$ and $C_1$ are also marked by dash circles.

Table 4.6: Current design of four target components

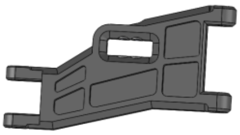
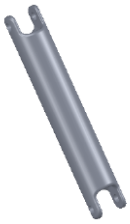
| Target component | Existing design | R/C racing car overview |
|---|---|---|
| Bumper (***R**01*) |  |  |
| Wheel (***R**02*) |  | |
| Suspension arm (***R**03*) |  | |
| Driveshaft (***R**04*) |  | |

Table 4.7: The AM design feature candidate list in the case study

| ***F**01* | ***F**02* | ***F**03* | ***F**04* | ***F**05* | ***F**06* | ***F**07* |
|---|---|---|---|---|---|---|
| Encapsulated bearing | Encapsulated spring | Honeycomb structure | Porous structure | Spiral structure | Topology optimized structure | Curved tunnel |
| ***F**08* | ***F**09* | ***F**10* | ***F**11* | ***F**12* | ***F**13* | ***F**14* |
| Integrated rotational joint | Wireframe | Lattice structure | Integrated socket | Spike | Perforated panel | Freeform surface |
| ***F**15* | ***F**16* | ***F**17* | ***F**18* | ***F**19* | ***F**20* | ***F**21* |
| Threaded surface | Snap-fit hinge | Thin-wall beam | Auxetic array | Internal ribbing | External ribbing | Weave structure |

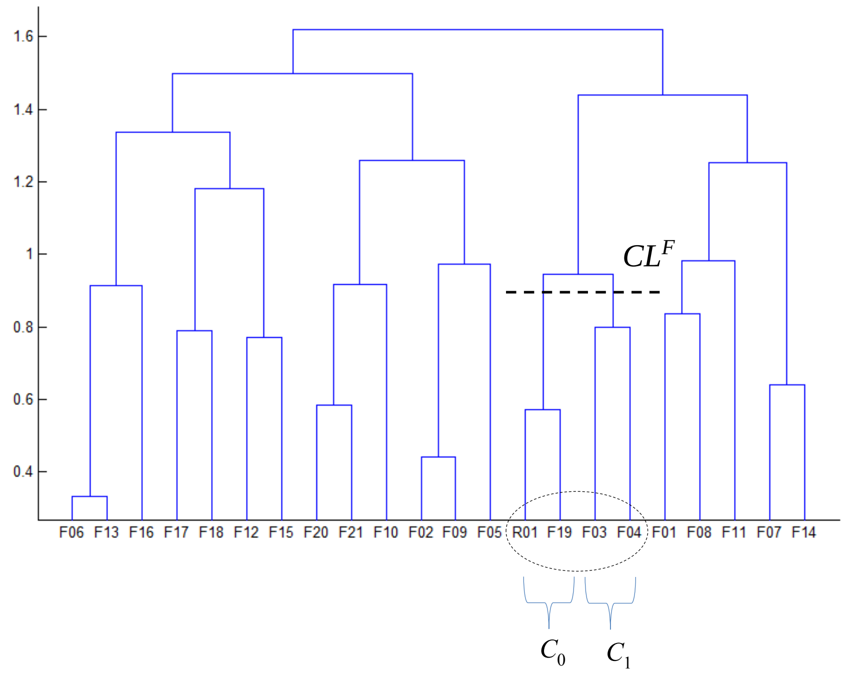



Figure 4.8: The resultant dendrogram for the bumper redesign

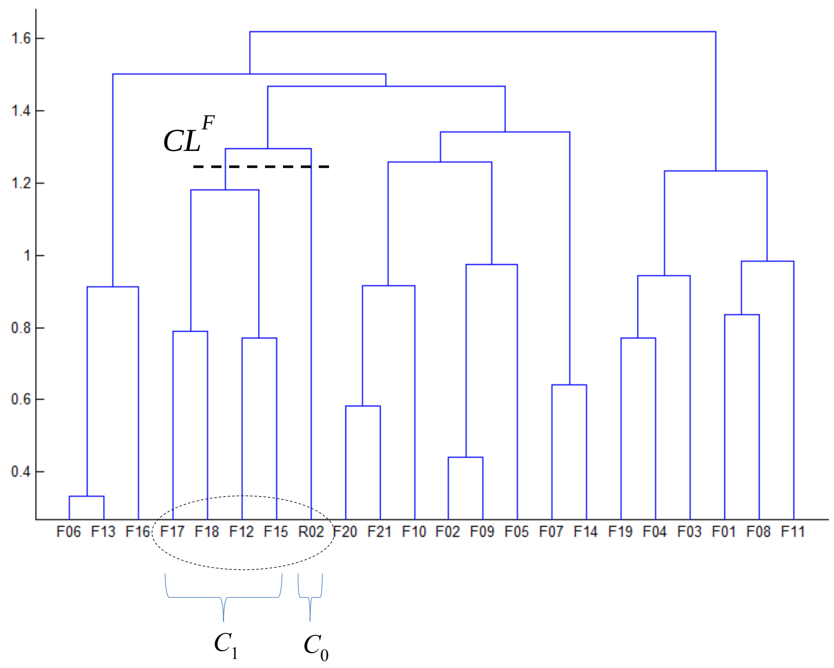



Figure 4.9: The resultant dendrogram for the wheel redesign

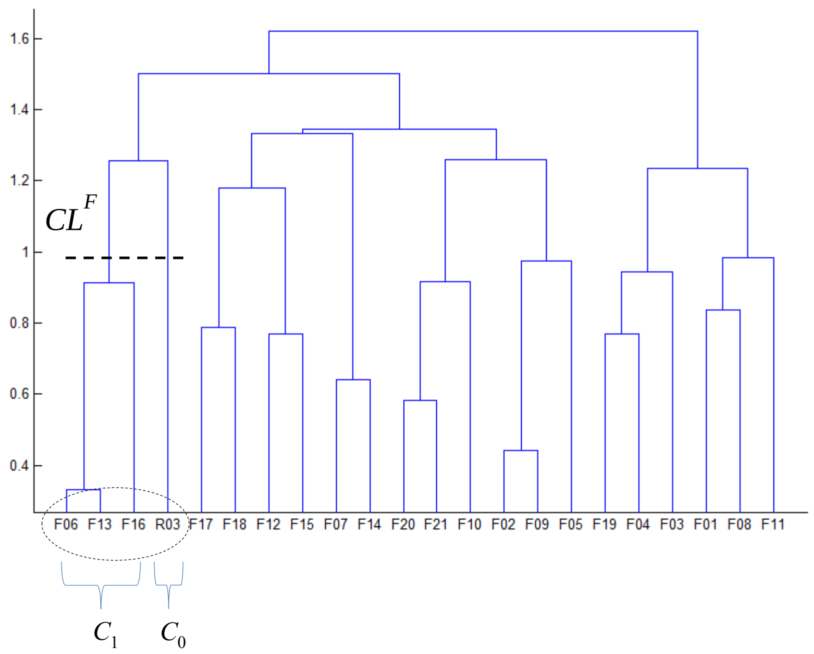



Figure 4.10: The resultant dendrogram for the suspension arm redesign

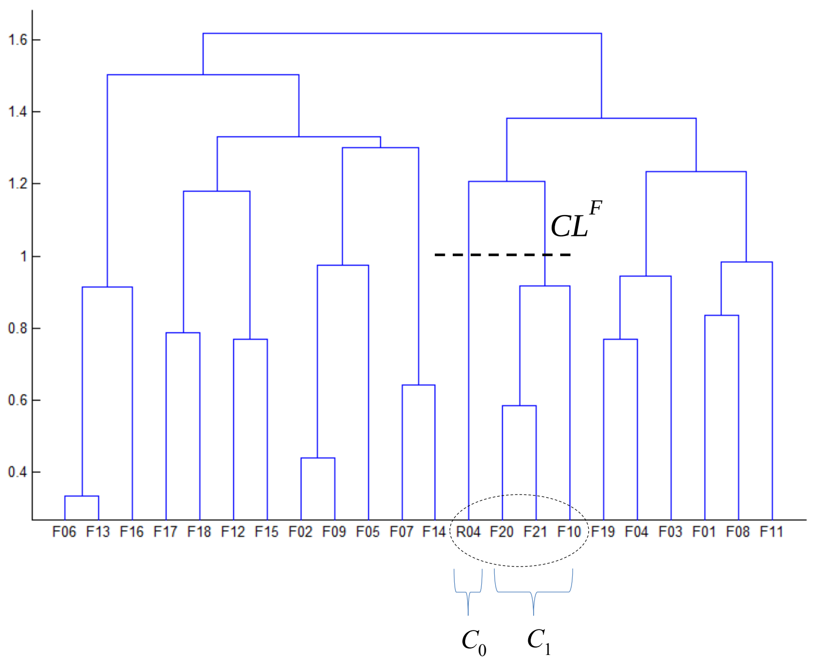



Figure 4.11: The resultant dendrogram for the driveshaft redesign

Based on the hybrid machine learning approach, AM design features were selected automatically for each target component, as listed in Table 4.8. The proposed design feature recommendation method is intended to provide conceptual design solutions to designers, especially inexperienced ones who are new to AM techniques. Therefore, in order to test the feasibility of the proposed method, in this case study, the calculation result of the computational method was supplied to a group of 4 novice designers (senior year students), each of which was given the task to design one of the 4 target components. In other words, the designers could obtain the recommended features without the necessity to manually explore and evaluate all the candidate features. These designers then determined the final AM design features (highlighted in bold) among those listed in Table 4.8 to be implemented in the real component design.

Table 4.8: Selected AM design features for target components

| **Target components** | **AM design features** |
|---|---|
| Bumper (***R****01*) | ***Honeycomb structure (F03)***, porous structure (***F****04*), internal ribbing (***F****19*) |
| Wheel (***R****02*) | Thin-wall beam (***F****17*), ***auxetic array (F18)***, spike (***F****12*), ***threaded surface (F15)*** |
| Suspension arm (***R****03*) | ***Topology optimized structure (F06)***, perforated panel (***F****13*), ***snap-fit hinge (F16)*** |
| Driveshaft (***R****04*) | ***External ribbing (F20)***, ***weave structure (F21)***, ***lattice structure (F10)*** |

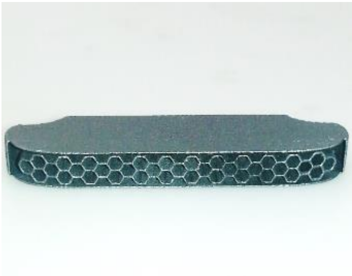
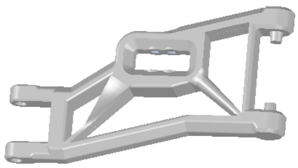
Computer-Aided Design (CAD) models of four target components redesigned with AM design features are shown in Table 4.9. The target components implemented with AM design features were manufactured by the SLM250HL machine (SLM Solutions GmbH) using AlSi10Mg alloy. The additive manufactured prototypes are also shown in Table 4.9. The prototypes were fabricated only to demonstrate the feasibility (or "manufacturability") of the designs in AM. However, the surface finish and accuracy of the additive manufactured prototypes were not sufficiently good for them to be assembled in real cars, which is a limitation in this work. Due to resource constraints, overcoming post-processing challenges was beyond the scope of this thesis. Development in post-processing techniques for surface finish and accuracy improvement will be part of the future work as discussed in Section 7.3.

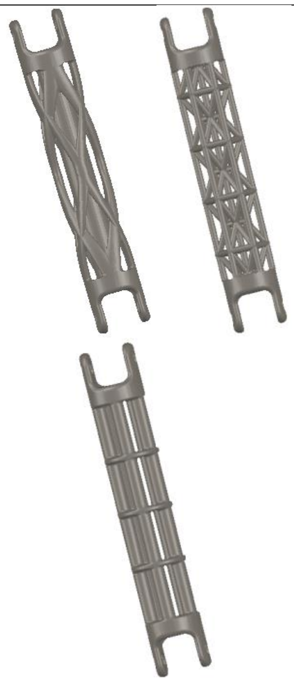
As demonstrated in the case study, the proposed hybrid machine learning approach is able to help inexperienced designers explore AM-enable design capabilities by recommending AM design features. Since the manual AM design feature screening process is replaced by the proposed computational method, designers can explore a large quantity of AM design features with less effort. When new AM design features and new industrial application examples are made available in the future, they can be used to update the AM design feature recommendation process by enlarging the pool of candidate AM design features and tuning the SVM classifier.

This chapter presents an AM design feature recommendation method that has been illustrated in the case study involving real component design. The proposed method is based on a computational algorithm that selects AM design features from the database in an automatic manner, which has not been done in previous research. The research of Maidin et al. [40, 107] can be used as the benchmarking work for comparison. In the above mentioned research, a database of AM design features was constructed for designers' reference. The AM design features in the database were categorized in terms of their major applications, so that the designers can narrow down their scope of searching. However, the

AM design feature selection task was still done empirically and manually by human designers who needed to browse through the lists of features. In comparison, as shown in this case study, the proposed computational method enables designers to quickly identify suitable AM design features without manually going through the database.

Table 4.9: Target components redesigned with AM design features

| Target components | New design with AM design features | Additive manufactured prototypes |
| --- | --- | --- |
| Bumper (***R****01*) |  (***F****03*) |  |
| Wheel (***R****02*) |  (***F****18*, ***F****15*) |  |
| Suspension arm (***R****03*) |  (***F****06*, ***F****16*) |  |

| | | |
|---|---|---|
| Driveshaft (***R****04*) |  (***F****21*) (***F****10*) (***F****20*) |  |

## 4.3 Summary

This chapter introduced a hybrid machine learning approach for AM design feature selection in the conceptual design stage. Design knowledge, including that of candidate AM design features and target components, was represented by binary codes. Hierarchical clustering was implemented to group similar AM design features and target components. The SVM classifier was trained using existing industrial application examples to iteratively update the dendrogram cut-off threshold, which defined final clusters that indicated AM design feature selection results and possible combinations of different AM design features. The proposed method was applied to design variable product platforms.

The proposed AM design feature selection method has presented the capability to organize and utilize design knowledge for the purpose of helping designers explore AM-enabled design space systematically. The proposed hybrid learning approach can identify reveal similarities between AM design features and learn from previous design examples computationally. The proposed approach provides designers (especially novices in AM) with valid conceptual design solutions.

However, the proposed AM design feature selection method does not evaluate the effect of cost, which is an important factor in planning platform strategies. In Chapter 5, the production cost of additive manufactured variable platforms will be investigated and incorporated into design parameter constraints.

# Chapter 5 AM Process Costing for Additive Manufactured Variable Platform Designs

Challenges exist in developing new product design methodologies to utilize AM-enabled design freedoms while limiting costs and resource consumptions at the same time. Although the design of an additive manufactured part can be adjusted without tooling or fixturing changes as those in traditional processes, large deviations from the reference part design may still introduce undesirably high cost increment due to significant adjustments of AM process settings or even additional trial-and-error runs to validate the new settings. Therefore, in this research, we introduce the concept of an additive manufactured variable product platform and its associated process setting platform, where all design and process setting adjustments based on the reference part are constrained within a bounded feasible space in order to limit cost increments. The objective of the research presented in Chapter 5 is to develop a design methodology for exploring AM-enabled design freedoms within a product family, while at the same time suppressing costs introduced by the freedoms. To achieve the objective, we need to find the feasible space of allowable AM process setting adjustment values, and then map the feasible space to platform module's design constraints. Platform modules implemented with AM design features can then be optimized to improve performance and customer perceived utilities in different market segments. In this chapter, the author proposes the concepts of AM process setting platform and variant. The increment of AM production cost is estimated based on AM process setting adjustments using a Fuzzy Time-Driven Activity-Based Costing (FTDABC) approach, with uncertainties expressed in fuzzy numbers. Time equations in the FTDABC are computed using an Adaptive Neuro-Fuzzy Inference System (ANFIS). The boundary of the process setting adjustment's feasible space is searched by solving a multiobjective optimization problem that maximizes design freedoms and minimizes cost increments. The extreme values of product platforms' design parameters are computed in a Mamdani model based on fuzzy design rules. Finally, the identified design constraints are used in the variable platform design optimization process to maximize part-worth customer utilities which are functions of performance indicators. Case studies on designing an R/C racing car family are used to demonstrate the proposed methodology. In contrast to the performance loss due to commonality in conventional product families, this research shows that the implementation of additive manufactured variable platforms for a product family is able to improve individual products' performance in a diversified market at controlled costs.

Chapter 5 is organized as follows. AM process settings in variable product platforms and their general relationship with AM production cost are discussed in Section 5.1. Section 5.2 describes the proposed methodologies to explore AM-enabled design freedoms within a product family and solve a variable platform design optimization to maximize customer-perceived utilities in different market segments. A case study is presented in Section 5.3 to illustrate the proposed methodology. Section 5.4 summarizes the research presented in Chapter 5.

## 5.1 AM Process Setting Platform and Variants – Fundamental Concepts

AM process settings for a variable product platform are allowed to be varied in order to realize differentiated design feature characteristics in multiple platform variants. Therefore, a set of AM process settings is defined as a *process setting platform* (*PcP*) in correspondence with a *variable product platform* in this research. For each individual product platform variant, a *process setting variant* (*PcV*) is to be identified. However, process settings' variation must be constrained in a reasonably small range, defined as a *feasible space* (*FS*), so that the production cost increment will remain low. The same machine, material, pre-process, operation sequence, and similar but non-identical data preparation settings and process parameters are used to fabricate all product platform variants in the family. By constraining the design space of AM, some of the customization benefits are sacrificed for the purpose of constraining or limiting AM production costs due to platform design variations. In this research, the AM process is treated as a known priori that is used to produce all platform variant modules in a family. Applying multiple AM processes in producing the same platform will significantly increase costs, which violates the principle objective of product family design.

In a more detailed definition, a *PcP* is the set of AM process's "properties" whose values are either common or treated as references; while the *FS* is the union of the *PcP* and allowable setting adjustment ranges. For an additive manufactured variable product platform module consisting of *N* possible platform variants, AM process settings of the reference platform design $\boldsymbol{X}_R$ is taken as the *PcP*. For the rest of the ($N$-$1$) variants, process settings *PcP* may need to be adjusted based on *PcP* to adapt to different platform variant design $\boldsymbol{X}_{PVi}$, such as fine tuning the build orientation and re-planning the support pattern. However, these adjustment operations may introduce extra cost increment due to labor, resource, and time consumptions. If the adjustment is too large, several try-and-error runs may be required to confirm the manufacturability, resulting in high cost increment, which needs to be avoided. The *increased* amount of cost (*CostInc*) due to process settings *adjustment* is a function of *PcV*, formulated as:

$$CostInc_{PcV} = CostInc\left(PcV\right) \tag{5.1}$$

$$PcV = PcP + \left\{\Delta Setting\right\} \tag{5.2}$$

$$CostInc\left(PcV\right) \leq \varepsilon \quad (\varepsilon \geq 0) \tag{5.3}$$

where $\{\Delta Setting\}$ represents AM process setting adjustment terms based on *PcP*; and $\varepsilon$ is the maximum allowable value of cost increment specified by the designer.

This research explores AM process settings feasible space ($FS = \{\Delta Setting\}_{allowable}$) in a product family, within which the setting adjustment values are constrained to limit the extra cost. The identified feasible space's boundary is then mapped to variable platform module's design parameters in a later step, which is used in the design optimization process for variable platform modules.

In this research, the author focuses on estimating cost increments based on AM process settings. It is assumed that different platform variant designs belonging to the same platform type share the same assembly interface or joints, the same material, and similar overall dimensions, and hence the assembly cost will be assumed to be the same for all platform variant modules. Therefore, the assembly costs will not be involved in AM production cost calculation, while only costs of producing individual parts are estimated.

A relevant work was done by Williams et al. [124], who used the process parameter platform method in the management of the production capacity of a 3D printer. More specifically, the production capacity was controlled by manipulating the batch size, process parameters, and machine type. In the above work, process parameters were set with the same values for all product variants in a family, and they were changed only when the 3D printer's capacity needs to change. However, in this research, the author considers the situation where designers may need to vary the process parameters for different product variants whose design variables may be different from each other, while the production cost may also change with the process parameters. Furthermore, this research takes uncertainties into consideration during cost estimation, which was not done in the previous work of Williams et al. [124].

## 5.2 AM Process Setting Feasible Space Exploration

The overall procedure of the proposed design methodology with corresponding data flow is summarized in the flowchart as shown in Figure 5.1. Sections 5.2.1 – 5.2.4 will discuss each step in details. A case study is illustrated in Section 5.3 to demonstrate the application of the methodology.

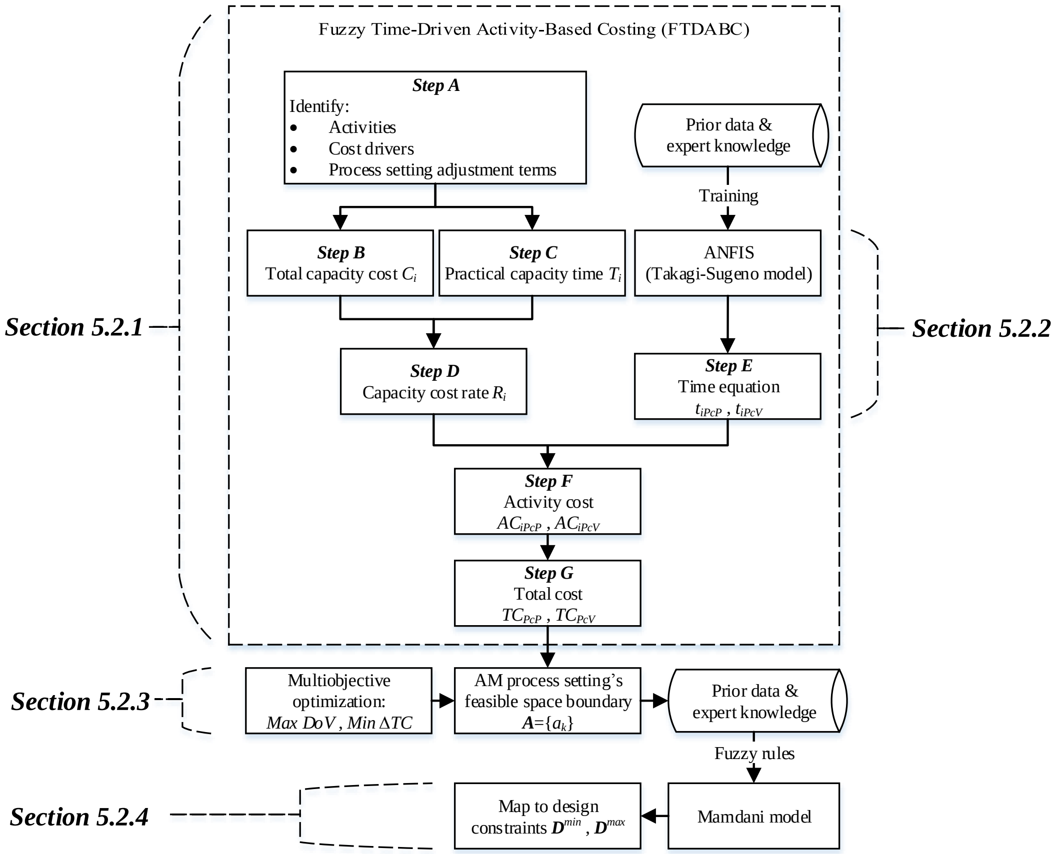



Figure 5.1: Overview of the proposed methodology and data flow

### 5.2.1 AM Process Settings Platform Variant Cost Estimation using the FTDABC

The general procedure of the standard Time-Driven Activity-Based Costing (TDABC) was first invented by Kaplan and Anderson [125] to calculate the organizational cost in various activities. In this research, A Fuzzy Time-Driven Activity-Based Costing (FTDABC) method is developed based on the standard TDABC to estimate AM production cost drivers in terms of time equations. Due to its good adaptability to any variations within activities, the proposed FTDABC is an appropriate tool to explore the relationship between cost increments and AM process setting adjustments. In literature, the fuzzy set theory has been used to model uncertainties of the total capacity cost and practical capacity time in the work of Chansaad et al. [126] and Sarokolaei et al. [127] respectively. In the proposed FTDABC, in addition to the above two types of quantities, fuzziness is also applied in formulating the uncertain time equation which is the major variable in determining total costs [125].

The detailed procedures (Steps A to G) of the proposed FTDABC in AM production are explained below. Some terminologies and notations are adopted from the standard TDABC approach [125, 126].

**Step A. Identify activities, cost drivers, and corresponding process setting adjustment terms in AM production**

As indicated in Figure 5.1, the first step of the FTDABC method is to identify activities, cost drivers, and process setting adjustment terms between the *PcP* and a *PcV* in AM production. Taking the SLM process as an example, two main activities in the production of variable platforms along with their corresponding cost drivers and process setting adjustment terms are shown in Figure 5.2. Each activity is explained as follows.

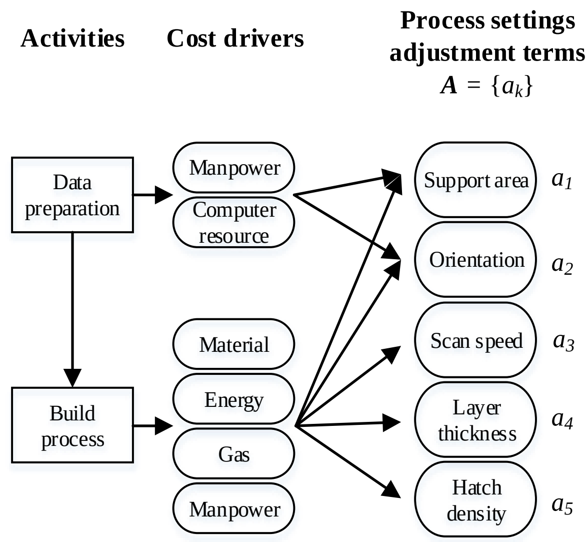



Figure 5.2: Activities, cost drivers, and process setting adjustment terms in AM (the SLM)

***Activity 1: Data preparation***. Prior to the fabrication, digital models of parts are processed either manually or automatically in software. Whenever a design is changed, the support pattern, build orientation, and part position on the build platform may need to be re-planned. The manpower and computer resource utilization required to make such modifications constitute the cost drivers in Activity 1.

***Activity 2: Build process***. During the fabrication, AM process parameters may be fine-tuned to meet geometry or property requirements in different platform variant modules. Changes in energy, gas, and material consumption as well as manpower requirements constitute the cost drivers in Activity 2.

It is noted that post-processing is necessary for most of the metal parts fabricated by AM. However, some post-processing activities, such as support material removal and surface grinding, needs to be done manually by human operators. In practice, the time and cost related to such manual work depend largely on the experience and skillfulness of the human operators, and they are hard or impractical to model as functions of the part's design. For those post-processing work that can be done by machines, such as electrochemical polishing and heat treatment, their process, time, and hence costs are determined by the materials instead of the part's design. In other words, the costs of electrochemical polishing and heat treatment are more likely to be constants instead of variables. Therefore, the costs of post-processing activities are not included in the proposed cost model, which is a limitation of the current work and may be addressed separately in future research.

As shown in Figure 5.2, the vector $\boldsymbol{A}=\{a_k\}=\{\Delta Setting_k\}$ is used to represent the five AM process setting adjustment terms shown in Fig. 2 where the notations $a_1$, $a_2$, …, $a_5$ are placed beside their corresponding terms. All $a_k$ in $\boldsymbol{A}$ are normalized real numbers, i.e. $|a_k| \in [0,1]$, while $a_k = 1$ indicates that the largest possible adjustment has been made in the $k^{th}$ AM process setting when a $PcV$ is derived from the $PcP$, and $a_k = 0$ indicates that no adjustment has been made. Detailed definition of each AM process setting adjustment term $a_k$ is presented as follows.

1) Support area adjustment $a_1$:

Platform module design variations may require support area adjustments. For example, when an overhang portion is enlarged in a particular platform variant, the support area also needs to be increased accordingly, and vice versa. Therefore, $a_1$ can be calculated as:

$$a_1 = \frac{SA_{PcV} - SA_{PcP}}{SA_{PcP\max} - SA_{PcP}} \tag{5.4}$$

where $SA_{PcP}$ and $SA_{PcV}$ represent support areas of the *PcP* and a derived *PcV* respectively. The maximum possible support area adjustment value $(SA_{PcP\max} - SA_{PcP})$ is obtained when the largest projection area of the *PcV* on the build substrate, i.e., $SA_{PcV\max}$, is completely supported during the AM fabrication process.

2) Orientation adjustment $a_2$:

Platform design variations may require adjustments in part orientation, in order to reduce support materials and improve surface smoothness [128]. It is noted that rotations about the *Z* axis (i.e. the axis parallel to the build direction) will not affect the build time and quality. Therefore, the orientation adjustment term $a_2$ is a function of rotations $\Delta\beta_X$ and $\Delta\beta_Y$ about the *X* and *Y* axis respectively, expressed as:

$$a_2 = \sqrt{\frac{1}{2}\left(\frac{\Delta\beta_X^{(PcV-PcP)}}{\pi}\right)^2 + \frac{1}{2}\left(\frac{\Delta\beta_Y^{(PcV-PcP)}}{\pi}\right)^2} \tag{5.5}$$

where the $(PcV - PcP)$ notation indicates that the orientation adjustment is made from the *PcP* to *PcV*. When both $\Delta\beta_X$ and $\Delta\beta_Y$ take the largest possible orientation adjustment value π, $a_2$ has its maximum value of 1.

3) Scan speed adjustment $a_3$:

In the SLM process, scan speed can be adjusted in order to change the build time, surface roughness, and part properties based on a platform variant's design requirements [21]. The scan speed adjustment $a_3$ is expressed as:

$$a_3 = \Delta V / \left|\Delta V\right|_{max} = \left(V_{PcV} - V_{PcP}\right) / \left|\Delta V\right|_{max} \tag{5.6}$$

where $V_{PcP}$ and $V_{PcV}$ are the scan speed values of the *PcP* and a derived *PcV*, respectively. $\left|\Delta V\right|_{max}$ is the maximum possible adjustment range, beyond which the as-built material may have inferior properties [23].

4) Layer thickness adjustment $a_4$:

Similar to the scan speed, the layer thickness in the AM can also be adjusted to change build time and part quality based on platform variant's design requirements. The layer thickness adjustment $a_4$ is expressed as:

$$a_4 = \Delta T / \left|\Delta T\right|_{max} = \left(T_{PcV} - T_{PcP}\right) / \left|\Delta T\right|_{max} \tag{5.7}$$

where $T_{PcP}$ and $T_{PcV}$ are the layer thickness values of the *PcP* and a derived *PcV* respectively. $\left|\Delta T\right|_{max}$ is the maximum possible adjustment range, beyond which the build process may fail due to too thick or too thin layer thickness.

5) Hatch density adjustment $a_5$:

Hatch density can be adjusted to create different weight and strength of platform variant modules. The hatch density adjustment $a_5$ is expressed as:

$$a_5 = \Delta H = H_{PcV} - H_{PcP} \tag{5.8}$$

where $H_{PcP}$ and $H_{PcV}$ are the hatch density values of the *PcP* and a derived *PcV* respectively. Since both $H_{PcP}$ and $H_{PcV}$ range from 0 to 100%, the $a_5$ value is also in the scale of [0, 1].

**Step B. Obtain the fuzzy total capacity cost**

For each activity, the total capacity cost per unit part within a period (e.g. one month) is recorded. A cost value usually varies over time (for instance, the February value being different from the January value) due to unpredictable fluctuations in company operations. Therefore, the total capacity cost can be modeled by a fuzzy set [126]. In this research, a triangular fuzzy number $C_i$ = [$C_{iS}$, $C_{iM}$, $C_{iL}$] is used for Activity $i$, as formulated in Eq. (5.9). The smallest value $C_{iS}$, the most probable value $C_{iM}$, and the largest value $C_{iL}$ are obtained from the company's historical data. The unit is a dollar. As indicated in Fig. X, the total capacity cost $C_i$ is a basic term required to calculate the capacity time rate $R_i$.

$$\begin{aligned} & C_i = \left[ C_{iS}, C_{iM}, C_{iL} \right] \\ & where\ i = 1\ and\ \ 2\ for\ Activity 1\ and\ \ 2,\ respectively \end{aligned} \tag{5.9}$$

**Step C. Obtain the fuzzy practical capacity time in each activity**

Similar to the total resource cost, the practical capacity time per unit part in Activity $i$ within a period can also be modeled in triangular fuzzy numbers $T_i$ = [$T_{iS}$, $T_{iM}$, $T_{iL}$] recorded in given historical data, as formulated in Eq. (5.10). The unit is an hour. As indicated in Fig. X, the practical capacity time $T_i$ is another basic term required to calculate the capacity time rate $R_i$.

$$\begin{aligned} & T_i = \left[ T_{iS}, T_{iM}, T_{iL} \right] \\ & where\ i = 1\ and\ \ 2\ for\ Activity 1\ and\ \ 2,\ respectively \end{aligned} \tag{5.10}$$

If a company is starting out an AM program, AM fabrication test runs need to be conducted in order to initialize [$C_{iS}$, $C_{iM}$, $C_{iL}$] and [$T_{iS}$, $T_{iM}$, $T_{iL}$] values. When more data are accumulated over time in the company's operation, [$C_{iS}$, $C_{iM}$, $C_{iL}$] and [$T_{iS}$, $T_{iM}$, $T_{iL}$] values can be updated. Such historical data accumulation is a common process in most corporate costing approaches [125].

**Step D. Calculate the fuzzy capacity cost rate**

The α-cut fuzzy arithmetic (division operation) [129] is conducted to calculate the fuzzy capacity cost rate $R_i$ for Activity $i$. Mathematically, $R_i$ is the ratio of the total capacity cost to the practical capacity time, as formulated in Eq. (5.11).

$$R_i = \mu(C_i / T_i)(r) = \begin{cases} \dfrac{rT_{iL} - C_{iS}}{(C_{iM} - C_{iS}) + (T_{iM} - T_{iL})r}, & C_{iS} / T_{iL} \le r \le C_{iM} / T_{iM} \\ \dfrac{C_{iL} - T_{iS}r}{(C_{iL} - C_{iM}) + (T_{iM} - T_{iS})r}, & C_{iM} / T_{iM} \le r \le C_{iL} / T_{iS} \end{cases} \tag{5.11}$$
$$where\ i = 1\ and\ 2\ for\ Activity 1\ and\ 2,\ respectively$$

where the operator $\mu(A \otimes B)$ represents a α-cut fuzzy arithmetic ($\otimes$ can be $+, -, \times$, or $\div$) between two fuzzy numbers $A$ and $B$. This notation will be used in the rest of this chapter.

The fuzzy capacity cost rate $R_i$ is considered as the "unit cost" that can be used to calculate the total cost by multiplying it with time consumptions in each activity. The formulation of time equations is introduced in Step E.

**Step E. Formulate the fuzzy time equation**

For Activity $i$, the time consumption $t_{iPcP}$ of the $PcP$ is expressed in triangular fuzzy numbers to model the uncertainty, as formulated in Eq. (5.12). While the $t_{iPcP}$ can be obtained as a constant from historical data of the reference platform design, the time consumption $t_{iPcV}$ of a $PcV$ is a function of process setting adjustments. In other words, any process setting adjustment due to part design changes has the possibility to increase or decrease the activity's time consumption. Mathematically, the time equation of a $PcV$ in Activity $i$ is formulated in Eq. (5.13).

$$t_{iPcP} = \tau_i = \left[\tau_{iS}, \tau_{iM}, \tau_{iL}\right] \tag{5.12}$$

$$t_{iPcV} = \tau_i \prod_j^N \left[1+\theta\left(\boldsymbol{a}^j\right)_i\right] = \tau_i \Theta_i = \left[\tau_{iS}\Theta_i, \tau_{iM}\Theta_i, \tau_{iL}\Theta_i\right]$$
$$where\ i=1\ and\ \ 2\ for\ Activity 1\ and\ \ 2,\ respectively \tag{5.13}$$

where $\boldsymbol{a}^j$ is a set of relevant process setting adjustments that are taken as variables in Activity $i$, represented as:

$$\boldsymbol{a}^j \subseteq \boldsymbol{A} \tag{5.14}$$

where $\boldsymbol{A} = \{a_1, a_2, a_3, a_4, a_5\}$ are process setting adjustment terms defined in Step A.

In the above formula, $\theta(\boldsymbol{a}^j)_i$ is the percentage increment of time consumption due to an adjustment in a process platform variant's setting $\boldsymbol{a}^j$. Each $\theta(\boldsymbol{a}^j)_i$ and hence the resultant $\Theta_i$ is a crisp number. The ANFIS method to calculate the value of $\theta(\boldsymbol{a}^j)_i$ will be discussed separately in Section 5.2.2, as marked in Figure 5.1.

**Step F. Calculate the activity cost**

The activity cost $AC_i$ of Activity $i$ is the α-cut fuzzy product of the activity time consumption and the capacity cost rate, as expressed in Eq. (5.15):

$$AC_{iPcP} = \mu\left(t_{iPcP} \times R_i\right), \text{ for process setting platform } (PcP)$$
$$AC_{iPcV} = \mu\left(t_{iPcV} \times R_i\right), \text{ for process setting variant } (PcV) \tag{5.15}$$
$$where\ i=1\ and\ \ 2\ for\ Activity 1\ and\ \ 2,\ respectively$$

where both $AC_{iPcP}$ and $AC_{iPcV}$ are fuzzy numbers.

**Step G. Calculate and defuzzify the total cost**

As shown in Fig. 1, the last step of the FTDABC method is to calculate the fuzzy total cost $TC_f$ in AM production, which is the α-cut fuzzy summation of activity cost values in Activity 1 and 2, as formulated in Eq. (5.16). The fuzzy total cost $TC_f = TC_f(r)$ is a fuzzy membership function, and the positive real number $r \in R^+$ is the variable in the fuzzy membership function, while each $r$ value has a corresponding membership degree determined by $TC_f(r)$. Although the fuzzy number $TC_f$ represents the total cost more realistically by preserving uncertainties, it cannot be directly used to formulate the objective function in the optimization problem presented in Section 5.2.3. Therefore, a crisp number is required, and $TC_f$ is defuzzified using the centroid method [130] to obtain the final crisp value $TC$, as formulated in Eq. (5.17). The ratio of integrals in Eq. (5.17) calculates the "center of gravity" of the fuzzy membership function $TC_f$, which results in the most expected crisp total cost $TC$ regardless of the cost distribution in $TC_f$ [130].

$$\begin{aligned} TC_{f\,PcP} &= \mu\left(AC_{1PcP} + AC_{2PcP}\right), \text{ for process setting platform } (PcP) \\ TC_{f\,PcV} &= \mu\left(AC_{1PcV} + AC_{2PcV}\right), \text{ for process setting variant } (PcV) \end{aligned} \tag{5.16}$$

$$\begin{aligned} TC_{PcP} &= \frac{\int_{r_{min}}^{r_{max}} TC_{f\,PcP}(r)\,r\,dr}{\int_{r_{min}}^{r_{max}} TC_{f\,PcP}(r)\,dr}, \text{ for process setting platform } (PcP) \\ TC_{PcV} &= \frac{\int_{r_{min}}^{r_{max}} TC_{f\,PcV}(r)\,r\,dr}{\int_{r_{min}}^{r_{max}} TC_{f\,PcV}(r)\,dr}, \text{ for process setting variant } (PcV) \end{aligned} \tag{5.17}$$

It is noted that the number of product units will not affect the FTDABC model, since its basic terms, i.e. "total capacity cost" $C_i$ and "practical capacity time" $T_i$, are the cost and time *per unit part* expressed in fuzzy membership functions with estimated probabilities.

### 5.2.2 Time Equation Formulation using ANFIS

In this section, the time equation formulation method in Step E is presented in details. An ANFIS-based expert system [131] is used to predict each activity's time consumption, given any combination of process setting adjustments.

An ANFIS is created to map each $\boldsymbol{a}^j$ to $\theta(\boldsymbol{a}^j)_i$ in time equations, and hence each activity in the FTDABC may contain multiple ANFIS. A set of "IF (antecedent) – THEN (consequence)" rules established from prior data are used to initialize a first-order Takagi-Sugeno (TS) fuzzy model. A sample set of rules is shown in Eq. (5.18) for illustration. The number of rules in an ANFIS must be equal to the number of output membership functions, which is 2 in Eq. (5.18).

$$\begin{aligned}
&Input\ Variable:\ \boldsymbol{a}^j = \{a_1, a_2\} \\
&R_1 : IF\left(a_1\, is\, \lambda_{11}\, AND\, a_2\, is\, \lambda_{21}\right) \\
&\quad THEN\, \theta\left(\boldsymbol{a}^j\right)_i{}' = p_{11}a_1 + p_{12}a_2 + q_1 \\
&R_2 : IF\left(a_1\, is\, \lambda_{12}\, AND\, a_2\, is\, \lambda_{22}\right) \\
&\quad THEN\, \theta\left(\boldsymbol{a}^j\right)_i{}'' = p_{21}a_1 + p_{22}a_2 + q_2
\end{aligned} \tag{5.18}$$

In Eq. (5.18), $\lambda_{kl}$ is a linguistic term (e.g. "High", "Low" etc.) defined by an antecedent fuzzy membership function, which is a Gaussian function in this research, as formulated in Eq. (5.19). The consequent is defined by a linear equation $\boldsymbol{p}_l^T \boldsymbol{a}^j + q_l$. The model output $\theta(\boldsymbol{a}^j)_i$ is the weighted average of the output function in all rules, as formulated in Eqs. (5.20) and (5.21), where different $\lambda_{kl}$ terms can be summed and multiplied as natural exponential functions.

$$\lambda_{kl}\left(a_k, c_{kl}, \sigma_{kl}\right) = \exp\left(-\frac{\left(a_k - c_{kl}\right)^2}{2\sigma_{kl}{}^2}\right) \tag{5.19}$$

$$\theta\left(\boldsymbol{a}^j\right)_i = \sum_{l=1}^{L} \varphi_l \left(\boldsymbol{p}_l^T \boldsymbol{a}^j + q_l\right) \tag{5.20}$$

$$\varphi_l = \frac{\prod_{k=1}^{K} \lambda_{kl}}{\sum_{l=1}^{L}\left(\prod_{k=1}^{K} \lambda_{kl}\right)} \tag{5.21}$$

where ($L=2$) is the total number of rules, ($K=2$) is the number of input variables (i.e. process setting adjustment terms {*a1*, *a2*}). In the above formulations, expert knowledge/experience is applied in (1) selecting relevant AM process setting adjustment terms that are taken as input to the TS model; (2) partitioning the input domain and thereby determining the number of fuzzy membership functions; (3) initializing the fuzzy rules with guessed parameters $(\boldsymbol{p}_l^T, q_l, c_{kl}, \sigma_{kl})$ that are to be computed later.

In the ANFIS, the TS model is organized in an ANN-like network as shown in Figure 5.3, which illustrates the rules as well as calculation steps in Eqs. (5.18) – (5.21) using five levels of nodes between the input $\boldsymbol{a}^j$ and the output $\theta(\boldsymbol{a}^j)_i$ [131, 132]. Given historical data pairs $(\boldsymbol{a}^j, \theta(\boldsymbol{a}^j)_i)$ are used to train the TS model by optimizing the parameters $(\boldsymbol{p}_l^T, q_l, c_{kl}, \sigma_{kl})$. A hybrid learning algorithm (a combination of least-squares estimates and gradient-based optimization) [131] is used to update the parameters.

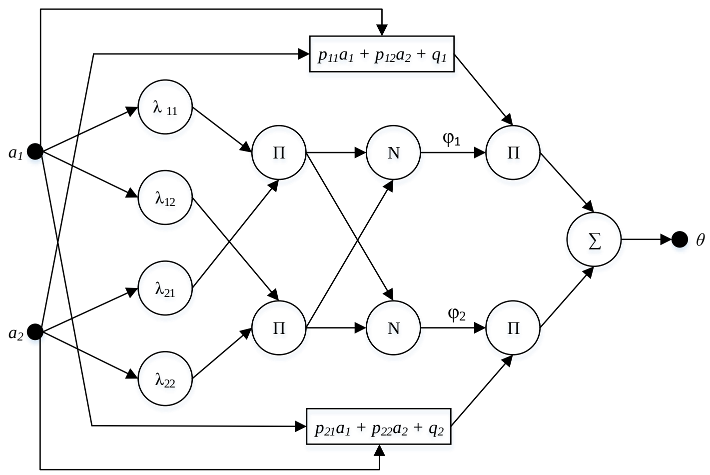



Figure 5.3: An example of ANFIS network

There are several previous papers that have introduced costing methods which enhanced the standard Time-Driven Activity-Based Costing by applying fuzzy logic [127, 133, 134]. However, compared to the existing methods, the FTDABC algorithm presented in this chapter is fundamentally different in terms of time equation formulation. In this work, the ANFIS is incorporated in the FTDABC to correlate time variations with AM process setting adjustments. The ANFIS-based time equation estimation method enables the proposed FTDABC to predict AM production costs, while existing methods in [127, 133, 134] focus more on cost measurement rather than cost prediction capability.

The performance of ANFIS is compared with other approaches in modeling the time equations. Since ANFIS is a combination of an artificial neural network (ANN) and a fuzzy rule based model (i.e. an inference system), it is usually compared with ANN in many literatures [135]. In this work, a backpropagation neural network (BP-ANN) implemented using MATLAB Neural Network Toolbox is tested for predicting time equation values. Another alternative modeling approach that is often compared with ANFIS is Gaussian process regression (GPR), which assumes that all variables follow normal distributions [136]. The ANFIS-GPR comparison can also be regarded as a comparison between fuzzy theory-based and probability theory-based modeling approaches [137]. The performances of ANFIS, ANN, and GPR are evaluated using the root mean square error (RMSE) and coefficient of determination ($R^2$) [135, 137] in Activities 1 and 2 as mentioned in the previous section. Table 5.1 lists the results which show that ANFIS has smaller RMSE and larger $R^2$ than the other two approaches in both activities, indicating a better accuracy of ANFIS in time equation modeling. The key advantage of ANFIS is that expert knowledge can be utilized in the modeling, whereas in the "black-box" type of regression or data fitting approaches like ANN and GPR, the input-output relation is completely unknown before training. As mentioned previously in this section, by using the expert knowledge in time equation modeling, only the relevant (but not all) AM process settings adjustment terms are selected as input variables, while the initial fuzzy rules and input domain partitions can be set with reasonably guessed initial parameters. Therefore, with the involvement of the expert knowledge, ANFIS shows a good accuracy.

Table 5.1: Performance comparison of ANFIS, ANN, and GPR

| | RMSE | | $R^2$ | |
|---|---|---|---|---|
| | Activity 1 | Activity 2 | Activity 1 | Activity 2 |
| ANFIS | 0.11 | 0.08 | 0.97 | 0.98 |
| ANN | 0.35 | 0.17 | 0.92 | 0.93 |
| GPR | 1.9 | 1.2 | 0.81 | 0.85 |

Compared to the conventional time-driven costing methods [125], there are several benefits of implementing the proposed ANFIS in the FTDABC. (1) Time consumptions can be estimated quickly, without the necessity to import exact CAD models into computer-aided manufacturing (CAM) software or other analyzing software [18, 27]. Therefore, a large enough number of the process setting adjustment values can be evaluated in the later stage of feasible space's boundary searching by multiobjective optimization. (2) Expert knowledge/experience and the company's historical data can be utilized to create an empirical model. When new empirical data are available in future, they can easily be used to update the existing ANFIS for improving the predicting performance. And (3) effects of uncertainties are captured in the model through the use of fuzzy membership functions.

The time-driven costing method has good adaptability to changes in activities. In other words, cost variations as a result of activity changes can be easily calculated by time-driven cost models via changing the time equation values [125]. Compared with previous AM costing approaches [29-31, 34, 138] that estimate costs based on specific part design with deterministic process settings, the proposed FTDABC method is advantageous in the following aspects: 1) it correlates AM production costs with relative AM process setting adjustments that can be treated as variables, 2) the established FTDABC and ANFIS models are independent of specific parts to be manufactured, and 3) the fuzzy analysis for uncertainties modeling makes the methodology suitable for practical operations. The FTDABC method will be used at the variable platform design stage to control and limit costs, as discussed in Sections 5.2.3 and 5.2.4.

### 5.2.3 AM Process Setting Adjustments' Feasible Space Boundary Searching

After the proposed costing model is established, the next task is to identify the feasible space boundary of the process setting adjustment values in order to constraint the cost increment, as indicated in Figure 5.1. The boundary searching procedure is formulated as a multiobjective optimization problem, whose input variable is the vector $\boldsymbol{A}$ containing all the five AM process adjustment terms defined in Section 5.2.1.

Recalling the total cost estimation method in the proposed FTDABC, we can compute the cost increment of a *PcV* as defined in Eqs. (5.1) – (5.3), in terms of process settings adjusted from those in the *PcP*:

$$CostInc = \Delta TC(\boldsymbol{A}) = TC_{PcV}(\boldsymbol{A}) - TC_{PcP} \tag{5.22}$$

In Eq. (5.22), $\Delta TC(\boldsymbol{A})$ is the objective function to be minimized in the optimization problem. It is allowed to take negative values, indicating a production cost decrement which is also possible to occur during process setting adjustments.

All variables $a_k$ in $\boldsymbol{A}$ are normalized into dimensionless real numbers, i.e. $|a_k| \in [0,1]$. A metric named "Degree of Variation (*DoV*)" is defined to measure the difference between a *PcV* and the *PcP*. The *DoV* is formulated as the Euclidean norm of the vector $\boldsymbol{A}$:

$$\begin{aligned} & DoV_{PcP \to PcV}(\boldsymbol{A}) := \|\boldsymbol{A}\|^2 = \sum_k a_k{}^2, \text{ where } k = 1,2,\ldots,5 \\ & |a_k| \in [0,1] \\ & DoV_{PcP \to PcV}(\boldsymbol{A}) \in [0,5] \end{aligned} \tag{5.23}$$

The *DoV* implies the size of the process setting's feasible space; hence it is to be maximized in the optimization problem in order to provide more freedoms in manipulating the AM production plan. The complete formation of the multiobjective optimization can be summarized as follows:

$$\begin{aligned} & \textit{Variable}: \ \boldsymbol{A} = \{a_k\} \\ & \textit{Objective functions}: \begin{cases} \min\limits_{\boldsymbol{A}} \ \Delta TC(\boldsymbol{A}) \\ \max\limits_{\boldsymbol{A}} \ DoV_{PcP \to PcV}(\boldsymbol{A}) \end{cases} \\ & \textit{Bound constraint}: |a_k| \in [0,1] \end{aligned} \tag{5.24}$$

A multiobjective GA [139] algorithm is used to find the *best-known Pareto set* of $\boldsymbol{A}$, as illustrated in Figure 5.4 where each point on the plot represents an instance of $\boldsymbol{A}$. GA is used due to the following reasons: 1) genetic mutations and crossovers can move the population away from local optima; 2) it is suitable for non-linear problem where analytical or differentiable fitness functions are not available; 3) its population-based optimization method is able to find solutions for a large number of decision variables.

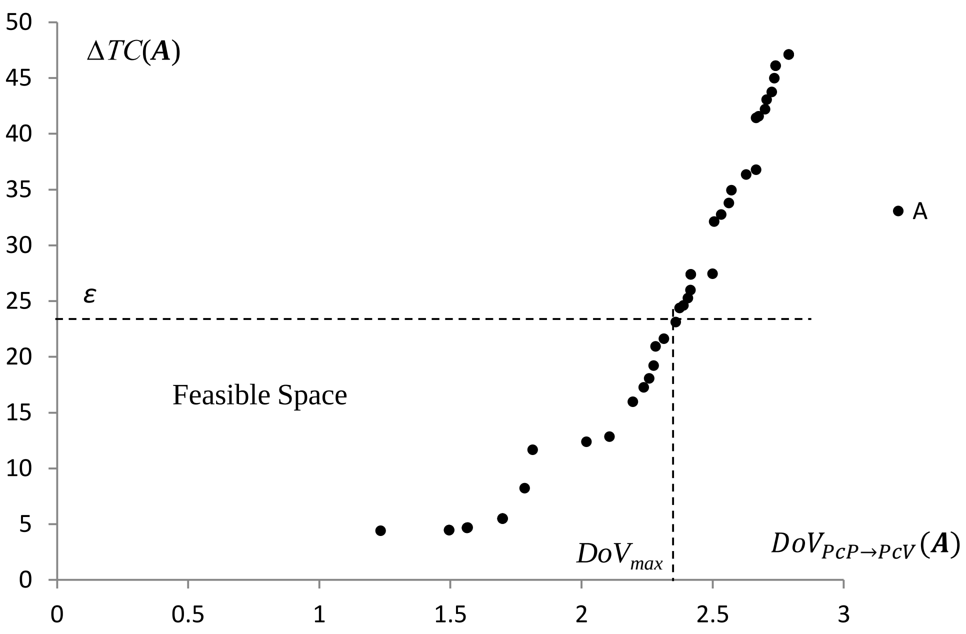



Figure 5.4: The feasible space boundary formed by the Pareto front and ε

A predetermined cost increment upper limit $\varepsilon$ is used to contain the feasible space for process setting adjustments. The Pareto front below the horizontal dash line $\Delta TC(\boldsymbol{A}) = \varepsilon$ defines the feasible space boundary. The $\varepsilon$ value can be changed, with the horizontal dash line shifting along the vertical axis, to define different feasible spaces. The $\varepsilon$ value also defines the maximum allowable $DoV_{max}$ (2.3 in Figure 5.4). Any process setting adjustment that has a *DoV* value larger than the $DoV_{max}$ is not allowed (infeasible) due to unacceptably high cost increment.

### 5.2.4 Mapping AM Process Settings to Variable Platform Design Constraints

In Section 5.2.4, the method of mapping from AM process settings to variable platform design constraints is introduced.

Recalling the product family's mathematical representation in Section 3.2, the $j^{th}$ platform variant $\boldsymbol{X}_{PV,ij}$ in the $i^{th}$ product variant of the family is defined as a vector of $n$ design parameters, each of which is a scalar $x_p^k$ (e.g. size, density, material, etc.). $\boldsymbol{X}_{PV,ij}$ is calculated as the product of the *platform design parameter variation matrix* $\boldsymbol{D}_{ij}$ and the *reference platform design parameter vector* $\boldsymbol{X}_{R,j}$, as formulated in Eq. (5.25). $\boldsymbol{D}_{ij}$ is a $n \times n$ diagonal matrix with its diagonal elements $d_{kk}$ being dimensionless positive real numbers. $\boldsymbol{X}_{R,j}$ represents an initial or currently existing design of the variable platform module, which is taken as the reference based on which all future new $\boldsymbol{X}_{PVij}$ designs are derived. Therefore, diagonal elements $d_{kk}$ in $\boldsymbol{D}_{ij}$ can be treated as the scale factor controlling the variation of each design parameter across different variants in the family.

$$\boldsymbol{X}_{PV,ij} = \boldsymbol{D}_{ij}\boldsymbol{X}_{R,j} \Rightarrow \begin{Bmatrix} x_p^1 \\ x_p^2 \\ \vdots \\ x_p^n \end{Bmatrix}_j = \begin{pmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{pmatrix}_j \begin{Bmatrix} x^1 \\ x^2 \\ \vdots \\ x^n \end{Bmatrix}_j \qquad (5.25)$$
$$where\ i = 1, 2, \ldots, N$$

Platform modules' design parameters are subject to constraints imposed by the production process. Such platform design constraints are expressed in an inequality equation shown in Eq. (5.26), where $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ contain the minimum and maximum allowable values of diagonal elements $d_{kk}$ in $\boldsymbol{D}_{ij}$, respectively.

$$\boldsymbol{D}_{ij}^{min} \leq \boldsymbol{D}_{ij} \leq \boldsymbol{D}_{ij}^{max} \qquad (5.26)$$

Although general relationships between AM process settings and certain product design constraints have been experimentally investigated in previous research [53, 56-58], high fidelity mathematical models have not been established. In practice, design rules in correspondence to AM process setting adjustments are created mainly empirically based on designers experience and knowledge. Therefore, in this research, a knowledge-based expert system is used to establish the mapping from the process setting adjustments' feasible space boundary to product platform design constraints. A Mamdani-type fuzzy inference system, or a Mamdani model [132], is applied to create the expert system. IF-THEN design rules with linguistic antecedent and consequent terms expressed in fuzzy membership functions are formulated in the Mamdani model to represent expert knowledge and experience. Major advantages of a Mamdani model are: (1) design rules can be easily implemented and updated, (2) the model structure is independent of AM process, only design rules are specifically defined for a particular AM process, and (3) uncertainties are well incorporated in the model by the use of fuzzy membership functions.

The proposed Mamdani model takes a point $\boldsymbol{A}^{*} = \{a_k\}^{*}$ on the identified boundary as the input, and the diagonal elements in $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ as outputs. The selection of $\boldsymbol{A}^{*}$ is performed by human designers, who determine the maximum tolerable cost increments.

AM design features are implemented in variable platform modules to utilize AM advantages and achieve superior performance [38]. A set of IF-THEN design rules is defined in the Mamdani model for *each* AM design feature fabricated by a particular AM process. As an example, design parameters and constraints of a honeycomb feature fabricated by the SLM process are expressed in Eq. (5.27). This feature can be used in a variable platform module subjected to design variations. Correspondingly, two example design rules $R_1$ and $R_2$ for a honeycomb feature are listed in Eq. (5.28) for illustration.

$$\boldsymbol{X}_{C,j} = \begin{Bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{Bmatrix} = \begin{Bmatrix} bounding\,box\,length \\ bounding\,box\,height \\ cell\,size \\ wall\,thickness \end{Bmatrix} = \begin{Bmatrix} B \\ H \\ C \\ T \end{Bmatrix}$$

$$\boldsymbol{D}_{i,j} = \begin{pmatrix} d_B & 0 & 0 & 0 \\ 0 & d_H & 0 & 0 \\ 0 & 0 & d_C & 0 \\ 0 & 0 & 0 & d_T \end{pmatrix}$$

$$\boldsymbol{D}_{i,j}^{min} = \begin{pmatrix} d_B{}^{min} & 0 & 0 & 0 \\ 0 & d_H{}^{min} & 0 & 0 \\ 0 & 0 & d_C{}^{min} & 0 \\ 0 & 0 & 0 & d_T{}^{min} \end{pmatrix}$$

$$\boldsymbol{D}_{i,j}^{max} = \begin{pmatrix} d_B{}^{max} & 0 & 0 & 0 \\ 0 & d_H{}^{max} & 0 & 0 \\ 0 & 0 & d_C{}^{max} & 0 \\ 0 & 0 & 0 & d_T{}^{max} \end{pmatrix} \tag{5.27}$$

$$\begin{aligned} & Input\ Variable:\ \boldsymbol{A} = \{a_k\} = \{\Delta Setting\} \\ & where\ k = 1,2,\ldots,5\ for\ five\ process\ setting\ adjustment\ terms \\ & R_1:\ IF\ (a_1\,is\,High) \\ & \quad THEN \begin{cases} d_L{}^{min} is\,Small; d_L{}^{max}\,is\,Large \\ d_H{}^{min} is\,Small; d_H{}^{max}\,is\,Large \\ d_C{}^{min} is\,Small; d_C{}^{max}\,is\,Large \\ d_T{}^{min} is\,Small; d_T{}^{max}\,is\,Large \end{cases} \\ & R_2:\ IF\ (a_3\,is\,Low \cup a_4\,is\,Low) \\ & \quad THEN \begin{cases} d_L{}^{max}\,is\,Small \\ d_H{}^{max}\,is\,Small \\ d_T{}^{max}\,is\,Small \end{cases} \end{aligned} \tag{5.28}$$

In the above $R_1$ and $R_2$ examples, recalling the definition of the SLM process setting adjustment terms $\boldsymbol{A}$ = {$a_k$} in Section 5.2.1, $a_1$, $a_2$, $a_3$, $a_4$, and $a_5$ represent support area adjustment, orientation adjustment, scan speed adjustment, layer thickness adjustment, and hatch density adjustment in the SLM respectively. $R_1$ is formulated based on the knowledge that if the support area of the variable platforms (i.e. the honeycomb feature) has a large allowable adjustment range (i.e. $a_1$ is "High"), the honeycomb feature's overall dimensions and cell size can be increased, which results in larger overhang portions. Similarly, $R_2$ is formulated based on the knowledge that if the SLM scan speed and layer thickness are not allowed to increase largely (i.e. $a_3$ and $a_4$ are "Low"), the honeycomb structure's overall dimension cannot be increased much since longer build time will be required which introduced high gas and energy consumption in the SLM process. Other rules except for $R_1$ and $R_2$ are not presented here due to the page limit. The mathematical approach of fuzzy inference based on multiple rules in a Mamdani model can found be in literature such as [132], hence it is not explained here.

Only the most influential process adjustment terms are contained in each Mamdani design rule as inputs. Such rules are defined as "incomplete rules" which are common in fuzzy expert systems [140, 141]. However, at least one rule is fired for any possible combination of input variables [140]. The Mamdani model is implemented in MATLAB®. A case study of honeycomb-based car bumper design will be presented in Section 5.3 to further illustrate the Mamdani model.

### 5.2.5 Optimizing Variable Platform within the Design Constraints

After design constraints of variable platform modules are identified in terms of $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$, design parameters of each platform variant in a product family can be optimized to maximize the customer perceived utility. A part-worth utility function of the $i^{th}$ product variant is formulated as:

$$U_i(\boldsymbol{D}_{ij}) = \sum\nolimits_{\varphi=1}^{\Phi} \beta_{p_\varphi}[p_\varphi(\boldsymbol{D}_{ij}, \boldsymbol{X}_{R,j}, \boldsymbol{X}_U)] \tag{5.29}$$

where $\boldsymbol{X}_U$ represents unique modules of the $i^{th}$ product that are not shared by others in the family; $p_\varphi$ is the $\varphi^{th}$ performance indictor of the product determined by both platform modules and unique modules; and $\beta_{p_\varphi}$ is the utility weight assigned to each $p_\varphi$. The customer perceived utility is a measure of product performance in terms of customer preference in a particular market segment.

The platform variant design optimization problem in a particular market segment is formulated as follows.

$$\begin{aligned}
&\textit{Variable}: \ \boldsymbol{D}_{i,j} \\
&\textit{Objective function}: \ \max \ U_i(\boldsymbol{D}_{ij}) \\
&\textit{Bound constraint}: \ \boldsymbol{D}_{ij}^{\min} \le \boldsymbol{D}_{ij} \le \boldsymbol{D}_{ij}^{\max}
\end{aligned} \tag{5.30}$$

## 5.3 Case Study

A family of TRAXXAS R/C racing cars [142] is used to demonstrate the application of the proposed design methodology. In the case study, the design of an individual variable platform module of the car is analyzed to explore its feasible space of AM process setting adjustments and the constraints of design parameters.

The honeycomb structure, a frequently used AM design feature, is applied in designing the bumper which is taken as a variable platform module shared by all cars in the family. The result of AM process setting feasible space exploration and design constraints mapping is presented in the case study.

The conceptual bumper design is shown in Figure 5.5. The honeycomb structure is implemented, which is a popular AM design feature with high strength to weight ratio, and has many found applications in automotive and aerospace [38]. The key design parameters of the honeycomb-based bumper are defined in Eq. (5.27) in Section 5.2.4, i.e. $\boldsymbol{X}_{R,j} = \left[B \ H \ C \ T\right]^T$. The reference design parameters $\boldsymbol{X}_R$ of the bumper are listed in Table 5.2. These design parameter values are taken as the baseline for other platform variants in the racing car family, based on which each platform variant can be optimized in its respective market segment.

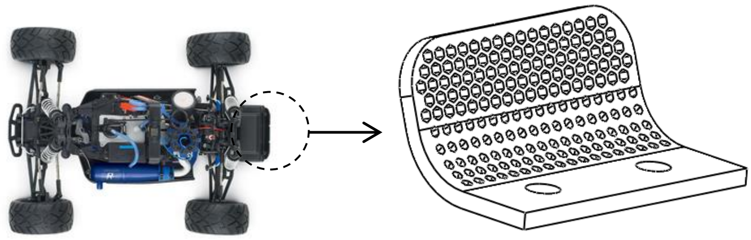


Figure 5.5: Bumper design with the honeycomb feature

Table 5.2: Reference design parameters of the bumper

| $k$ | Key design parameters | $\boldsymbol{X}_R$ |
|---|---|---|
| 1 | Bounding box length ($B$) | 55 mm |
| 2 | Bounding box height ($H$) | 20 mm |
| 3 | Cell size ($C$) | 3 mm |
| 4 | Wall thickness ($T$) | 2 mm |

The SLM process setting of this reference bumper design is treated as the *PcP*, with the values listed in Table 5.3. The initial orientation of the reference bumper in the *PcP* is considered "0", and any orientation adjustment of a *PcV* for a platform variant relative to the *PcP* will have a positive value (i.e. $a_2 > 0$), as defined in Section 5.2.1.

Table 5.3: The SLM process settings in the PcP of the reference bumper design

| Support Area ($mm^2$) | Orientation | Scan speed (mm/s) | Layer thickness (mm) | Hatch density (%) |
|---|---|---|---|---|
| 210 | 0 | 550 | 0.05 | 80 |

The FTDABC method with the ANFIS model for time equation formulation is illustrated below in Section 5.3.1. The resultant AM process setting adjustments' feasible space boundary is presented in Section 5.3.2. The variable platform (i.e. the bumper) design constraints are calculated in Section 5.3.3.

### 5.3.1 Cost Estimation Using the FTDABC Method with the ANFIS Model

At the data preparation stage (Activity 1) in Step A, the CAD software SolidWorks is used to create and modify digital models; Magics is used to prepare STL files including support structures; and AutoFab is the CAM software used to set SLM process parameters and to generate ready-to-print codes readable by the AM machine. During the build process (Activity 2), SLM 250HL manufactured by SLM Solutions GmbH will be used to produce components in aluminium alloys. Argon gas is continuously purged into the build chamber. Engineers need to monitor the build process regularly when the machine is running from the start to the end.

The ANFIS model is used to formulate time equations in the FTDABC method. In the case study, ANFIS training data in form of $(\boldsymbol{a}^j, \theta(\boldsymbol{a}^j)_i)$ were obtained from prior practices of producing parts using SLM. The process settings ($\boldsymbol{a}^j$) were set and adjusted in the CAM software (Magics® and Build Processor®). The resultant time variations values ($\theta(\boldsymbol{a}^j)_i$) in different activities (including data preparation and build process) as described in Section 5.2.1 were recorded. The result of ANFIS training is presented as follows. For illustration, a first-order output function in Activity 1 is written in Eq. (5.31); with its trained parameters and corresponding rules listed in Table 5.4.

$$\begin{aligned}
&\theta(\boldsymbol{a}^1)_1 = \theta(\{a_1, a_2\})_1 = p'a_1 + p''a_2 + q' \\
&\theta(\boldsymbol{a}^2)_1 = \theta(\{a_5\})_1 = p'''a_5 + q'' \\
&where \begin{cases} a_1 = \{\Delta \text{support area}\} \\ a_2 = \{\Delta \text{orientation}\} \\ a_5 = \{\Delta \text{hatch density}\} \end{cases}
\end{aligned} \tag{5.31}$$

Table 5.4: First-order output functions' parameters in corresponding rules

| **Antecedent (IF)** | **Output function (THEN)** | | | | |
|---|---|---|---|---|---|
| | $\theta(\boldsymbol{a}^1)_1$ | | | $\theta(\boldsymbol{a}^2)_1$ | |
| | *p'* | *p''* | *q'* | *p''* | *q''* |
| $a_1$ is Low AND $a_2$ is Low | 2.570 | 0.986 | 0.070 | | |
| $a_1$ is Medium OR $a_2$ is Medium | 1.139 | 2.041 | -0.296 | | |
| $a_1$ is High OR $a_2$ is High | 0.956 | 1.295 | -0.503 | | |
| $a_5$ is High | | | | 0.664 | 0.002 |
| $a_5$ is Medium | | | | 0.587 | -0.157 |
| $a_5$ is High | | | | 1.365 | -0.865 |

The surface generated by ANFIS is shown in Figure 5.6, which illustrates the approximated relationship between $\{a_1,a_2\}$ on the two horizontal axes and $\theta$ on the vertical axis.

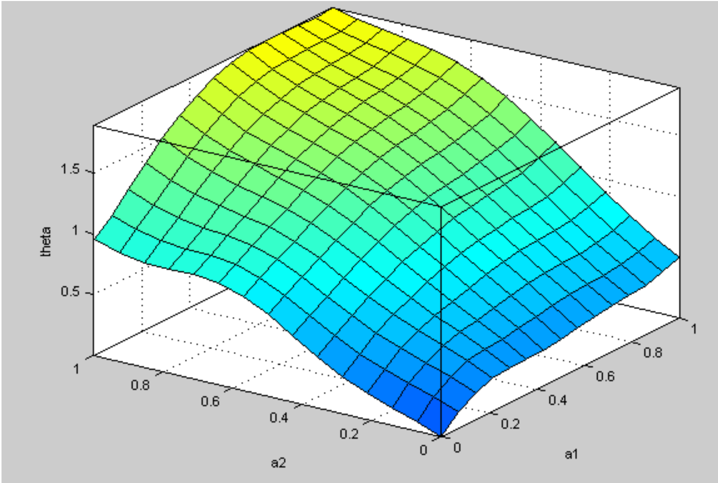



Figure 5.6: ANFIS surface of $\theta(\{a_1,a_2\})_1$

The fuzzy time consumptions in Activity 1 (Data Preparation) and Activity 2 (Build Process) of producing the reference bumper design are listed in Table 5.5. With $\tau_i$ and $\theta(\boldsymbol{a}^j)_i$ obtained above, time equations $t_{iPcV} = \tau_i \prod_j^N [1+\theta(\boldsymbol{a}^j)_i]$ can be calculated with given process setting adjustment terms $\boldsymbol{A} = \{a_k\}$.

Table 5.5: Fuzzy time consumption $\tau_i = [\tau_{iS}, \tau_{iM}, \tau_{iL}]$

| Activity ($i$) | $\tau_{iS}$ (hr) | $\tau_{iM}$ (hr) | $\tau_{iL}$ (hr) |
|---|---|---|---|
| Data preparation ($i$ = 1) | 0.6 | 1.0 | 1.2 |
| Build process ($i$ = 2) | 4.0 | 4.6 | 5.5 |

The total capacity cost $C_i$ and practical capacity rate $T_i$ in Activity 1 (Data Preparation) and Activity 2 (Build Process) of the SLM production obtained from historical data are listed in Table 5.6. The fuzzy membership functions of resultant capacity cost rate $R_i$ as the α-cut ratio of $C_i$ to $T_i$ are plotted in Figure 5.7. The cost of each activity and the resultant total cost can then be calculated using fuzzy capacity cost rate $R_i$ and time equations $t_{iPcV}$, with any process setting adjustments $\boldsymbol{A}$ as the input variable.

Table 5.6: The total capacity cost and practical capacity rate in the SLM production

| Activity ($i$) | $C_i$ (\$/unit) | | | $T_i$ (hr) | | |
|---|---|---|---|---|---|---|
| | $C_{iS}$ | $C_{iM}$ | $C_{iL}$ | $T_{iS}$ | $T_{iM}$ | $T_{iL}$ |
| Data preparation ($i$ = 1) | 50 | 80 | 90 | 0.6 | 1.4 | 1.9 |
| Build process ($i$ = 2) | 95 | 150 | 210 | 2 | 5 | 11 |

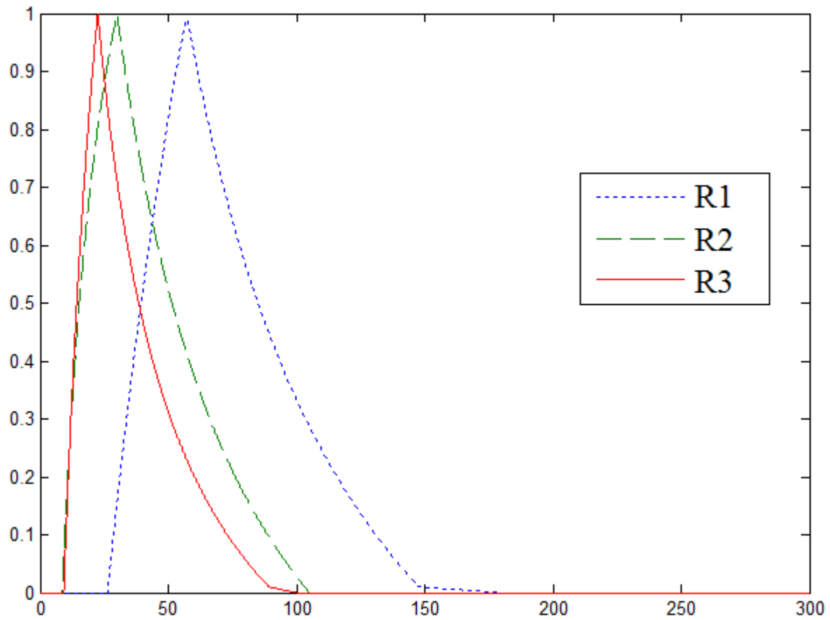



Figure 5.7: Calculated fuzzy membership functions of capacity cost rate

### 5.3.2 Process setting adjustments' feasible space boundary searching

The multiobjective optimization problem formulated in Eq. (5.24) is solved. As explained in Section 5.2.1, the solution space of the decision variables ( $\boldsymbol{A} = \{a_k\}$ ) is bounded by the constraint $|a_k| \in [0,1]$. The Pareto set computed by multiobjective GA is shown in Figure 5.8, which consists of 50 Pareto-optimal solutions indicated by the dots. The $\varepsilon$ value is set at \$30 per unit. The Pareto front bound by the horizontal dash line $\varepsilon$ = 30 forms the feasible space boundary of AM process setting adjustment terms. The maximum possible Degree of Variation obtained is $DoV_{max}$ = 2.57. It is noted that time and cost predictions are generated by the FTDABC method during the iterative solving process of the multiobjective optimization problem, where accuracy test using ground truth will not be practical since over hundreds of predicted values are calculated in each GA iteration [139].

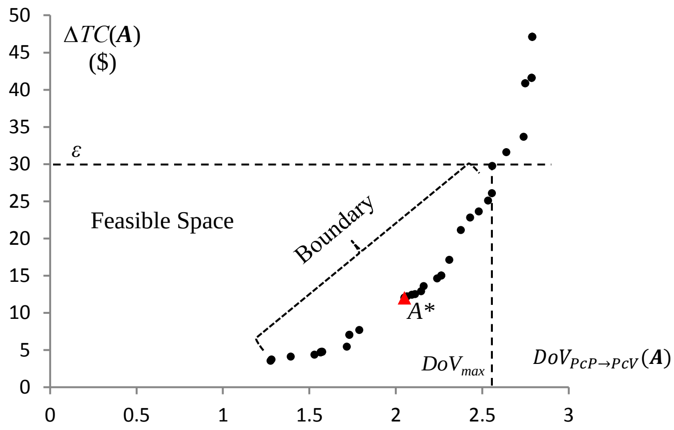



Figure 5.8: Identified feasible space with the boundary

### 5.3.3 Honeycomb features' design parameter constraints

Using the fuzzy rules formulated for honeycomb feature design mentioned in Eqs. (5.27) and (5.28) in Section 5.2.4, constraints of design parameters in terms of $\boldsymbol{D}_{i,j}^{min}$ and $\boldsymbol{D}_{i,j}^{max}$ are derived from the Mamdani model. Assuming that the company wants to cap the cost increment of producing each variable honeycomb bumper platform at \$13 per unit, a point $\boldsymbol{A}^*$ (the red triangular marker in Figure 5.8) can be selected on the process setting adjustment's feasible space boundary, so that the maximum acceptable cost increment is ($\Delta TC$= \$12.41 < \$13) and the maximum possible design freedom is ($DoV$ = 2.09). Values of $\boldsymbol{A}^*$ vector are shown in Eq. (5.32), which can be explained in terms of maximum allowable changes in a *PcV* relative to the *PcP*. For example, the first element in $\boldsymbol{A}^*$ is $a_1^*$ = 0.088, which indicates that in the SLM production of bumper variants in a product family, the support area in the *PcP* can be increased by no more than 0.088 (or 8.8%) of the largest possible support area adjustment value (1990 mm$^2$). If the support area is increased beyond the allowable limit, high extra cost may be induced.

$$\boldsymbol{A}^* = \{a_k^*\} = \{0.088, 0.870, -0.421, -0.818, 0.2\} \tag{5.32}$$

After the AM process setting adjustments' feasible space boundary has been identified above, the next step is to calculate the variable platform design constraints using the Mamdani model introduced in Section 5.2.4. $\boldsymbol{A}^*$ is input to the Mamdani model implemented with expert rules such as those in Eq. (5.28), and the design constraints in terms of $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ are computed. The results are expressed as follows.

$$
\begin{aligned}
\boldsymbol{D}_{ij}^{min} &= \begin{pmatrix} d_B{}^{\min} & 0 & 0 & 0 \\ 0 & d_H{}^{\min} & 0 & 0 \\ 0 & 0 & d_C{}^{\min} & 0 \\ 0 & 0 & 0 & d_T{}^{\min} \end{pmatrix} \begin{pmatrix} 0.54 & 0 & 0 & 0 \\ 0 & 0.65 & 0 & 0 \\ 0 & 0 & 0.52 & 0 \\ 0 & 0 & 0 & 0.58 \end{pmatrix} \\
\boldsymbol{D}_{ij}^{max} &= \begin{pmatrix} d_B{}^{\max} & 0 & 0 & 0 \\ 0 & d_H{}^{\max} & 0 & 0 \\ 0 & 0 & d_C{}^{\max} & 0 \\ 0 & 0 & 0 & d_T{}^{\max} \end{pmatrix} \begin{pmatrix} 1.52 & 0 & 0 & 0 \\ 0 & 1.55 & 0 & 0 \\ 0 & 0 & 2.05 & 0 \\ 0 & 0 & 0 & 2.57 \end{pmatrix}
\end{aligned}
\tag{5.33}
$$

$$
\begin{aligned}
\left[B^{\min}\ H^{\min}\ C^{\min}\ T^{\min}\right]^T &= \left[0.54B\ 0.65H\ 0.52C\ 0.58T\right]^T \\
\left[B^{\max}\ H^{\max}\ C^{\max}\ T^{\max}\right]^T &= \left[1.52B\ 1.55H\ 2.05C\ 2.57T\right]^T
\end{aligned}
\tag{5.34}
$$

Recalling Eq. (5.27), the diagonal elements of $\boldsymbol{D}_{i,j}^{min}$ and $\boldsymbol{D}_{i,j}^{max}$ calculated in Eq. (5.33) represent the minimum and maximum allowable scale factors applied to the platform design parameters ($\boldsymbol{X}_{R,j} = \left[B\ H\ C\ T\right]^T$) . The vector $\boldsymbol{X}_{R,j}$ with given values represents the initial or a currently existing design of the honeycomb bumper; and it can be taken as the reference based on which all variants of honeycomb bumper design in the family can be derived. Therefore, the bound constraint of any bumper variant's design parameters can be explicitly expressed in the form of Eq. (5.34). The result of the proposed methodology provides a guideline in the concurrent design of product and process, helping designers to explore design space and restrict cost increment at the same time.

Recalling the reference design parameter $\boldsymbol{X}_{R,j}$ listed in Table 5.7, the minimum and maximum bumper design parameters in platform variants can be calculated based on the result in Eq. (5.34) and the relationship $\boldsymbol{X}_{PV_i,j} = \boldsymbol{D}_{i,j}\boldsymbol{X}_{R,j}$ formulated in Eq. (5.25). The results are shown in Table 5.7.

Table 5.7: Minimum and maximum bumper design parameters in platform variants

| Design parameter | *B* (mm) | *H* (mm) | *C* (mm) | *T* (mm) |
|---|---|---|---|---|
| Minimum | 29.7 | 13.0 | 1.6 | 1.2 |
| Maximum | 83.6 | 31.0 | 6.2 | 5.1 |

The bumper design in Section 5.3 illustrates the concepts of variable product platform, process setting platform, and the proposed cost-driven design method. The same concepts and methods can be generalized to other components if their designs can be adjusted to meet different requirements and they can be fabricated by AM. The SLM process was used in this work not to simplify the problem, since it is actually more complex than many other commercial AM processes (especially plastic-based processes) in terms of the adjustability of process settings. SLM was selected because it can produce functional metallic parts and has great potential in industry. The proposed method can also be applied to other AM processes, such as electron beam melting (EBM) or fused deposition modeling (FDM) whose process settings are less flexible in commercial machines, which may make the proposed design process even simpler.

### 5.3.4 Comparing Optimized Variable Platforms and Consistent Platforms

To illustrate the process of optimizing variable platforms as described in Section 5.2.5, a family of two R/C racing cars (Table 5.8) were designed for two different market segments: “short-course race truck” and “on-road sedan”. Short-course race trucks ran on rugged grounds with random presence of sands and rocks. While on-road sedans ran on flat, smooth, and relatively straight tracks. In this case study, multiple variable platform modules fabricated by AM were optimized to maximize the customer utility.

Table 5.8: The R/C racing car family with two market segments

| **Market Segment 1** | **Market Segment 2** |
|---|---|
| Short-course race truck | On-road sedan |
|  |  |

Variable platform modules shared by both cars in the family included: (1) bumper, (2) chassis, and (3) wheel. These three modules were chosen as variable platforms because: (1) the modules were fundamental components in cars; (2) the modules could have similar but non-identical geometries in different product variants; and (3) the design parameters of the modules affected various aspects of car performance. Values in the *Reference Platform Design Parameter Vector* $\boldsymbol{X}_{R,j}$ of each variable platform module are listed in Table 5.9. Table 5.9 also contains the design parameter constraints ($\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$) calculated by the method introduced in Section 5.2.

Table 5.9: Key design parameters of variable platform modules

| Platform module ($j$) | $k$ | Key design parameters | $\boldsymbol{X}_{R,j}$ | $d_{kk}{}^{min}$ | $d_{kk}{}^{max}$ |
|---|---|---|---|---|---|
| Bumper ($j$=1) | 1 | Bounding box length ($B$) | 55 mm | 0.54 | 1.52 |
| | 2 | Bounding box height ($H$) | 20 mm | 0.65 | 1.55 |
| | 3 | Cell size ($C$) | 3 mm | 0.52 | 2.05 |
| | 4 | Wall thickness ($T$) | 2 mm | 0.58 | 2.57 |

| | | | | | |
|---|---|---|---|---|---|
| Chassis ($j$=2) | 1 | Overall length ($L$) | 300 mm | 0.86 | 1.22 |
| | 2 | Overall width ($W$) | 100 mm | 0.75 | 1.20 |
| | 3 | Volume fraction ($V$) | 70% | 0.67 | 1.41 |
| Wheel ($j$=3) | 1 | Average cell size ($R$) | 5 mm | 0.56 | 2.30 |
| | 2 | Cell wall thickness ($D$) | 2 mm | 0.87 | 2.51 |

Four performance indicators were used as factors in customer utility calculation: the Weight-Reduction Indicator $W$, Acceleration Indicator $X$, Range Indicator $Y$, and Shock-Resistance Indicator $Z$. Part-worth utility functions are formulated in Eq. (5.35) for both short-course race trucks ($U_1$) and on-road sedans ($U_2$). The part-worth values were obtained by surveying student participants about their preferences. Both $U_1$ and $U_2$ are objective functions that are maximized in variable platform design optimization.

$$\begin{aligned} \max : U_1(\boldsymbol{D}_{1j}) &= 0.23W_1 + 0.27X_1 + 0.19Y_1 + 0.31Z_1 \\ \max : U_2(\boldsymbol{D}_{2j}) &= 0.29W_2 + 0.29X_2 + 0.26Y_2 + 0.16Z_2 \end{aligned} \tag{5.35}$$

Genetic algorithm (GA) implemented in MATLAB® was used to find optimal platform variant design parameters in each product variant in its corresponding market segment. The optimization result is shown in Table 5.10, where diagonal element values $d$ of the optimal *Platform Design Parameter Variation Matrix* $\boldsymbol{D}_{ij}$ and the corresponding maximum customer utility $U_i(\boldsymbol{D}_{optimal})$ are listed. Two product variants in the R/C racing car family were optimized independently in their corresponding market segments, and hence the variable platform design parameters of the truck are different from those of the sedan, as shown in the first two rows of Table 5.10. In order to compare the performance of variable product platforms with consistent platforms implemented in conventional product families, another design optimization problem was formulated with the objective function written as the linear combination of $U_1$ and $U_2$ (Eq. (5.36)). For simplicity, both weight values $\omega_1$ and $\omega_2$ were assumed 1.

$$\begin{aligned} &\max:\ Y = \omega_1 U_1(\boldsymbol{D}_{1j}) + \omega_2 U_2(\boldsymbol{D}_{2j}) \\ &where\ \boldsymbol{D}_{1j} \equiv \boldsymbol{D}_{2j} \end{aligned} \tag{5.36}$$

As indicated in Eq. (5.36), $\boldsymbol{D}_{1j}$ equals $\boldsymbol{D}_{2j}$ since design parameters of consistent platforms were shared by all product variants in the family. Optimal consistent platform design parameters are shown in the bottom row of Table 10, which also shows utility values $U_i(\boldsymbol{D}_{1j}\equiv\boldsymbol{D}_{2j})$ when consistent platforms are applied. In Table 10, the positive values of the percentage difference ($\%(\Delta U)_i$) between $U_i(\boldsymbol{D}_{optimal})$ and $U_i(\boldsymbol{D}_{1j}\equiv\boldsymbol{D}_{2j})$ (Eq. (5.37)) indicate that the application of variable platforms has improved customer utility in both market segments.

$$\%(\Delta U)_i = \frac{U_i\left(\boldsymbol{D}_{optimal}\right) - U_i(\boldsymbol{D}_{1j} \equiv \boldsymbol{D}_{2j})}{U_i(\boldsymbol{D}_{1j} \equiv \boldsymbol{D}_{2j})} \times 100\% \tag{5.37}$$

Table 5.10: Optimal design parameters and resultant maximum customer utilities

| Market segment | Bumper | | | | Chassis | | | Wheel | | Customer utility | | $\%(\Delta U)_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $d_B$ | $d_H$ | $d_C$ | $d_T$ | $d_L$ | $d_W$ | $d_M$ | $d_R$ | $d_D$ | $U_i(\boldsymbol{D}_{optimal})$ | $U_i(\boldsymbol{D}_{1j}\equiv\boldsymbol{D}_{2j})$ | |
| Variable platforms (Short-course race truck) | 0.54 | 0.65 | 2.01 | 0.63 | 0.89 | 0.75 | 0.67 | 0.75 | 0.99 | 1.78 | 1.61 | 10.6% |
| Variable platforms (On-road sedan) | 0.55 | 0.66 | 1.59 | 0.58 | 0.86 | 0.75 | 0.67 | 0.84 | 2.08 | 2.29 | 1.97 | 16.2% |
| Consistent platforms ($\boldsymbol{D}_{1j} \equiv \boldsymbol{D}_{2j}$) | 0.55 | 0.67 | 2.03 | 0.81 | 0.86 | 0.78 | 0.70 | 0.88 | 2.45 | | | |

Table 5.11 shows the costs of three optimized variable platforms for short-course race truck ($C_1(\boldsymbol{D}_{optimal})$) and on-road sedan ($C_2(\boldsymbol{D}_{optimal})$), and the costs of optimized consistent platforms ($C(\boldsymbol{D}_{1j}\equiv\boldsymbol{D}_{2j})$). The percentage cost difference $\%(\Delta C)_i$ between an optimized variable platform and its consistent platform counterparts is calculated as:

$$\%(\Delta C)_i = \frac{C_i\left(\boldsymbol{D}_{optimal}\right) - C(\boldsymbol{D}_{1j} \equiv \boldsymbol{D}_{2j})}{C(\boldsymbol{D}_{1j} \equiv \boldsymbol{D}_{2j})} \times 100\% \tag{5.38}$$

Table 5.11: Optimal design parameters and resultant maximum customer utilities

| | AM production costs ($/unit) | | | $\%(\Delta C)_i$ | | |
|---|---|---|---|---|---|---|
| | Bumper | Chassis | Wheel | Bumper | Chassis | Wheel |
| $C_1(\boldsymbol{D}_{optimal})$ – Optimized variable platform (short-course race truck) | 26 | 85 | 96 | -25.7% | -4.5% | -14.3% |
| $C_2(\boldsymbol{D}_{optimal})$ – Optimized variable platform (on-road sedan) | 32 | 82 | 100 | -8.6% | -7.9% | -10.7% |
| $C(\boldsymbol{D}_{1j}\equiv\boldsymbol{D}_{2j})$ – Optimized consistent platform | 35 | 89 | 112 | | | |

The negative $\%(\Delta C)_i$ values shown in Table 5.11 indicate that the optimized variable platforms have lower cost than conventional consistent platforms. Therefore, combining the results shown in Tables 5.10 and 5.11, it can be demonstrated that the proposed cost-driven design methodology for variable platforms has achieved both performance improvements and cost savings.

## 5.4 Summary

In Chapter 5, the author introduced the concept of AM process setting platforms and variants. AM production cost increment was estimated using the proposed FTDABC approach, with time equations calculated from a trained ANFIS. The feasible space's boundary of AM process setting adjustment terms was identified by solving a multiobjective optimization problem that minimized cost increment and maximized design freedoms. A Mamdani model implemented with fuzzy design rules was used to calculate the extreme values of variable platform modules' design parameters. A case study on designing a family of R/C racing cars illustrated the procedures of AM process setting feasible space exploration and platform design constraint identification. The proposed methodology can help designers explore AM-enabled design space while controlling production cost at the same time. As proven by the case study, compared to conventional product families with the drawback of performance compromise due to consistent platform sharing, the implementation of additive manufactured variable platforms demonstrated product performance improvements at lower costs.

# Chapter 6 Design Evaluation and Optimization of a Product Family

The flexibility of AM processes makes them suitable for producing customized products in various industries [101]. Therefore, AM techniques have good potential to be applied in product family development, where both product commonality and variety are desired to satisfy diversified customer requirements at low costs [4]. The objective of the work described in this chapter is to investigate the effect of applying AM in the product family development in terms of costs and performances. When both additive manufactured parts and those fabricated by traditional techniques coexist in the list of module candidates, novel design evaluation and optimization methods are developed to provide product family design guidelines.

In this chapter, the Complex Commonality Index (*CCI*) is formulated to indirectly measure cost savings in product families. The calculation of *CCI* considers the effect modular commonality, parametric commonality, and cost information in design and production. The market share (*MS*) of a product family is formulated using a logit model based on customer perceived utilities. *MS* is used as the performance metric of the family. A multiobjective design optimization problem is set up to optimize platform module selections and corresponding engineering design parameters simultaneously, with maximizing the commonality and performance metrics as two conflicting objective functions. The NSGA-II with a proposed chromosome encoding pattern is used to search for Pareto-optimal designs. The proposed product family design evaluation and optimization methodology is illustrated in the case study of designing a family of R/C racing cars. The optimization result provides designers with feasible alternative design proposals to satisfy different performance and commonality requirements. It is observed that Pareto-optimal solutions form distinct groups, each of which contains multiple product family designs sharing the same modular configuration. Variable and consistent platform modules coexist in Pareto-optimal designs; while variable platforms are more favoured when higher performance and lower commonality level are desired. Two types of product families differing in the allowable modular configuration are optimized and compared. It is found that the incorporation of additive manufactured variable platforms can significantly improve a product family's performance while its commonality level is decreased only slightly, which implies the benefit of implementing AM in the product families. The sensitivity analysis is conducted to study the effect of AM-related costs on design optimization result. The result suggests that additive manufactured variable platforms will be more frequently used in the product families if AM techniques become cheaper and more standardized in the future. It is expected that the proposed method can help designers understand the potential impact of AM techniques in product family design.

In some literature, the economic efficiency of product family design can be evaluated in terms of profit which is the difference between the total revenue and total cost [75]. However, successful computation of the total revenue, cost, and the resultant profit requires detailed information and knowledge about productions, R&D, logistics, marketing strategies, customer behaviors, and so on. Such comprehensive information is complex, dynamic, and not always accessible by designers to make credible profit calculation. Therefore, instead of the real profit value, numerical metrics are developed in this research to indirectly indicate a product family's costs and performances, in order for designers to evaluate or compare their designs even when comprehensive information of the company and the market is not available. The commonality metric is formulated in Section 6.1, and the performance metric is formulated in Section 6.2.

## 6.1 Product Family Evaluation – the Commonality Metric

Various commonality metrics have been used in product family evaluation, aiming to measure the degree of commonality which leads to cost savings [64, 70, 71, 80, 84]. However, these metrics evaluate only either module selections or components' engineering design parameters. They are not able to measure the commonality of the product family when both module selection and component design parameters are jointly evaluated as a combined factor. Although some of these metrics, such as the Comprehensive Metric of Commonality (*CMC*) developed in [70], contain cost information, they consider only the absolute component cost itself, while relative values of cost savings due to platform sharing are not involved in the evaluation of a product family's commonality. Particular product family design is considered more economically efficient than one another if it can save cost more effectively while improving or at least preserving performances. Therefore, in this research, cost savings resulted from implementing commonality is treated as an important factor in evaluating product families. Furthermore, additive manufactured variable platforms enable higher flexibility and complexity in the product family design. Therefore, a novel method other than those in literature is needed to measure the cost effectiveness of newly configured product families.

In Section 6.1, a novel commonality metric for additive manufactured product family design is proposed. The metric is named *Complex Commonality Index* (*CCI*). The word "complex" is used to indicate that three different levels of commonality contribute to the index value, i.e. 1) the commonality of modular configurations, 2) the commonality of modules' engineering design parameters, and 3) the commonality of manufacturing processes. In addition, unit cost savings from the commonality are also incorporated in the *CCI* formulation.

### 6.1.1 Formulating the Complex Commonality Index (*CCI*)

Information and data required for the *CCI* formulation include: 1) the list of all module candidates manufactured by either AM or traditional processes, represented as $\boldsymbol{X}_L$, 2) product family design variables $\boldsymbol{B}_i$ and $\boldsymbol{D}_{ij}$, whose values will be iteratively updated during optimization process, 3) cost information of each module candidate; and 4) cost information for setting up and managing different production lines.

Mathematically, the proposed *CCI* is a function of $\boldsymbol{B}_i$ and $\boldsymbol{D}_{ij}$. Relevant cost-related data will be calculated separately as described in Section 6.1.2, hence they are not treated as decision variables in *CCI* formulation. The mathematical formulation of *CCI* is shown in Eq. (6.1).

$$
\begin{aligned}
&CCI\left(\boldsymbol{B}_i, \boldsymbol{D}_{ij}\right) \\
&= \sum_{q=1}^{Q}\left[\sum_{j}\left(\frac{c_{jj}-1}{N-1}\cdot \Delta c_{mj}\cdot \sum_{k=1}^{n_{Pj}}\frac{1}{\sigma_{jk}+1}\right)+K_q\cdot \Delta C_q\right] \\
&= \sum_{q=1}^{Q}\left[\sum_{j}\left(\frac{c_{jj}-1}{N-1}\cdot \Delta c_{mj}\cdot \mu_j\right)+K_q\cdot \Delta C_q\right] \\
&= \sum_{j=1}^{M}\left(\frac{c_{jj}-1}{N-1}\cdot \Delta c_{mj}\cdot \mu_j\right)+\sum_{q=1}^{Q}\left(K_q\cdot \Delta C_q\right) \\
&\forall i\in\{1,2,\ldots,N\} \quad \forall j\in\{1,2,\ldots,M\}
\end{aligned}
\tag{6.1}
$$

Terms in Eq. (6.1) are explained in details as follows, with their formulation shown in Eq. (6.2) as follows.

$$
\begin{aligned}
&\boldsymbol{C}=\sum_{i=1}^{N}\boldsymbol{B}_i \;\Rightarrow\; c_{jj}=\sum_{i=1}^{N}b_{jj,i}=0,1,\ldots,N\\
&\sigma_{jk}=\left[\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(d_{kk}-\bar{d}\right)^2}\,\right]_j\\
&K_q=\begin{cases}1: when\, c_{jj}=N\, or\, 0\\ 0: when\, c_{jj}\in\left(0,N\right)\end{cases}\\
&c_{mj}\left(AM\right)>c_{mj}\left(non-AM\right)\\
&\Delta c_{mj}\left(AM\right)<\Delta c_{mj}\left(non-AM\right)\\
&i=1,2,\ldots,N\ and\ j=1,2,\ldots,M
\end{aligned}
\tag{6.2}
$$

$N$ is the number of product variants in the family, $M$ is the total number of all types of module candidates on the list, and $c_{jj}$ is a diagonal element of the *Modular Commonality Matrix* $\boldsymbol{C}$. In Eqs. (6.1) and (6.2), $\sigma_{jk}$ is the standard deviation of $j^{th}$ module candidate's $k^{th}$ design variable, across all $N$ product variants in the family. The value of $\sigma_{jk}$ is calculated from the diagonal elements of $\boldsymbol{D}_{ij}$. The term $\mu_j=\sum_{k=1}^{n_{Pj}}\frac{1}{\sigma_{jk}+1}$ with $\sigma_{jk}$ in the denominator position implies the commonality of engineering design parameters, so that a larger $\mu_j$ (smaller $\sigma_{jk}$) indicates higher similarity in design parameters of different products. $Q$ is the total number of module types, and each type can have two classes of alternative designs: additive manufactured modules (potential to be variable platforms) and those fabricated by traditional processes (potential to be consistent platforms).

A reference family $\boldsymbol{F}^0$ is assumed to consist of $N$ completely identical product variants, based on which any other product families can be derived by modifying any one of the product variants. Two manners of modifications are allowed: 1) replacing original modules with totally different alternatives, and 2) tuning original modules' design parameters. Both types of design variations will decrease the family's commonality (or increase its variety), which requires extra costs in product family development [79]. In the proposed *CCI* formulation, $\Delta c_{m_j}$ is the estimated cost *increment* per unit part introduced by replacing the $j^{th}$ module of any product variant in the reference family $\boldsymbol{F}^0$, and the newly formed family $\boldsymbol{F}'$ will contain product variants with multiple different modules. In other words, $\Delta c_{m_j}$ also represents the estimated cost *saving* due to a module being shared as a platform in the family. The product family will have a larger *CCI* value if its implemented platforms are low-cost modules (indicated by small $c_{m_j}$) with high cost saving potential (indicated by large $\Delta c_{m_j}$). For an additive manufactured variable platform module, its design parameters are allowed to vary in different product variants independently. Such variations require extra efforts (e.g. time, manpower, and resources) for drawing, analyzing, and validating etc., which introduce extra cost. Generally, the degree of variation itself, indicated by the standard deviation $\sigma_{jk}$, is used as a factor in the *CCI* formulation, since the correlation between the extra cost and the degree of design parameter variations is difficult to model and predict accurately. A large $\sigma_{jk}$ and hence a smaller $\mu_j$ implies a lower level of commonality of the product family. For a consistent platform, $\sigma_{jk}$ is 0 and $\mu_j$ is 1; while for a variable platform, $\sigma_{jk}$ can be larger than 0 and $\mu_j$ can be smaller than 1.

$\Delta C_q$ is the cost increment required for operating two different manufacturing processes, i.e. an AM process and a traditional process, for the $q^{th}$ type of module. $\Delta C_q$ values are determined by extra manpower, resource, and inventory costs. In other words, $\Delta C_q$ also represents estimated cost savings when only one manufacturing process (either an AM or a traditional process) is implemented for the $q^{th}$ module type. In this situation, *all* product variants use the same $j^{th}$ module candidate from the list $\boldsymbol{X}_L$, indicated by the corresponding $c_{jj}$ being $N$ and the $c_{kk}$ of other module candidates of the same type ($q^{th}$ type) being 0. In the *CCI* formulation, the binary value $K_q$ becomes 1 to "enable" the cost saving term $\Delta C_q$ when a manufacturing process is shared (i.e. process commonality is achieved); and it becomes 0 to "disable" the $\Delta C_q$ term when different manufacturing processes need to be operated, which results in a smaller *CCI* value.

As discussed above, a larger *CCI* value implies higher commonality. The *CCI* can be considered as an indicator of three levels of commonality in a product family: 1) manufacturing process level commonality, represented by the $K_q \Delta C_q$ term; 2) modular level, represented by the $\frac{c_{jj}-1}{N-1} \cdot \Delta c_{mj}$ term; and 3) parametric level, represented by the $\mu_j$ term. In the *CCI* formulation, $\sigma_{jk}$ is added by 1 to prevent the denominator from being 0 when $\sigma_{jk}$ is 0. $c_{jj}$ is subtracted by 1, so that the $\left(\frac{c_{jj}-1}{N-1} \cdot \Delta c_{mj} \cdot \mu_j\right)$ term is 0 when $c_{jj}$ is 1, which represents a unique module.

In the *CCI* formulation, the $\Delta c_{mj}$ and $\Delta C_q$ terms represent the *unit* cost savings on the component and manufacturing process levels respectively, while the $c_{jj}$, $\mu_j$, and $K_q$ terms represent the number (or frequency) of shared features (components, parameters, or processes). Therefore, in fact, the *CCI* value reflects the total cost saving in a product family due to multiple levels of commonality. Profit (i.e. the difference of total revenue and total cost) is not computed in this research. The determination of selling prices (that largely affect the total revenue) is complicated and requires even more information than that in *CCI*, which is inaccessible in the early design stage.

As discussed in previous research [70, 71, 80, 84], it is a conventional practice to formulate commonality metrics on a pre-defined scale such as [0,1]. To follow this convention, the original *CCI* value calculated in Eq. (6.1) is normalized by dividing it by its maximum possible value $CCI_{max}$, as formulated in Eq. (6.3). The maximum $CCI_{max}$ value is obtained when all platforms in the family are consistent platform modules fabricated by traditional processes. In this situation, $c_{jj}$ values for corresponding consistent platforms are equal to $N$, while $c_{jj}$ values for corresponding variable platforms are 0. In addition, all $\sigma_{jk}$ values are 0 due to the consistency of engineering design parameters in all product variants; and all $K_q$ values are 1 since only one traditional process is operated for producing the $q^{th}$ type of module. The maximum value of the normalized $CCI$ is 1, indicating the highest achievable commonality of the family. In summary, a product family with high commonality is achieved by selecting platform module with no or little variations in design parameters, low production cost, and large cost savings/increment in case of design changes.

$$
\begin{aligned}
CCI_{normalized} &= \frac{CCI}{CCI_{max}} \in (0,1] \\
CCI_{normalized}{}^{max} &= \frac{CCI_{max}}{CCI_{max}} = 1
\end{aligned}
\tag{6.3}
$$

### 6.1.2 Costing

This section introduces a method to estimate $c_{m\,j}$ and $\Delta c_{m\,j}$ values used in the $CCI$ formulation.

From the manufacturer's point of view, a product lifecycle costs (LLC) are raised in four general phases: design, production, service, and end-of-life. In this research, only the unit design cost ($c_{m\,j}{}^{Des}$) and unit production cost ($c_{m\,j}{}^{Pro}$) are considered in the unit module cost ($c_{m\,j}$) calculation for the $j^{th}$ module, as shown in Eq. (6.4). Service and end-of-life costs are related to after-sale customer activities which are difficult to estimate at the product design stage, hence they are not involved in the $c_{mj}$ calculation.

$$c_{mj} = c_{mj}^{Des} + c_{mj}^{Pro} \tag{6.4}$$

As discussed by Wang et al. [79], the cost of a product family will be raised by changing either product design or customer order. This numerical amount of cost *increment* $\Delta c_{mj}$ per unit of the $j^{th}$ module due to design variations can also be considered as cost *savings* brought by commonality. Similarly to the calculation of $c_{mj}$, the $\Delta c_{mj}$ value is estimated as the summation of the extra design cost $\Delta c_{mj}^{Des}$ and extra production cost $\Delta c_{mj}^{Pro}$ per unit module, as written in Eq. (6.5).

$$\Delta c_{mj} = \Delta c_{mj}^{Des} + \Delta c_{mj}^{Pro} \tag{6.5}$$

The $\Delta c_{mj}^{Des}$ value is estimated as:

$$\Delta c_{mj}^{Des} = \frac{\left( \sum_{l}^{L} R_l t_l \right)_j}{V} \tag{6.6}$$

where $L$ is the total number of tasks (e.g. drawing, simulation, and assembly scheduling etc.) at the design phase to change a product variant's design by replacing the original module with a different alternative from the list $\boldsymbol{X}_L$. $R_l$ is the time rate of the cost of doing the $l^{th}$ task, and $t_l$ is the average time span required to complete the $l^{th}$ task. $V$ is the estimated production volume of each product variant. Since the aforementioned design tasks need to be carried out only *before* the designs are put into production, the term $\left( \sum_{l}^{L} R_l t_l \right)$ represents the non-recurring cost that does not increase with production volume. Therefore, when $V$ is large, the different between $\Delta c_{m_j}^{Des}$ of an additive manufactured platform and that of a consistent platform is trivial, so that $\Delta c_{m_j}^{Des}$ values of both types of platform modules can be treated as equal.

At the production phase, the extra cost introduced by producing multiple different module designs in the same manufacturing line is the summation of extra cost in fabrication and that in maintaining the inventory. The average cost increment per unit product $\Delta c_{m_j}^{Pro}$ can be obtained by dividing the overall value by the production volume, as shown in Eq. (6.7). The $\Delta c_{m_j}^{Pro}$ can also be considered as the cost saving brought by commonality.

$$\Delta c_{m_j}^{Pro} = \frac{\left(\Delta C_{Fabricate} + \Delta C_{Inventory}\right)_j}{V} \tag{6.7}$$

The extra cost $\Delta C_{Inventory}$ is the sum of extra amounts in three main categories of inventory costs: ordering costs, holding costs, and shortage costs, as formulated in Eq. (6.8). Inventory powder or wire-based raw materials for additive manufactured components have lower procurement prices (smaller $\Delta C_{Ordering}$ ) compared to inventory spare parts and tools for traditional manufacturing techniques. Materials in AM have higher recycling rates and less wastage than traditional processes [27]. Furthermore, in AM processes, the same raw material can be used to produce different part geometries of platform variants. Therefore, less inventory stock with smaller $\Delta C_{Holding}$ is required for additive manufactured components compared to their counterparts fabricated by traditional processes. With the assumption that no shortage is expected, the $\Delta C_{Shortage}$ is neglected in the computation of $\Delta C_{Inventory}$. Based on the above discussion, $\Delta C_{Inventory}$ values of additive manufactured components are smaller than those of components fabricated by traditional processes.

$$\Delta C_{Inventory} = \Delta C_{Ordering} + \Delta C_{Holding} + \Delta C_{Shortage} \tag{6.8}$$

The estimation of $\Delta C_{Fabricate}$ can be carried out using the time-driven activity-based costing method [125], as represented in Eq. (6.9). The extra cost $\Delta C_{Fabricate}$ per unit product is raised from three activities required for realizing different designs in the same manufacturing line: 1) digital data preparation for different designs, 2) build process with different settings, and 3) postprocess with different tools and work sequences. In Eq. (6.9), $AC_a$ represents the cost in each of the aforementioned three activities, while $R_a$ and $t_a$ are the time rate of costs and the timespan of the activity respectively. Various cost drivers (e.g. manpower, material, and energy etc.) are included in each time rate term $R_a$.

$$\Delta C_{Fabricate} = \sum_{a=1}^{A=3} AC_a = \sum_{a=1}^{A=3} \left( R_a t_a \right) \tag{6.9}$$

For additive manufactured components, the flexibility of AM techniques enables multiple different designs to be fabricated using the same process settings [30]. In other words, variations in a variable platform's design do not necessarily require significant changes in the production process. For traditional processes, however, a design change usually result in significant adjustments in the process, such as developing new/extra tools, which is a significant cost increment [29]. Therefore, $\Delta C_{Fabricate}$ values of additive manufactured variable platform modules are smaller than that of consistent platform modules. In other words, the commonality-induced cost saving brought by sharing the design of an additive manufactured variable platform is smaller than that of a consistent platform module counterpart.

Based on the discussion in this section, the term $\Delta c_{mj}/c_{mj}$ for a variable platform module has a smaller value than that for a consistent platform module counterpart.

## 6.2 Product Family Evaluation – the Performance Metric

Besides cost savings, another objective in platform-based product family design is to satisfy customer needs and improve the product family's competitiveness over other alternative products in the market [64, 68, 69, 71]. This section introduces the formulation to estimate a product family's market share (*MS*), which is used as the performance metric and an objective function in product family design optimization.

### 6.2.1 Formulating the Market Share

Information and data required for the market share estimation include: 1) the size of each market segment in terms of customer populations; 2) customer preferences in each market segment, reflected by utility functions; and 3) performances of competing products from other manufacturers.

The *MS* is defined as the estimated percentage of the total number of customers in the entire market that will purchase the products from the family to be designed. The mathematical expression of the *MS* is shown in Eq. (6.10), where $S_i$ is the size (customer population) of the $i^{th}$ market segment, and $Pr_i$ is the probability for a customer to choose the product variant over competing alternatives from other manufacturers. A larger *MS* value indicates a higher competitiveness of the product family in the market.

$$MS = \frac{\sum_{i=1}^{N} S_i \cdot Pr_i}{\sum_{i=1}^{N} S_i} \tag{6.10}$$

### 6.2.2 Estimating the Customer Choice Probability

The customer choice probability $Pr_i$ of a product variant positioned in the $i^{th}$ market segment can be modeled by a mixed logit model [143], as formulated in Eq. (6.11).

$$Pr_i = \int L_i(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\boldsymbol{\theta}) d\boldsymbol{\beta} \tag{6.11}$$

where $L_i(\boldsymbol{\beta})$ is the logit probability defined in Eq. (6.12). $L$ is the total number of alternative products, including the $i^{th}$ product variant from the family to be designed, and $(L - 1)$ other products from competitors.

$$L_i(\boldsymbol{\beta}) = \frac{e^{A_i(\boldsymbol{\beta})}}{\left[\sum_{l=1}^{L} e^{A_l(\boldsymbol{\beta})}\right]_i} \tag{6.12}$$

In Eq. (6.12), $A_i(\boldsymbol{\beta})$ represents the observed part of customer perceived utility $U_i$ on the $i^{th}$ product variant. $A_i(\boldsymbol{\beta})$ is modeled by a part-worth utility function [143] defined in Eq. (6.13).

$$\begin{aligned} U_i &= A_i(\boldsymbol{\beta}) + \boldsymbol{\varepsilon_i} \\ A_i(\boldsymbol{\beta}) &= \boldsymbol{\beta} \cdot \boldsymbol{p}_i(\boldsymbol{B}_i, \boldsymbol{D}_{ij}, \boldsymbol{X}_R) \\ &= \sum_{\varphi=1}^{\Phi} \beta_{p_\varphi} [p_\varphi(\boldsymbol{B}_i, \boldsymbol{D}_{ij}, \boldsymbol{X}_R)] \end{aligned} \tag{6.13}$$

where $\boldsymbol{\varepsilon_i}$ is the independent identically distributed extreme value, which can be assumed zero for simplicity [75]. $\boldsymbol{p}_i$ is the vector of totally $\Phi$ key attribute level which are functions of the product family's design variables $\boldsymbol{B}_i$, $\boldsymbol{D}_{ij}$, and $\boldsymbol{X}_R$. $p_\varphi$ is the $\varphi^{th}$ element (attribute level) of $\boldsymbol{p}_i$. $\boldsymbol{\beta}$ is a vector of utility weights following density functions $f(\boldsymbol{\beta}|\boldsymbol{\theta})$, such as Gaussian functions. $\boldsymbol{\theta}$ is the vector of parameters (e.g. means and variances) defining the distribution of $\boldsymbol{\beta}$. $\beta_{p_\varphi}$ is the $\varphi^{th}$ utility weight value assigned to the attribute level $p_\varphi$.

Combining Eqs. (6.10) to (6.13), the *MS* can now be formulated in the form shown in Eq. (6.14), which is a function of $\boldsymbol{B}_i$, $\boldsymbol{D}_{ij}$, and $\boldsymbol{X}_R$.

$$\begin{aligned} & MS\left(\boldsymbol{B}_i, \boldsymbol{D}_{ij}, \boldsymbol{X}_R\right) \\ & = \frac{\sum_{i=1}^{N}\left\{S_i \int \frac{\exp\left[\boldsymbol{\beta}\cdot\boldsymbol{p}_i\left(\boldsymbol{B}_i,\boldsymbol{D}_{ij},\boldsymbol{X}_R\right)\right]}{\left[\sum_{l=1}^{L}\exp\left(\boldsymbol{\beta}\cdot\boldsymbol{p}_l\right)\right]_i} f\left(\boldsymbol{\beta}|\boldsymbol{\theta}\right) d\boldsymbol{\beta}\right\}}{\sum_{i=1}^{N} S_i} \\ & \in (0,1) \\ & \forall i \in \{1,2,\ldots,N\} \quad \forall j \in \{1,2,\ldots,M\} \end{aligned} \tag{6.14}$$

where $\boldsymbol{p}_l$ is the $l^{th}$ competing product's attribute level.

For simplicity, the density function $f(\boldsymbol{\beta}|\boldsymbol{\theta})$ can take a binary form expressed in Eq. (6.15), with utility weights $\boldsymbol{\beta}$ simplified to be constant numbers denoted by $\boldsymbol{\beta}_0$. The identification of each $\beta_{p_\theta}$ value can be done through market surveys followed by conjoint analysis [81].

$$\begin{aligned} & f\left(\boldsymbol{\beta}|\boldsymbol{\theta}\right) = \begin{cases} 1, & \boldsymbol{\beta} = \boldsymbol{\beta}_0 \\ 0, & \boldsymbol{\beta} \neq \boldsymbol{\beta}_0 \end{cases} \\ & Pr_i = L_i\left(\boldsymbol{\beta}_0\right) = \frac{e^{A_i\left(\boldsymbol{\beta}_0\right)}}{\left[\sum_{l=1}^{L} e^{A_l\left(\boldsymbol{\beta}_0\right)}\right]_i} \end{aligned} \tag{6.15}$$

With the above manipulation, the mixed logit is converted to a standard logit model [143], and the *MS* formulation is simplified to the form written in Eq. (6.16). Obviously, a product family with better performances (indicated by larger attribute levels) will have higher customer choice probability values and hence larger market share.

$$
\begin{aligned}
& MS\left(\boldsymbol{B}_i, \boldsymbol{D}_{ij}, \boldsymbol{X}_R\right) \\
& = \frac{\sum_{i=1}^{N} \left\{ S_i \dfrac{\exp\left[\boldsymbol{\beta}_0 \cdot \boldsymbol{p}_i\left(\boldsymbol{B}_i, \boldsymbol{D}_{ij}, \boldsymbol{X}_R\right)\right]}{\left[\sum_{l=1}^{L} \exp\left(\boldsymbol{\beta}_0 \cdot \boldsymbol{p}_l\right)\right]_i} \right\}}{\sum_{i=1}^{N} S_i} \\
& \in (0,1) \\
& \forall i \in \{1, 2, \ldots, N\} \quad \forall j \in \{1, 2, \ldots, M\}
\end{aligned}
\tag{6.16}
$$

## 6.3 Product Family Design Optimization

Efficient product family design is expected to satisfy diversified customer requirements at low cost. Therefore, the product family design is to be optimized to maximize the commonality metric and performance metric which are conflicting objective functions due to the trade-off between cost and performance. The overall framework of tasks and information flow in the optimization is illustrated in Figure 6.1 . This section presents the design optimization process of a product family, which allows coexistence of consistent platforms and additive manufactured variable platforms. Platform module selection and engineering design parameters are simultaneously optimized using a multiobjective genetic algorithm (GA) with a proposed encoding pattern.

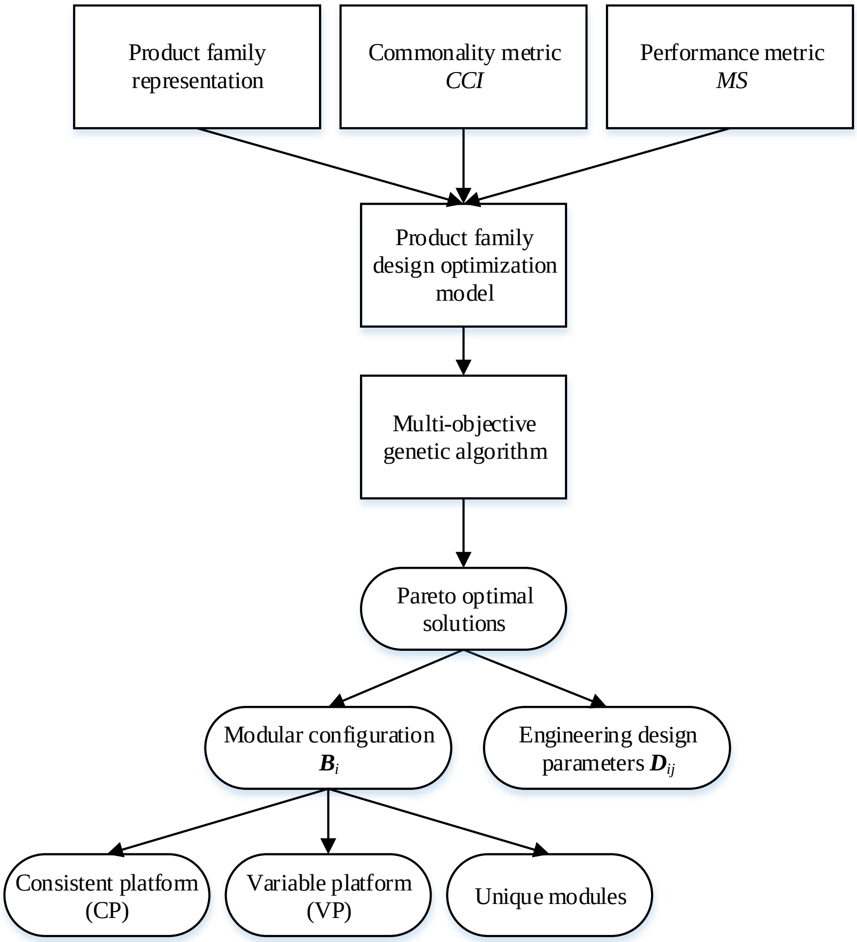



Figure 6.1: The overall framework of tasks and information flow in the optimization

### 6.3.1 The Multiobjective Design Optimization Problem

The complete product family design optimization problem is formulated in Eq. (6.17). It is assumed that the reference modular design parameters $\boldsymbol{X}_R$ are given. The decision variables are $\boldsymbol{B}_i$ and $\boldsymbol{D}_{ij}$ for each $i^{th}$ product variant each $j^{th}$ module candidate. Two objective functions are the metrics *CCI* and *MS* formulated in Sections 6.1 and 6.2.

$$
\begin{aligned}
&Given: && \boldsymbol{X}_R, \boldsymbol{D}_{ij}^{min}, \boldsymbol{D}_{ij}^{max} \\
&Variables: && \begin{cases} \boldsymbol{B}_i \\ \boldsymbol{D}_{ij} \end{cases} \quad \forall i \in \{1,2,...,N\} \, \forall j \in \{1,2,...,M\} \\
&Objectives: && \begin{cases} max: CCI\left(\boldsymbol{B}_i, \boldsymbol{D}_{ij}\right)_{normalized} \in (0,1] \\ max: MS\left(\boldsymbol{B}_i, \boldsymbol{D}_{ij}, \boldsymbol{X}_R\right) \in (0,1) \end{cases} \\
&Subject\ to: && \begin{cases} h_1: \ b_{jj} = [0,1] \\ h_2: \ \sum_{j}^{M_q} b_{jj}^q = 1 \\ h_3: \ \boldsymbol{D}_{aj} = \boldsymbol{D}_{bj} \ \ a \neq b, \ for \ CP \\ g_1: \ \boldsymbol{D}_{ij}^{min} \leq \boldsymbol{D}_{ij} \leq \boldsymbol{D}_{ij}^{max} \end{cases}
\end{aligned}
\tag{6.17}
$$

Decision variables $\boldsymbol{B}_i$ and $\boldsymbol{D}_{ij}$ are subject to several constraints expressed in $h_1$, $h_2$, $h_3$, and $g_1$. As specified in $h_1$, the diagonal elements $b_{jj}$ of $\boldsymbol{B}_i$ are assigned the binary values 0 and 1. In $h_2$, the term $b_{jj}^q$ represents the $\boldsymbol{B}_i$ elements corresponding to the same $q^{th}$ module type which has a total number of $M_q$ module candidates (including both variable and consistent platforms). $h_2$ indicates that one and only one module must be selected among the $M_q$ candidates for the $q^{th}$ module type. $h_3$ represents the constraint that the design parameters of a consistent platform module in different product variants must be identical. In $g_1$, the extreme values $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ are obtained from the manufacturability analysis which ensures that the designed component can be successfully fabricated by the designated manufacturing technique (AM or traditional processes). Both $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ are pre-determined and treated as given values in the optimization problem.

### 6.3.2 Multiobjective Genetic Algorithm

The product family design optimization problem formulated in Section 6.3.1 is solved by a multiobjective genetic algorithm (GA), which is suitable for nonlinear and mixed-integer problems. The GA variant used in this research is the NSGA-II, a controlled elitist genetic algorithm developed. NSGA-II searches for Pareto-optimal solutions by non-dominated sorting, followed by crowding distance sorting to preserve diversity during the convergence to the final Pareto front. Crossover and mutation operations are used to update the chromosome population. The general solving procedure of NSGA-II is shown in Figure 6.2. The detailed computation method can be found in [144].

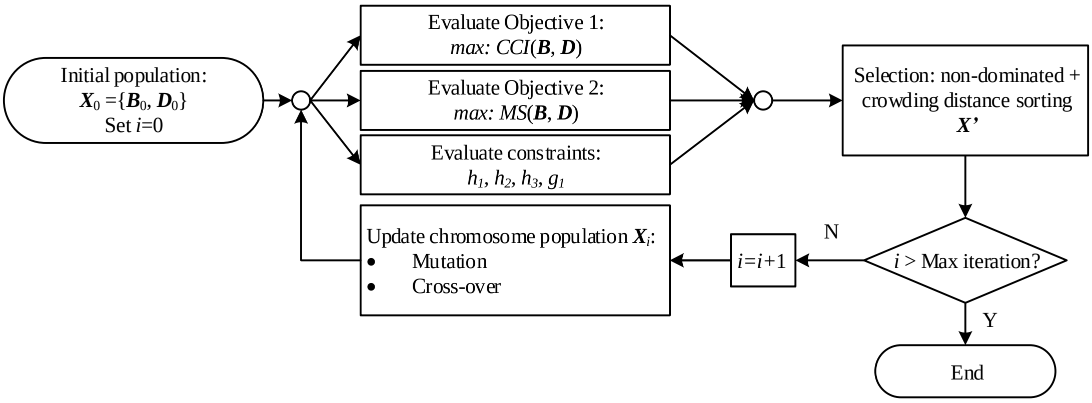



Figure 6.2: The solving procedure of NSGA-II

For joint optimization of module selection and design parameters, a chromosome is encoded to contain the information of each product variant's modular configuration and each module's engineering design parameters. A representative chromosome with the proposed encoding pattern is illustrated in Figure 6.3.

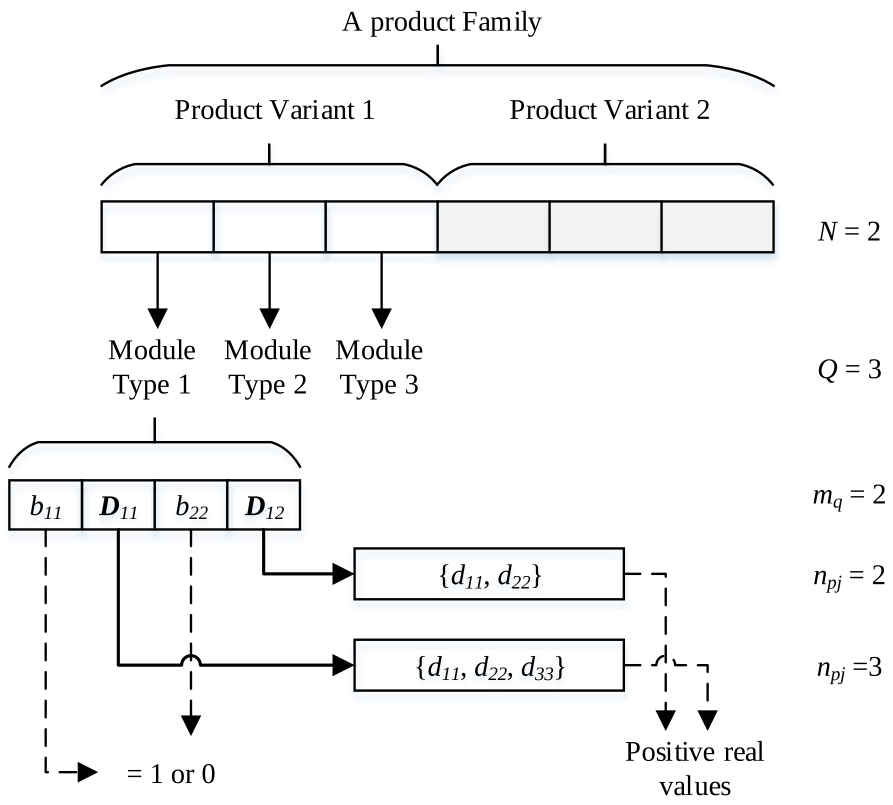



Figure 6.3: A representative chromosome

A chromosome representing a product family consists of $N$ sections, each of which represents an individual product variant. A section is further partitioned into $Q$ sub-sections, each of which represents a particular module type. A sub-section contains a string of two types of genes: 1) diagonal elements $b_{jj}$ of the Module Selection Matrix $\boldsymbol{B}_i$, that take are binary bits; and 2) modules' engineering design parameters $\boldsymbol{D}_{ij}$ that are arrays of positive real (or floating-point) values. The total length of a chromosome is calculated in Eq. (6.18), where $m_q$ is the number of platform module candidates (including variable and consistent platforms) that belong to the $q^{th}$ type of module. $n_{Pj}$ is the number of engineering design parameters in the $j^{th}$ module.

$$\Theta = N\sum_{q=1}^{Q}\left[\sum_{j=1}^{m_q}\left(1+n_{Pj}\right)\right] \tag{6.18}$$

Two stopping criteria are defined for the NSGA-II. The optimization process stops either when the maximum allowable number of iterations (set as 800) has been reached. The product family design optimization process is implemented in MATLAB®.

### 6.3.3 Optimizing Two Types of Product Families

Two types of product families are introduced as follows, which are optimized and compared against each other.

**Type I** is a product family with the configuration proposed in this research that allows implementations of both CP and additive manufactured VP modules. In the production of Type I product families, both AM and traditional processes can be employed. The optimization result is a set of Pareto-optimal solutions.

**Type II** is a conventional product family that allows the implementation of only CP modules. In the production of Type II product families, only existing (or original) traditional manufacturing processes are employed. Mathematically, $b_{jj}$ corresponding to consistent platforms are always 1, while $b_{jj}$ corresponding to variable platforms are always 0. Furthermore, $\boldsymbol{D}_{ij}$ of the same platform module in different product variants are always identical, resulting in all $\sigma_{jk}$ being 0. Therefore, in Type II product families, *CCI* takes the maximum constant value ( $CCI_{normalized}^{max} = 1$ ). A product family design optimization problem is now a single objective problem with the *MS* as the only fitness function. The optimization result is the global optimal solution for the family design. The design optimization problem for Type II product families is formulated in Eq. (6.19).

$$
\begin{aligned}
&Given: && \boldsymbol{X}_R, \boldsymbol{D}_{ij}^{min}, \boldsymbol{D}_{ij}^{max} \\
&Variables: && \boldsymbol{D}_{ij} \quad \forall i \in \{1,2,...,N\} \quad \forall j \in \{1,2,...,M\} \\
&Objectives: && max\ MS\left(\boldsymbol{D}_{ij}\right) \in [0,1] \\
&Subject\ to: && \begin{cases} h_1: \ b_{jj} = [0,1] \\ h_2: \ \sum_{j}^{M_q} b_{jj}^{q} = 1 \\ h_3: \ \boldsymbol{D}_{aj} = \boldsymbol{D}_{bj} \ \ a \neq b, \ for\ CP \\ g_1: \ \boldsymbol{D}_{ij}^{min} \leq \boldsymbol{D}_{ij} \leq \boldsymbol{D}_{ij}^{max} \end{cases}
\end{aligned}
\tag{6.19}
$$

## 6.4 Case Study and Computational Experiments

A family of R/C racing cars is to be designed as a case study to illustrate the proposed product family design evaluation and optimization methodology. Computational experiments are also carried out with the assumed parameters to compare different types of product families and predict future trends of product family development strategies.

### 6.4.1 Variable Platform Module Candidates

Three product variants of the R/C racing car family are positioned in three different market segments ($N$ = 3): "monster truck (P1)", "short-course race truck (P2)", and "on-road sedan (P3)", as shown in Table 6.1. Monster trucks run on extremely rough and bumpy terrains spread with large obstacles; short-course race trucks run on rugged grounds usually with the random presence of sands and rocks; while on-road sedans run on flat, smooth, and relatively straight tracks. Different application scenarios of racing cars lead to diversified customer needs that are reflected in utility functions.

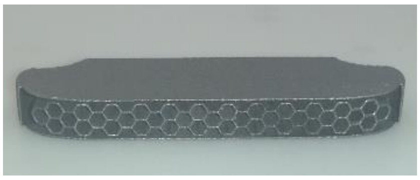
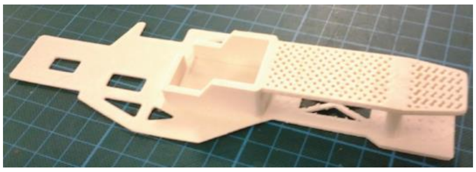
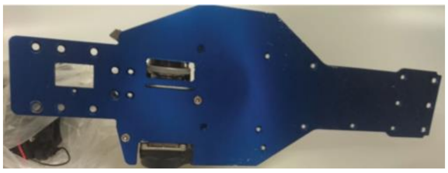
Table 6.1: The R/C racing car family with three product variants

| **Market Segment 1** | **Market Segment 2** | **Market Segment 3** |
|---|---|---|
| Monster truck (P1) | Short-course race truck (P2) | On-road sedan (P3) |
|  |  |  |

In this case study, the module candidate list $\boldsymbol{X}_L$ is assumed to contain three types ($Q$ = 3) of modules: the bumper, the chassis, and the wheel, which are all potential platforms. These three types of modules are chosen because: 1) they are fundamental components in cars, 2) they can take similar but not strictly identical geometries in different variants, and 3) their design parameters may have significant effects on various aspects of car performances.

It is also assumed that each type of module has two alternatives, i.e. an additive manufactured module that can be used as a VP, and a module fabricated by a traditional process that can be used as a CP. Therefore, $\boldsymbol{X}_L$ contains totally six ($M$ = 6) module candidates. Conceptual design drawings for each of the module candidates are listed in Table 6.2, together with their manufacturing processes. Table 6.2 also presents AM design features on each additive manufactured module that are expected to enhance module performances.

Table 6.2: VP and CP module candidates

| $q$ | **Module types** | **VP module candidate** | **CP module candidate** |
|---|---|---|---|
| 1 | Bumper |  $j = 1$ |  $j = 2$ |
| 2 | Chassis |  $j = 3$ |  $j = 4$ |
| 3 | Wheel |  $j = 5$ |  $j = 6$ |

Module candidates' key design parameters are identified empirically by experienced designers. Reference design parameters $\boldsymbol{X}_R$ are listed in Table 6.3. Table 6.3 also contains the diagonal elements $d_{kk}^{min}$ and $d_{kk}^{max}$ in $\boldsymbol{D}_{ij}^{min}$ and $\boldsymbol{D}_{ij}^{max}$ respectively, which are used as bound constraints in the design optimization problem.

Table 6.3: Key design parameters of VP and CP modules

| **Module type (*q*)** | **Platform type (*j*)** | ***k*** | **Key design parameters** | $X_R$ | $d_{kk}^{min}$ | $d_{kk}^{max}$ |
|---|---|---|---|---|---|---|
| Bumper (*q* = 1) | VP (*j* = 1) | 1 | Width (*B*) | 55 mm | 0.54 | 1.52 |
| | | 2 | Height (*H*) | 20 mm | 0.65 | 1.55 |
| | | 3 | Cell diameter (*C*) | 3 mm | 0.52 | 2.05 |
| | | 4 | Cell-wall thickness (*T*) | 2 mm | 0.58 | 2.57 |
| | CP (*j* = 2) | 1 | Width (*B*') | 55 mm | 0.60 | 1.40 |
| | | 2 | Height (*H*') | 20 mm | 0.8 | 1.45 |
| Chassis (*q* = 2) | VP (*j* = 3) | 1 | Length (*L*) | 300 mm | 0.86 | 1.22 |
| | | 2 | Width (*W*) | 100 mm | 0.75 | 1.20 |
| | | 3 | Thickness (*P*) | 3 mm | 0.50 | 1.50 |
| | | 4 | Volume fraction (*V*) | 70% | 0.67 | 1.41 |
| | CP (*j* =4) | 1 | Length (*L*') | 300 mm | 0.60 | 1.10 |
| | | 2 | Width (*W*') | 100 mm | 0.80 | 1.40 |
| | | 3 | Thickness (*P'*) | 3 mm | 0.50 | 1.50 |
| Wheel (*q* = 3) | VP (*j* = 5) | 1 | Diameter (*S*) | 90 mm | 0.70 | 2.20 |
| | | 2 | Cell diameter (*R*) | 5 mm | 0.56 | 2.30 |
| | | 3 | Cell-wall thickness (*D*) | 2 mm | 0.87 | 2.51 |
| | CP (*j* = 6) | 1 | Diameter (*S'*) | 90 mm | 0.70 | 2.20 |

### 6.4.2 Numerical Factors in Product Family Evaluation

In this section, numerical factors used in *CCI* and *MS* formulations are obtained from the R/C racing car family.

Estimated $\Delta c_{mj}$ and $\Delta C_q$ values are listed in Table 6.4. These values will be used in the computation of *CCI* during the product family design optimization process. It is assumed that the product volume of each product variant is $V = 1000$.

Table 6.4: $\Delta c_{mj}$ and $\Delta C_q$ values for CCI calculation

| **Module type (*q*)** | $\Delta C_q$ **(\$/unit)** | **VP/CP module candidate (*j*)** | $\Delta c_{mj}$ **(\$/unit)** |
|---|---|---|---|
| Bumper ($q$ = 1) | 10 | VP ($j$ = 1) | 3 |
| | | CP ($j$ = 2) | 7 |
| Chassis ($q$ = 2) | 14 | VP ($j$ = 3) | 5 |
| | | CP ($j$ =4) | 12 |
| Wheel ($q$ = 3) | 17 | VP ($j$ = 5) | 3 |
| | | CP ($j$ = 6) | 16 |

Four attributes contribute to the customer utility calculation: Weight-Reduction Indicator *W*, Acceleration Indicator *X*, Range Indicator *Y*, and Shock-Resistance Indicator *Z*. The standard logit model, as formulated in Eq. (6.15), is used to estimate customer choice probabilities. $\boldsymbol{\beta}$ is the vector of utility weights assigned to the aforementioned four attributes, as written in Eq. (6.20). The computation of $\boldsymbol{\beta}$ is beyond the scope of this research. Therefore, utility weight values are assumed known a priori, as listed in Table 6.5 for monster trucks, short-course race trucks, and on-road sedans.

$$\boldsymbol{\beta} = \left[\beta_W, \beta_X, \beta_Y, \beta_Z\right]^T \tag{6.20}$$

Table 6.5: Utility weights for three product variants in each corresponding market segment

| **Market segment (*i*)** | ***W*** $\beta_W$ | ***X*** $\beta_X$ | ***Y*** $\beta_Y$ | ***Z*** $\beta_Z$ |
|---|---|---|---|---|
| Monster truck (1) | 0.4 | 0.5 | 0.2 | 0.9 |
| Short-course race truck (2) | 0.6 | 0.7 | 0.5 | 0.8 |
| On-road sedan (3) | 0.9 | 0.9 | 0.8 | 0.5 |

It is assumed that each market segment consists of ($L$ = 3) alternative products including the one to be designed ($l$ = 1). Two other products in the same market segment are from competing manufacturers ($l$ = 2 and 3), whose utility values $A_2$ and $A_3$ are assumed known and constant, as listed in Table 6.6.

Table 6.6: Utility values of two other competing products in the each market segment

| **Market segment (*i*)** | $A_2$ | $A_3$ |
|---|---|---|
| Monster truck (1) | 1.5 | 2 |
| Short-course race truck (2) | 2.2 | 2.8 |
| On-road sedan (3) | 3.0 | 4.0 |

The assumed size, i.e. customer population, of an imaginary market is listed in Table 6.7.

Table 6.7: Assumed market segment size

| | **Monster truck (*i* = 1)** | **Short-course race truck (*i* = 2)** | **On-road sedan (*i* = 3)** | **Total** |
|---|---|---|---|---|
| $S_i$ | 2,000 | 4,500 | 5,000 | 11,500 |

Therefore, the *MS* of the product family is computed as:

$$
\begin{aligned}
MS &= \frac{\sum_{i=1}^{N} S_i \cdot Pr_i}{\sum_{i=1}^{N} S_i} \\
&= \frac{2,000Pr_1 + 4,500Pr_2 + 5,000Pr_3}{11,500} \qquad (6.21) \\
&\in (0,1)
\end{aligned}
$$

With the numerical factors obtained above, the *CCI* and *MS* values can be calculated during the optimization process, as the product family decision variables $\boldsymbol{B}_i$ and $\boldsymbol{D}_{ij}$ are updated in each iteration.

### 6.4.3 Optimization Results

As discussed in Section 6.3.2 and 6.3.3, in the design of a Type I family, module selections (represented by $\boldsymbol{B}_i$) and engineering design parameters (represented by $\boldsymbol{D}_{i,j}$) are optimized simultaneously in the same run, using the proposed chromosome encoding method. The optimization process is stopped at the 179th iteration. The resultant Pareto front is shown in Figure 6.4, which contains the non-dominated solutions that maximize $CCI_{normalized}$ and *MS*. Each point on the Pareto front represents a set of decision variables $\boldsymbol{B}_i$ and $\boldsymbol{D}_{ij}$ than define a Pareto-optimal product family design.

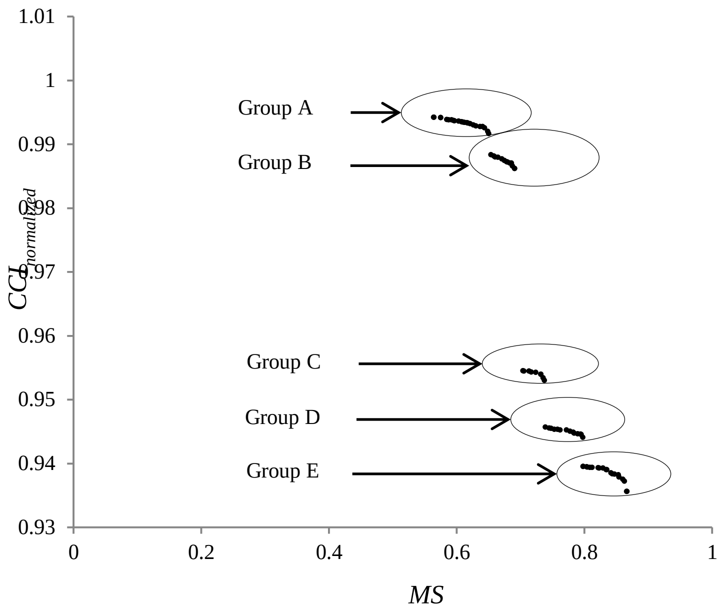



Figure 6.4: The Pareto front of the product family design optimization

The Pareto solutions form five groups with clear gaps in the *CCI-MS* plot, as marked by circles in Figure 6.4. All Pareto-optimal designs in a group have the same modular configuration, as shown in Table 6.8. The notation "CP" represents a consistent platform, "VP" represents a variable platform, "$U_{VP}$" represents an additive manufactured unique module that used to be a variable platform candidate on the list $\boldsymbol{X}_L$, and "$U_{CP}$" represents a unique module fabricated by traditional processes that used to be a consistent platform candidate. P1 (monster truck), P2 (short-course race truck), and P3 (on-road sedan) are the three product variants.

Table 6.8: Platform module selection in different groups of Pareto solutions

<table>
<tr><th rowspan="2">Group</th><th colspan="3">Bumper</th><th colspan="3">Chassis</th><th colspan="3">Wheel</th></tr>
<tr><th>P1</th><th>P2</th><th>P3</th><th>P1</th><th>P2</th><th>P3</th><th>P1</th><th>P2</th><th>P3</th></tr>
<tr><td>A</td><td colspan="3">CP</td><td>$U_{VP}$</td><td colspan="2">CP</td><td colspan="3">CP</td></tr>
<tr><td>B</td><td colspan="3">CP</td><td colspan="2">VP</td><td>$U_{CP}$</td><td colspan="3">CP</td></tr>
<tr><td>C</td><td colspan="3">CP</td><td>$U_{CP}$</td><td colspan="2">VP</td><td colspan="3">CP</td></tr>
<tr><td>D</td><td colspan="3">CP</td><td colspan="3">VP</td><td colspan="3">CP</td></tr>
<tr><td>E</td><td colspan="3">VP</td><td colspan="3">VP</td><td colspan="3">CP</td></tr>
</table>

The module selections shown in Table 6.8 reflect the values of $\boldsymbol{B}_i$. Several observations can be made from the optimization result, discussed as follows.

(1) A Pareto-optimal product family design can contain a mixture of consistent platforms, additive manufactured variable platforms, and unique modules.

(2) The same product variant can have different sets of module selections in different groups of Pareto-optimal product families. For example, P1 (monster truck) in Group A implements the consistent bumper platform, an additive manufactured unique chassis module and the consistent wheel platform; while the same product variant P1 in Group E implements the variable bumper platform, the variable chassis platform, and the consistent wheel platform.

(3) For the same type of module, the selection of variable or consistent platform differs in different groups of Pareto-optimal families. For example, the consistent bumper platform is implemented in all product variants in Group A, B, C, and D; while the variable bumper platform is implemented in Group E.

4) The popularity of variable platforms increases from Groups A to E, while the resultant *MS* increases and the commonality level *CCI* decreases. The term “popularity” is inferred by 1) the number of implemented variable platform modules and 2) the number of product variants that implement each variable platform module, as shown in Table 6.9. The positive correlation between *MS* and the popularity of variable platforms can be explained by the fact that variable platform modules with performance-enhancing AM design features usually can have higher customer utilities. However, since design parameters of variable platforms are allowed to change

independently in different product variants, the more variable platforms are used, the lower the commonality level is. Furthermore, as discussed in Section 6.1.2, *CCI* decreases as more variable platforms with higher cost $c_{mj}$ and lower cost saving $\Delta c_{mj}$ are implemented in the family.

Table 6.9: Popularity of variable platforms in different Pareto solution groups

| Group | No. of implemented VP | No. of product variants that implement the VP |
|---|---|---|
| A | 0 | 0 |
| B | 1 (chassis) | 2 (P1 + P2) |
| C | 1 (chassis) | 2 (P2 + P3) |
| D | 1(chassis) | 3 (P1 + P2 + P3) |
| E | 2 (bumper + chassis) | 3 (P1 + P2 + P3) for bumper<br>3 (P1 + P2 + P3) for chassis |

Within each group, different points have different values of engineering design parameters $\boldsymbol{D}_{i,j}$. It can be observed that the sub-Pareto-front within each group shows a trend of decreasing *CCI* with increasing *MS*. Such a trend can be explained by the trade-off between commonality and performances, which is a commonly observed phenomenon in any product families [9, 64].

Different parameters (e.g. market sizes and customer preferences) in *MS* formulation will affect the optimization result. This is because when the parameters change, the objective function of the optimization problem is also changed. However, the methods of setting up and solving the problem remain the same no matter what the numerical values are used, as long as the information required for *CCI* and *MS* calculations can be obtained.

### 6.4.4 Comparison of Commonality Metrics

To validate the effectiveness of the proposed metrics in product family design, the optimization solution presented in Section 6.4.3 is compared with the benchmarking results obtained by using other commonality metrics from literatures. The Commonality Index (*CI*) proposed by Martin and Ishii [145] has been used in many literatures to represent component sharing within product families. The formulation of *CI* is shown below:

$$CI = 1 - \frac{u - \max p_j}{\sum_{j=1}^{v_n} p_j - \max p_j} \tag{6.22}$$

where $u$ is the number of unique parts, $p_j$ is the number of parts in the $j^{th}$ product variant, and $v_n$ is the number of product variants in the family. *CI* ranges from 0 to 1, and a larger *CI* value indicates higher commonality (or fewer unique components).

Another representative metric is the Product Line Commonality Index (*PCI*) proposed by Kota et al. [146]. Unlike *CI* that measures only component sharing, *PCI* also represents the commonality in design parameters, materials, and processes. The formulation of *PCI* is shown below:

$$PCI = \frac{\sum_{i=1}^{P} n_i \times f_{1i} \times f_{2i} \times f_{3i} - \sum_{i=1}^{P} \frac{1}{n_i^2}}{\left(P \times N\right) - \sum_{i=1}^{P} \frac{1}{n_i^2}} \tag{6.23}$$

where $P$ is the number of components in each product variant, $N$ is the number of product variants in a family, $n_i$ is the number of product variants that share the same $i^{th}$ component. $f_{1i}$ is the size and shape factor for the $i^{th}$ component, defined as the percentage of products sharing the $i^{th}$ component ($n_i$) that have identical size and shape. The minimum value of $f_{1i}$ is $1/n_i$ and the maximum value is 1. $f_{2i}$ is the material and manufacturing process factor, and $f_{3i}$ is the assembly and fastening schemes factor for the $i^{th}$ component. The definitions of $f_{2i}$ and $f_{3i}$ are similar to that of $f_{1i}$. *PCI* ranges from 0 to 1, and a larger *PCI* value indicates a higher level of commonality.

As described in Section 6.3, the proposed *CCI* can be applied as an objective function in the product family design optimization problem that identifies the optimal combination of modules and design parameters. For comparison, *CI* and *PCI* are applied in the same optimization problem in place of *CCI*, and the results are summarized in Table 6.10. The maximum *MS* values (indicating the best performance of the product family) obtained by applying different commonality metrics are shown in Table 6.10 along with the corresponding optimal numbers of unique modules (U), consistent platforms (CP), and variable platforms (VP) as well as the numbers of product variants implemented with each U, CP, and VP.

Table 6.10: Optimization result summary for different commonality metrics

| | *CI* | *PCI* | *CCI* |
|---|---|---|---|
| Commonality metrics values | 0.88 | 0.81 | 0.93 |
| Max. *MS* | 0.72 | 0.78 | 0.87 |
| # of unique modules (U) | 1 (chassis) | 1 (chassis) | 0 |
| # of product variants with U | 3 for chassis | 1 for chassis | 0 |
| # of consistent platforms (CP) | 2 (wheel & bumper) | 2 (wheel & bumper) | 1 (wheel) |
| # of product variants with CP | 3 for wheel<br>3 for bumper | 3 for wheel<br>3 for bumper | 3 for wheel |
| # of variable platforms (VP) | 0 | 1 (chassis) | 2 (bumper & chassis) |
| # of product variants with VP | 0 | 2 for chassis | 3 for bumper<br>3 for chassis |

It can be observed from Table 6.10 that the maximum *MS* value obtained by using *CCI* is larger than those obtained using *CI* and *PCI*. As discussed in Section 6.2, a high *MS* is desirable since it implies that the product family well meets the customer requirements in different market segments. Furthermore, by comparing the design solutions (i.e. the number of U, CP, and VP) obtained by using *CI*, *PCI*, and the proposed *CCI*, it can be found that *CCI* results in the fewest unique modules (U), the fewest consistent platforms (CP), and the largest amount of variable platforms (VP). Similarly, *CCI* also results in the largest number of product variants that use the variable platforms. It is also seen that *PCI* has more preference to variable platforms than *CI*, although not so much as *CCI*. The reasons of the abovementioned differences in results using different commonality metrics are discussed as follows. In the definition of *CI*, a shared component is not accepted as a platform if any of its design parameters are changeable. The diversification within a product family is completely achieved by applying unique modules in different product variants. Therefore, no variable platforms are allowed in the product family. In the definition of *PCI*, design parameter changes in shared component are acceptable and recognized using the factors $f_{xi}$. However, the degree of parameters changes/variations across different product variants is not measured. Any parameter change, no matter how small or how large it is, will cause the same amount of drop in the $f_{xi}$ value (note: $f_{xi}$ is a discrete variable defined as the ratio of two integers). In the proposed *CCI*, the degree of parameter variations is recognized along with the other levels of commonality (i.e. the modular commonality and process commonality). In the design solution obtained by using *CCI*, variable platforms with slight changes in design parameters are preferred, and the diversification in the product family can be achieved by using variable platforms instead of unique modules. Therefore, according to the above result and discussion, the proposed *CCI* is more suitable than *CI* and *PCI* in designing additive manufactured variable platforms whose design parameters are allowed to be varied within the family to achieved higher performance.

### 6.4.5 Comparison between the Type I and Type II Product Families

To illustrate the effect of implementing additive manufactured variable platform modules, the design optimization result obtained in Section 6.4.3 for a Type I family is compared with that of a Type II conventional product family which only allows the implementation of consistent platforms (as described in Section 6.3.3).

In order to form a Type II car family, the additive manufactured bumper, chassis, and wheel are removed from the module candidate list $\boldsymbol{X}_L$, while all consistent platform modules are implemented in the family ($M$ = $Q$ = 3). Therefore, matrices $\boldsymbol{B}_i$ are now pre-defined constants; and matrices $\boldsymbol{D}_{ij}$ are the only decision variables. The *CCI* value is now a constant that has the maximum achievable value ($CCI_{normalized}{}^{max} = 1$). The maximum *MS* value is 0.562, obtained from solving the single objective optimization problem in Eq. (6.19). This Type II family's optimization result is marked by Point A on the *CCI-MS* plot in Figure 6.5, together with the Pareto front obtained from the Type I product family. Average values of $CCI_{normalized}$ and *MS* for the Type I family are calculated and marked by Point B on the *CCI-MS* plot. The $CCI_{normalized}$ and *MS* values for both Types I and II families are listed in Table 6.11.

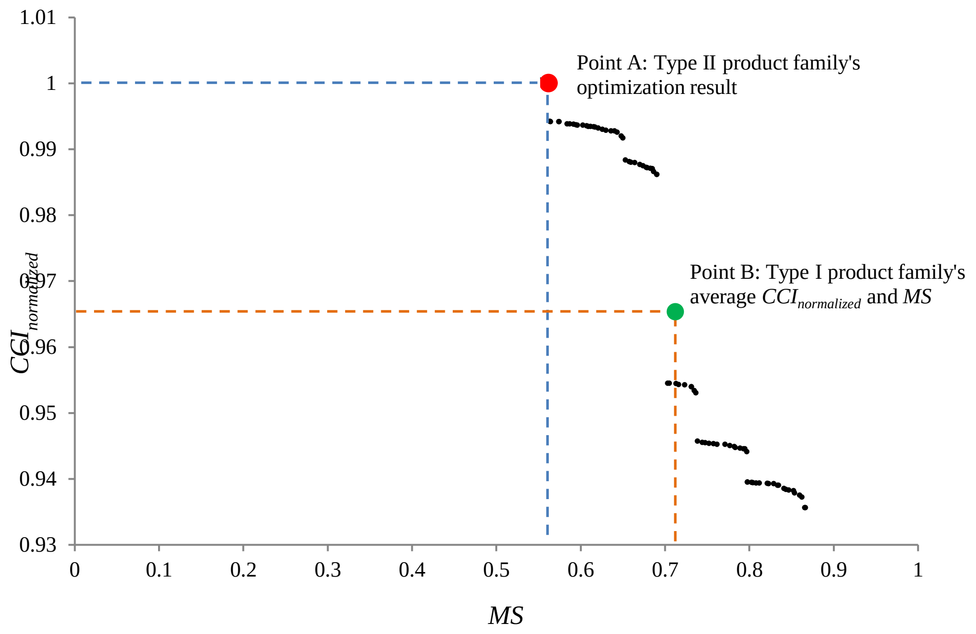



Figure 6.5: The Type I and II product families' optimization result

Table 6.11: $CCI_{normalized}$ and $MS$ values of the Type I and II product families

| | **Type I family** | **Type II family** |
|---|---|---|
| $CCI_{normalized}$ | 0.965 (average) | 1 |
| $MS$ | 0.714 (average) | 0.562 |

Compared with the Type II conventional product family that allows only consistent platforms, the Type I family's Pareto-optimal designs have an average decrease of only 3.5% in *CCI*, but a significant increase of 27.6% in *MS*. Therefore, the implementation of additive manufactured variant platforms shows the ability to improve the product family's performance and market share without significantly deteriorating the commonality.

## 6.5 Summary

In this chapter, commonality and performance metrics were formulated to evaluate product family designs. A multiobjective design optimization problem was set up to simultaneously optimize platform module selections and engineering design parameters. The NSGA-II was used to search for Pareto solutions. The proposed product family design evaluation and optimization methods were illustrated in a case study on R/C racing car family, whose Pareto-optimal configurations were found to contain both variable and consistent platform modules. The application of additive manufactured product platforms had significantly improved a product family's performance while the commonality level was decreased only slightly, which implied the benefit of implementing AM in product families. The sensitivity analysis suggested that additive manufactured variable platforms will be more favored in product family design if AM techniques would be cheaper and more standardized in the future. The optimization result provided designers with feasible alternative design proposals to satisfy different performance and commonality requirements of a product family.

The proposed design framework can be applied in other product family design cases if the platform components can be manufactured by AM. The proposed AM design feature selection method is useful as long as the loading conditions, required properties, and design objectives of the target components are clearly defined. Similarly, the proposed AM costing, design evaluation and optimization methods are feasible in other design scenarios as long as the basic information of a product family (i.e. the component list, reference design parameters, market segments, and unit costs etc.) can be obtained.

# Chapter 7 Conclusions

## 7.1 Research Summary

This thesis presented a design framework for the product families implemented with additive manufactured variable platforms. The concept of an additive manufactured variable platform (VP) was proposed. In contrast to a conventional consistent platform (CP), the design of a VP module was allowed to be varied across different product variants in the same product family. A novel product family configuration was proposed, which contained VP, CP, and unique modules. A mathematical model was formulated to represent the product family, whose module configuration and design parameters were represented by matrices and vectors that can be used as input variables in the design optimization stage.

An automated and intelligent method is needed to help designers explore AM-enabled design space and select appropriated AM design features at the conceptual design stage. Therefore, in this research, the author proposed a hybrid machine learning approach for AM design feature selection. A hierarchical clustering was performed on coded AM design features and target components, resulting in a dendrogram. Existing industrial application examples were used to train a binary classifier that extracted the final sub-cluster from the dendrogram. The AM design features contained in the final sub-clusters were selected. A case study on designing an R/C racing car family was carried out. It was demonstrated that the proposed hybrid machine learning approach was able to provide novice designers with feasible conceptual design solutions.

After AM design features are selected, platform design constraints of additive manufactured VP modules were identified in order to control and limit cost increment due to VP design variations in the product family. The author introduced the concept of AM process settings platforms and variants. AM production cost drivers were identified, and a Fuzzy Time-Driven Activity-Based Costing (FTDABC) approach was developed to estimate the cost increment due to process setting adjustments. Time equations in the FTDABC were predicted in a trained adaptive neuro-fuzzy inference system (ANFIS). The process setting adjustment's feasible space boundary was identified by solving a multiobjective optimization problem. Design parameter limitations of a VP module were computed in a Mamdani-type expert system, which could be used as constraint functions in the VP design optimization.

For product family design evaluation, the Complex Commonality Index (*CCI*) was formulated to measure the design commonality in the manufacturing process level, modular level, and parametric level commonality. The *CCI* metric reflected the cost saving potentials of a product family. The market share (*MS*) was formulated to measure a product family's performance in the market which contains multiple competing alternative products. A multiobjective design optimization problem was developed to maximize the commonality and performance. The product family's modular configuration and engineering design parameters were optimized using a multiobjective genetic algorithm. A family of R/C racing cars was designed in the case study. Optimization results indicated that Pareto-optimal designs contained both variable and consistent platforms simultaneously. The implementation of additive manufactured variable platforms demonstrated a significant improvement in a product family's market share at the price of only a slight decrease in commonality. Sensitivity analysis results suggested that additive manufactured platforms would be more favored against traditionally fabricated platforms as AM techniques become cheaper and more standardized in the future.

## 7.2 Contributions and Significance

The major contribution of this research is the development of a product family design framework which implements additive manufactured variable platforms. The proposed design framework provides enterprises with guidelines in designing performance and cost efficient product families by utilizing AM capabilities. Through the process of developing the individual methodologies in the design framework, other sub-contributions are made and their significance is highlighted as follows.

- **Propose a novel platform concept:**

  An additive manufactured variable platform allows more design flexibilities than conventional consistent platforms.

- **Develop an AM design feature selection method:**

  The proposed hybrid machine learning approach enables automatic and intelligent AM design feature selection. As new AM design features are created in the future, they can be added to the growing database and provided to designers. Furthermore, when more industrial applications examples are made available in the future, they can be used to update the database and refine the AM design feature selection result.

- **Develop a cost prediction approach using FTDABC:**

  Compared with existing AM costing approaches, the proposed FTDABC method has the following advantages: 1) it correlates various cost drivers with adjustable AM process settings in the production of VP modules; 2) the time equation based cost estimation model is independent of the specific part to be manufactured; and 3) the fuzzy time-driven analysis is able to model uncertainties, which makes the FTDABC suitable for practical operations.

- **Develop a AM process setting adjustments' feasible space boundary searching approach:**

  The proposed feasible space boundary searching method enables designers to control and limit cost increments during the design of platform variants.

- **Propose *CCI* to measure a product family's design commonality:**

  The *CCI* is developed to measure product family commonality and indicates the cost saving potential. Compared with existing commonality metrics, the *CCI* has the following advantages: 1) it is able to measure multiple levels of commonality, i.e. manufacturing process level, modular level, and parametric level; 2) its formulation considers the relative values of cost savings due to platform sharing, hence it is a better indicator of a product family's economic efficiency compared with existing metrics which use only absolute component costs; and 3) it is able to measure the design commonality of a product family with both CP and VP modules.

- **Identify optimal product family configuration with both VP and CP modules implemented:**

  The proposed GA chromosome encoding pattern enables the simultaneous optimization of both modular structure and engineering design parameters in multiple product variants within the family. The multiobjective design optimization result provides designers with solutions to create alternative product family designs which can satisfy different performance and commonality requirements.

## 7.3 Limitations and Recommended Future Work

Limitations and recommended future work of this research are discussed as follows:

- **Dynamics of market conditions:**

  In this research, the proposed product family design framework assumes static market conditions, while the fluctuations of market conditions over time are not taken account. In future research, it is recommended to study how the dynamics of market conditions and customer requirements will affect the optimal product family configurations, with the purpose of producing product families that can achieve long term success in the market.

- **Collaborative design of the customized product families:**

  In this research, design engineers are assumed to be the only participants in the product family development process. However, involvement of other professionals and even direct customers in product design is made possible by the availability of affordable AM equipment and easy-to-use design software. In future work, the collaboration mechanism among multiple participating parties can be added to the product family design framework.

- **Extend to product families with additive manufactured electronic components:**

In this thesis, the proposed design framework is only applied to design mechanical modules in a product family. As the technologies of additive manufacturing of electric circuitry are advancing [147, 148], AM is gaining the potential to fabricate batteries, sensors, and many other electro-mechanical elements directly from raw materials. In future work, it is recommended to modify the product family design framework to accommodate additive manufactured electronic components.

- **Development of secondary post-processing techniques for additive manufactured components.**

  AM prototypes fabricated in this thesis are inferior in surface finish and dimensional accuracy compared with machined parts. In future work, efforts will be made in the development of efficient secondary post-processing techniques to improve the surface finish and accuracy. A possible research area is the mechanical and/or electro-chemical polishing of additive manufactured metallic parts. Another possible research direction can be the design and implementation of a hybrid manufacturing machine that contains both AM and milling tools.

# List of Publications

## Journals

- Yao, X., Moon, S.K., and Bi, G., 2016, "A Cost-Driven Design Methodology for Additive Manufactured Variable Platforms in Product Families", ASME Journal of Mechanical Design, Vol. 138, No. 4, pp. 041701, doi: 10.1115/1.4032504.

- Yao, X., Moon, S.K., and Bi, G., 2016, "A Hybrid Machine Learning Approach for Additive Manufacturing Design Feature Recommendation", Rapid Prototyping Journal (accepted).

- Yao, X., Moon, S.K., and Bi, G., 2016, "Multidisciplinary Design Optimization to Identify Additive Manufacturing Resources in Customized Product Development", Journal of Computational Design and Engineering (in press).

- Lei, N., Yao, X., Moon, S.K., and Bi, G., 2016, "An Additive Manufacturing Process Model for Product Family Design", Journal of Engineering Design Vol. 27, No. 11, pp. 751-767.

- Yao, X., Moon, S.K., and Bi, G., "Commonality and Performance Metrics to Evaluate and Optimize the Design of Additive Manufactured Product Families", International Journal of Manufacturing Research (accepted).

- Yao, X., Moon, S.K., and Bi, G., "Effects of Heat Treatment on Microstructures and Tensile Properties of IN718/TiC Nanocomposite Fabricated by Selective Laser Melting", International Journal of Precision Engineering and Manufacturing (under review).

- Yao, X., Moon, S.K., and Bi, G., "Effects of the TiC Nanoparticle Content on Microstructures and Tensile Properties of Selective Laser Melted IN718/TiC Nanocomposites", Rapid Prototyping Journal (under review)

## Conference Papers

- Yao, X., Moon, S.K., Bi, G, and Wei, J., "An Additive Manufacturing Resource Allocation Framework for Multi-Material Part Design," 26th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), Seoul, Korea, June 27-30, 2016.

- Yao, X., Moon, S.K., and Bi, G., "Multidisciplinary Design Optimization for Additive Manufactured Customized Products," 2nd

International Conference on Progress in Additive Manufacturing (Pro-AM), Singapore, May 17-19, 2016.

- Yao, X., Moon, S.K., Bi, G., and Son, H., "Commonality and Performance Metrics of Product Families Implemented with Additive Manufactured Variable Platforms," International Conference on Innovative Design and Manufacturing (ICIDM), Auckland, New Zealand, January 24-26, 2016.

- Yao, X., Moon, S.K., and Bi, G., "The Additive Manufacturing Process Setting's Feasible Space Exploration and Association with Variable Product Platform Design," ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), Boston, MA, USA, August 2-5, Paper No.DETC2015-46665, 2015.

- Yao, X., Moon, S.K., and Bi, G., "Additive Manufacturing Design Future Selection for Variable Product Platforms," International Conference on Engineering Design (ICED), Milan, Italy, July 27-30, 2015.

# References

[1] Gibson, I., Rosen, D. W., and Stucker, B., 2010, *Additive Manufacturing Technologies*, Springer, New York, USA.

[2] Gibson, I., Rosen, D. W., and Stucker, B., 2010. "Design for Additive Manufacturing." *Additive Manufacturing Technologies*, Springer US, pp. 299-332.

[3] Seepersad, C. C., 2014, "Challenges and Opportunities in Design for Additive Manufacturing," 3D Printing and Additive Manufacturing, **1**(1), pp. 10-13.

[4] Cameron, B., and Crawley, E., 2014. "Crafting Platform Strategy Based on Anticipated Benefits and Costs." *Advances in Product Family and Product Platform Design*. T. W. Simpson, et al., eds., Springer New York, pp. 49-70.

[5] Pirmoradi, Z., Wang, G. G., and Simpson, T., 2014. "A Review of Recent Literature in Product Family Design and Platform-Based Product Development." *Advances in Product Family and Product Platform Design*. T. W. Simpson, et al., eds., Springer, New York, USA, pp. 1-46.

[6] Simpson, T., Siddique, Z., and Jiao, J., 2006. "Platform-Based Product Family Development." *Product Platform and Product Family Design*. T. Simpson, et al., eds., Springer, New York, USA, pp. 1-15.

[7] Corl, M., Parsons, M., and Kokkolaras, M., 2014. "Optimal Commonality Decisions in Multiple Ship Classes." *Advances in Product Family and Product Platform Design*. T. W. Simpson, et al., eds., Springer New York, pp. 625-645.

[8] Shooter, S., 2006. "Ice Scraper Product Family Development at Innovation Factory." *Product Platform and Product Family Design*. T. Simpson, et al., eds., Springer US, pp. 459-473.

[9] Willcox, K., and Wakayama, S., 2003, "Simultaneous Optimization of a Multiple-Aircraft Family," Journal of Aircraft, **40**(4), pp. 616-622.

[10] Marion, T., and Simpson, T., 2006. "Platform Leveraging Strategies and Market Segmentation." *Product Platform and Product Family Design*. T. Simpson, et al., eds., Springer US, pp. 73-90.

[11] de Weck, O. L., Suh, E. S., and Chang, D., 2003. "Product Family and Platform Portfolio Optimization," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, Illinois USA, 2-6 Sep 2003, pp. 175-185.

[12] Song, B., Dong, S., Deng, S., Liao, H., and Coddet, C., 2014, "Microstructure and Tensile Properties of Iron Parts Fabricated by Selective Laser Melting," Optics & Laser Technology, **56**, pp. 451-460.

[13] Vilaro, T., Abed, S., and Knapp, W., 2008. "Direct Manufacturing of Technical Parts Using Selective Laser Melting: Example of Automotive Application," *12th European Forum on Rapid Prototyping*, Paris, France, 5-6 Mar 2008.

[14] Poprawe, R., Hinke, C., Meiners, W., Schrage, J., Bremen, S., and Merkt, S., 2015. "Slm Production Systems: Recent Developments in Process Development, Machine Concepts and Component Design." *Advances in Production Technology*. C. Brecher, ed., Springer International Publishing, pp. 49-65.

[15] Cheng, W., Fuh, J. Y. H., Nee, A. Y. C., Wong, Y. S., Loh, H. T., and Miyazawa, T., 1995, "Multi-Objective Optimization of Part-Building Orientation in Stereolithography," Rapid Prototyping Journal, **1**(4), pp. 12-23.

[16] Canellidis, V., Giannatsis, J., and Dedoussis, V., 2009, "Genetic-Algorithm-Based Multi-Objective Optimization of the Build Orientation in Stereolithography," The International Journal of Advanced Manufacturing Technology, **45**(7-8), pp. 714-730.

[17] Strano, G., Hao, L., Everson, R. M., and Evans, K. E., 2011, "Multi-Objective Optimization of Selective Laser Sintering Processes for Surface Quality and Energy Saving," Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture, **225**(B9), pp. 1673-1682.

[18] Verma, A., and Rai, R., 2014. "Computational Geometric Solutions for Efficient Additive Manufacturing Process Planning," *Proc. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2014)*, Buffalo, USA, 17 - 20 Aug 2014.

[19] Kempen, K., Thijs, L., Van Humbeeck, J., and Kruth, J. P., 2012, "Mechanical Properties of Alsi10mg Produced by Selective Laser Melting," Physics Procedia, **39**, pp. 439-446.

[20] Strano, G., Hao, L., Everson, R. M., and Evans, K. E., 2013, "A New Approach to the Design and Optimisation of Support Structures in Additive Manufacturing," The International Journal of Advanced Manufacturing Technology, **66**(9-12), pp. 1247-1254.

[21] Dadbakhsh, S., Hao, L., Jerrard, P. G. E., and Zhang, D. Z., 2012, "Experimental Investigation on Selective Laser Melting Behaviour and Processing Windows of in Situ Reacted Al/Fe2o3 Powder Mixture," Powder Technology, **231**, pp. 112-121.

[22] Alrbaey, K., Wimpenny, D., Tosi, R., Manning, W., and Moroz, A., 2014, "On Optimization of Surface Roughness of Selective Laser Melted Stainless Steel Parts: A Statistical Study," Journal of Materials Engineering and Performance, **23**(6), pp. 2139-2148.

[23] Gu, D., Chen, H., Jia, Q., Dai, D., Andres, G., Andreas, W., Ingomar, K., Zhong, M., and Reinhart, P., 2014, "Combined Strengthening of Multi-Phase and Graded Interface in Laser Additive Manufactured Tic/Inconel 718 Composites," Journal of Physics D: Applied Physics, **47**(4), pp. 045309.

[24] Sun, J., Yang, Y., and Wang, D., 2013, "Parametric Optimization of Selective Laser Melting for Forming Ti6al4v Samples by Taguchi Method," Optics & Laser Technology, **49**, pp. 118-124.

[25] Yadroitsev, I., Krakhmalev, P., and Yadroitsava, I., 2014, "Selective Laser Melting of Ti6al4v Alloy for Biomedical Applications: Temperature Monitoring and Microstructural Evolution," Journal of Alloys and Compounds, **583**, pp. 404-409.

[26] Thijs, L., Sistiaga, M. L. M., Wauthle, R., Xie, Q. G., Kruth, J. P., and Van Humbeeck, J., 2013, "Strong Morphological and Crystallographic Texture and Resulting Yield Strength Anisotropy in Selective Laser Melted Tantalum," Acta Materialia, **61**(12), pp. 4657-4668.

[27] Verma, A., and Rai, R., 2013. "Energy Efficient Modeling and Optimization of Additive Manufacturing Processes," *Solid Freeform Fabrication Symposium*, Austin, Texas, USA, 12-14 Aug 2013.

[28] Leuders, S., Thöne, M., Riemer, A., Niendorf, T., Tröster, T., Richard, H. A., and Maier, H. J., 2013, "On the Mechanical Behaviour of Titanium Alloy Tial6v4 Manufactured by Selective Laser Melting: Fatigue Resistance and Crack Growth Performance," International Journal of Fatigue, **48**, pp. 300-307.

[29] Hopkinson, N., and Dickens, P., 2003, "Analysis of Rapid Manufacturing—Using Layer Manufacturing Processes for Production," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, **217**(1), pp. 31-39.

[30] Ruffo, M., Tuck, C., and Hague, R., 2006, "Cost Estimation for Rapid Manufacturing - Laser Sintering Production for Low to Medium Volumes," Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, **220**(9), pp. 1417-1427.

[31] Lindermann, C., Jahnke, U., Moi, M., and Koch, R., 2012. "Analyzing Product Lifecycle Costs for a Better Understanding of Cost Drivers in Additive Manufacturing," *Solid Freeform Fabrication Symposium*, Austin, Texas, USA, 6-8 Aug 2012.

[32] Lindermann, C., Jahnke, U., Moi, M., and Koch, R., 2013. "Impact and Influence Factors of Additive Manufacturing on Product Lifecycle Costs," *Solid Freeform Fabrication Symposium*, Austin, Texas, USA, 12-14 Aug 2013.

[33] Williams, C., Allen, J., Rosen, D. W., and Mistree, F., 2006. "Process Parameter Platform Design to Manage Workstation Capacity." *Product Platform and Product Family Design*. T. Simpson, et al., eds., Springer, New York, USA, pp. 421-455.

[34] Atzeni, E., and Salmi, A., 2012, "Economics of Additive Manufacturing for End-Usable Metal Parts," The International Journal of Advanced Manufacturing Technology, **62**(9-12), pp. 1147-1155.

[35] Mellor, S., Hao, L., and Zhang, D., 2014, "Additive Manufacturing: A Framework for Implementation," International Journal of Production Economics, **149**, pp. 194-201.

[36] Bourhis, F. L., Kerbrat, O., Dembinski, L., Hascoet, J.-Y., and Mognol, P., 2014, "Predictive Model for Environmental Assessment in Additive Manufacturing Process," Procedia CIRP, **15**, pp. 26-31.

[37] Bourhis, F., Kerbrat, O., Hascoet, J.-Y., and Mognol, P., 2013, "Sustainable Manufacturing: Evaluation and Modeling of Environmental Impacts in Additive Manufacturing," The International Journal of Advanced Manufacturing Technology, **69**(9-12), pp. 1927-1939.

[38] Rosen, D. W., 2014, "Research Supporting Principles for Design for Additive Manufacturing," Virtual and Physical Prototyping, **9**(4), pp. 225-232.

[39] Yang, S., and Zhao, Y., 2015, "Additive Manufacturing-Enabled Design Theory and Methodology: A Critical Review," The International Journal of Advanced Manufacturing Technology, **80**(1-4), pp. 327-342.

[40] Bin Maidin, S., Campbell, I., and Pei, E. J., 2012, "Development of a Design Feature Database to Support Design for Additive Manufacturing," Assembly Automation, **32**(3), pp. 235-244.

[41] Su, X., Yang, Y., Xiao, D., and Chen, Y., 2013, "Processability Investigatation of Non-Assembly Mechanisms for Powder Bed Fusion Process," The International Journal of Advanced Manufacturing Technology, **64**(5-8), pp. 1193-1200.

[42] Brackett, D., Ashcroft, I., and Hague, R., 2011. "Topology Optimization for Additive Manufacturing," *Solid Freeform Fabrication Symposium*, Austin, Texas, USA, 8-10 Aug 2011, pp. 6-8.

[43] Namasivayam, U. M., and Seepersad, C. C., 2011, "Topology Design and Freeform Fabrication of Deployable Structures with Lattice Skins," Rapid Prototyping Journal, **17**(1), pp. 5-16.

[44] Yan, C., Hao, L., Hussein, A., Bubb, S. L., Young, P., and Raymont, D., 2014, "Evaluation of Light-Weight Alsi10mg Periodic Cellular Lattice Structures Fabricated Via Direct Metal Laser Sintering," Journal of Materials Processing Technology, **214**(4), pp. 856-864.

[45] Yan, C., Hao, L., Hussein, A., Young, P., and Raymont, D., 2014, "Advanced Lightweight 316l Stainless Steel Cellular Lattice Structures Fabricated Via Selective Laser Melting," Materials & Design, **55**, pp. 533-541.

[46] Wang, D., Yang, Y., Liu, R., Xiao, D., and Sun, J., 2013, "Study on the Designing Rules and Processability of Porous Structure Based on Selective Laser Melting (Slm)," Journal of Materials Processing Technology, **213**(10), pp. 1734-1742.

[47] Nguyen, J., Park, S.-i., and Rosen, D., 2013, "Heuristic Optimization Method for Cellular Structure Design of Light Weight Components," International Journal of Precision Engineering and Manufacturing, **14**(6), pp. 1071-1078.

[48] Oxman, N., 2011, "Variable Property Rapid Prototyping," Virtual and Physical Prototyping, **6**(1), pp. 3-31.

[49] Brackett, D., Panesar, A., Ashcroft, I., Wildman, R., and Hague, R., 2013. "An Optimization Based Design Framework for Multi-Functional 3d Printing," *24th Annual International Solid Freeform Fabrication Symposium*, Austin, Texas, USA, 12-14 Aug 2013.

[50] Hu, Y., Fadel, G. M., Blouin, V. Y., and White, D. R., 2006, "Optimal Design for Additive Manufacturing of Heterogeneous Objects Using Ultrasonic Consolidation," Virtual and Physical Prototyping, **1**(1), pp. 53-62.

[51] Nagel, J. K. S., and Liou, F. W., 2010, "Designing a Modular Rapid Manufacturing Process," Journal of Manufacturing Science and Engineering, **132**(6), pp. 061006-061006.

[52] Kumke, M., Watschke, H., and Vietor, T., 2016, "A New Methodological Framework for Design for Additive Manufacturing," Virtual and Physical Prototyping, **11**(1), pp. 3-19.

[53] Hague, R., Mansour, S., and Saleh, N., 2004, "Material and Design Considerations for Rapid Manufacturing," International Journal of Production Research, **42**(22), pp. 4691-4708.

[54] Adam, G. A. O., and Zimmer, D., 2015, "On Design for Additive Manufacturing: Evaluating Geometrical Limitations," Rapid Prototyping Journal, **21**(6), pp. 662-670.

[55] Rezaie, R., Badrossamay, M., Ghaie, A., and Moosavi, H., 2013, "Topology Optimization for Fused Deposition Modeling Process," Procedia CIRP, **6**, pp. 521-526.

[56] Samperi, M., Chernow, E., Simpson, T. W., Joshi, S., and Talbot, M., 2013. "Towards a Process Workflow for Designing and Fabricating Parts Using Additive Manufacturing," *RAPID 2013*, Pittsburg, Pennsylvania.

[57] Vayre, B., Vignat, F., and Villeneuve, F., 2012, "Designing for Additive Manufacturing," Procedia CIRP, **3**, pp. 632-637.

[58] Vayre, B., Vignat, F., and Villeneuve, F., 2013, "Identification on Some Design Key Parameters for Additive Manufacturing: Application on Electron Beam Melting," Procedia CIRP, **7**, pp. 264-269.

[59] Kranz, J., Herzog, D., and Emmelmann, C., 2015, "Design Guidelines for Laser Additive Manufacturing of Lightweight Structures in Tial6v4," Journal of Laser Applications, **27**(S1), pp. S14001-1-16.

[60] Zimmer, D., and Adam, G., 2011. "Direct Manufacturing Design Rules." *Innovative Developments in Virtual and Physical Prototyping*, CRC Press, pp. 545-551.

[61] Liu, X., and Rosen, D., 2010. "Ontology Based Knowledge Modeling and Reuse Approach of Supporting Process Planning in Layer-Based Additive Manufacturing," *International Conference on Manufacturing Automation*, 13-15 Dec. 2010, pp. 261-266.

[62] "Astm Standard Wk38342 New Guide for Design for Additive Manufacturing (under Development)," accessed 14 Jun 2016, https://www.astm.org/DATABASE.CART/WORKITEMS/WK38342.htm

[63] Chowdhury, S., Messac, A., and Khire, R. A., 2011, "Comprehensive Product Platform Planning (Cp3) Framework," Journal of Mechanical Design, **133**(10), pp. 101004, doi:10.1115/1.4004969.

[64] Moon, S. K., Park, K., and Simpson, T., 2014, "Platform Design Variable Identification for a Product Family Using Multi-Objective Particle Swarm Optimization," Research in Engineering Design, **25**(2), pp. 95-108.

[65] Luo, X., Yang, W., Kwong, C., Tang, J., and Tang, J., 2014, "Linear Programming Embedded Genetic Algorithm for Product Family Design Optimization with Maximizing Imprecise Part-Worth Utility Function," Concurrent Engineering, **22**(4), pp. 309-319.

[66] Sawai, K., Nomaguchi, Y., and Fujita, K., 2015, "Fundamental Framework toward Optimal Design of Product Platform for Industrial Three-Axis Linear-Type Robots," Journal of Computational Design and Engineering, **2**(3), pp. 157-164.

[67] Yifei, T., Zhaohui, T., Song, M., Lu, Q., and Zhibing, G., 2015, "Research on Customer-Oriented Optimal Configuration of Product Scheme Based on Pareto Genetic Algorithm," Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, **229**(1), pp. 148-156.

[68] Jiao, J., Zhang, Y., and Wang, Y., 2007, "A Generic Genetic Algorithm for Product Family Design," Journal of Intelligent Manufacturing, **18**(2), pp. 233-247.

[69] Fujita, K., and Yoshida, H., 2004, "Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes," Concurrent Engineering, **12**(2), pp. 105-118.

[70] Thevenot, H. J., and Simpson, T. W., 2007, "A Comprehensive Metric for Evaluating Component Commonality in a Product Family," Journal of Engineering Design, **18**(6), pp. 577-598.

[71] Ji, Y., Tang, T., Yu, C., and Qi, G., 2014, "A Joint Optimization Strategy for Scale-Based Product Family Positioning," International Journal of Computational Intelligence Systems, **7**(sup2), pp. 3-14.

[72] Suh, E., de Weck, O., and Chang, D., 2007, "Flexible Product Platforms: Framework and Case Study," Research in Engineering Design, **18**(2), pp. 67-89.

[73] Suh, E., de Weck, O., Kim, I., and Chang, D., 2007, "Flexible Platform Component Design under Uncertainty," Journal of Intelligent Manufacturing, **18**(1), pp. 115-126.

[74] Kashkoush, M., and ElMaraghy, H., 2014, "Product Family Formation for Reconfigurable Assembly Systems," Procedia CIRP, **17**, pp. 302-307.

[75] Bryan, A., Wang, H., and Abell, J., 2013, "Concurrent Design of Product Families and Reconfigurable Assembly Systems," Journal of Mechanical Design, **135**(5), pp. 051001-051001-16.

[76] Agyapong-Kodua, K., Wahid, B. M., and Weston, R. H., 2011, "Towards the Derivation of an Integrated Process Cost-Modelling Technique for Complex Manufacturing Systems," International Journal of Production Research, **49**(24), pp. 7361-7377.

[77] Park, J., and Simpson, T. W., 2007, "Toward an Activity-Based Costing System for Product Families and Product Platforms in the Early Stages of Development," International Journal of Production Research, **46**(1), pp. 99-130.

[78] Park, J., and Simpson, T. W., 2005, "Development of a Production Cost Estimation Framework to Support Product Family Design," International Journal of Production Research, **43**(4), pp. 731-772.

[79] Wang, L., Song, B., Li, X., and Ng, W. K., 2007. "A Product Family Based Life Cycle Cost Model for Part Variety and Change Analysis," *International Conference on Engineering Design, ICED'07*, Paris, France, 28 - 31 August 2007.

[80] Johnson, M. D., and Kirchain, R., 2010, "Developing and Assessing Commonality Metrics for Product Families: A Process-Based Cost-Modeling Approach," Engineering Management, IEEE Transactions on, **57**(4), pp. 634-648.

[81] Chen, S., Wu, S., Zhang, X., and Dai, H., 2014, "An Evolutionary Approach for Product Line Adaptation," International Journal of Production Research, **52**(20), pp. 5932-5944.

[82] Hsiao, S.-W., and Tsai, H.-C., 2005, "Applying a Hybrid Approach Based on Fuzzy Neural Network and Genetic Algorithm to Product Form Design," International Journal of Industrial Ergonomics, **35**(5), pp. 411-428.

[83] Chandrasegaran, S. K., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F., and Gao, W., 2013, "The Evolution, Challenges, and Future of Knowledge Representation in Product Design Systems," Computer-Aided Design, **45**(2), pp. 204-228.

[84] Nanda, J., Thevenot, H. J., Simpson, T. W., Stone, R. B., Bohm, M., and Shooter, S. B., 2007, "Product Family Design Knowledge Representation, Aggregation, Reuse, and Analysis," AI EDAM, **21**(02), pp. 173-192.

[85] Tucker, C. S., and Kim, H. M., 2008, "Optimal Product Portfolio Formulation by Merging Predictive Data Mining with Multilevel Optimization," Journal of Mechanical Design, **130**(4), pp. 041103-041103.

[86] Tucker, C., 2014. "Quantifying the Relevance of Product Feature Classification in Product Family Design." *Advances in Product Family and Product Platform Design*. T. W. Simpson, et al., eds., Springer New York, pp. 147-177.

[87] Agard, B., and Kusiak, A., 2004, "Data-Mining-Based Methodology for the Design of Product Families," International Journal of Production Research, **42**(15), pp. 2955-2969.

[88] Moon, S. K., Simpson, T. W., Shu, J., and Kumara, S. R. T., 2009, "Service Representation for Capturing and Reusing Design Knowledge in Product and Service Families Using Object-Oriented Concepts and an Ontology," Journal of Engineering Design, **20**(4), pp. 413-431.

[89] Chen, Y.-M., and Wei, C.-L., 1997, "Computer-Aided Feature-Based Design for Net Shape Manufacturing," Computer Integrated Manufacturing Systems, **10**(2), pp. 147-164.

[90] Xue, D., and Dong, Z., 1997, "Coding and Clustering of Design and Manufacturing Features for Concurrent Design," Computers in Industry, **34**(1), pp. 139-153.

[91] Myung, S., and Han, S., 2001, "Knowledge-Based Parametric Design of Mechanical Products Based on Configuration Design Method," Expert Systems with Applications, **21**(2), pp. 99-107.

[92] Moon, S. K., 2008, "A Strategic Module-Based Platform Design Method for Developing Customized Families of Products and Services," Ph.D. thesis, The Pennsylvania State University, University Park, PA, USA,

[93] Chin, K.-S., Chan, A., and Yang, J.-B., 2008, "Development of a Fuzzy Fmea Based Product Design System," The International Journal of Advanced Manufacturing Technology, **36**(7-8), pp. 633-649.

[94] Huang, H.-Z., Bo, R., and Chen, W., 2006, "An Integrated Computational Intelligence Approach to Product Concept Generation and Evaluation," Mechanism and Machine Theory, **41**(5), pp. 567-583.

[95] Sun, J., Kalenchuk, D. K., Xue, D., and Gu, P., 2000, "Design Candidate Identification Using Neural Network-Based Fuzzy Reasoning," Robotics and Computer-Integrated Manufacturing, **16**(5), pp. 383-396.

[96] Zhang, F., and Xue, D., 2002, "Distributed Database and Knowledge Base Modeling for Concurrent Design," Computer-Aided Design, **34**(1), pp. 27-40.

[97] Zhang, L., Jiao, J., and Helo, P., 2006. "Integrated Product and Process Family Data Modeling for Product Lifecycle Management," *IEEE International Conference on Industrial Informatics*, 16-18 Aug. 2006, pp. 531-536.

[98] Jiao, J., Zhang, L., and Pokharel, S., 2006. "Process Platform and Production Configuration for Product Families." *Product Platform and Product Family Design*. T. Simpson, et al., eds., Springer US, pp. 377-402.

[99] Jiao, J., Zhang, L., and Pokharel, S., 2007, "Process Platform Planning for Variety Coordination from Design to Production in Mass Customization Manufacturing," Engineering Management, IEEE Transactions on, **54**(1), pp. 112-129.

[100] Wang, L., Fu, X.-y., and Zhong, S.-s., 2013. "Generalized Process Family Modeling Based on Process Platform." *Proceedings of 20th International Conference on Industrial Engineering and Engineering Management*. E. Qi, et al., eds., Springer Berlin Heidelberg, pp. 183-196.

[101] Bourell, D. L., Rosen, D. W., and Leu, M. C., 2014, "The Roadmap for Additive Manufacturing and Its Impact," 3D Printing and Additive Manufacturing, **1**(1), pp. 6-9.

[102] Ma, J., and Kim, H. M., 2014. "Predictive, Data-Driven Product Family Design," *ASME IDETC/CIE*, Buffalo, New York, USA, August 17-20.

[103] Khajavirad, A., and Michalek, J. J., 2008, "A Decomposed Gradient-Based Approach for Generalized Platform Selection and Variant Design in Product Family Optimization," Journal of Mechanical Design, **130**(7), pp. 071101-071101.

[104] Khajavirad, A., Michalek, J. J., and Simpson, T. W., 2009, "An Efficient Decomposed Multiobjective Genetic Algorithm for Solving the Joint Product Platform Selection and Product Family Design Problem with Generalized Commonality," Structural and Multidisciplinary Optimization, **39**(2), pp. 187-201.

[105] Hernandez, G., Allen, J. K., and Mistree, F., 2003, "Platform Design for Customizable Products as a Problem of Access in a Geometric Space," Engineering Optimization, **35**(3), pp. 229-254.

[106] Khire, R. A., and Messac, A., 2008, "Selection-Integrated Optimization (Sio) Methodology for Optimal Design of Adaptive Systems," Journal of Mechanical Design, **130**(10), pp. 101401-101401.

[107] Maidin, S. B., 2011, "Development of a Design Feature Database to Support Design for Additive Manufacturing (Dfam)," Ph.D. thesis, Loughborough University, Loughborough, UK.

[108] Thompson, M. K., Moroni, G., Vaneker, T., Fadel, G., Campbell, R. I., Gibson, I., Bernard, A., Schulz, J., Graf, P., Ahuja, B., and Martina, F., 2016, "Design for Additive Manufacturing: Trends, Opportunities, Considerations, and Constraints," CIRP Annals - Manufacturing Technology, **65**(2), pp. 737-760.

[109] "Unimagined Possibilities with Lasercusing," accessed 6 Jun 2016, http://www.concept-laser.de/en/industry.html

[110] "Industries & Markets," accessed 6 Jun 2016, http://www.eos.info/industries_markets

[111] "Cast Studies," accessed 6 Jun 2016, https://www.3dsystems.com/case-studies?field_industries_tid=All

[112] "3d Printers and 3d Printing News," accessed 6 Jun 2016, http://www.3ders.org/

[113] Abbas, O. A., 2008, "Comparisons between Data Clustering Algorithms," Int. Arab J. Inf. Technol., **5**(3), pp. 320-325.

[114] Cai, R., Zhang, Z., Tung, A. K. H., Dai, C., and Hao, Z., 2014, "A General Framework of Hierarchical Clustering and Its Applications," Information Sciences, **272**, pp. 29-48.

[115] Singhal, A., 2001, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., **24**(4), pp. 35-43.

[116] Biggio, B., Bulò, S. R., Pillai, I., Mura, M., Mequanint, E. Z., Pelillo, M., and Roli, F., 2014. "Poisoning Complete-Linkage Hierarchical Clustering," *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop*, Springer Berlin Heidelberg, Joensuu, Finland, 20-22 Aug 2014, pp. 42-52.

[117] Tibshirani, R., Walther, G., and Hastie, T., 2001, "Estimating the Number of Clusters in a Data Set Via the Gap Statistic," Journal of the Royal Statistical Society: Series B (Statistical Methodology), **63**(2), pp. 411-423.

[118] Kotsiantis, S. B., 2007, "Supervised Machine Learning: A Review of Classification Techniques," Informatica, **31**(3), pp. 249-268.

[119] Fatima, S., Mohanty, A. R., and Naikan, V. N. A., 2015. "Multiple Fault Classification Using Support Vector Machine in a Machinery Fault Simulator." *Vibration Engineering and Technology of Machinery*. J. K. Sinha, ed., Springer International Publishing, Switzerland, pp. 1021-1031.

[120] Khan, L., Awad, M., and Thuraisingham, B., 2007, "A New Intrusion Detection System Using Support Vector Machines and Hierarchical Clustering," The VLDB Journal, **16**(4), pp. 507-521.

[121] Li, H., Liang, Y., and Xu, Q., 2009, "Support Vector Machines and Its Applications in Chemistry," Chemometrics and Intelligent Laboratory Systems, **95**(2), pp. 188-198.

[122] Delgado, M. F., Cernadas, E., Barro, S., and Amorim, D., 2014, "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?," Journal of Machine Learning Research, **15**(1), pp. 3133-3181.

[123] Vapnik, V. N., 1999, "An Overview of Statistical Learning Theory," Neural Networks, IEEE Transactions on, **10**(5), pp. 988-999.

[124] Williams, C. B., Mistree, F., and Rosen, D. W., 2011, "A Functional Classification Framework for the Conceptual Design of Additive Manufacturing Technologies," Journal of Mechanical Design, **133**(12), pp. 121002-121002-11.

[125] Kaplan, R. S., and Anderson, S. R., 2004, "Time-Driven Activity-Based Costing," Harvard Business Review, **82**(11), pp. 131-140.

[126] Chansaad, A., Rattanamanee, W., Chaiprapat, A., and Yenradee, P., 2012. "Fuzzy Time-Driven Activity-Based Costing Model in an Uncertain Manufacturing Environment," *Proc. Asia Pacific Industrial Engineering & Management Systems Conference*, Phuket, Thailand, 2-5 Dec 2012.

[127] Sarokolaei, M. A., Saviz, M., Moradloo, M. F., and Dahaj, N. S., 2013, "Time Driven Activity Based Costing by Using Fuzzy Logics," Procedia - Social and Behavioral Sciences, **75**, pp. 338-345.

[128] Strano, G., Hao, L., Everson, R. M., and Evans, K. E., 2011, "Multi-Objective Optimization of Selective Laser Sintering Processes for Surface Quality and Energy Saving," Proceedings of the Institution of Mechanical Engineers Part: B-Journal of Engineering Manufacture, **225(**B9), pp. 1673-1682.

[129] Dutta, P., Boruah, H., and Ali, T., 2011, "Fuzzy Arithmetic with and without Using A-Cut Method: A Comparative Study," International Journal of Latest Trends in Computing, **2**(1), pp. 99-107.

[130] Leekwijck, W. V., and Kerre, E. E., 1999, "Defuzzification: Criteria and Classification," Fuzzy Sets and Systems, **108**(2), pp. 159-178.

[131] Jang, J. S. R., 1993, "Anfis: Adaptive-Network-Based Fuzzy Inference System," Systems, Man and Cybernetics, IEEE Transactions on, **23**(3), pp. 665-685.

[132] Babuška, R., 2003. "Neuro-Fuzzy Methods for Modeling and Identification." *Recent Advances in Intelligent Paradigms and Applications*. A. Abraham, et al., eds., Physica-Verlag HD, pp. 161-186.

[133] Chansaad, A., Rattanamanee, W., Chaiprapat, A., and Yenradee, P., 2012. "Fuzzy Time-Driven Activity-Based Costing Model in an Uncertain Manufacturing Environment," *Proc. Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference, Phuket, Thailand*.

[134] Esmalifalak, H., Albin, M. S., and Behzadpoor, M., 2015, "A Comparative Study on the Activity Based Costing Systems: Traditional, Fuzzy and Monte Carlo Approaches," Health Policy and Technology, **4**(1), pp. 58-67.

[135] Azeez, D., Ali, M. A. M., Gan, K. B., and Saiboon, I., 2013, "Comparison of Adaptive Neuro-Fuzzy Inference System and Artificial Neutral Networks Model to Categorize Patients in the Emergency Department," SpringerPlus, **2**, pp. 416.

[136] Mathur, N., Glesk, I., and Buis, A., 2016, "Skin Temperature Prediction in Lower Limb Prostheses," IEEE Journal of Biomedical and Health Informatics, **20**(1), pp. 158-165.

[137] Asnaashari, E., Asnaashari, M., Ehtiati, A., and Farahmandfar, R., 2015, "Comparison of Adaptive Neuro-Fuzzy Inference System and Artificial Neural Networks (Mlp and Rbf) for Estimation of Oxidation Parameters of Soybean Oil Added with Curcumin," Journal of Food Measurement and Characterization, **9**(2), pp. 215-224.

[138] Yim, S., and Rosen, D., 2012. "Build Time and Cost Models for Additive Manufacturing Process Selection," *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Chicago, IL, USA, 12-15 Aug 2012.

[139] Konak, A., Coit, D. W., and Smith, A. E., 2006, "Multi-Objective Optimization Using Genetic Algorithms: A Tutorial," Reliability Engineering & System Safety, **91**(9), pp. 992-1007.

[140] Guillaume, S., 2001, "Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review," Fuzzy Systems, IEEE Transactions on, **9**(3), pp. 426-443.

[141] Guillaume, S., and Charnomordic, B., 2011, "Learning Interpretable Fuzzy Inference Systems with Fispro," Information Sciences, **181**(20), pp. 4409-4427.

[142] "Traxxas Products," accessed 22 Jan 2015, http://traxxas.com/products

[143] Train, K. E., 2009, *Discrete Choice Methods with Simulation*, Cambridge University Press, Cambridge, UK.

[144] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-Ii," Evolutionary Computation, IEEE Transactions on, **6**(2), pp. 182-197.

[145] Martin, M. V., and Ishii, K., 1996. "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation," *Proc. ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, CA, USA, August 18-22.

[146] Kota, S., Sethuraman, K., and Miller, R., 1998, "A Metric for Evaluating Design Commonality in Product Families," Journal of Mechanical Design, **122**(4), pp. 403-410.

[147] Lopes, A. J., MacDonald, E., and Wicker, R. B., 2012, "Integrating Stereolithography and Direct Print Technologies for 3d Structural Electronics Fabrication," Rapid Prototyping Journal, **18**(2), pp. 129-143.

[148] "The World's First 3d Electronics Printer," accessed 5 June 2015, http://www.voxel8.co/

# Appendix A

## The Product Family of Nitro-Powered R/C Racing Cars

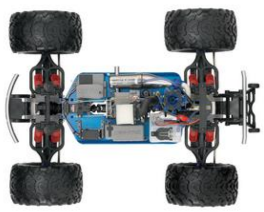
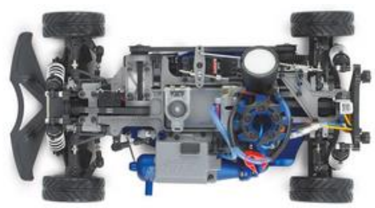
Traxxas's market segmentation grid of nitro-powered R/C land vehicles is shown in Table A-1. In the market segmentation grid, the market segments are plotted horizontally, while price/performance tiers are plotted vertically, and the market niche is defined in the intersection of each price/performance tier with each market segment. The maximum speed of the car increases from the left to the right in the market segment grid.

Table A-1: Market segmentation of Traxxas's nitro-powered R/C land vehicles

| | | | |
|---|---|---|---|
| High-end ($400-600) | Revo 3.3 ($599.95, TRX-3.3 Engine) | Slayer Pro 4X4 ($424.95/Amazon, TRX 3.3 Engine) | Nitro 4-Tec 3.3 ($449.99, TRX 3.3 Engine) |
| Mid-range ($300-400) | T-Maxx Classic ($399.99, TRX 2.5 Engine) | Nitro Splash ($349.99, TRX 3.3 Engine) | N.A. |
| Low-end ($200-300) | Nitro Stampede ($299.99, TRX Pro 1.5 Engine) | N.A | N.A |
| | Monster truck | Short-course race truck | On-road sedan |

Horizontal leveraging of the market set is shown in Table A-2, depicting the *high-end* tier which is included in all four market segments. The high-end cars in all four market segments use exactly the same engine.

Table A-2: Horizontal leveraging of the market segment (high-end tier)

| Monster truck (T-Maxx 3.3) | Short-course race truck (Slayer Pro 4X4) | On-road sedan (Nitro 4-Tec 3.3) |
|---|---|---|
|  |  |  |
|  |  |  |
|  |  |  |