UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A PLATFORM CASCADING METHOD FOR SCALE BASED PRODUCT FAMILY DESIGN

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By
JIJU ANDREWS NINAN
Norman, Oklahoma
2007

UMI Number: 3261114

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

A PLATFORM CASCADING METHOD FOR SCALE
BASED PRODUCT FAMILY DESIGN

A DISSERTATION APPROVED FOR THE
SCHOOL OF AEROSPACE AND MECHANICAL ENGINEERING

BY

Dr. Zahed Siddique, Chair

Dr. Kuang-Hua Chang

Dr. Kurt Gramoll

Dr. Suleyman Karabuk

Dr. Shivakumar Raman

Dr. Alfred G. Striz

Dedication

Dedicated to my father, late Mr. P.K. Ninan and my mother Mrs. Susan Ninan

ACKNOWLEDGEMENTS

First and foremost I offer my sincerest gratitude to my advisor, Dr. Zahed Siddique, who has supported me throughout this research with his guidance and knowledge. With his guidance, he induced in me, the necessary diligence and inspiration required to make this project successful. I could not have wished for a friendlier advisor than Dr. Siddique, who has always been very understanding, inspirational and patient in times of all my hardships. I thank you for giving me the opportunity to work with you over these years.

I would like to mention a special thanks to Dr. Suleyman Karabuk for all the help he has provided me in writing the optimization formulations. Without his guidance and his expertise in Mathematical Programming, I could not have achieved this goal.

I would also like to thank Dr. Kuang-Hua Chang, Dr. Kurt Gramoll, Dr. Shivakumar Raman, and Dr. Alfred G. Striz for serving in the committee and for their valuable suggestions and guidance in writing this dissertation. Their suggestions have helped me greatly in bringing this work in final form.

I would like to thank my family members, Yamini and my friends for the words of encouragement when I needed it. All these years spent away from home have been hard. But your thoughts and prayers have kept me focused in my goals. Thanks to Drs. Conrad and Mary Hornbuckle for providing me with a home away from home. With your support and encouragement following this path seemed so natural.

I would also like to thank all my teachers for their guidance over the years. Special thanks to my teacher and mentor Mr. V. Jayadevan for providing the mathematical foundation from which I have built my career.

I thank all my colleagues at the University of Oklahoma, ArvinMeritor and Hydril for all the inputs and suggestions. Last but not least, I would like to thank all the people who have helped me attain my dream of becoming an Engineer.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this dissertation, a product family design method for scale based products leveraged from multiple platforms is presented. A product family is a set of related products derived from a product platform. Product family design involves designing the platform and also leveraging the different product variants from the platform. A common approach to the product family design is to treat it as a design optimization problem, so that tradeoff analysis can be performed between commonality and individual product performance.

A product family based on a single platform may lead to poor performance of the product family. A better approach is to leverage the products from multiple platforms. This approach offers more challenges to the designer. The designer must now determine: the optimum number of product platforms that are required, the values of platform parameters for each platform, the products that are leveraged from each platform and the value of scale parameters.

In this dissertation, a Platform Cascading Method (PCM) is presented which is capable of designing the family of products based on multiple platforms. PCM is comprised of three stages: (1) Single platform stage (2) Evaluation stage and (3) Cascading stage. In PCM, the family is first leveraged using a single platform. The non platform scale based design problem has the structure of a Mixed Integer Non Linear Problem (MINLP) due to the combinatorial nature of the platform commonality parameters and continuous product parameters. Solving MINLPs are not straightforward and require high amount of expertise and time in solving the problem and hence transform to high product lead time. In PCM, the non platform specified product family design formulation is converted from a MINLP to a NLP by relaxing the platform commonality parameters to continuous parameters and then mathematically constraining to produce discrete results in the end.

In the evaluation stage, evaluation functions are used to evaluate the family of products leveraged from the platform. After Evaluation of the products leveraged, PCM uses a cascading formulation to generate subsequent platforms from the initial platform. Cascading generates new platform by converting the platform parameters from the previous platform to a scale parameter to leverage the set of products that have poor performance. The method is illustrated using two examples: (1) axial pump product family design and (2) universal electric motor product family design. The method can be easily implemented in gradient based optimization tools and can be used design scale based product families in a time efficient manner.

CHAPTER 1

PRODUCT PLATFORM SCALING ISSUES

Product family design features designing a family of products built around a common platform. The key element in successfully deriving variety and maximum commonality is the product platform design from which the product family is leveraged. In this chapter, a brief introduction is provided to product family concepts and product family design methods. Research questions and objectives addressed through this dissertation are presented along with a brief description of the proposed approach. In the last section, organization of the dissertation is provided.

1.1 Introduction to Product Family Design

Product development enterprises normally offer a range of products varying from low cost-low performance to high cost-high performance products to serve different market segments. Traditionally, the product varieties were individually designed and manufactured to suit the requirements of the particular market segment. Each product had different components even though they served the same or similar function. The product families lacked commonality among products in the portfolio. Lack of commonality among the different products resulted in high cost in design, manufacturing and inventory. These costs could be reduced or eliminated by sharing components and parts among the different family members. Many companies started (re)designing their product lines as a result of these advantages in using a platform to support the family.

A *product family* is a set of related products derived from a product platform, which is "a collection of the common elements, especially the underlying core technology, implemented across a range of products (Meyer and Lehnerd, 1997). A product family is comprised of a set of variables, features or components that remain constant from product to product (*product platform*) and others that vary from product to product.

There are two basic approaches to product family design (Meyer and Lehnerd, 1997) (1) top-down (proactive platform) wherein a company strategically manages and develops a family of products based on a product platform and its derivatives and (2) bottom-up (reactive redesign) wherein a company redesigns a group of distinct products to standardize components and improve economies of scale and scope. Based on the product differentiating factors, product families can be classified as (1) *modular product* *families* - wherein product family members are instantiated by adding, substituting, and or removing one or more functional modules from the product platform and (2) *scalable product families* - wherein scaling variables are used to “stretch” or “shrink” the product platform in one or more dimensions to obtain the different product variants.

Product family members or *product instances* are leveraged from the product platform to serve different market segments. Each family member is designed for a particular market or has a certain performance. Savings in costs certainly comes at the expense of loss in performance of individual products as forcing commonality among the family members results in products to under perform. Thus product family design is a trade off between cost and performance. The designers must also balance the commonality of the products in the family with the individual distinctiveness of each product in the family. Normally a product family design process includes: (1) designing the platform and (2) designing the individual product variants from the platform. Therefore, product family design should focus on the design of the entire family and platform, as well as the individual products. Several researchers have treated the design of product families as a design optimization problem. The advantage of this methodology is that designers can maintain a balance between commonality and cost. The platform and family members can simultaneously be optimized for performance, cost and commonality while designing the products.

1.2 Classification of Product Family Optimization Methods

The optimization formulations currently available for product family design can be classified as

(1) Single stage and Multi-stage optimization methods (Simpson, 2004)

(2) Platform-specified and Non platform-specified design (Simpson, 2004, Fellini *et al.*, 2006)

(3) Single platform and multi-platform design (Simpson, 2004)

1.2.1 Single Stage and Multi-Stage Optimization

Based on the number of stages involved in the design process, product family optimization methods can be categorized as (a) Single stage design optimization and (b) Multi-stage design optimization. Single stage (Figure 1a) approaches seek to optimize the product platform and corresponding family of products simultaneously, while multi-stage approaches (Figure 1b) optimize the platform first and then instantiate the individual products from the platform. Single stage optimization usually requires only one optimization run, but the size of the optimization problem increases tremendously as the number of parameters and number of products in the family increases. Multi-stage optimization breaks the larger problem into smaller sub-problems. They require at least one optimization run for determination of the platform and '*n*' optimization runs to leverage the '*n*' products in the family from the platform (Figure 1b).

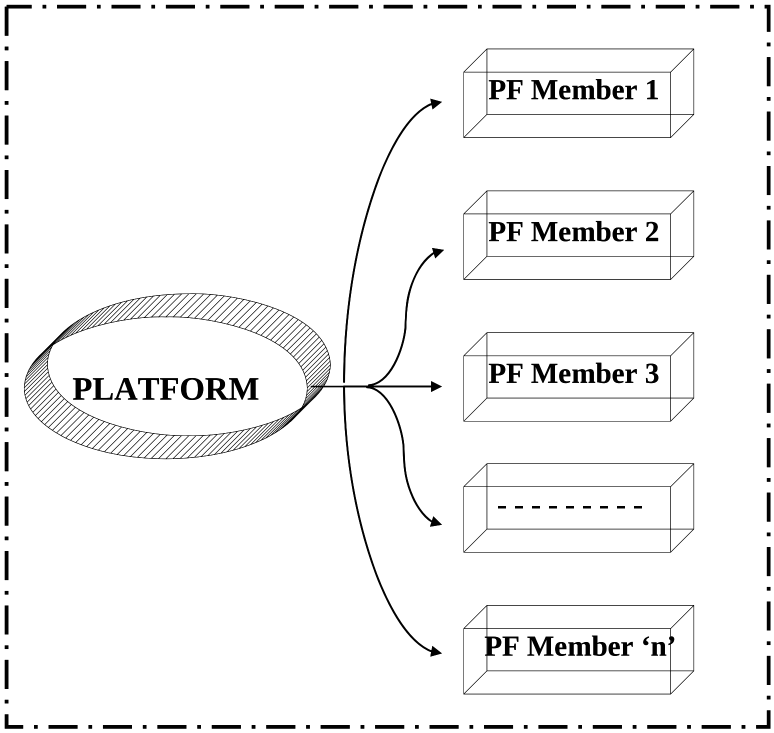


Figure 1.1a: Single Stage Optimization

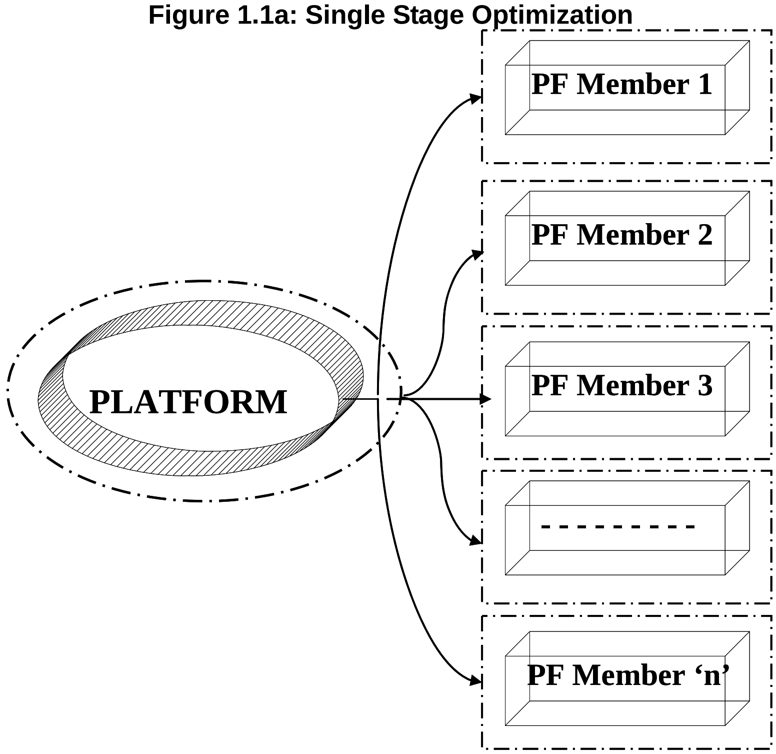
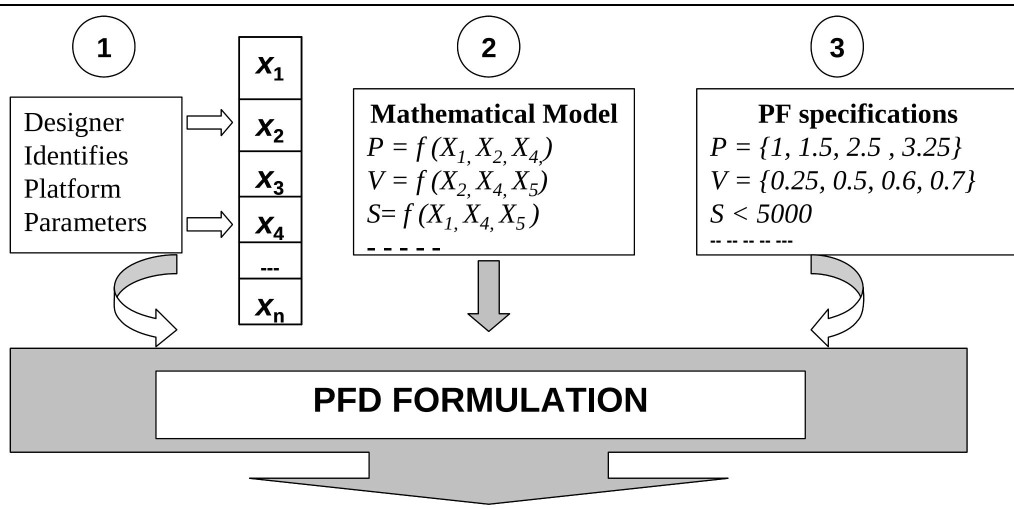


Figure 1.1b: Multi-Stage Optimization

1.2.2 *Platform Specified and Non-Platform Specified Optimization*

Another way of classifying the product family design optimization problem is according to the level of information provided by the designer as: (a) *platform-specified* and (b) *non-platform specified*. Sometimes the platform is specified by the designer, which means the values of the platform parameters and scale parameters for each product instances are determined using the formulation. When the platform is not specified by the user, the configuration of the platform also has to be identified, which involves determining the parameters that constitute the platform and their optimum value. The second class of problem (non-platform specified) is more difficult to solve, as the

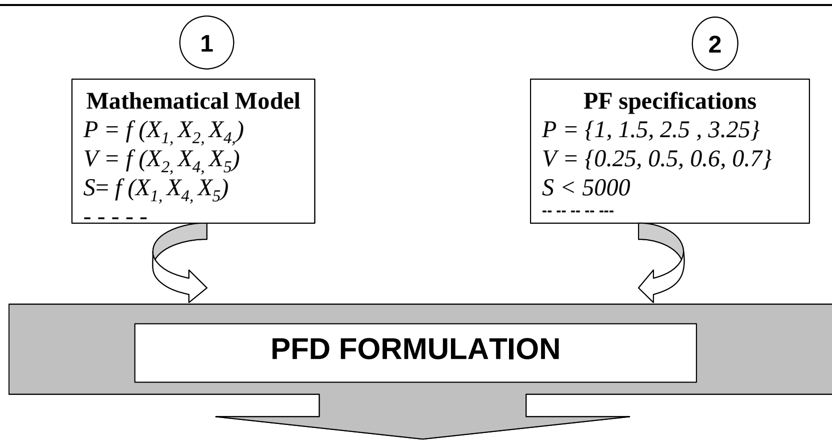


	Product 1	*Product 2*	*Product 3*	*Product 4*
x_1	56	53	61	67
x_2	***0.5***	***0.5***	***0.5***	***0.5***
x_3	12	14	15	13
x_4	***0.28***	***0.28***	***0.28***	***0.28***
--	----	------	----	---
P	***1***	***1.5***	***2.5***	***3.25***
V	***0.25***	***0.5***	***0.6***	***0.7***
S	***3500***	***3700***	***5000***	***5000***

Figure 1.2: Platform-Specified Product Family Optimization

formulation should determine the combination of platform parameters from all the possible combinations of product parameters and also determine their optimum values.

Figure 1.2 shows a platform specified product family design optimization formulation. The inputs to the formulation are (1) identification of platform parameters (2) the underlying mathematical model and (3) product family design specifications. The designer first selects the platform parameters from the product parameters $x_1, x_2, .. x_n$. In the case shown in Figure 1.2, x_2 *and* x_4 are identified by the designer as the platform parameters. The mathematical model relates the product performance to product parameters. The model also specifies the bounds for each product parameter. The product family specifications specify the performance and other requirements for each



	Product 1	***Product 2***	***Product 3***	***Product 4***
x_1	56	53	61	67
x_2	***0.5***	***0.5***	***0.5***	***0.5***
x_3	12	14	15	13
x_4	***0.28***	***0.28***	***0.28***	***0.28***
--	----	------	----	---
P	***1***	***1.5***	***2.5***	***3.25***
V	***0.25***	***0.5***	***0.6***	***0.7***
S	***3500***	***3700***	***5000***	***5000***

Figure 1.3: Non-Platform Specified Product Family Optimization

product variety. The formulation outputs the optimum value of product parameters for each product family member and their performance. In Figure 1.2, it can be seen that the platform parameters share the same value throughout the family.

In case of non-platform specified optimization (Figure 1.3), the designer does not identify the platform parameters; instead, the formulation determines the best combination of platform parameters from all the possible combinations of product parameters.

1.2.3 Single Platform Optimization and Multi platform Optimization

Early research on product family design was based on the assumption that all the product instances can be generated from a single platform successfully. But considerable loss in performance was noted in many product families as compared to individually designed products. To address this issue, the product families can be designed around multiple platforms, to minimize the loss of efficiency due to "commonalization". The additional design objectives in the case of multi-platform design are to decide the minimum number of platforms required to support the family and select the platform from which each product family member is leveraged. The platforms may be specified or non-specified as in the case of single platforms. The major differences between single platform optimization and multi-platform optimization are explained in Section 1.3.

1.3 Moving Towards Multi-Platform Design - Challenges

In a single platform approach all the products in the family are leveraged from the same platform. If a product parameter is a platform parameter, then it will be shared across all the products in the family. There is only one possible combination of parameter sharing within the family. Either the parameter is shared throughout the family (all the ‘n’ products in the family will have the same value for the platform parameters) or it will have unique values for different products in the family.

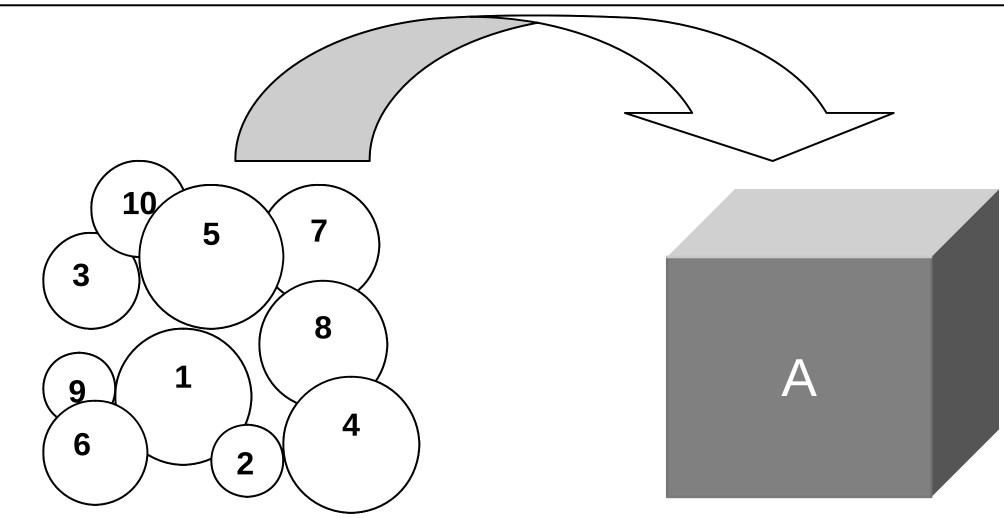


Figure 1.4: Single Platform Analogy

Single platform approach is analogous to the situation shown in Figure 1.4. The idea is to pack as many crystal balls into the box as possible while minimizing the possibility of them breaking during transportation. Here the balls are analogous to product parameters and the box represents the platform. Considering that each ball can either be included or excluded in the box, there are two possibilities for each ball. If there are ‘n’ balls there are ‘2^n’ possibilities, ignoring the different possible arrangements of the balls within the box.

The single platform approach may cause poor performance of individual product family members. A single platform might not be sufficient to leverage successfully all of the products in the family (Dai and Scott, 2005). Hence the situation of multi- platform approach arises. In the multi-platform approach the products are leveraged from two or more platforms so that the loss of performance due to commonalization can be reduced. The multi platform product family design optimization problem is relatively new to the research community and has not been studied by many researchers.

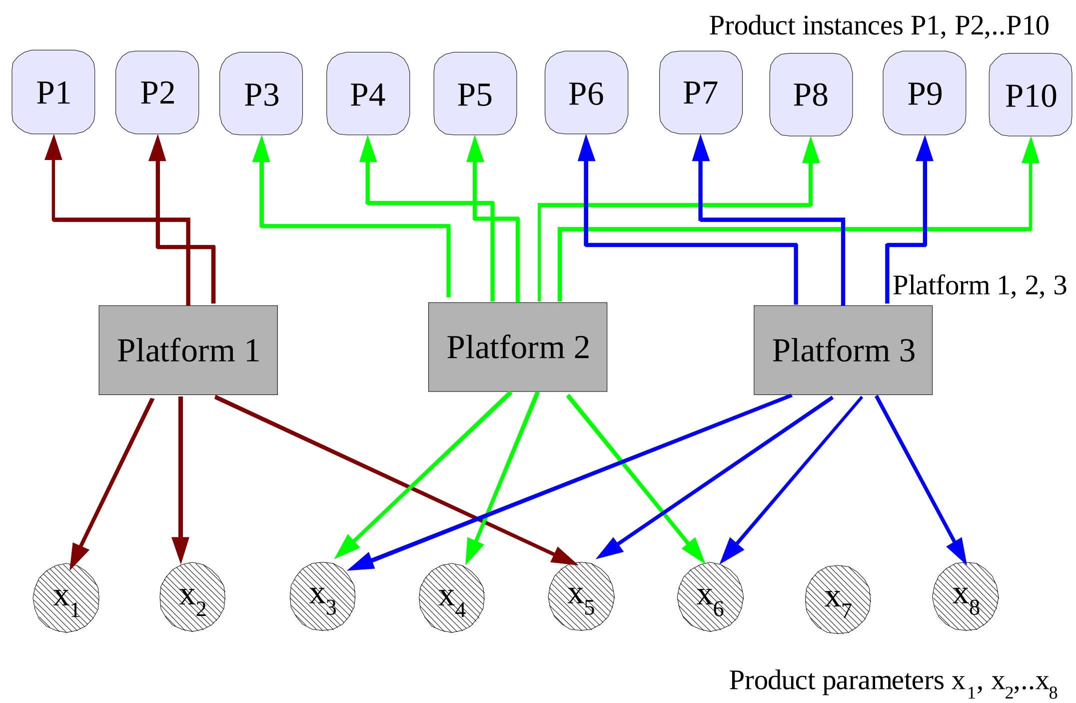


Figure 1.5: Hypothetical Multi-Platform Design with Three Platforms

Figure 1.5 shows a hypothetical situation for a scale based product family with ten product variants. There are eight design variables associated with the products. It is assumed here that platforms 1, 2, and 3 are sufficient to satisfy the entire range of products with minimal loss of efficiency. Product instances p_1 *and* p_2 are derived from platform 1, p_3, p_4, p_5, p_8 *and* p_{10} from platform 2, and p_6, p_7 *and* p_9 from platform 3. Here the product parameters x_1, x_2 *and* x_5 constitute platform 1, x_3, x_4 *and* x_6 forms platform 2 and x_3, x_4, x_6 *and* x_8 form platform 3. Here it is assumed that this combination of platform and scale variables will generate the family of products with minimal loss of performance from the target.

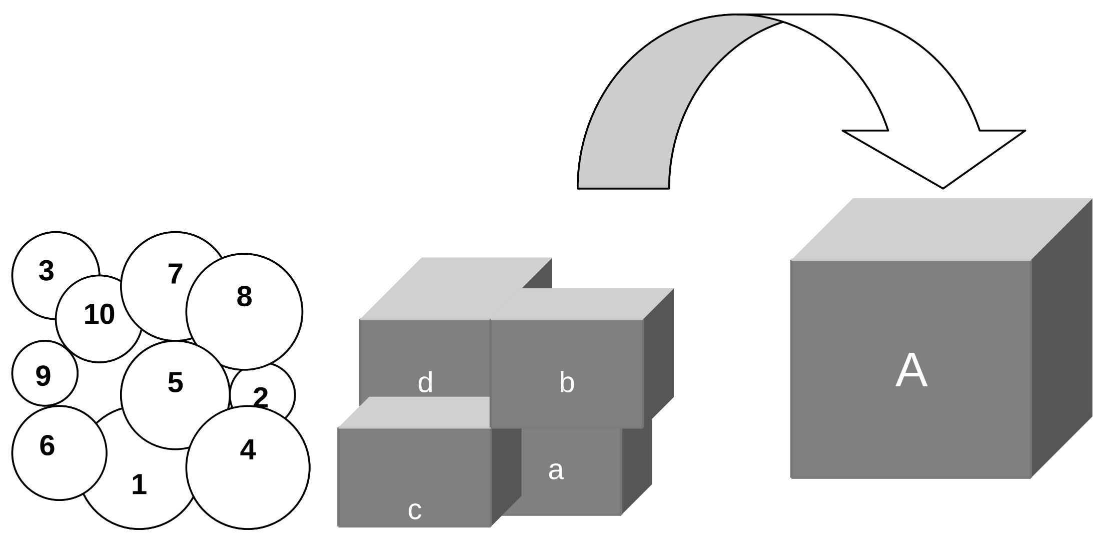


Figure 1.6: Multi-Platform Analogy

In multi-platform design, the challenges are to find: (1) the minimum number of platforms that can serve the family of products with minimal loss of performance (2) the platform from which each product is leveraged and (3) which parameters constitute the platform parameters for each platform. The same analogy shown in Figure 1.4 can be extended to the multi-platform case as shown in Figure 1.6. Here the crystal balls are first packed into smaller boxes which in turn are placed in a bigger box. The multi-platform problem adds another dimension of combinatorial nature to the single platform problem.

1.4 Objectives and Research Questions

The overall objective of this research is *to develop a more efficient product family design approach so that scale based product families based on multiple platform(s) can be designed*. The focus of the dissertation will be on multi-platform product family design. Manufacturing techniques/processes relating to product families are not considered in this dissertation; instead, it will be assumed that increasing commonality will lead to increased cost savings. The first research question that will be addressed through this dissertation is:

> ***RQ1)*** *How do we represent a family of products supported by a single platform using a mathematical programming model and identify a solution technique, so that tradeoffs between commonality and performance can be performed to support product family design?*

A mathematical programming model, capable of addressing trade-off decisions between commonality and performance of product family members will be used to design the product family. Specifications of product family members will be captured using objectives and constraints in the mathematical model. The model will force the commonality of the platform parameters in the corresponding product instances using commonality constraints. The model will be capable of exploring all possible combinations of platform commonality to determine the optimum platform configuration(s) and product instances.

RQ2) *How do we extend the mathematical model to design product families supported by multiple platforms?*

The second research question is related to extending the approach to multi-platform design. Some of the issues that need to be addressed in multi-platform design are (1) determining the number of platforms that are required to support the family (2) identifying the platform from which each product can be leveraged (3) determining the configurations of each platform and the values of platform parameters (4) determining the values of the scale variables for each product instance and (5) establishing measures and comparing the product family derived from the multiple-platform design to that of single platform design.

As the number of platforms increases, it is natural to assume that product development cost also increases. The cost savings associated with commonality in product family design will not be studied due to time constraints. It will be assumed that each design variable has equal preference in being treated as a platform. That is, minimizing the loss of performance due to commonality will be the criteria for the selection of parameters as the platform parameters. The following are the sub-research questions that follow from Research Question 2.

RQ2.1) How do we extend the single platform representation as sub-problem for deciding configuration of the multiple platforms?

RQ2.2) How do we extend the mathematical model to evaluate the optimum number of platforms?

RQ2.3) How do we maintain a relationship between the different platforms so that commonality between the different platforms can be established?

RQ2.4) What are the ideal scenarios that determine when a multi-platform approach should be used?

Following are the objectives related to modeling and the solution of a multiple-platform scale-based product family design problem that are addressed in this research.

O1) *Develop a mathematical programming model that represents a scale-based product family in terms of decision variables (design variables), constraints and objectives.*

O2) *Capture the commonality of the platform components/parameters*

O3) *Extend the model to identify the platform parameters resulting from performing a trade-off between commonality and performance for different possible platform configurations and identify the best platform configuration for given set of requirements.*

O4) *Identify solution techniques/algorithms to solve the model so that a product family satisfying the design requirements can be generated.*

O5) *Evaluate the product family in terms of the deviation of the actual performance from the target performance and generate multiple platforms if necessary.*

O6) *Extend the formulation to evaluate the optimum number of platforms and their configurations in the case of multi-platform design.*

O7) *Evaluate the performance of the product family*

1.5 Proposed Multi-Platform Cascading Approach

The approach taken to solve the multi-platform design problem is described in this section. The inputs to the formulation are the parametric description of the products, constraints related to the performance of the products, and the underlying mathematical model relating the product parameters to the constraints and objectives. The proposed approach consists of three stages:

(1) Single platform stage

(2) Platform evaluation and

(3) Cascading

Figure 1.7 shows the flow chart of the proposed approach.

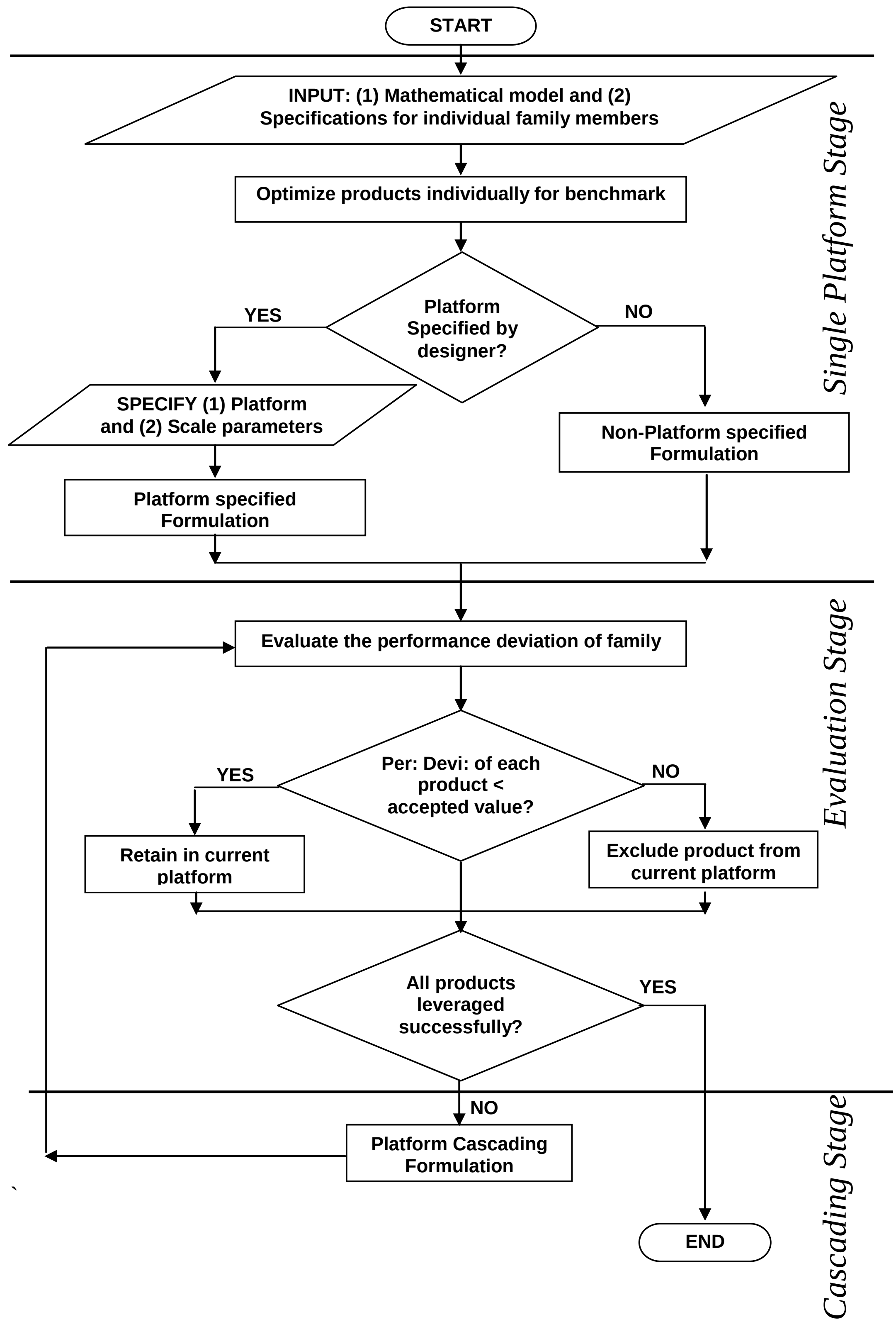


Figure1.7: Flow Chart of the Proposed Approach

Stage 1: Single Platform Stage

The starting point for the proposed approach is a product family based on a single platform assumption. First, the products are designed under the assumption that a single platform is sufficient to scale all the products in the family.

The two possible design cases involve (1) the designer specifying the platform parameters and (2) the formulation exploring the optimum platform for the family of products. To capture the commonality of parameters, equality constraints will be used that will constrain platform parameters to take the same value for all the products in the family. The optimization formulation will try to find the optimum values of the both the scale parameters and platform parameters while minimizing the deviation of performance from the target.

In the first stage, since a single platform assumption is used, if a parameter is shared it will be shared among all of the products in the family. In a platform-specified case, the commonality will be modeled only for the parameters selected by the designer as platform parameters. In a non-platform specified case, all the parameters have the option to be a platform or scale parameter. The formulation will be capable of exploring several combinations so that maximum commonality can be achieved while minimizing the loss of performance. To accomplish this, the equality constraints need to be turned on while evaluating a particular parameter as a platform parameter, and turned off while they become scale parameters. Binary decision variables, which correspond to each product parameter, will be used to turn the platform commonality equality constraints ON/OFF. For a single platform approach, *'n-1'* equality constraints will be required to link all the *'n'* products for each product parameter. The objectives in the formulation will be to maximize commonality while minimizing the loss of performance due to commonality.

Stage 2: Platform Evaluation Stage

In this stage, the resulting product family from the previous stage is evaluated. The loss of performance due to commonality of each product or the family as a whole will be evaluated. The benchmark for comparison is the set of products designed individually for maximum performance. A threshold value for performance deviation will be selected and the products with a higher deviation will be segregated out.

Stage 3: Platform Cascading Stage

In the platform cascading stage, one of the platform parameters obtained in the first stage will be selected and relaxed to a scale parameter. The objective is to ensure commonality between the first and the succeeding platform while generating improved products. The resulting products have to be evaluated as in Stage 2. The platform cascading can lead to three possible scenarios: (1) all of the products show improved performance and the loss of performance is within acceptable values compared to the benchmark, (2) some products show improvement while others do not, and (3) none of the products improve their performance. In the first case, further design iterations will not be required and the product family design process is complete, comprising of two platforms, one cascaded from the other. In the second case, the designer may wish to segregate the non-conforming products and cascade the platform again until all of the products have acceptable performance. In the third case, several iterations may be carried out until a product family with acceptable performance can be reached.

1.6 Organization of Dissertation

To facilitate this discussion, an overview of the chapters in this dissertation is shown in Figure 1.8. Having laid the foundation by introducing the research questions and objectives for the work in this chapter, the next chapter contains a literature review of research related to product family design and different scale-based product family design methods. A matrix to differentiate the existing work based on the approach, modeling assumptions, number of supported platforms and solution technique employed is

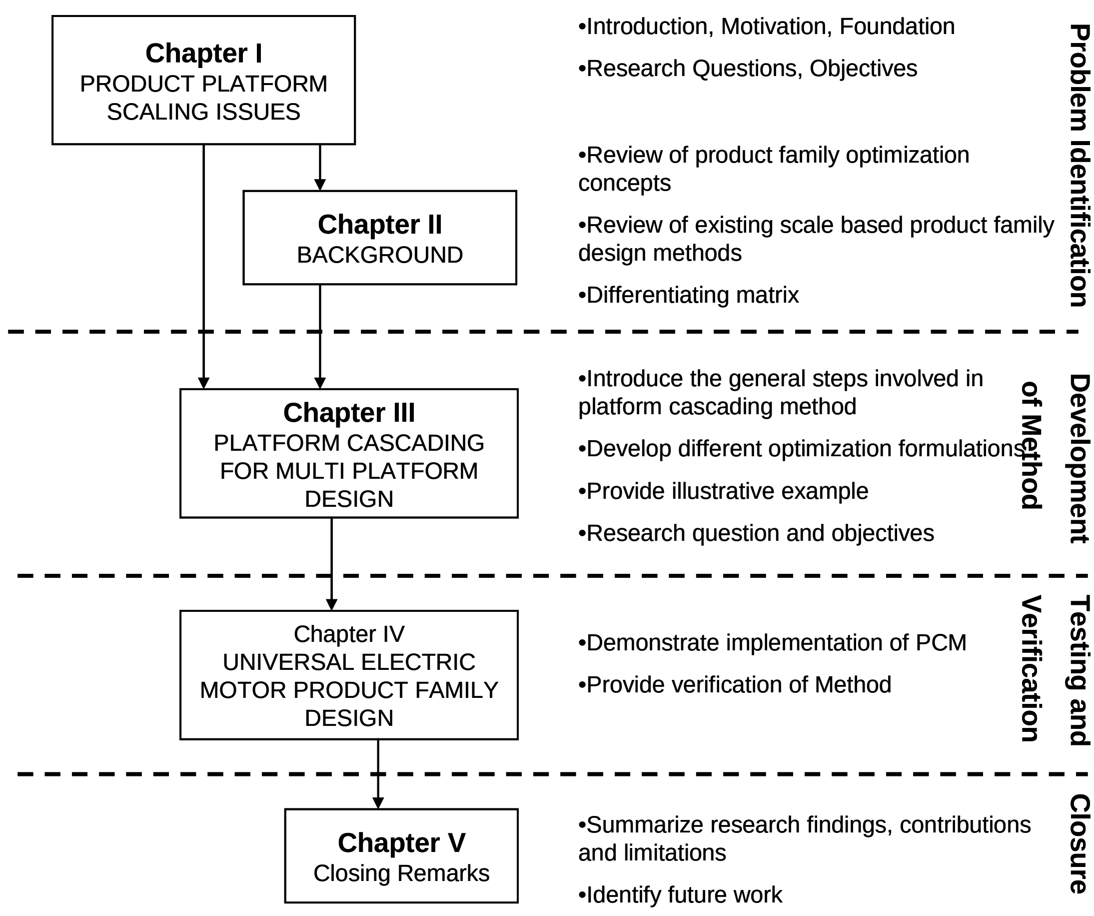


Figure 1.8: Overview of Chapters

provided towards the end of Chapter 2. The matrix helps to establish the uniqueness of the work presented in this dissertation.

The general objective and steps of Platform Cascading Method (PCM) is presented in Chapter 3. Section 3.1 gives an overview of the PCM. Section 3.2 highlights the issues and design problems associated with the different activities of PCM. An illustrative example of an axial pump product family design is presented in Section 3.3. This illustrative example is used to explain the general steps of PCM presented in Section 3.4. Section 3.4 also explains how the research questions are answered using PCM and the objectives that are achieved though this research.

Application of PCM to the design of a Universal Electric Motor product family to illustrate the use of PCM for scale based product family design is presented in Chapter 4. The results obtained from PCM are compared to that of existing work in Section 4.3.

Chapter 5 is the final chapter and contains a summary of the work, emphasizing answers to the research questions and objectives of the work. Limitations of PCM and possible avenues of future work are discussed in Sections 5.3 and 5.4 respectively.

CHAPTER 2

BACKGROUND

This section presents a review of relevant literature. Section 2.1 is a review of product family concepts present in literature. Section 2.2 focuses on product family optimization methods. Since the focus of the proposal is on scalable product family design, in-depth reviews of existing scale-based product family design methodologies are presented in Section 2.2.1. The different methods are explained in detail and differences between them are captured using a differentiating matrix in Section 2.2.2.

2.1 Product Family Concepts

One of the earliest development and application of the product family concept was reported by Lehnerd and Meyer (1997). In 1971, Black & Decker launched the Double Insulation Program to redesign the universal motor field assembly. Universal motor field assembly was one of the key sub-sets of Black & Decker's universal motor. Their goal was to create a single basic motor design that could be adapted to produce a broad range of power to serve infrequent household users, frequent household users, and even professional tradesmen (Meyer and Lehnerd, 1997). Martin and Ishii (2004) started to investigate commonality, modularity, and standardization. Simpson (1998) related change in form and function to highlight mutability, modularity, and robustness which he suggested as the core characteristics of product families. Chen *et al.* (1994) suggested designing flexible product architectures to enable small product changes to increase product variety. Stadzisz and Henrioud (1997) described a methodology for the integrated design of product families and assembly processes. Stadzisz and Henrioud (1997) defined “A product family is considered as a set of similar products whose main functions are identical”. However, product variations and their assembly plans demand flexibility in a common assembly process for the product family. Stadzisz and Henrioud (1997) proposed reduction of this required flexibility in the assembly process as a design criterion because it required more capital investment and brought productivity reduction. The basic concept of family of products or multi-products approach is to obtain the biggest set of similar products through the most standardized set of base components and production processes.

2.1.1 *Modular and Scalable Product Architecture*

Modularity is the concept of separating a system into independent parts or modules, which can be treated as logical units. Ulrich and Tung (1991) provide a summary of different types of modularity. They also stated that modularity depends on two characteristics of a design: (1) Similarity between the physical and functional architecture of the design and (2) Minimization of incidental interactions between physical components. Complete modularity is achieved when there is a one-to-one correspondence between the physical and functional architectures. In their book "Product Design and Development" (Ulrich and Eppinger, 2004), the authors described the different stages of product development. Ulrich and Eppinger (2004) focus on different aspects of product architecture and how they can be used to develop modular products. According to the authors, a modular architecture implements one or a few functional elements in their entirety and the interactions between the chunks of modules are well defined and are generally fundamental to the primary functions of the product.

The opposite of a modular structure is an integral architecture. Modular chunks allow changes of a product to be made to only a few functional elements that have little relation to other elements in the product. Products built around modular product structures can be more easily varied without adding tremendous complexity to the manufacturing system. Modular product architecture also facilitates component standardization. Ulrich and Eppinger (2004) also propose a four-step methodology for establishing the product architecture. Another approach to designing product families is to develop a parametrically scalable product platform (Rothwell and Gardiner, 1990). This platform can then be scaled in one or more parameters to develop the product family. Sabbagh (1996) in his work shows Boeing successfully scaling the design to come up with a family of products satisfying different capacities and flight ranges. Similarly, Rolls Royce scaled its RTM322 engine to realize a family of engines with different thrust outputs and specific fuel consumption RG90. Meyer and Lehnerd (1997) explains how Black & Decker designed a universal motor platform that could be scaled along its stack length to generate a wide variety of power outputs while significantly increasing economies of scale and reducing labor costs. Naughton *et al.* (1997) explains Honda's intention of building a world car with an ingenious frame that allows the auto maker to shrink or expand the overlying car without starting from the ground up by coming up with a platform--by far the most expensive part of a new car--that can be bent and stretched into markedly different vehicles. Other industrial applications of both modularity (Sanderson and Uzumeri, 1997; Pine *et al*., 2000; Pine, 1993; Feitzinger and Lee, 1997; Kobe, 1997; Wilhelm, 1997) and scale-based approaches (Meyer, 1997; Rothwell and Gardiner, 1990; Sabbagh, 1996) can be found in the literature.

2.1.2 Commonalization

Commonality is one of the primary objectives to develop platforms for a set of similar products. Current approaches to providing families of products through the use of common platforms mainly focus on increasing commonality and standardization. Wheelwright and Clark (1992) suggested designing "platform projects" that are capable of meeting the needs of a core group of customers but are easily modified into derivatives through addition, substitution, and removal of features. McGrath (1995) also emphasized a well designed product platform is very important for a family of products. At the same time, parts commonality had been viewed as a means of cost reduction.

McDermott and Stock (1994) in their paper described how the use of common parts could shorten the product development cycle for savings in both time and money in the manufacturing process. MacDuffie *et al.* (1996) investigated how variety affected manufacturing within the automotive industry by studying empirical data; he reported that part complexity has a negative impact on productivity. From the literature it is evident that increasing commonality in the platform for a set of products has obvious benefits. In the following sub-sections, some of the terminologies related to product family design are presented.

2.1.3 Standardization

The main concept behind developing a common platform is to provide a standard platform for a set of similar products. Thus, the concept of standardization certainly applies to platform commonality. Standardization and platform commonization have a very strong relationship between them. To develop a common platform, standardization is one of the required characteristics. In the context of platform commonization, standardization has to be achieved in several levels: "Standardization of components ", "Standardization of module interfaces ", "Standardization of assembly process".

2.1.4 Modularity

The concept of modularity generally applies to the relationship between functions and structure. In many cases, the concept of functional modules does not apply readily when the architecture is integral. The general lesson appears to be that, to achieve both variety and standardization, it is necessary to go beyond the conventional view of functional and structural modules to include assembly and other life-cycle considerations. This broader view of modularity enables the isolation of required variety into appropriate module types (structural, functional, assembly, etc.).

2.1.5 Mutability

Mutability is the capability of the system to be contorted or reshaped in response to changing requirements or environmental conditions without a change in function. This is the characteristic that enables a platform to be used across models. Robustness and mutability are two of the characteristics that are desired in common platforms. Although some of the product variety concepts do not apply to platform commonization, most of them do apply and are related.

2.1.6 Robustness

Robustness implies insensitivity to small variations and does not dictate a change in form or a change in function (Simpson, 1998). Robustness is a characteristic that is desired in components, assembly process, and module interfaces for common platforms. From the common platform view point robustness refers to insensitivity to small variations for: (1) components - As an example, a small change in the length of the platform will not require any change in the components/subassemblies of the platform; (2) module interfaces - As an example, the interfaces with the engine and platform will not require any change when a different type of engine will be used with the platform; (3) assembly process - As an example, a small change in the dimension in a component can be accommodated using the same assembly line without any changes.

2.2 Product Family Optimization

Several researchers have used optimization approaches to design a family of products to arrive at a suitable a product platform and also the product varieties. Optimization approaches are used to perform trade-offs between commonality (the underlying platform) and the performance of the product variants. Most researchers studied the effect of commonality on individual product performances like cost, efficiency, strength, reliability of the product variants. Other performances like environmental effects have also been considered in product family optimization. Ortega *et al.* (1999) presented a decision support approach to perform trade-off analyses in the design of a family of environmentally conscious oil filters by modeling performance in terms of economical and environmental goals. They determined a baseline oil filter models to suit an existing family of vehicles, which can meet environmental requirements at competitive costs.

Optimization approaches have been used to design both modular and scale based product families. In case of modular product family design, Allada and Jiang (2002); Blackenfelt (2000); Cetin and Saitou (2004); Chang and Ward, Fujita (2002); Fujita *et al.* (1993, 1996, 1998, 1999, 2003); Fujita and Ishi (1997); Fujita and Yoshida (2004); Rai and Allada (2003); Kokkolaras *et al.* (2004), optimization approaches were primarily used to: (1) identify functional and variational modules (2) optimize module interfaces (3) optimize the modular platforms and (4) optimize module diversions. Stone *et al.* (2000) presented a heuristic method to identify modules for modular product architectures, which was later extended to identify functional and variational modules. Allada and Jiang (2002) used Dynamic Programming model to arrive at a module configuration for an evolving family of products. Blackenfelt (2000) used robust design techniques to maximize profit and balance commonality within a family of lift tables. Fujita *et al.* (2001) developed a simulated annealing technique for optimizing module diversions for the case of television receiver circuit product family. Since scale-based product family design is the focus of this thesis, the existing scale-based product family design methods are examined in detail in next section.

The product family optimization approaches reported have been applied to a variety of sample problems. These example product family design problems fall under the category of (1) *consumer products,* such as knives (Rai and Allada, 2003), drills (Li and Azarm, 2002), nail guns (Nelson *et al.,* 2001), vacuum cleaners (Jiang and Allada, 2001), and automobile systems (Fellini e*t al.,* 2002; Kokkolaras *et al.,* 2004), (2) *industrial products,* such as chillers (Hernandez *et al* 2001), flow control valves (Farrell and Simpson, 2003), electric motors (Simpson *et al.*, 2001; Nayak *et al*., 2002; Messac *et al.*, 2002; Dai and Scott, 2005), and axial displacement pumps (Bhandare and Allada, 2006), and (3) *aerospace related products,* such as aircraft (Fujita and Yoshida, 2001; Simpson and D'souza, 2004), and spacecraft (Gonzales *et al.*, 2000). Some of the methods use simple analytical models to represent the relation between product parameters and performances while some require complex design and synthesis tools. A classification of the different product family example problems can be found in (Scott *et al.*, 2006).

Efforts are underway to develop a product family test bed comprised of different product family example problems (Allada *et al.*, 2006). Some of the example problems consider a non-uniform market demand for the product variants, while others assume a uniform demand for all the products in the family. Both the examples used in this dissertation assume a uniform market demand. A comprehensive review and classification of product family optimization methods can be found in Simpson (2003).

2.2.1 Product Family Design Methods for Scale-Based Families

In this section, existing scale-based product family design problems are presented. These scale-based product families are examined in detail to understand the existing approaches in terms of modeling, capabilities, solution algorithms, and limitations. This helps to differentiate the work presented in this diissertationwith that of existing methods. A matrix differentiating the different approaches is presented Section 2.2.2 (Table 2.2). The matrix captures the main differences between different methods based on the product family concepts introduced in Chapter 1 and also the differences in the optimization approach adopted. The different scale-based methods investigated are presented in the following sub-sections.

2.2.1.1 Product Platform Concept Exploration Method (PPCEM)

PPCEM (Simpson *et al.*, 2000) is a multi-stage method for design of scale-based product families. The inputs to the PPCEM are the (1) overall design requirements and (2) identification of the platform and scale variables. The formulation returns the optimized product platform and the product family instances. Design of a ten-electric motor family is used as an example to demonstrate the method as presented in the literature. The task is to design a family of ten electric motors with Torque = {0.05, 0.10, 0.125, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.5} Nm and each having a 300 Watt power output and sharing a common platform. The different steps involved in PPCEM are explained here in the context of the case study:

1. *Create the market segmentation grid:* A market segmentation grid shows the division of the market into different segments (Meyer and Lehnerd, 1997) and is used to identify the leveraging opportunities for a platform to generate products

that cater to different segments. Different strategies like horizontal leveraging, vertical leveraging, and beachhead approach are presented in the literature. In the example problem, a vertical leveraging is used. The universal electric motors are intended to be used in the Low Cost - Low Performance, Mid-Range and High Cost-High Performance power tools.

2. *Classify factors and ranges:* Map the overall design requirements and market segmentation grid into appropriate factors and identify corresponding ranges for each. For the sample problem the torque and power requirements and constraints for the individual motors are listed. The ranges of the design variables are also fixed at this stage.
3. *Build and validate meta models:* This step is optional and is used if mathematical or simulation model of the products are computationally expensive. The mathematical model for the sample problem is presented in Appendix A. The underlying mathematical model is not computationally expensive.
4. *Aggregate product platform specifications:* In this step, the target means of performance of the platform are set by finding the mean of the performance of the individual motors. Then, the standard deviations of the performances of the products are also found. In the motor example, the stack lengths of the motor are selected as the scale parameter. The mean and standard deviation of stack length which result in a mean Torque of 0.2425 Nm and standard deviation of 0.13675 Nm (calculated from the individual motor requirements) are evaluated.
5. *Generate the platform and variants:* Generation of the platform and the product variants is a two-stage process. At first, a decision support problem is formulated

so as to find the mean and standard deviation of the scale variables and which can satisfy the goals as closely as possible. The solution also returns the values of the platform variables. This is the first stage of the design process. From this, the range of the scale variables are found out [- 3 , +3]. In the next stage, compromise DSP is formulated to derive the individual members of the family using the platform. The platform variable values are held to those found in the first stage and the scale variable values which can satisfy the individual product performance for each product are found out. This process is repeated as many times as there are products in the family. In the sample problem, there are ten motors in the family; hence, the process is repeated ten times to instantiate the motors from the family. Therefore, the total design process involves at least 1+10 =11 optimization runs.

Assumptions in the formulation

Several assumptions are taken in PPCEM. In Step 4, the mean and standard deviation of the scale variables are found using the following assumptions:

1. The mean of the performance corresponds to the value of the function that describes the performance at the mean of the scale variables. For the sample problem, the mean power, mean efficiency, and mean mass are calculated as the power, efficiency, and mass, respectively, for the mean length. This condition is true only in the case of linear functions, while the underlying function is highly non-linear.

i.e. if $h = f(x_1, x_2, x_3, \ldots x_n)$ then ... (2.1a)

$$\mu_h = f(\mu_{x_1}, \mu_{x_2}, \mu_{x_3}, \ldots \mu_{x_n}) \quad \text{... (2.1b)}$$

2. The standard deviation of performance is approximated using the first-order Taylor series approximation, assuming the deviation is small. For example, the standard deviation of torque is related to the standard deviation of stack length by using the equation.

$$\sigma_T = \left| \frac{\partial T}{\partial \mu_L} \right| \sigma_L \quad \text{... (2.2)}$$

2.2.1.2 Variation Based Methodology for Product Family Design (VBPDM)

VBPDM (Nayak *et al.*, 2002) is an extension to PPCEM. In PPCEM, the platform and scale variables were identified by the designer. The designer had to use engineering knowledge or use trial and error to select the scale and platform variables.

VBPDM does not require the designer specifying the platform. VBPDM utilizes variational methods to identify the platform. The family members are then instantiated from the platform. Like PPCEM, VBPDM is also a two-stage approach, with the first stage being the identification of the platform and its parameter values. In the second stage, the different product instances and instantiated from the platform using the scale variables. In the platform selection process, a decision support problem is formulated to find the mean and standard deviation of the design variables that result in the range of performance of the product family. A multi-objective model using Goal Programming (Winston, 1994) is used. The target means and standard deviations of the performance are calculated from the design specifications. The deviation of the actual mean and standard deviations of the family is captured using deviation variables. These deviation variables are then minimized so as to bring the actual performance close to the target values. The mean of performance is approximated using Equations 2.1(a) and b). The standard deviation of performance is calculated using the Taylor series approximation in Equation 2.2 extended to the multi- variable case as

$$\sigma_h = \sqrt{\left(\frac{\partial h}{\partial x_1}\right)^2 \sigma_{x_1}{}^2 + \left(\frac{\partial h}{\partial x_2}\right)^2 \sigma_{x_2}{}^2 + \ldots + \left(\frac{\partial h}{\partial x_n}\right)^2 \sigma_{x_n}{}^2} \qquad \text{... (2.3)}$$

The commonality goal tries to reduce the standard deviation of each design variable to a target value of zero. The deviation from the target is captured using deviation variables and is minimized in the objective function. The different degrees to which the commonality goal is satisfied by the different system variables provides an indication of which variables are to be made platform. The ratio of the standard deviation to the mean is used to make this decision. A threshold value of 10% of the mean value is considered a small enough value for a design variable to be considered a platform parameter. This means that the required range of target performance can be achieved using a small variation in the design variables or, in other words, fixing the value of these design variables to the mean value will not result in a large performance loss across the family.

The second stage is similar to that of PPCEM. The platform variables are held to the mean values obtained in the first stage and product instances are derived from the platform. The universal motor product family design case study was performed as in the case of PPCEM. The radius of the motor, the thickness of the stator, and the number of turns of the armature wire were found to be suitable scale parameters. VBPDM does not consider any manufacturing or process parameters in the optimization process. In

PPCEM, the scale length was selected as the platform parameter as it offered several manufacturing advantages and, hence, greater cost savings. The authors also compared the values of performances of the resulting families from the two methods. The VBPDM motors showed better efficiencies and also reduced mass. VBPDM also required at least 11 optimization runs to solve the sample problem.

2.2.1.3 Multi-Criteria Optimization in Product Platform Design

Nelson *et al.* presented a multi-criteria optimization model to take trade-off decisions in product family designs [Nelson *et al.* (2001)]. The authors showed how to generate the Pareto set in the case of product family design, where each product family member has different objective functions. The concepts were demonstrated by implementing the method in a two-member nail gun product family design problem. No generic method was presented which could be applicable to all product family members. The product family consisted of two nail guns. The nail gun A is an industrial quality gun for use by professional carpenters, and nail gun B is an entry level, less expensive model. Model A is the flagship model, and the objective in its design is to maximize the size of the nail that can be driven into the wood. Model B is intended for the casual user, and the design objective is to maximize user comfort by minimizing the recoil that the user experiences.

Figure 2.1 shows the Pareto set for the case of a nail gun two-product family. f_A and f_B are the objective functions for nail gun A (Maximize the size of the nail) and gun B (Minimize the recoil), respectively. Point 'X' shows the null platform point which represents the individual optimum for guns A and B with no commonality. The extreme points (f_A^*, f_B) and (f_A, f_B^*) are the solution of the optimization problem with only one of the scalar functions f_A or f_B as the objective and enforcing commonality of parts.

The distance between the utopia point and the null platform point is an indication of the cost of commonality. This gives an understanding of the cost of commonality while

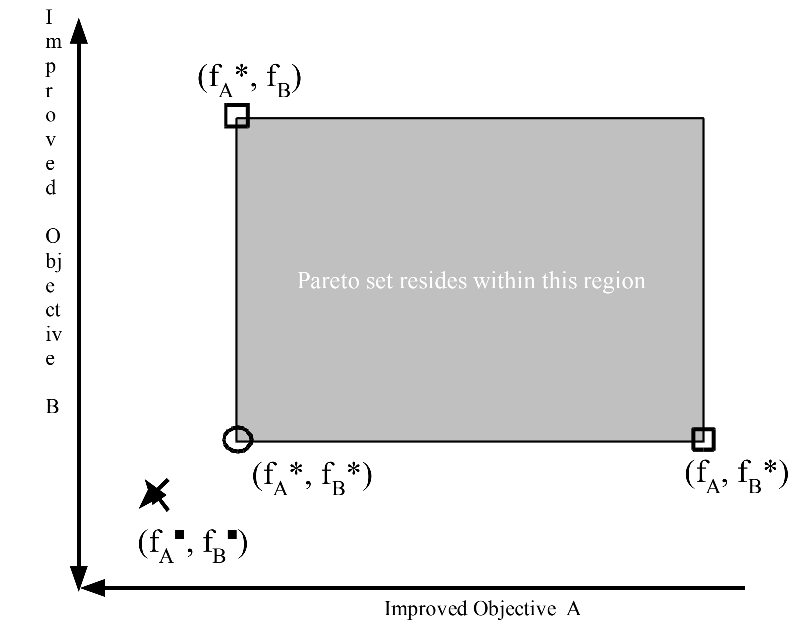


Figure 2.1: Bounds on the Pareto Set (Adapted from Nelson *et al*, 2001)

sharing different components. The disadvantage of the method is the difficulty in visualizing and generating a Pareto set when there are more than two products in the family and when many combinations of platform and scale variables are possible.

2.2.1.4 Product Family Penalty Function Using Physical Programming (PFPF)

In this method, Messac *et al.* (2002) use physical programming (Messac, 1996) for product family design. The difference between PFPF and PPCEM is that, in PPCEM, the scale variables of the product family need to be known prior. In PFPF, the scale factors are identified first.

In this approach, Messac *et al.* (2002) extended PPCEM by introducing a Product Family Penalty Function (PFPF) to aid in the platform decision process. During optimi-zation, PFPF will penalize the parameters that are not common throughout the product family while optimizing the desired objectives. This allows the identification of design variables to be kept common. The parameters that can be easily held to a constant value without affecting design objectives are grouped together to form the platform. The authors showed the use of Physical Programming for product family design. Physical Programming (Messac, 1996) has the capability of handling multi-objective optimization problems in a simple and user friendly way. Instead of weighting different objectives, physical programming lets the user specify the ranges of different degrees of desirability for different objectives. The authors presented two methods: (1) multiple-formulation method and (2) single formulation method. The variation of design variables for the family is captured using the formula

$$\mathrm{var}_i = \sqrt{\sum_{j=1}^{k} \frac{\left(x_i^j - \overline{x_i}\right)^2}{k}} \qquad \text{... (2.4a)}$$

$$\text{Where } \overline{x_i} = \sum_{j=1}^{k} \frac{\overline{x_i}}{k} \qquad \text{... (2.4b)}$$

x^j_i is the i^{th} parameter for j^{th} product.

1. Multi-formulation method: First, the motors are individually optimized without any commonality of parameters to study the optimum configuration and performance. Different optimization runs are carried out to minimize the varia-tion of each design parameter and still attain the target performance. Other para-meters are left unconstrained and the performance loss is observed. The design variable that causes the largest decrease in performance is selected as the scaling

variable. For the sample UEM design problem, radius was selected as the scale variable using the formulation explained above.

2. Single formulation method: In the single formulation method, all the design constrains and objectives are considered at the same time. For the motor example, the single formulation method optimizes the performance of the family of motors while minimizing the variation of each design variable throughout the family.

The authors reported that using both the methods, radius had the highest percentage variation and, hence, the best candidate for selection as the scale variable.

2.2.1.5 Product Platform Design Through Sensitivity Analysis and Cluster Analysis

Dai and Scott (2005) presented a method for product family design using cluster analysis and sensitivity analysis. He presented a multiple-platform design method where design variables may be shared among variants using any possible combination of sub-sets. Sensitivity analysis is performed to help select the candidate platform design variables. Then cluster analysis is performed on each design variable candidate to evaluate the performance loss due to commonalization and then to determine the platform configuration.

Following are the steps involved in product family design using the method:

1. Design products individually, obtaining optimal design solutions for each of the individual product variants without any platform constraints.
2. Perform sensitivity analysis for each design variable with respect to overall design performance.
3. Perform cluster analysis to group design variables, incorporating the sensitivity information acquired in Step 2.
4. Select variables for commonalization and fix the platform by determining their values.
5. Optimize all the product variables in the family by determining the values of the remaining variables.
6. Compare the product family design solutions obtained in Step 1 to determine if the performance loss from commonalization is allowable. If the performance loss is unacceptable, consider a different cluster (Dai and Scott, 2005).

The method was applied to the universal electric motor product family design problem. The authors presented results that showed motors with improved efficiency and lesser mass than that of PPCEM motors. The motors were based on multiple-platform rather than single platform in case of PPCEM. For the specific example the following steps/sub-steps were reported by the authors:

1. Optimize the 10 motors individually (10x1= 10 Optimization runs).
2. Compute sensitivity of each design variable for 10 motors (10x8x2=160 optimization runs).

3. Generate 10 quadratic approximation curves to the represent variation of objective functions for each of the 8 parameter (8x10 = 80 approximation curves).
4. Cluster the design parameters using the data in the above steps and then derive the platform.
5. Use the platform to generate 10 motors that meet specific target performance (10x1=10 optimization runs).

It is evident that the suggested method is impractical to implement in the case of product family design problems with many design variables and variants. As the problem size increases, the complexity of the problem will also tremendously increase. A more straightforward method needs to be developed which is easier and practical to use.

2.2.1.6 Assessing Variable Levels of Platform Commonality Within a Product Family Using Multi-Objective Genetic Algorithm

Simpson and Dsouza (2004) presented a product family design approach using genetic algorithm. The presented approach simultaneously designs the family of products while considering varying levels of commonality within the product family. The presented approach is a single-stage optimization method where the platform parameters and their values are simultaneously arrived at in a single optimization run. The method was applied to the design of three family-general aviation aircraft to accommodate 2, 4, or 6 people and satisfy certain customer requirements. The approach is a Genetic Algorithms-based extension of PFPF presented in Section 2.2.1.4.

Genetic algorithms are modified to include the commonality controlling genes in the chromosome and the commonality functions in the fitness function. If there are 'n' design variables and 'm' products in the family, then there are L= "n + mn" genes in the chromosome. The first 'n' genes are the commonality controlling parameters corresponding to each design variable. These genes take value of 0 or 1. A gene value of 1 denotes that the corresponding design variable is shared across the family. The variation of design variables captured using the PFPF is included in the fitness function of GA and minimized. Mutation and Crossover operators were used to generate an offspring design from parent chromosomes.

The disadvantages of this approach are the inherent disadvantages of genetic algorithms. Genetic algorithms are unsuitable for large problems (Goldberg, 1999) and, hence, cannot be used when there is large number of product family members or there are many design variables. Genetic algorithms are heuristic in nature and do not guarantee optimum solution. Moreover, GA have to be fine-tuned for each problem by adjusting parameters like population size, crossover and mutation rate, etc. which can vary from problem to problem. The advantage is that they are very easy to implement and can be parallelized to speed up the search.

2.2.1.7 Commonality Decisions in Product Family Design

The distance between the different points gives the loss of performance due to commonality for different platform configurations. They identified the different component possibilities and modeled the distance between the corresponding points to null platform design and constrained it to be less than a specified factor. In other words, loss of performance due to commonality should be less than a user specified value. The objective function consisted of the sum of all the possible shared components represented by $\sum_{(i,j)} \eta_{ij}^{pq}$. This represents possible sharing of components *'p'* and *'q'* between products *'i'* and *'j'*.

The term $\sum_{(i,j)} \eta_{ij}^{pq}$ is computed by the equation

Table 2.1: Platform Selection Formulation (Fellini *et al*)

Max:

$$\sum D_{\alpha}(x_i^p - x_j^q) \quad \forall p, q \in P, (i, j) \in S^{pq},\ p < q$$

Subject to:

$$g^p(x^p) \leq 0$$

$$h^p(x^p) = 0,$$

$$f_p(x^p) \geq (1 - L^p) f^{p,0}$$

$$\sum_{(i,j)} \eta_{ij}^{pq} = \sum S^{pq} - \sum_{(i,j)pq} D_o(x_i^p - x_j^q) \qquad \text{... (2.5a)}$$

$$\text{Where } D_0(x_i^p - x_j^q) = \begin{cases} 0 & if \quad x_i^p = x_j^q \\ 1 & otherwise \end{cases} \qquad \text{... (2.5b)}$$

To address the combinatorial nature D_0 was approximated using a differentiable function D_α

$$D_\alpha(x_i^p - x_j^q) = 1 - \frac{1}{\left(\dfrac{x_i^p - x_j^q}{\alpha}\right)^2 + 1} \qquad \text{... (2.6)}$$

Function D_α captures the distances between the designs in terms of commonality. After the problem is solved, the values of candidate platform parameters are compared.

They are assumed to be shared if their difference is between a specified tolerance. This step identifies the platform or sharing in case of different products. Once the component commonality is established, a new formulation is executed minimizing the distance between the null platform set and the Pareto set corresponding to the selected platform S_{pq}. The commonality between the parameters is established as hard constraints.

The method introduced by Fellini *et al.* (2006) has similarities with the Platform Cascading Method introduced in this thesis. PCM relies on platform cascading to generate multiple platforms which is an unique approach to product family design. The idea of converting the discrete platform commonality variables to continuous variables to enable execution in a gradient based optimizer is employed in both the formulations. But the formulations differ vastly in approach and modeling. The following are major differences between PCM and the one reported by Fellini (2006):

1. In Fellini *et al.* (2006), the commonality between different components are identified between components in corresponding product pairs, extending it to all possible components for all possible product pairs (Figure 2.2). The colored strike through in Figure 2.2 represents the actual sharing of components in the case of (a) Fellini et al. and (b) PCM. In case of Fellini *et al*. sharing of parameters between just any two product pairs are possible. This can lead to a large number of possibilities to be modeled for product families with many components and many family members. Moreover the sharing of components between just two product pairs as opposed to the entire family might not result in

$$\cancel{X_{11} = X_{12}},\ X_{11} = X_{13},\ X_{12} = X_{13}$$

$$X_{21} = X_{22},\ X_{21} = X_{23},\ \cancel{X_{22} = X_{23}}$$

$$\cancel{X_{31} = X_{32}},\ X_{31} = X_{33},\ \cancel{X_{32} = X_{33}}$$

(a) Fellini *et al.*

$$X_{11} = X_{12} = X_{13}$$

$$\cancel{X_{21} = X_{22} = X_{23}}$$

$$X_{31} = X_{32} = X_{33}$$

(b) PCM

Figure 2.2: Modeling of Commonality and Possible Sharing in Cases of (a) Fellini *et al.* and (b) PCM

real manufacturing advantage. For a family *‘m’* products there are ‘$\frac{m\,(m\text{-}1)}{2}$’ possible commonalities corresponding to each parameter. In PCM, commonality is modeled for all parameters/components in as a single constraint relating all the products considered. The latter approach ensures that if a parameter is shared it is shared across all the products in the platform.

2. Fellini *et al.* (2006) performs the design in two stages: (1) Identification of component sharing and (2) generation of products from the identified platforms. The different stages are employed in PCM to determine the products that are leveraged from each platform. Once the products that will be leveraged from the current platform are established, a single stage optimization (single platform formulation, platform formulation) simultaneously generates the platform and leveraged products.

3. In Fellini *et al.* (2006) the loss of performance due to commonality is treated as a constraint in the platform decision stage to arrive at possible platforms. In PCM, the loss of performance due to commonality is used to decide the products that will be leveraged from each platform.
4. In PCM commonality exists between the different platforms due to cascading, whereas in Fellini *et al.* the different platforms are not interrelated.

2.2.1.8 Axial Pump Product Family Design

Bhandare and Allada (2006) introduced the axial pump product family design problem and later made it available to the product family research community through the product family design test bed (Allada *et al.,* 2006). Bhandare and Allada (2006) introduced the problem as a platform-specified problem. The design objectives were (1) to determine the optimum number of platforms and (2) to evaluate the optimum value of platform parameters and scale parameters, required to successfully leverage a family of five Axial Piston pumps. The authors evaluated the "loss of performance due to platforming" by attaching a cost function to the customer dissatisfaction due to variation of performance from target. The continuous product parameters were discretized and then the cost function was evaluated at different points (exhaustive enumeration) to evaluate the loss of performance for different values of product parameters and different number of platforms. They considered a non-uniform demand for product variants. The demands for each product variants were captured into the cost function for computing the total cost of product variants. Since the platform and scale parameters were identified by the design, it can be classified as a platform-specified multi-platform design method.

The Axial Pump product family design problem with some modifications is used as an illustrative example in Chapter 3.

2.2.2 Differentiating Matrix for Existing Scale-based Product Family Design Methods

Table 2.2 shows a matrix differentiating the different approaches presented in Sections 2.2.1 to 2.2.8. The matrix captures the main differences between different methods based on the product family concepts introduced in Chapter 1, the differences in the optimization formulation adopted, and solution algorithm used. In the matrix, the following acronyms are used: SS= Single Stage optimization process, MS= Multi-Stage optimization process, Prob= Probabilistic optimization algorithm, Det= Deterministic solution algorithm, SP= Single Platform design, MP= Multiple-Platform design, UEM= Uniform Electric motor, GAA = General Aviation Aircraft, ASF= Automotive Side Frame and NG= Nail Gun, APF= Axial Pump Family

Table 2.2: Differentiating Matrix for Existing Scale-Based Product Family Design Methods

Authors	***Stages***	***Platform Specified***	***No. of Platforms***	***Solution Algorithm***	***Case Study***	***Remarks***
						SP= Single Platform, MP= Multi-Platform, Det = Deterministic, Prob= Probabilistic, UEM= Universal Electric motor, GAA = General Aviation Aircraft, NGF= Nail Gun Family, ASF= Automotive side frame, APF= Axial Pump Family
(1) Simpson *et al.(2000)*	MS	Yes	SP	Det	UEM	Platform to be identified by the designer, Several approximations in the model, 11 optimization runs required
(2) Nayak *et al.(2002)*	MS	No	SP	Det	UEM	Platform configuration determined by the formulation. Two-stage optimization process involving many optimization runs, Many approximations made in the model
(3) Nelson *et al.(2001)*	MS	Yes	SP	Det	NGF	For generation of Pareto curve in case of product family design
(4) Messac *et al.(2002)*	SS	Yes	SP	Det	UEM	Formulation determines the scale variables, Linearizing assumptions made, Several optimization runs required
(5) Dai & Scott(2005)	MS	No	MP	Det	UEM	Very long and tedious design process, uses sensitivity analysis and clustering
(6) Simpson *et al.(2004)*	SS	No	SP	Prob	GAA	Combines PFPF and Genetic Algorithms, Based on probabilistic solution technique
(7) Fellini *et al.(2004)*	MS	NO	SP	Det	ASF	Extension of Nayak *et al.*, It's a multi-stage design process capable of identifying the platform configura-tion, Relaxes the MINLP using an approximation function
(8) Bhandare & Allada(2006)	N/A	Yes	MP	N/A	APF	Uses exhaustive enumeration

2.3 Summary

In this chapter, background information related to product families, product family design, and product family optimization was presented. Different scale-based product family design methods reported were presented in Section 2.2.1. A table differentiating the different approaches was presented in the beginning of Section 2.2.2. Some of the concepts presented in the background information were used in developing the model presented in Chapter 3. The Axial Pump design example (Bhandare and Allada, 2006) and the universal motor example presented by Simpson *et al.* (2001) are used as illustrative examples in this dissertation.

CHAPTER 3

PLATFORM CASCADING FOR MULTI-PLATFORM DESIGN

In this chapter, a platform cascading method will be introduced for multi-platform design. A general formulation applicable to all scalable product families is presented in Section 3.4. The general steps are illustrated using an axial pump family design problem. The axial pump design problem is presented in Section 3.3.

3.1 Platform Cascading for Multi-Platform Design–Overall Approach

In most cases, a single platform is insufficient to design a family of products while using the platform approach. A single platform approach assumes that when a com-ponent or a product parameter is shared, it is shared across all products in the family. As the number of products in the family increases or as the portfolio of different products varies considerably, a single common platform approach may lead to inferior product families. A common platform that can serve the entire family of products can cause some product family members to perform poorly. In a multi-platform approach, the family members are leveraged from more than one platform so that products with minimal loss of performance can be derived. Cost efficiency of a single platform design may be higher compared to a multi-platform design due to the fact that an increase in the number of the platforms will lead to an increase in cost of the derived product family. In a multi-platform design, it is therefore necessary to design the family of products using an optimum number of platforms. Also in the case of multi-platform design, the combination of products that are leveraged from each platform and the configuration of each platform that leads to a family of products with minimal loss of performance need to be decided.

In this section, a cascading method for multi-platform design will be presented. The proposed method is a three-stage design process. The first step of the design approach is to design the entire family of products based on single platform approach (Single platform stage). The performance of the family members is then evaluated by comparing them with the benchmarks using predetermined criteria. Benchmark products have the same specification as the corresponding family members, but they are designed individually (without commonality). Family members that perform poorly are segregated and separated out from the current platform. This stage of the design process is referred to as the Evaluation stage. In the last stage of the design process, the initial platform is cascaded by relaxing one of the platform parameters to a scale parameter to arrive at a new platform to support the products that were separated out. Cascading the platforms helps to attain commonality between different platforms and, hence, is assumed to achieve higher cost savings. In the cascading approach, all of the platform parameters will share the same value for different platforms. The resulting products are again evaluated and the platform is again cascaded if necessary. The design process is iterative and can be continued until a family of products with acceptable performance can be reached. The method is applicable only to scale-based product families.

The specifications of product family members and the underlying mathematical model are specified by the designer. The mathematical model is usually comprised of the range of possible values of the design parameters and parametric relation between the design parameters and the responses. The Platform Cascading Method returns the parametric description of the product family members, the configurations of different platforms, the platform from which each product is leveraged, the performance of each product family member, and their performance loss due to commonality.

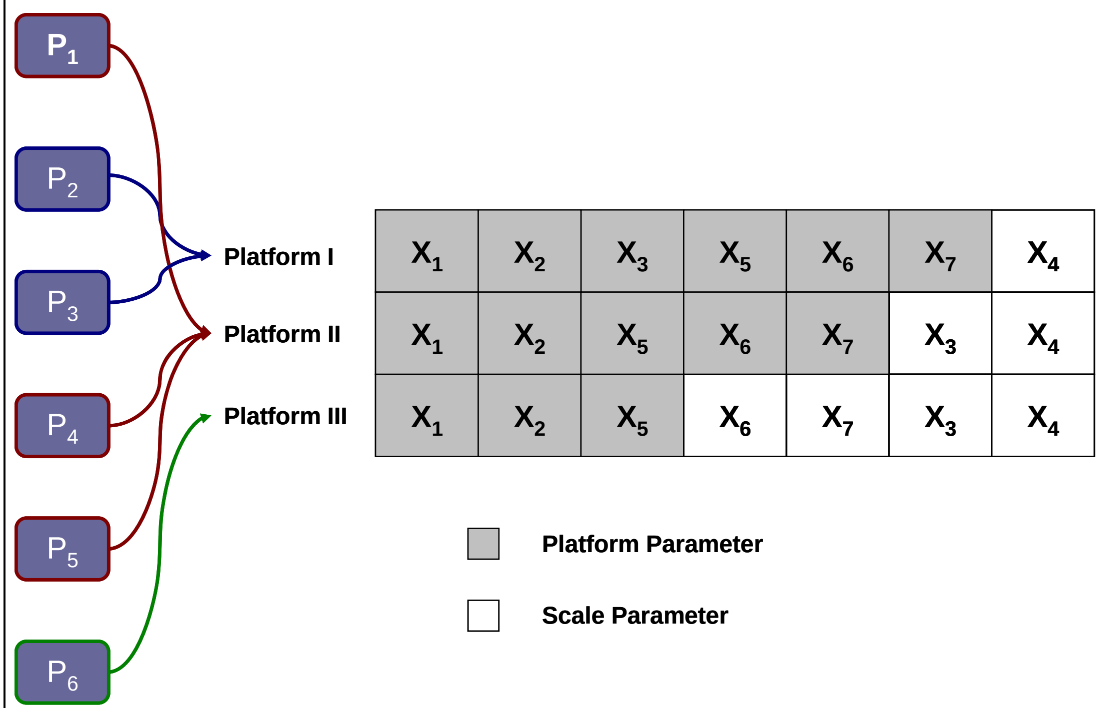


Figure 3.1: Platform Cascading

The three stages in the design process and the mathematical foundations for the proposed method are explained in detail in the following sub-sections. The method is explained with reference to a hypothetical multi-platform, scale-based product family developed using the cascading approach as shown in Figure 3.1. The hypothetical product family is a scale-based product family comprising of six products $\{P_1, P_2, .. P_6\}$. The product parameters related to the family of products are $x_1, x_2, .. x_7$.

In scale-based product family architecture, each product instance $\{P_1, P_2, ..P_m\}$ of the family can be uniquely and completely described by the same set of product parameters $x_1, x_2, .. x_n$. These parameters describe the attributes of the physical components present in the products. If the values of any of the parameters are constant throughout the family (in the case of single platform) or a sub-set of products (in the case of multi-platform), the parameter is said to be a platform parameter. Ideally, one would like to hold all the parameters relating to a particular component constant to constitute a platform; however, Nelson *et al.* (2001) shows that holding any of the parameters constant can still result in manufacturing advantages.

The parameters related to the entire product family can be represented using x_{ij} representations. Parameter x_{ij} represents the i^{th} product parameter for the j^{th} product. Extending the above notations to the family of products represented in Figure 3.1, the entire family of products can be represented using the vector of design parameters (X)

$$X = \begin{pmatrix} x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16} \\ x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26} \\ x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36} \\ x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46} \\ x_{51}, x_{52}, x_{53}, x_{54}, x_{55}, x_{56} \\ x_{61}, x_{62}, x_{63}, x_{64}, x_{65}, x_{66} \\ x_{71}, x_{72}, x_{73}, x_{74}, x_{75}, x_{76} \end{pmatrix}$$

The design task is to find the value of the parameters in X that result in an optimum product family with maximum commonality and minimal loss in performance. The following are the three stages in the proposed design process: To arrive at the optimum design, the Platform Cascading Method uses several optimization formulations at different stages of the design process. Optimization formulations help to perform trade-offs between commonality and performance and arrive at the optimum design points.

Stage 1: Single Platform Stage

The starting point of the formulation is designing the entire product family using a single platform. In the single platform stage, all the products are leveraged using one platform. Platform parameters have the same value for all of the products in the family.

There are two possible cases for the single platform case:

1) The platform-specified case

In a platform-specified case, the designer specifies the platform parameters. The aim of the optimization formulation is to arrive at an optimum X ($\dot{X}$) which enforces commonality of the specified platform parameters throughout the family and also minimizes the loss of performance due to commonality.

2) Non-platform specified case.

In the case of non-platform specified formulation, the aim of the formulation is to explore different levels of commonality and perform trade-offs between commonality and the loss of performance of family members. The goals are to arrive at a suitable product platform and to leverage the product family members using the platform.

In the case of scale-based product families, platform commonality can be modeled mathematically for an entire family by using the following equality condition.

$$x_{ij} = x_{ij+1} \vee j, j \neq m \text{ If } i \in x_p \qquad \text{... (3.1)}$$

Where x_p is the set of platform variables.

In Figure 3.1, x_1 is a platform parameter, hence the platform commonality can be captured throughout the entire product family by extending the Equation 3.1 to the example case as

$$x_{11} = x_{12} = x_{13} = x_{14} = x_{15} = x_{16} \qquad \text{... (3.2)}$$

To represent the sharing of parameters, a set of binary decision variables (0, 1) corresponding to each product parameter will be introduced. These platform commonality decision parameters are represented by y_i.

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix}$$

$$y_i = \begin{cases} 1 \text{ when the parameter is a platform parameter} \\ 0 \text{ when the parameter is a scale parameter} \end{cases}$$

y_i parameters can be used to turn ON/OFF the commonality of the corresponding parameters. In the platform specified case, y_i values of the platform parameters are set to 1 to enforce commonality. In the non-platform specified case, the y_is help to explore the levels of commonality by turning platform commonality ON/OFF for different parameters. The formulation will try to maximize the commonality by performing a trade-off between the maximum number of platform parameters and the loss of performance of family members.

For the purpose of establishing benchmarks for the evaluation stage and also to provide a good starting point for the single platform formulation, products instances are individually optimized, subject to design and performance requirements of the corresponding product instance. The individual optimization formulation tries to find the optimum value of the product parameters in the case of each product variety. The formulation tries to minimize undesired performances and maximize desired performances subject to the performance requirements/constraints of each product variety. The formulation will be run as many times as there are products in the family. The individual optimum corresponds to the best performance that can be achieved subject to requirements of the products.

In the proposed method, the first product platform forms the basis of the subsequent platforms. In the hypothetical case presented in Figure 3.1, the first stage of the design process returned a family of products with platform parameters x_1, x_2, x_3, x_4, x_5 *and* x_6. The only scale parameter is x_7.

Prod 1	***Prod 2***	***Prod 3***	***Prod 4***	***Prod 5***	***Prod 6***
x_1^*	x_1^*	x_1^*	x_1^*	x_1^*	x_1^*
x_2^*	x_2^*	x_2^*	x_2^*	x_2^*	x_2^*
x_3^*	x_3^*	x_3^*	x_3^*	x_3^*	x_3^*
x_4^*	x_4^*	x_4^*	x_4^*	x_4^*	x_4^*
x_5^*	x_5^*	x_5^*	x_5^*	x_5^*	x_5^*
x_6^*	x_6^*	x_6^*	x_6^*	x_6^*	x_6^*
x_7^*	x_7^*	x_7^*	x_7^*	x_7^*	x_7^*
Per_1^*	Per_1^*	Per_1^*	Per_1^*	Per_1^*	Per_1^*
Per_2^*	Per_2^*	Per_2^*	Per_2^*	Per_2^*	Per_2^*
Per_3^*	Per_3^*	Per_3^*	Per_3^*	Per_3^*	Per_3^*

Starting point for Single Platform Formulation

Benchmark Performance

Figure 3.2: Individual Optimum of Products

Stage 2: Evaluation Stage

In the platform evaluation stage, the performance of the product family members are evaluated against the benchmark products. The benchmarks can either be derived using the individual optimum performance as explained earlier or they can be specified by the designer. In this stage, the performance of each family member leveraged using the platform is compared to that of the performance of benchmark products.

$$\Delta_j = f_{family,j}(Per\,1, Per\,2, \ldots Per\,n) - f_{benchmark,j}(Per\,1^*, Per\,2^*, \ldots Per\,n^*) \qquad (3.4)$$

There may be more than one performance measure concerned with the products. Hence a representative function of the performances will be used while comparing the performance of the products. A threshold value will be identified by the designer as an acceptable loss of performance. The products whose deviations (Δ_j) fall within the acceptable limit will be retained in the current platform. Products whose deviations do not conform to acceptable limits are excluded from the platform.

With reference to Figure 3.2, there are three measures related to the product. The deviation function presented in Equation 3.4 can be extended to the hypothetical case as.

$$\Delta_j = f_{family,j}(Per\,1, Per\,2, Per\,3) - f_{benchmark,j}(Per\,1^*, Per\,2^*, Per\,3^*) \quad \text{... (3.5)}$$

The threshold value will influence the number of products that will be retained in the current platform. In this method, it is assumed that the designer specifies a reasonable threshold value for loss of performance. Developing a strategy to arrive at an efficient threshold value will be a subject of future research.

After evaluation of the products leveraged from the current platform, three possibilities exists: (1) All the resulting products have a performance loss within acceptable limits (2) Some products have a performance loss within acceptable limits while others do not and (3) None of the products have a performance loss within acceptable limits. Based on the evaluation of the products leveraged from the cascaded platforms, the designer may take his next course of action for each case as follows:

Case 1:

All of the products have been successfully leveraged and the designer may exit out of the loop

Case 2:

In Case 2, there are two possibilities: (1) the designer may leverage the products which have acceptable performance from the platform. The resulting platform is then cascaded again to leverage the non-conforming products (2) the current platform may be cascaded again to leverage all the products together.

Case 3:

The existing platform has to be cascaded until product family members with acceptable performance are derived.

The choice between options (1) and (2) in Case 2 is dependant on the additional cost of developing another platform and the manufacturing processes involved. A suitable index capable of capturing product family development costs in these cases may be developed to help the designer in making this decision. This is a subject of future research.

Figure 3.1 shows that, upon evaluation of the resulting family products, it was found that P_2 *and* P_3 had loss of performance within acceptable limits. Products P_2 *and* P_3 were retained to be leveraged from the first platform while P_1, P_4, P_5 *and* P_6 will be separated out.

Stage 3: Cascading Stage

In this stage, only the non-conforming products separated out after the evaluation is considered. Let p_{ck} be the set of products being considered for leveraging from the platform 'k'. Let x_{pk} denote the platform parameters for the current platform pp_k. The idea is to arrive at a new platform pp_{k+1}, which consists of platform parameters

x_{pk+1} formed by relaxing one of the platform parameters in x_{pk} to a scale parameter ($x_{pk+1} \subseteq x_{pk}$). Initially the value of platform parameters in x_{pk+1} is held to the same value as that of x_{pk}.

In the hypothetical example shown Figure 3.1, p_{c2} consists of the products $\{P_1, P_4, P_5 \textit{ and } P_6\}$. The platform parameters x_{p1} corresponding to pp_1 are $\{x_1, x_2, x_3, x_5, x_6, x_7\}$. A product platform pp_2 was cascaded from pp_1 by converting x_3 to a scale parameter. All the remaining products (P_1, P_4, P_5, P_6) except P_6 were found to have performance loss within acceptable limits.

At this point, the designer may wish to continue to cascade Platform No.1 until all the products have acceptable performance, or he may leverage the conforming products (P_1, P_4, P_5) using Platform No.2 and leverage P_6 using platform No.3. Here, a third platform was cascaded from Platform No.2 by transforming $x_6 \textit{ and } x_7$ to scale variables to leverage P_6. Platform pp_3 is a result of cascading pp_2 twice and consists of parameters $\{x_1, x_2 \textit{ and } x_5\}$. The proposed method is iterative and relies on the designer's judgment in comparing and evaluating different leveraging options.

3.2 Addressing Issues Related to the Product Family Optimization Problem

Optimization problems are normally classified as linear or non-linear problems according to the nature of objective functions and constraints. Linear problems have linear objective functions and constraints; hence, they easier to solve than non-linear problems due to their inherent properties (Winston, 1994). The relations between product parameters and performances (like mass, stress, etc.) in product family design problems are usually non-linear in nature. This makes the product family optimization a non-linear optimization problem. The general form of a platform specified product family optimization problem is shown in F1 (Table 3.1).

Here, $f(x_{ij})$ is the objective to be achieved, like maximizing the performance of the product etc. The product constraints $g(x_{ij})$ are related to individual products or the family as a whole. The commonality constraints ensure the sharing of platform parameters between different members in the family.

Table 3.1: The Platform Specified, Scale-Based Product Family Optimization Problem

Maximize:

$f(x_{ij})$ Objective to be achieved

Subject to:

$g(x_{ij})$ Product constraints

$x_{ij} = x_{ij+1} \vee j, j \neq m, and \vee i \in x_p$ Commonality constraints

$l \leq x_{ij} \leq u$ Bounds on the design variable (F1)

From Formulations F1, it can be seen that the size of the optimization problem increases as the number of design parameters and the number of products increases. Commercial optimization tools are currently available for solving large non-linear problems. A non-linear problem of form F1 may have many local optimum points. Commonly used optimization solution methods like gradient-based optimization solvers have a tendency to converge to a local optimum. While gradient-based methods guarantee optimality (local), global optimum points are hard to reach using these methods.

Heuristic methods like Tabu Search (Glover and Laguna, 1993), Simulated Annealing (Kirkpatrick *et al.*, 1983), and Genetic Algorithms (Goldberg, 1999) help to arrive at global solutions reasonably fast for small problems. The lack of available commercial tools, the necessity to adapt algorithms for specific problems, and the inability to solve problems of large size are limiting their application in practical design problems.

Table 3.2: The Non Platform Specified Scale Based Product Family Optimization Problem

<u>*Maximize:*</u>

$f(x_{ij}) + \sum y_i$ Objective to be achieved

<u>*Subject to:*</u>

$g(x_{ij})$ Product constrains

$x_{ij} = x_{ij+1} \vee j, j \neq m, and \vee i \in x_p$ Commonality constraints (Single platform assumption)

$l_{ij} \leq x_{ij} \leq u_{ij}$ Bounds on the design variable

$$y_i = \begin{cases} 1 & when\ X_i\ is\ a\ platform \\ 0 & when\ X_i\ is\ not\ a\ platform \end{cases} \qquad \text{(F2)}$$

In the case of non-platform specified problems, the formulation should explore different possible combinations of platform variables and select the best possible combination with maximum commonality. The objectives in the case of a non-platform specified product family optimization problem are (1) minimizing the performance loss and (2) increasing the commonality. This can be achieved by introducing a component in the objective function for maximizing commonality as shown in F2 (Table 3.2). The ~~terms~~ in the objective $\sum y_i$ is aimed at increasing the commonality and $f(x_{ij})$ is aimed at minimizing performance loss due to commonality. In Formulation F2, it is assumed that if a parameter is shared, it is shared across all of the products in the family.

Table 3.3: Basic Form of a MINLP Problem

Minimize:

$Z = f(x, y)$

Subject To:

$g_l(x, y) \leq 0 \; l \in L$

$$x \in X, \; y \in Y \qquad \text{(F3)}$$

The general formulation F2 falls under the category of a Mixed Integer Non-Linear Problem (MINLP). MINLP's have the form F3 (Table 3.3) when represented in algebraic form (Grossmann, 1990). Here, $f(x, y)$ *and* $g(x, y)$ are differentiable functions, L is the index set of inequalities, and x is continuous and y is discrete. In the case of F2, $y_i s$ are 0-1 variables. The use of MINLP is a natural approach to formulating problems where it is necessary to simultaneously optimize the system structure (discrete) and parameters (continuous) (Bussieck and Pruessner, 2003).

Due to combinatorial nature, Formulation F2 cannot be solved with commonly used gradient-based optimization algorithms. Discrete problems solved with gradient-based optimizers are solved as continuous problems that produce a discrete result at the end. Formulation F2 will not execute in gradient-based solvers because the y_i variables have to be either 0 or 1 and cannot be analyzed for values in between (for example, $y_i = 0.23$) which is a requirement for gradient-based optimization.

MINLPs require specialized algorithms and solution methods because they combine the difficulties of the sub-classes, the combinatorial nature of Mixed Integer

Programs, and the difficulty in solving non-linear programs (convex and non-convex). Outer Approximation (Duran and Grossmann, 1986), Branch and Bound (Quesada and Grossmann, 1992), Generalized Benders Decomposition (Sahinidis and Grossmann, 1991) and Extended Cutting Plane methods (Pettersson, F. and Westerlund, 1995) are some of the methods capable of solving MINLPs (Horst *et al*, 2001). Information on MINLPs, their solution methods, commercial packages, and recent advancements can be found in Bussieck and Pruessner, 2003.

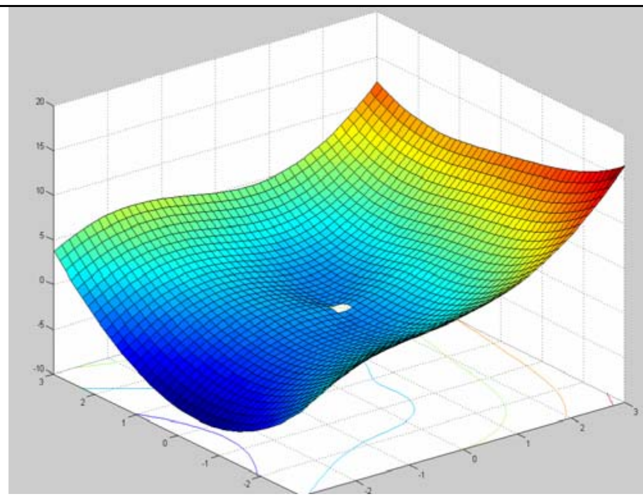


Figure 3.3: An Example of Non-Convex Space

The nature of the feasible region of F2 adds to the complexity of the problem. In a practical product family design setting, the feasible region may be non-convex. Figure 3.3 shows a sample non- convex region. There may be several local optimum points in the feasible region. Therefore, global optimization techniques need to be applied to Formulation F2. Not all MINLP codes available can solve non-convex problems. Commercial tools like BARON® are available for solution of non-convex MINLPs in polynomial form (Tawarmalani and Sahinidis, 2002). Global optimum solutions are obtained by using convex relaxations of the original problem. Choice of the solution method or algorithm is very critical in solving optimization problems. Solving the MINLP problems of this nature to global optimality can consume a lot of time and require a high level of expertise. Moreover, MINLP codes available today need fine tuning for particular problems. The difficulty in applying these methods to product family design problem is that they require a high level of expertise and time in solving the problem. This will transform to high product lead time.

The Platform Cascading Method presented in Section 3.4 uses several optimization formulations at different stages of the design process. The formulations are developed keeping in mind the different practical limitations of the design optimization discussed above. The formulations are aimed to arriving at optimum product families quickly and with relative ease of formulation and implementation. The different stages in the PCM are explained step by step along with the optimization formulations in Section 3.4. The method is illustrated using the axial displacement product family design case example introduced in Section 3.3

3.3 Illustrative Example - Axial Displacement Pump Product family (Adapted from Bhandare and Allada, 2006)

Pumps are devices that transfer mechanical energy into fluid power. They are classified primarily based on the type of motion that causes a transfer of energy. The axial piston pump uses reciprocating motion to transfer energy. It is a positive displacement pump with the designs available to obtain both fixed and variable displacements. A Fixed displacement type pump has been considered for the present case study.

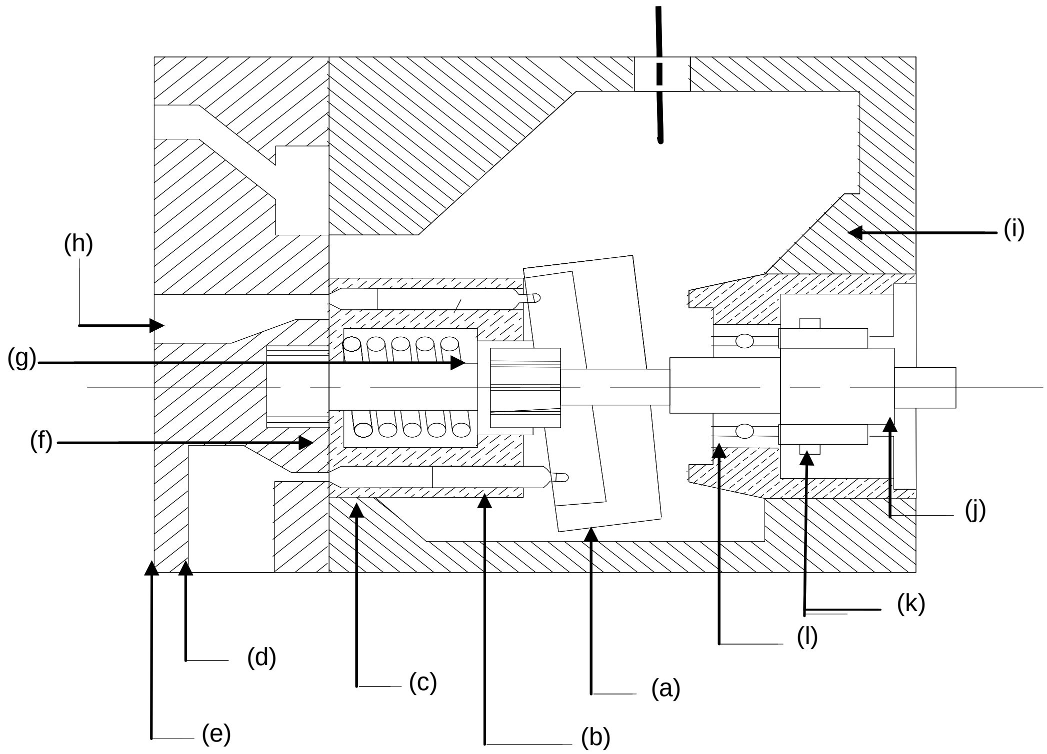
In the present example, various displacement requirements for the individual axial piston pumps have been considered. The problem considers the manufacturing cost of the axial piston pumps and aims to minimize cost by commonalizing the values of the design variables.

Table 3.4 lists the different technical parameters pertaining to the five variants of the axial piston pumps. The product variants have displacement requirements of 38, 51, 65, 75, and 90 cc. The acceptable loss in performance for each product introduced through platforming is also given in Table 3.4. The performance measure of each pump is assumed to be solely dependent on (1) Displacement of the pump and (2) Cost of the pump. Bhandare and Allada used the following design variables to link to the performance of the pumps:

Swash plate angle (9-21 degrees)

Diameter of the plunger (14-30 mm)

The number of plungers (5, 6, or 7)



(a)Swash plate assembly
(including shoe plate)
(a) Piston (Plungers) with shoes
(b)Cylinder block
(c)Inlet
(d)Valve plate

(a)Spring
(b)Splined washer
(c)Outlet
(d)Housing
(e)Drive shaft
(f)Shaft seal
(g)Shaft bearing

Figure 3.4: Schematic of the Axial Pumps (Adapted from Bhandare and Allada, 2006)

The major components of a typical fixed-displacement axial piston pump are shown in Figure 3.4. A valve plate contains an inlet and an outlet port and functions as the back cover. A rotating group consists of a cylindrical block splined to a drive shaft, splined spherical washer, springs, pistons with shoes, swash plate, and shoe plate. The spring forces the cylindrical block against the valve plate, while the spherical washer pushes against the shoe plate. This action holds the piston shoes against the swash plate, ensuring that the pistons reciprocate as the cylinder turns. The swash plate is stationary in a fixed-displacement design. For every rotation of the shaft, there is a change in the angle of the swash plate that leads to a fixed amount of suction and discharge of the fluid. This discharge is controlled by the design parameters affecting the displacement (swash plate angle, number of plungers, diameter of the plunger). The displacement of an axial piston pump is dependent on the following design parameters: the diameter of the plunger, the swash plate angle perpendicular to the axis of rotation, the number of plungers used, and the pitch circle diameter for the imaginary circle encompassing the plungers.

Axial displacement pumps mainly find application is open and closed center hydraulic systems. They are employed in systems like loading cranes, generator drives, compressor drives, drives for air conditioning systems, fan drives, etc.

The present example of an axial piston pump was provided by Bhandara and Allada through the product family test bed (Allada *et al.*, 2006). It provides ample scope for the researchers to extend it in various possible directions. Certain modifications were made to the original case study for completeness and also to enable it to fit the description of a scale-based product family. Following are the modifications made to the axial pump case study for implementation in this thesis:

1. The market demand for various pumps is not considered in this case study. A uniform demand is assumed for all of the product variants.
2. A new design variable, ‘inside diameter of the plunger’ is introduced. If the thickness of the plunger is held constant while the ‘outside diameter of the plunger’ is allowed to vary, any tangible cost saving could not be assumed as per the manufacturing

process. Hence the additional design variable (inside diameter) is introduced on the assumption that drilling the same size hole for all of the products in the family can introduce cost savings. The new parameters related to the plunger are (1) Outside diameter of the plunger (od_p) and (2) Inside diameter of the plunger (id_p). The changes do not change the overall structure of the problem, but will provide an opportunity for better cost savings.

3. The cost of the bearing is excluded in this case study to keep the problem size small. It is assumed that any bearing size can be manufactured and that they all have constant width.

4. In the original case study, the thickness of the plunger is computed by the following equations (Equation A.15 in Appendix A)

$$t_p = \frac{d_i - d_{i\_p}}{2} \; and \; t_p = \frac{P_i \times d_i \times fos_p}{20 \times \sigma_{p\_mat}} \qquad \text{... (3.6)}$$

5. The above equations are modified to accommodate the inside diameter into the equation.

$$\sigma_p = \frac{\left(\frac{od_p - id_p}{2}\right)}{P_i \times d_i} \; and \; t_p = \frac{od_p - id_p}{2} \qquad \text{... (3.7)}$$

6. Additional constraints were introduced to:

   (1) Reflect the changes in the above equation and then changing the stress from a hard constraint to a soft constraint assuming a safety factor of 2.0 as specified.

$$\sigma_p \leq 175 Mpa \qquad \text{... (3.8)}$$

(2) To introduce the manufacturing constraint of inability to manufacture very thin-walled plungers and also to generate only the feasible geometry such that the Outside Diameter is always greater than Inside Diameter by at least 4mm.

$$od_p - id_p \geq 4 \qquad \text{... (3.9)}$$

7. The modeling of the stress in the piston:

Bhandare and Allada (2006) modeled the plunger as a thin-walled pressure vessel to compute the stress (axial). The equation used (A.15) in Appendix A is derived from a more general equation for stress in a cylinder subjected to inside pressure for the case of thin-walled vessels (Norton, 1996). Since the optimization algorithm searches for design space that might not fall under the classification of thin walls (ID/OD < 0.1), the generic version of the equation which is applicable to thin- and thick-walled vessels as shown below is used in the model

$$\frac{\sigma_{p\_mat}}{fos_p} \geq \frac{P_i \times (\frac{id_p}{2})^2}{(\frac{od_p}{2})^2 - (\frac{id_p}{2})^2} \qquad \text{... (3.10)}$$

With the above modifications, the design requirements for the axial pump product family can be summarized as shown in Table 3.4.

Table 3.4: Design Requirement for the Axial Pump Family

Product Variant	Pressure	Driver Speed	Displacement	Acceptable Loss in Displacement	Stress in Plunger	Geometric Feasibility
	(bar)	(rpm)	(cc/rev)	(+/-) %	(Mpa)	(mm)
P_1	350	2650	38	10	175	4
P_2	400	2700	51	10	175	4
P_3	350	2500	65	10	175	4
P_4	350	2400	75	10	175	4
P_5	350	2200	90	10	175	4

The following are the design parameters and their bounds

1) Swash plate angle (9-21 degrees)
2) Outside diameter of the plunger (14-30 mm)
3) The number of plungers (5, 6, or 7)
4) Inside Diameter of the plunger (2-26 mm)

The objective for the product family design problem is to find suitable platform(s) that can be used to generate the family of products with minimal loss of performance. The cost of the pumps is derived from the cost of the material and from the manufacturing operations as explained in Appendix A. The design variables swash plate angle, outside diameter of the plunger and inside diameter of the plunger are continuous while the number of plungers is discrete in nature.

3.4 General Formulation

The PCM is a multi-stage optimization method for the design of scalable product families. The inputs to the formulation are: (1) the specification of the product family members (2) the underlying mathematical model that relates the product parameters to performances and (3) the identification of platform parameters (optional). PCM does not require the identification of platform parameters by the designer; however, it allows the designer the flexibility of being able to specify the platform. The outputs from the method are (1) the different product platforms and the products that are leveraged from it

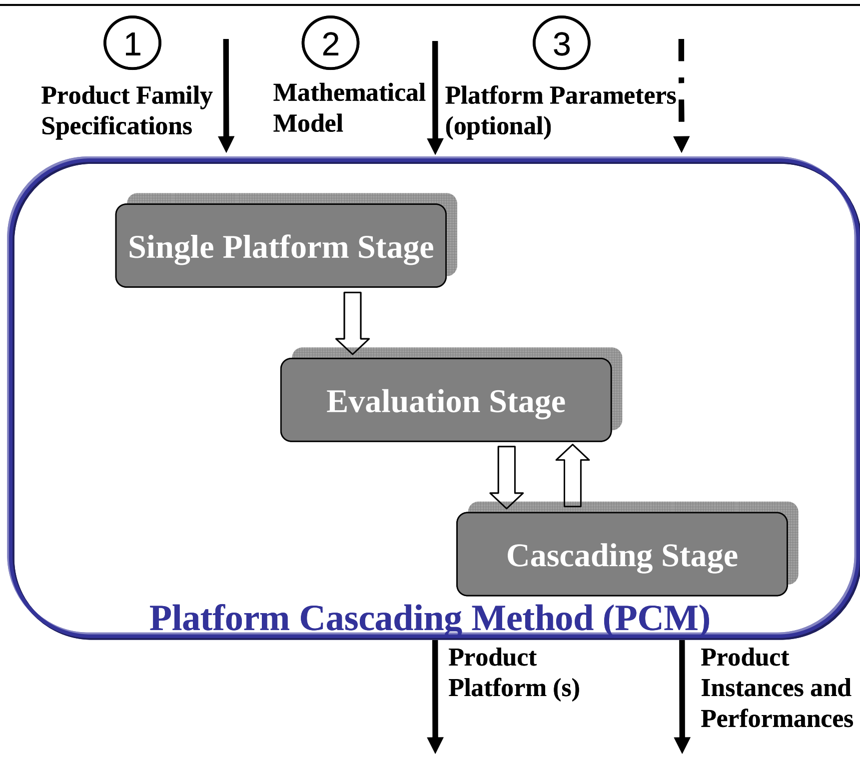


Figure 3.5: Platform Cascading Method Inputs and Outputs

and (2) the product family instances and their performances. Other secondary information like the loss of performance due to commonality in comparison to benchmarks and the best possible performance of the products can be obtained from the method. As evident from earlier discussions, the method is comprised of different stages. The method is only applicable towards scalable product families, wherein each product instance in the family can be completely described by the same set of product parameters. Hence the method will fall under the category of a multi-stage, non-platform specified, scale-based product family design method.

Table 3.5: General Steps in PCM

$PF = \{p_1, p_2, ..., p_m\}$

$Y = (y_1, y_2, ..., y_n)$

$PP = \{pp_k / pp_k$ is the set of product platforms from which the family is derived$\}$

$PP = \{pp_1, pp_2, ...pp_f\}$

$P_{ck} = \{p_{ck} / p_{ck}$ is the set of products considered for leveraging from platfrom 'k'$\}$

$X_{pk} = \{x_{pk} / x_{pk}$ is the set of platform parameters for platform 'k'$\}$

C_{ik} is the set of platform parameter values in platform 'k' $(if\ i \in x_{pk})$

N_k = Cardinality of x_{pk}

<u>1. Single Platform Stage</u>

1: Execute "Individual Optimization formulation"

2: $k = 1$

3: $P_{c1} = PF, \quad P_{c1}' = \phi$

4: Execute "Platform Specified/Non Platform Specified Formulation"

5: $k = k + 1$

<u>2. Evaluation stage</u>

6: $\Delta_j = f_{benchmark,j}(w_1 z_1 + w_1 z_1 + ... w_n z_n) - f_{family,j}(w_1 z_1 + w_1 z_1 + ... w_n z_n) \quad \forall j \in P_{ck-1}$

7: All $\Delta_j values \leq \eta ; P_{ck} = \{\}$; Goto 10:

<u>Case 2</u>: Some $\Delta_j values \leq \eta$ & other $\Delta_j values > \eta$ then (a) / (b)

(a) Include products with $\Delta_j values > \eta\ in\ P_{ck}$; $k = k + 1$; Goto 8:

(b) Include all products in P_{ck} ; $k = k - 1$; Goto 8:

<u>Case 3:</u> No $\Delta_j values \leq \eta ; k = k - 1$; Goto 8:

<u>3. Cascading Stage</u>

8: Execute "Platform Cascading Formulation"

9: Go to 6:

10: End

In the general formulation shown in Table 3.5, PF is the set of product family members consisting of the product instances $p_1, p_2, ..., p_m$. Y is the set of platform commonality variables corresponding to each of the product parameters $x_1, x_2, ..., x_n$. PP is the set of product platforms used to leverage the products. Initially, the number of platforms required is unknown. X_{pk} is the set of platform variables for each platform.

In the axial displacement example, PF consists of the products p_1, p_2, p_3, p_4 and p_5. These correspond to pumps with displacements of 38, 51, 65, 75, and 90 cc. The product parameters x_1, x_2, x_3 and x_4 are the (1) swash plate angle, (2) outside diameter of the plunger, (3) number of plungers, and (4) inside diameter of the plunger, respectively. Therefore, $Y = (y_1, y_2, y_3, y_4)$ are the commonality decision variables for the product family. $\mathrm{N_k}$ represents the number of platform parameters in each platform pp_k. The three stages of the PCM, general steps, and the optimization formulations used are shown in the following sub-sections along with their application to the case study.

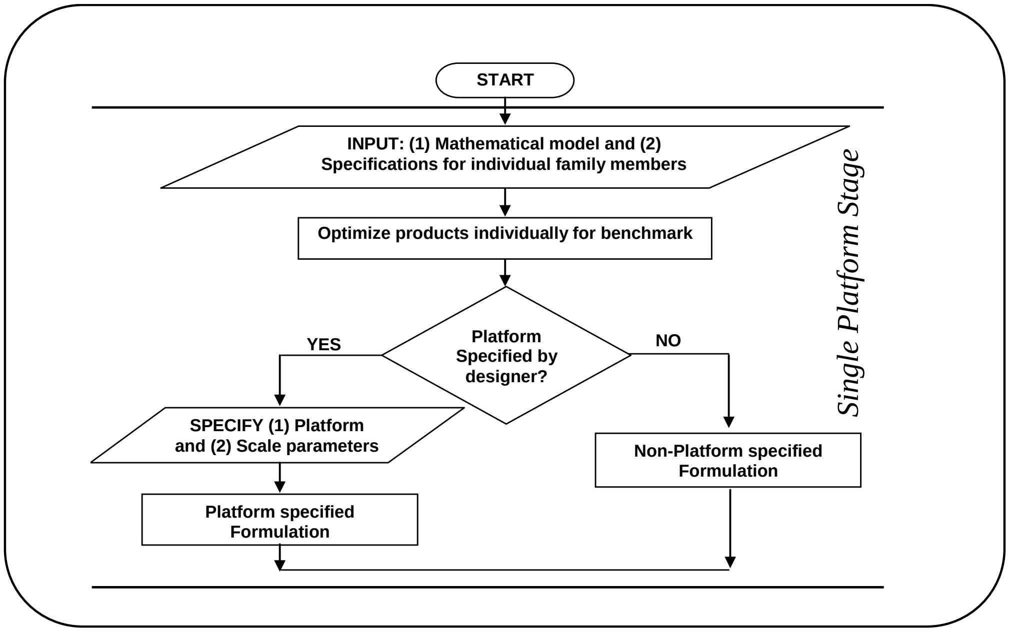


Figure 3.6: Single Platform Stage

3.4.1 Stage 1: Single Platform Stage

The first step in the single platform stage is the determination of individual optimum for the product instances. Each of the product instances are optimized individually using the individual optimization formulation shown in Table 3.6. The objective of the formulation is to find the best performances that can be obtained for each product instance. In most cases, there are more than one performance measures that need to be maximized/minimized while designing products; hence, an objective function consisting of weighted performances is used. The different weights in the objective function help to prioritize the different performances according to their relative importance.

Table 3.6: Individual Optimization of Product Instances

Given:
Mathematical model
Product constraints

Minimize
$f(w_1 z_1, w_2 z_2, ... w_n z_n)$

Subject to:
$g_t(X) \leq 0 \quad \forall\, r \in R$
$l_i \leq x_i \leq u_i \quad \forall\, i \in I$
(Formulation repeated 'm' times)

$g(X)$ are the constraints that need to be satisfied by each product. l_i *and* u_i represent the lower bound and upper bound, respectively, of the product parameters x_i. The formulation is repeated 'm' times for each of the product instances. At the end of each run the optimum value of each performance measure $(z_1, z_2, ..z_n)$ and the optimum value of the product parameters x_i are noted. This information is used in the subsequent steps.

In the case study, the objective to be minimized is the positive deviation of the cost of the pumps from the target cost. The cost is a non-linear function of the product parameters given by Equation A7 in Appendix A. Since the case study is single objective in nature, the objective function is straightforward and need not be weighted as in case of multi-objective problems. The products are required to have displacements of 38, 51, 65, 75 and 91 cc respectively. The displacement requirements are modeled as a constraint in the formulation. The other constraints related to the products are (1) stress in the plunger and (2) geometric and manufacturing feasibility.

The product parameters are bounded as shown in Table 3.7 (Constraint 4). The parameter x_3 represents the number of plungers in the pump and hence can only take integer values of 5, 6 and 7.

Table 3.7: Single Platform Optimization Formulation Applied to Axial Pump Product Family

Indices

i = Design parameters, $i \in I$, $I \in \{1, 2, .3, 4\}$. These indices correspond to the swash plate angle, outside diameter of the plunger, number of plungers and inside diameter of the plunger respectively.

j = Product family members, $j \in J$, $J \in \{1, 2, .. 5\}$

Parameters

P_j is the pressure requirement for product 'j', $P_j \in P$, $P = \{350, 400, 350, 350, 350\}$

Variables

C_j is the cost of pump 'j'. $C_j = f(x_1, x_2, x_3, x_4)$ given by Equation (A.7)

Given

Mathematical model. See Appendix A

Minimize

Total Cost (C_j)

Subject to

(1) Displacement requirement for the product family

$$\frac{1}{4000 \tan(x_1)(2x_2 - x_4)x_3^2 x_2^2} = \{38, 51, 76, 75, 91\}$$

(2) Stress in the plunger ≤ 175 Mpa

$$\frac{P_j \times (\frac{x_4}{2})^2}{(\frac{x_2}{2})^2 - (\frac{x_4}{2})^2} \le 175 Mpa$$

(3) Plunger feasible geometry and manufacturability constraint

$x_2 - x_4 \ge 4$

(4) Limits on the design variables

$9.0 \le x_1 \le 21.0$

$14.0 \le x_2 \le 31.0$

$5.0 \le x_3 \le 7.0, x_3 = I\text{nteger}$

$2.0 \le x_4 \le 30.0$

Formulation repeated for j=1, 2 … 5

The formulation was implemented using VRAND® Visual DOC®, a commercially available non-linear optimization tool. The formulation was solved using its gradient-based solver. Modified Method of Feasible Directions (MMFD) was selected as the solution algorithm. Visual DOC®, gradient-based solver has the ability to handle discrete sets of data points for design variables as long as the problem is not combinatorial in nature.

Table 3.8 shows the results obtained after individual optimization. Different starting points were used for each case so that the global optimum can be reached. The total cost resulting from the optimization run is used to serve as the benchmark for the product family. The values of product parameters resulting at the individual optimum are used as starting point for the single platform stage to enable speedy convergence. The stress values on the plungers are well below their maximum allowed value.

After the benchmarks and starting points are established using the results from the individual optimization formulation, the number of platform counter 'k' is initiated ($k = 1$). In the single platform stage, the design intent is to design the entire family using a single platform. Therefore P_{c1} includes all the products in the family ($P_{c1} = PF, \quad P_{c1'} = \phi$). In this example P_{c1} comprises of the family members $\{p_1, p_2, p_3, p_4 \text{ and } p_5\}$. The two formulations developed for the single platform design are (1) the platform specified formulation, where the designer identifies the platform parameters and (2) the non-platform specified formulation, where the designer does not identify the platform parameters. In the axial pump case study, the product platform is not provided. Therefore, non-platform specified design formulation is used to derive the platform from which the products can be leveraged. The general

Table 3.8: Results of Individual Optimization of Axial Pumps

	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
X1 (PA)	9.00	11.29	10.02	11.52	10.16
X2 (OD)	28.73	30.00	30.00	30.00	30.00
X3 (Np)	5.00	5.00	6.00	6.00	7.00
X4 (ID)	24.73	26.00	26.00	26.00	26.00
Displ	37.74	51.00	65.00	75.00	89.74
Cost	36.08	48.55	58.08	65.85	76.10
Stress	135.12	160.71	140.63	140.63	140.63
Geom	4.00	4.00	4.00	4.00	4.00

form of the non-platform specified formulation is shown in Table 3.9. The platform specified formulation differs from the non-platform specified formulation in the objective function and the setting of y_i parameter values. These are explained in the following section.

In the general formulation, $x_1, x_2, ..., x_n$ are the parameters that define the products. If there are j products in the family then x_{ij} represents the i^{th} parameter for product family member *'j'*. In the general formulation presented below, platform commonality parameters y_is are used to force the commonality of the platform variables. Platform parameters have the same value throughout the product family. If x_1 is a product parameter and x_{11}, x_{12} *and* x_{13} are the values of parameter x_1 for the product instances 1, 2 and 3 in a three member product family, then $x_{11} = x_{12} = x_{13}$, if x_1 is a platform parameter. The commonality condition can be achieved in the formulation using the binary variable y_1 and the following constraints.

$$(x_{11} - x_{12})y_1 = 0 \text{ and } (x_{12} - x_{13})y_1 = 0 \qquad \text{... (3.10)}$$

$$\text{Where } y_1 = \begin{cases} 1 & \text{when } \mathrm{x}_1 \text{ is a platform} \\ 0 & \text{when } \mathrm{x}_1 \text{ is not a platform} \end{cases}$$

Parameter y_1 takes the value of 1 when x_1 is a platform and it takes the value of 0 when it is not. This constraint imposes the following restriction on the values of x_{11}, x_{12} *and* x_{13}. When x_1 is a platform, parameter $y_1 = 1$. Equation 3.10 becomes

$$(x_{11} - x_{12})1 = 0 \text{ and } (x_{12} - x_{13})1 = 0 \qquad \text{... (3.11)}$$

$$\Rightarrow x_{11} = x_{12} = x_{13}$$

When x_1 is not a platform $y_1 = 0$, Equation 3.10 becomes

$$(x_{11} - x_{12})1 = 0 \text{ and } (x_{12} - x_{13})1 = 0$$

The constraint becomes invalid and x_{11}, x_{12} *and* x_{13} can take any value. If there are '*j*' products in the family there will be '*j-1*' platform commonality constraints.

There may be several objectives to be considered while designing products, such as minimizing mass, increasing efficiency, and reducing stress in components. Hence the formulation is designed to be capable of handling multi objective decision making. A Goal Programming Model (Winston, 1994) is adopted to address the multiple objectives of the product family design model. In Goal Programming, the target values are identified for each objective. The deviation of the actual objective value from its targets is captured using deviation variables. Deviation variables d_{lj}^{+} and d_{lj}^{-} are the positive and negative deviation of actual attainment $A_{lj}(x)$ from the target G_{lj} respectively. Both d_{lj}^{-} and $+d_{lj}^{+}$ are constrained to have only non negative values.

If $A_{lj}(x) \leq G_{lj}$ (underachievement) then $d_{lj}^{-} > 0$ *and* $d_{lj}^{+} = 0$

If $A_{lj}(x) \geq G_{lj}$ (overachievement) then $d_{lj}^{+} > 0$ *and* $d_{lj}^{-} = 0$

and If $A_{lj}(x) = G_{lj}$ (exactly satisfied) then $d_{lj}^{+} = 0$ *and* $d_{lj}^{-} = 0$

When values larger than the target are undesirable, the positive deviations are minimized in the objective function and vice versa. To keep the actual values close to the target both negative and positive deviations are minimized.

Table 3.9: General Formulation for Non-Platform Specified Optimization

Indices

$j =$ Product family members, $j \in J, J = \{1, 2, 3...m\}$

$t =$ Product Constraints, $t \in T, T = \{1, 2, 3...s\}$

$l =$ System goals, $l \in L, L = \{1, 2, 3...p\}$

Variables

x_{ij} is the parameter 'i' in product 'j'

$y_1, y_2, y_3, ...y_n$ are the commonality parameters corresponding to each parameter in I

G_{lg} is the target goal of objective l for product j

d_{lj}^{+} is the positive deviation of $\mathrm{l^{th}}$ goal for $\mathrm{j^{th}}$product

d_{lj}^{-} is the negative deviation of $\mathrm{l^{th}}$ goal for $\mathrm{j^{th}}$product

w_{lj} is the weights for the deviation variables $d_{lj}^{+/-}$in the objective function

w_i is the weights for the commonality parameters in the objevtive function

Objective

$$\sum_{l=1}^{p} \sum_{j=1}^{m} f(w_{lj}, d_{lj}^{-}, d_{lj}^{+}) - \sum_{i=1}^{n} w_i y_i$$

Subject to

(1) $x_{ij}^{lower} \leq x_{ij} \leq x_{ij}^{upper}, \quad \forall i \in I$ and $\forall j \in J$ Bounds on the design variable

(2) $0 \leq y_i \leq 1$

$$y_i = \begin{cases} 1 & \text{when } \mathrm{x_i} \text{ is a platform} \\ 0 & \text{when } \mathrm{x_i} \text{ is not a platform} \end{cases}$$

(3) $g_t(x) = 0, \qquad t = 1,...s$ Constraints relating to individual products

(4) $(x_{ij} - x_{ij+1})y_i = 0 \quad \forall i \in I$ and $\forall j \in J, j \neq m$ Commonality constraints

(5) $y_i^2 - y_i = 0 \quad \forall \quad i \in I$ Constraints for converting y_i *to* continuous variables

(6) $A_{lj}(x) + d_{lj}^{-} + d_{lj}^{+} = G_{lj} \quad \forall l \in L \quad and \quad \forall j \in J,$ Objectives transformed to system goals

(7) $d_{lj}^{-}, d_{lj}^{+} \geq 0,\ d_{lj}^{-}.d_{lj}^{+} = 0 \quad \forall l \in L$ and $\forall j \in J$ Non negativity of deviation variables

The term $\sum_{i=1}^{n} w_i y_i$ maximizes the number of platform parameters. Different terms in the objective function are weighted so that all of them are given equal priority while optimization is performed.

In the case of the axial pump family, there are four product parameters and five products in the family. When applied to the case study, the single platform formulation will return the optimum value of x_{ij} parameters considering maximum commonality and minimum performance deviation. Positive deviation of cost, positive and negative deviation of displacement and the sum of commonality parameters are the components of the objective function. The benchmark cost obtained from the individual optimization (Table 3.8) is used as targets in the formulation. The deviation from targets for each of the pumps is captured using the deviation variables $d_{11}^{+/-}, d_{12}^{+/-}, d_{13}^{+/-}, d_{14}^{+/-} \, and \, d_{15}^{+/-}$ while $d_{21}^{+/-}, d_{22}^{+/-}, d_{23}^{+/-}, d_{24}^{+/-} \, and \, d_{25}^{+/-}$ are the deviations of displacement. Since a cost that is higher than the target is undesirable, the positive deviation variables $d_{11}^{+}, d_{12}^{+}, d_{13}^{+}, d_{14}^{+} \, and \, d_{15}^{+}$ are minimized in the objective function.

Table 3.10: Single Platform Formulation Applied to Axial Displacement Pump Family

Indices

i = Design parameters, $i \in I, I \in \{1,2,3,4\}$.

These indices correspond to the swash plate angle, outside diameter of the plunger, number of plungers and inside diameter of the plunger respectively.

j = Product family members, $j \in J, J \in \{1, 2, .. 5\}$.

These products represent the product family with displacements of 38, 51, 65, 75, 91cc.

Parameters

P_j is the pressure requirement for product 'j'

D_j is the actual displacement of family member 'j'

D_j^* is the target displacement for pump 'j'

C_j is the actual cost of product 'j'

C_j^* is the cost of product 'j' from established from benchmark

Variables

x_{ij} is the parameter 'i' in product 'j', $1 = 1,2,3,4$ and $j = 1,2,...5$

C_j is the cost of product j. given by Equation (7) in Appendix A

G_{1j} is the target cost of product 'j'= C_j^*

G_{2j} is the target displacement of product $j = \{38, 51, 65, 75, 91\text{cc}\}$

d_{1j}^+ is positive deviation of cost goal for jth product

d_{2j}^+, d_{2j}^- is the positive, negative deviation of displacement goal for jth product

w_{lj} weights for the deviation variables $d_{lj}^{+/-}$ in the objetive function

w_i weights for the *c*ommonality paramters in the objective function

Given

Mathematical model. See Appendix A

Minimize

$$Z = \sum_{j=1}^{5} w_{1j} d_{1j}^+ + \sum_{j=1}^{5} w_{2j} d_{2j}^+ + \sum_{j=1}^{5} w_{2j} d_{2j}^+ + \sum_{i=1}^{4} w_1 y_i$$

Subject to

(1) Bounds on the design variables

$9.0 \le x_{1j} \le 21.0$

$14.0 \le x_{2j} \le 31.0$

Table 3.10 continued…

$5.0 \le x_{3j} \le 7.0, x_3 = Integer$

$2.0 \le x_{4j} \le 30.0 \quad \forall j \in J$

(2) Platform commonality decision variables

$0 \le y_i \le 1$

$$y_i = \begin{cases} 1 & \text{when } x_i \text{ is a platform} \\ 0 & \text{when } x_i \text{ is not a platform} \end{cases}$$

(3) 'Integerizing' constraints

$y_i^2 - y_i = 0 \quad \forall i \in I$

(4) Platform commonality constraint

$(x_{ij} - x_{ij+1})y_i = 0 \quad \forall i \in I$ and $j \in J, j \ne 5$

(5) Displacement requirement for the product family

$$\frac{1}{4000 \tan(x_{1j})(2x_{2j} - x_{4j})x_{3j}^2 x_{2j}^2} = \{38, 51, 76, 75, 91\}$$

(6)Stress in the plunger $\le$ 175 Mpa

$$\frac{P_j \times (\frac{x_{4j}}{2})^2}{(\frac{x_{2j}}{2})^2 - (\frac{x_{4j}}{2})^2} \le 175 Mpa \quad \forall j \in J$$

(7) Plunger feasible geometry and manufacturability constraint

$x_{4j} - x_{2j} \ge 4 \quad \forall j \in J$

(8) Target cost of the pumps

$C_j / C_j^* + d_{1j}^- - d_{1j}^+ = 1.0 \quad \forall j \in J$

(9) Target displacement of the pumps

$D_j / D_j^* + d_{2j}^- - d_{2j}^+ = 1.0 \quad \forall j \in J$

(10) Non negativity of deviation variables

$d_{1j}^-, d_{1j}^+, d_{2j}^-, d_{2j}^+ \ge 0 \quad \forall j \in J$

In the case of displacement, both negative and positive deviation is undesirable;

Table 3:11: Results of Single Platform Optimization

	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
X_1	9.01	9.01	11.42	13.04	13.90
X_2	28.52	28.52	28.52	28.52	28.52
X_3	6.00	6.00	6.00	6.00	6.00
X_4	23.79	23.79	23.79	23.79	23.79
Displ	51.00	51.00	65.00	74.49	77.43
Cost	49.83	49.95	61.42	69.34	71.80
Stress	115.04	131.48	115.04	115.04	93.32
Geom	4.73	4.73	4.73	4.73	4.73

hence both are minimized in the objective function. The sum of commonality parameters y_1, y_2, y_3 *and* y_4 are maximized in the objective. To model commonality, four constraints are required for each of the four product parameters. The initial value of the x_{ij} parameters is the individual optimum of the products. The initial values of the y_i parameters are held at 0.5 so that the platform configuration is unbiased. The formulation selects the parameters that least influence the performance loss of the products and drives their corresponding y_i to 1. Other y_i parameters are driven to 0 thus making them scale parameters.

Figure 3.7 (a), (b), (c) and (d) show the variation of $x_{1j}, x_{2j}, x_{3j},$ *and* x_{4j} at different design iterations. It can be seen that all the platform parameters are held to the same value for different products in the family, whereas the scale parameters have different values for different product family members. Table 3.11 shows the value of different parameters resulting from the single platform optimization. Platform 1 consists of parameters $x_2, x_3, and\ x_4$ whose values are equal to 28.52, 6.0 and 23.79 respectively. The product variants also satisfy the condition for stress in the plunger and the geometric and manufacturing feasibility.

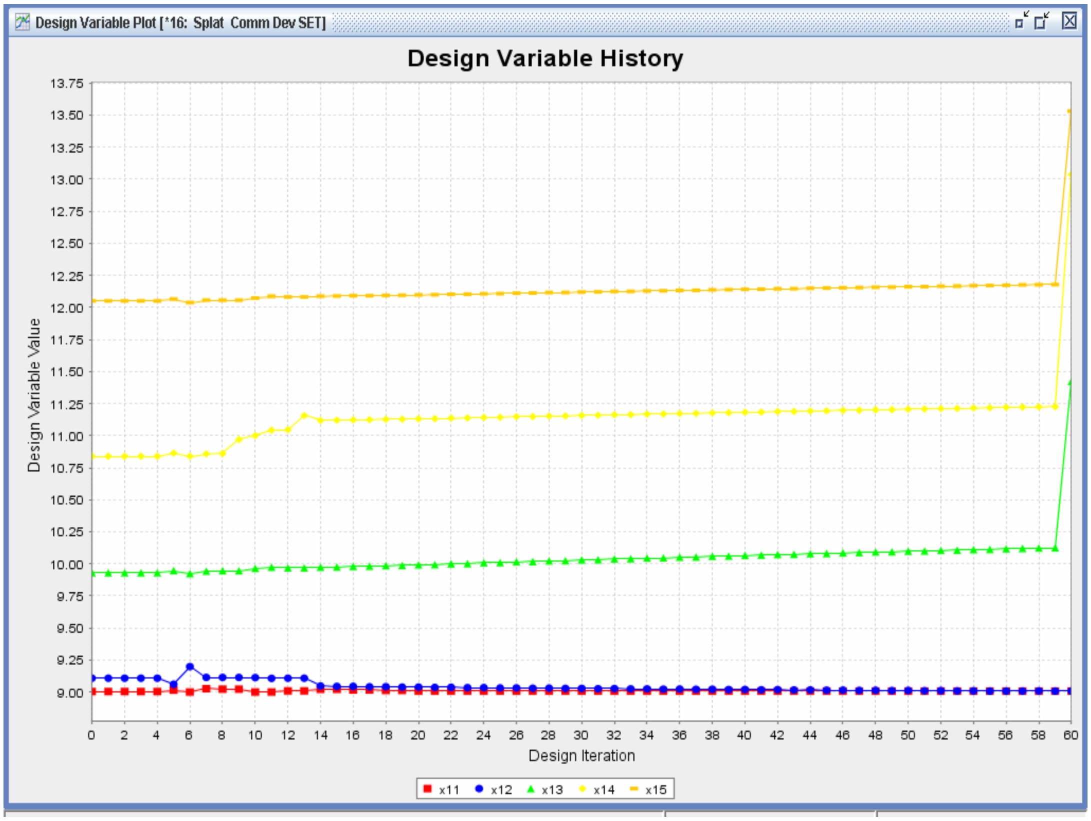


Figure 3.7 (a): Variation of x_{1j} Parameters for Different Design Iterations

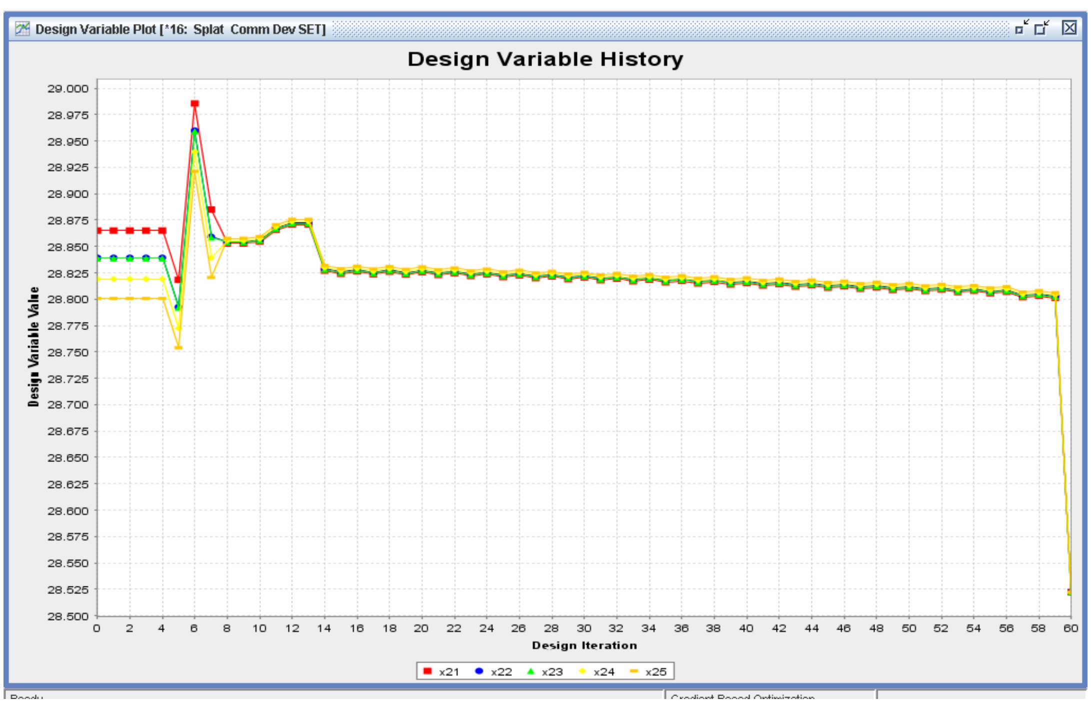


Figure 3.7(b): Variation of x_{2j} Parameters for Different Design Iterations

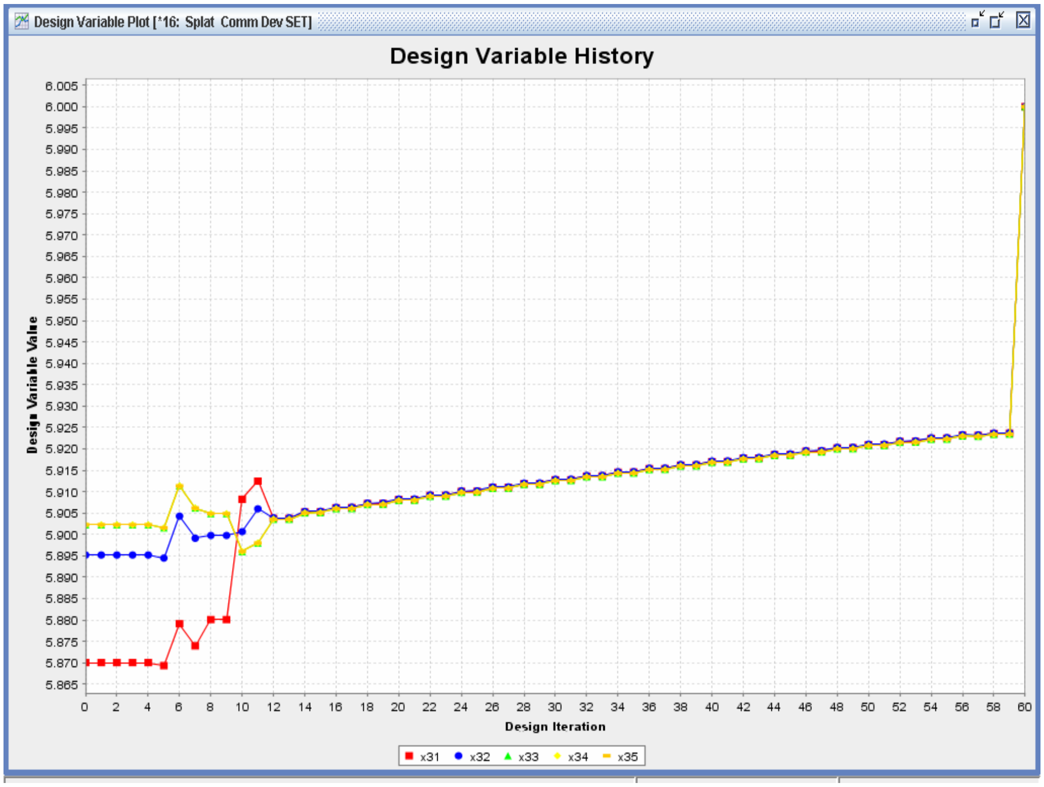


Figure 3.7 (c): Variation of x_{3j} Parameters for Different Design Iterations

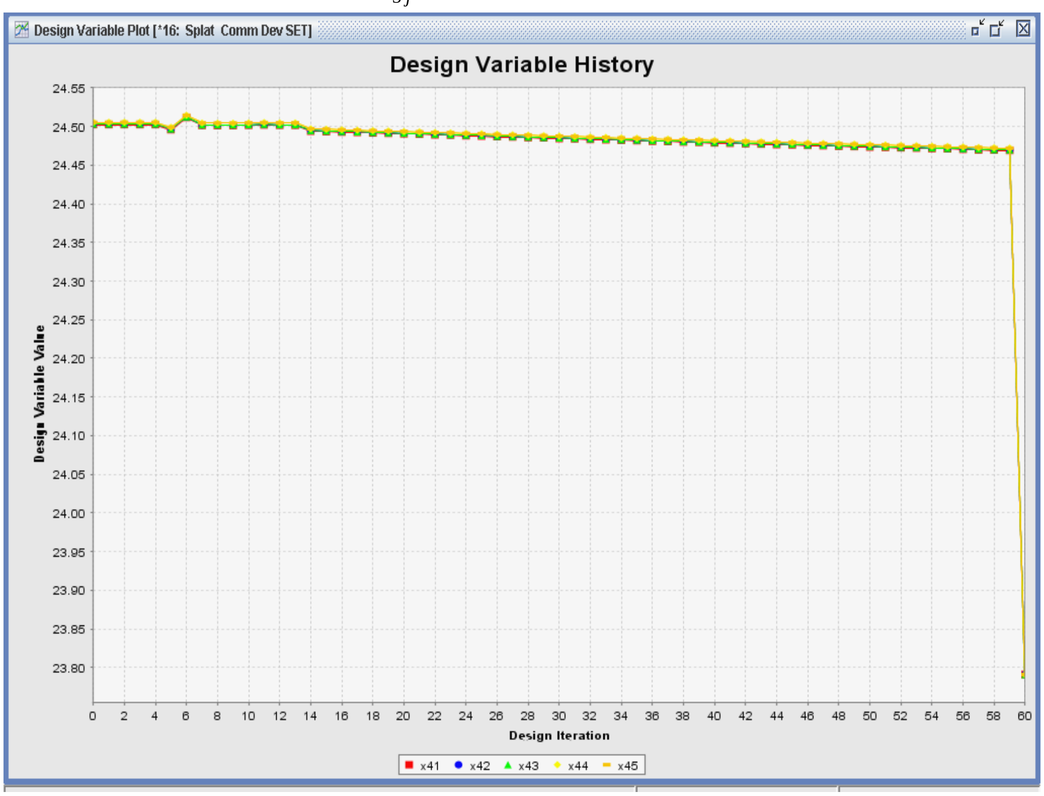


Figure 3.7(d): Variation of x_{3j} Parameters for Different Design Iterations

The single stage step of the PCM helps to answer Research question 1. The developed single platform optimization formulation performs trade-offs between

> ***RQ1)*** *How do we represent a family of products supported by a single platform using a mathematical programming model and identify a solution technique, so that tradeoffs between commonality and performance can be performed to support product family design?*

commonality and performance loss due to commonality, and arrives at an optimum platform configuration. The product family members are leveraged from the platform in a single stage. The formulation developed is easy to implement in gradient-based optimization methods.

Objectives O1, O2, O3 and O4 were achieved through the single platform stage of the design process. A mathematical programming model capable of representing a scale based product family in terms of decision variables, constraints and objectives (O1) was introduced. The commonalities of platform parameters were modeled by introducing binary platform commonality decision parameters and forcing commonality of corresponding family members through equality constraints (O2). To perform trade-offs between platform commonality and loss of performance, binary platform commonality parameters were treated initially as continuous and then constrained to only accept values of 0 or 1 values in the end. This approach enables the formulation to be implemented in gradient-based optimization methods (O3 and O4). The second stage of the PCM, evaluation stage is explained in next subsection.

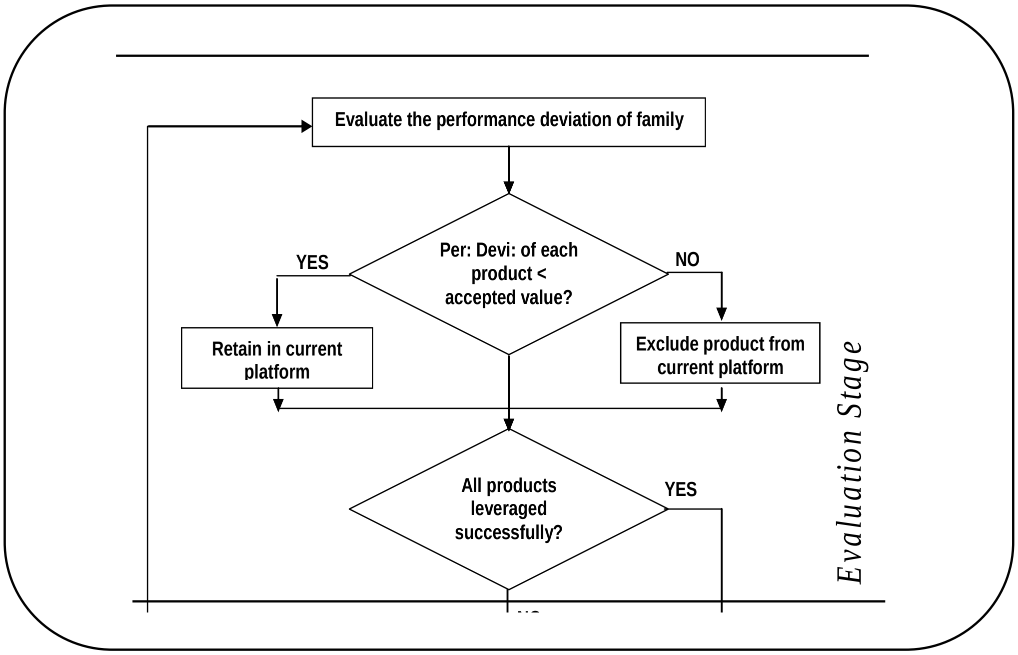


Figure 3.8: Evaluation Stage

3.4.2 Stage 2: Evaluation Stage

In the evaluation stage products leveraged from the platform are compared against the benchmark products. Let $z_1, z_2, ...z_p$ be the performance measures considered in the objective function, $z_{1j}, z_{2j} .... z_{pj}$ be their value for product 'j' and $z^*_{1j}, z^*_{2j} .... z^*_{pj}$ be the value of their corresponding benchmark. The performance of the products is evaluated using the function

$$\Delta_j = \pm(N_1 z^*_{1j} - N_1 z_{1j}) \pm (N_2 z^*_{2j} - N_2 z_{2j}) .... \pm (N_p z^*_{pj} - N_p z_p) \qquad ... (3.12)$$

Here $N_1, N_2, ... N_p$ are the factors used to normalize the performances for comparison. The following sign manipulations are performed to each of the components in the function depending on the nature of each desired performance measure.

For positive valued targets:

When a performance higher than target is desired and the performance measure obtained for product ‘j’ is higher than target a negative sign is assigned, and when performance obtained is lower than target, a positive sign is assigned.

When the performance measure is desired to be exactly equal to the target, a positive sign is assigned.

For negative valued targets:

When performance higher than target is desired and the performance measure obtained for product ‘j’ is higher than target, a negative sign is assigned the when performance obtained for product ‘j’ is lower than target a negative sign is assigned.

When a performance measure is desired to be exactly equal to the target, a positive sign is assigned. Δ_j values are calculated for each product leveraged from and current platform. Following are the cases that represent the possible scenarios that result after Δ_j.

Case 1: All $\Delta_j values \leq \eta$

In this case all the products have performance within the acceptable limits,, hence further iterations or platforms are not required. The design iterations may be considered complete.

Case 2: Some $\Delta_j \ values \leq \eta$

In case 2 some of the products satisfy the set case for product performance while others do not. The designer has two possible options:

(a) Include products with acceptable performance ($\Delta_j values \leq \eta$) to be leveraged from the current platform, separate the non-conforming ($\Delta_j values > \eta$) to be leveraged from the consecutive platform. The platform count is now incremented by 1 $k = k+1$ and then the platform cascading formulation is repeated with the nonconforming products.

(b) Include both conforming and nonconforming products and cascade platform P_{ck-1} further. The advantage is that this keeps the number of platforms lesser than case (b). ; Goto 10:

<u>Case 3:</u> No $\Delta_j values \leq \eta$; $P_{ck} = P_{ck-1}$; Execute cascading formulation.

In this step none of the products are conforming. The only option is to cascade the platform further until conforming products are attained.

For the axial pump case study the performances considered in the objective function are cost and displacement. The benchmark displacement and cost are normalized and the same normalization factors are used for corresponding family members. The limiting value of Δ_j^{Total} = 0.1 was set as the acceptable loss in performance due to commonality. The evaluation function shown in Equation 3.12 is extended to the present case study as

$$\Delta_j = \pm(N_j^1 \times \mathrm{Cost}_\mathrm{j}^* - N_j^1 \times \mathrm{Cost}_\mathrm{j}) \pm (N_j^2 \times \mathrm{Displacement}_\mathrm{j}^* - N_j^2 \times \mathrm{Displacement}_\mathrm{j})$$

... (3.13)

Table 3.12 shows the computation of Δ_j values for product family members leveraged from platform 1. Pumps 4 and 5 are found to have higher values for Δ_j values, hence

Table 3.12: Evaluation of Products Leveraged from Platform 1

	Displacement (Normalized)				Cost (Normalized)					
	Weights	Bench:	Family	ΔDispl	Weights	Bench:	Family	ΔCost	ΔTotal	Feasibility
Pump 1	0.0263	1.0000	1.3421	0.3421	0.0277	1.0000	1.3809	0.3809	0.7230	N
Pump 2	0.0196	1.0000	0.9795	0.0205	0.0206	1.0000	1.0290	0.0290	0.0495	Y
Pump 3	0.0154	1.0000	0.9450	0.0550	0.0172	1.0000	1.0576	0.0576	0.1126	Y
Pump 4	0.0133	1.0000	0.9246	0.0754	0.0152	1.0000	1.0531	0.0531	0.1285	Y
Pump 5	0.0110	1.0000	0.8508	0.1492	0.0131	1.0000	0.9435	-0.0565	0.0927	N*

* The displacement does not satisfy design requirement of deviation <10% hence infeasible

they are excluded from the current platform and are then considered for leveraging from the second platform to the targets.

In this stage of PCM, Research Question 2.4 is partly answered. In PCM, a multi-platform design is adopted, if it is found that the loss of performance of the products in the family is higher than acceptable. Otherwise it is considered that a single platform is sufficient to leverage the family of products. Objective 5, presented in Chapter I, is achieved by employing an evaluation function, which compares the normalized performances of the family members to that of benchmarks.

3.4.3 Platform Cascading Stage

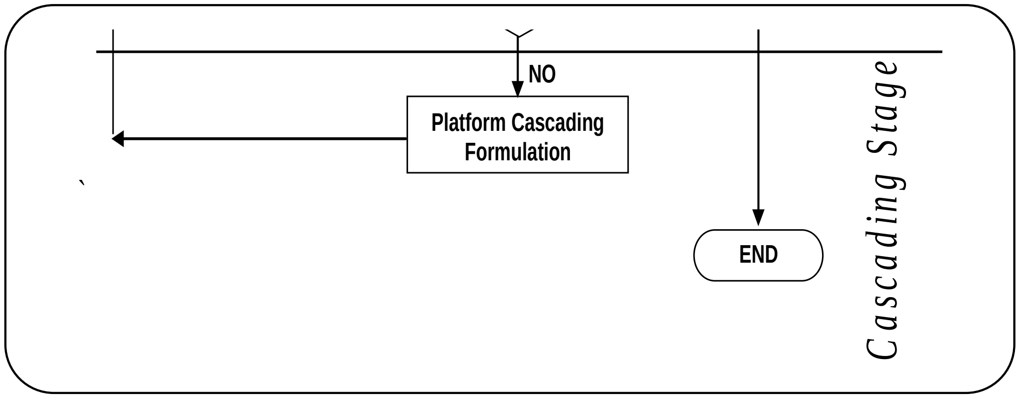


Figure 3.9: Cascading Stage

In the cascading stage, a cascading formulation is used to cascade the previous platform. The platform cascading formulation, (Table 3.13) starts from the design points from the previous platform. All the platform parameters from the previous platform (PP_{k-1}) are initially held to the values from the previous platform (C_{ik-1}), which is accomplished by the following constraints.

$$(x_{ij} - C_{ik-1})y_i = 0, \quad \forall i \in x_{pk-1} \; and \; \forall j \in p_{ck} \qquad \text{... (3.14)}$$

Here C_{ik-1} corresponds to the value of the platform parameters in the previous platform. The values of y_i parameters are held to 1; hence Equation 3.14 imposes the equality of x_{ij} parameters to their corresponding C_{ik-1} values.

The objective of the formulation is to improve the performance of the products by relaxing the previous platform. The formulation tries to select a platform parameter that minimizes the deviation of performance the most and upon conversion to a scale parameter. For this two constraints are introduced.

$$\sum y_i^3 \le N_{k-1} - 1, \quad \forall i \in x_{pk-1} \qquad \text{... (3.15a)}$$

$$\sum y_i \ge N_{k-1} - 1, \quad \forall i \in x_{pk-1} \qquad \text{... (3.15b)}$$

Here, N_{k-1} is the number of platform parameters in the previous platform; the formulation selects one of the parameters that can be converted to scale parameters. To satisfy the above constraints, only ($N_{k-1} - 1$) y_i parameters should be equal to 1 and the remaining ones should be 0. These conditions cannot be satisfied by constraints 3.15a alone, since it can accept any combination of values of y_i parameters whose sum is equal to $N_{k-1} - 1$. The y_i parameters should only accept binary values (0 or 1) to represent the sharing of parameters and not fractional values. Hence the constraint in 3.15b ensures that the values of y_i parameters are either 0 or 1. The objective function in this case is minimization of deviation parameters. The rest of the constraints are the same as the single platform formulation.

Table 3.13: General Platform Cascading Formulation

Minimize

$$\sum_{l=1}^{p}\sum_{j=1}^{m} f(w_{lj}, d_{lj}^{-}, d_{lj}^{+})$$

Subject to

(1) $x_{ij}^{lower} \le x_{ij} \le x_{ij}^{upper}, \quad \forall \quad i \in I \quad and\ \forall \quad j \in p_{ck}$, Bounds on the design variable

(2) $0 \le y_i \le 1$

$$y_i = \begin{cases} 1 & \text{when } \mathrm{x_i} \text{ is a platform} \\ 0 & \text{when } \mathrm{x_i}\ \textit{i}\text{s not a platform} \end{cases}$$

(3) $g_t(x) = 0,1,...s$, Individual product constraints

(4) $(x_{ij} - C_{ik-1})y_i = 0, \quad \forall \quad i \in x_{pk-1}\ and\ \forall j \in p_{ck}$, Commonality constraint

(5) $\sum y_i^3 \le N_{k-1} - 1, \quad \forall\ i \in x_{pk-1}$

$\sum y_i \ge N_{k-1} - 1, \quad \forall\ i \in x_{pk-1}$, Cascading constraints

(6) $A_{lj}(x) + d_{lj}^{-} + d_{lj}^{+} = G_{lj} \quad \forall l \in L$ and $j \in J$, Objectives transformed to system goals

(7) $d_{lj}^{-}, d_{lj}^{+} \ge 0, \quad d_{lj}^{-}.d_{lj}^{+} = 0 \quad \forall\ l \in L$ and $j \in J$, Non negativity of deviation variables

In the case of axial piston pumps, the objective function consists of minimization of positive deviation of cost from the targets and both the positive and negative deviation of displacement from targets. The design variables have the same bounds for all of the formulations. Since y_1 (swash plate angle) is a scale parameter in the first platform, it will remain a scale parameter in the subsequent platforms. Hence y_1 is assigned a value of 0. Parameters x_{2j}, x_{3j} *and* x_{4j} are held initially to 29.33, 6, and 23.79 respectively.

Table 3.14: Platform Cascading Formulation Applied to Pump Case study

Minimize

$$\sum_{j=1,5} f(d_{1j}^{+}) + \sum_{j=1,5} f(d_{2j}^{+}, d_{2j}^{-})$$

Subject to

(1)Bounds on the design variables

$$9.0 \le x_{1j} \le 21.0$$

$$14.0 \le x_{2j} \le 31.0$$

$$5.0 \le x_{3j} \le 7.0, x_3 = \text{Integer}$$

$$2.0 \le x_{4j} \le 30.0 \quad j = 1,5$$

(2) Platform commonality decision variables

$$y_1 = 0, \quad 0 \le y_2, y_3, y_4 \le 1$$

(3) Platform commonality constraint (cascading)

$$(x_{2j} - 29.33)y_2 = 0 \quad j = 1, 5$$

$$(x_{3j} - 6)y_3 = 0 \quad j = 1, 5$$

$$(x_{4j} - 23.79)y_4 = 0 \quad j = 1, 5$$

(4) Cascading constraints

$$\sum y_i^3 \ge 2$$

$$\sum y_i \le 2 \quad i = 2, 3, 4$$

(5) Displacement requirement for the product family

$$\frac{1}{4000 \tan(x_{1j})(2x_{2j} - x_{4j})x_{3j}^2 x_{2j}^2} = \{38, 91\} \quad j = 1,5$$

(6) Stress in the plunger < 175 Mpa

$$\frac{P_j \times (\frac{x_{4j}}{2})^2}{(\frac{x_{2j}}{2})^2 - (\frac{x_{4j}}{2})^2} \le 175 Mpa \quad j = 1,5$$

(7) Plunger feasible geometry and manufacturability constraint

$$x_{4j} - x_{2j} \ge 4 \quad j = 1, 5$$

Table 3.14 Contnd….

(8) Target cost of the pumps

$C_j / C_j^* + d_{1j}^- - d_{1j}^+ = 1.0 \quad j = 1,5$

(9) Target displacement of the pumps

$D_j / D_j^* + d_{2j}^- + d_{2j}^+ = 1.0 \quad j = 1,5$

(10) Non negativity of deviation variables

$d_{1j}^{-/+}, d_{2j}^{+/-} \geq 0 \quad j = 1,5$

Platform commonality constraints in the case of cascading parameters ensure that all the platform parameters have the same value as the preceding platform. For the first platform, three parameters were held as platform parameters. For the second platform, two out of the previous three parameters will be selected as the platform parameters. Cascading constraints select two corresponding y_i parameters and holds their value to one and forces the other to zero. The first cascading constraint can be satisfied by a combination of y_i parameters with fractional values (say $y_2 = 3/4; y_3 = 1/2; y_4 = 3/4$). Such a combination of parameter sharing does not physically make sense. The second constraint is introduced to ensure that two of the y_i parameters are assigned a value of 1 and the other one is assigned a value of zero. The rest of the constraints are the same as that of the single platform stage.

Figure 3.10 shows the variation of y_i parameters for different design iterations. The formulation returned selections of y_3 *and* y_4 as platform parameters with their values equal to the corresponding values in Platform 1. Pump 5 has performance within acceptable limits.

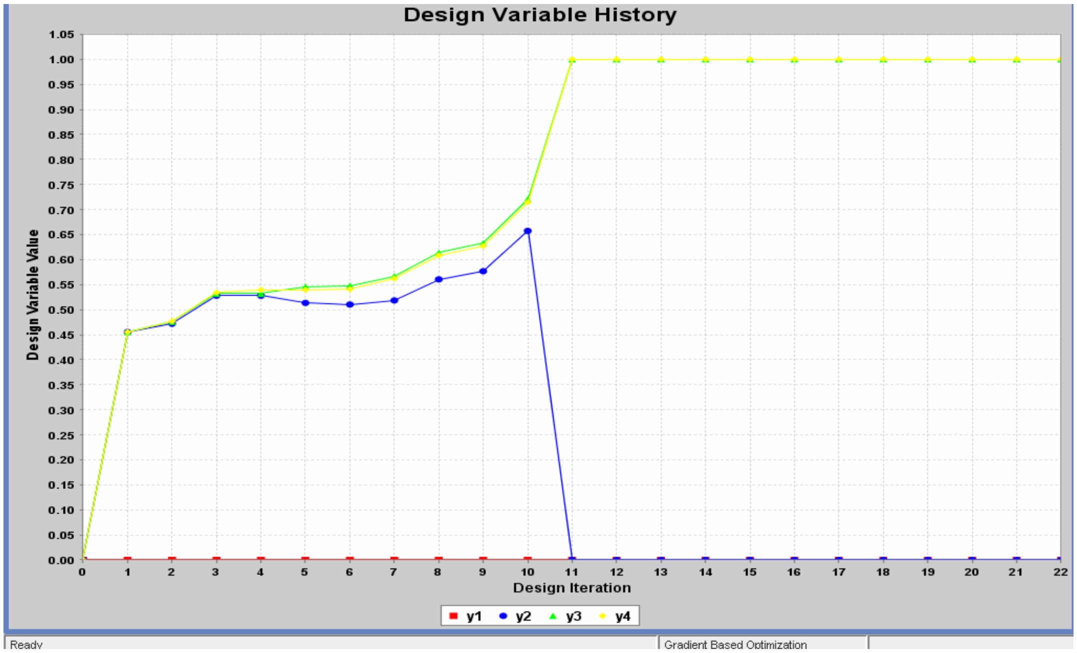


Figure 3.10: Variation of y_2, y_3 *and* y_4 Values During Various Design Iterations

The formulation was repeated with only one platform parameter, but no platform could be arrived which could leverage both pumps 1 and 5 (Table 3.16). Hence, it was

Table 3.15: Results from Platform Cascading (Platform 2) for Pumps 1 and 5

	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
X_1	9.01	-----	-----	-----	13.90
X_2	27.28	-----	-----	-----	30.00
X_3	6.00	-----	-----	-----	6.00
X_4	23.79	-----	-----	-----	23.79
Displ	47.62	-----	-----	-----	90.97
Cost	48.06	-----	-----	-----	79.04
Stress	131.01	-----	-----	-----	93.32
Geom	4	-----	-----	-----	6.208

decided to leverage pump 5 using platform 2, consisting of x_3 *and* x_4 as platform parameters, and then leverage pump 5 using x_4 as the platform parameter.

The value of the parameters for product family members leveraged from different platforms is superimposed in Table 3.17. It can be seen that x_4 is common for all the products in the family. Parameter x_3 is common for pumps 2, 3, and 4 and x_2 is common for pumps 2, 3, and 4. The cascading strategy for the axial product family is shown in Table 3.18. It shows the platform parameters for each platform, the products that are leveraged from them and the cascading relation between each platform.

Table 3.16: Evaluation of Products Leveraged from Platform 2

	Displacement (Normalized)				Cost (Normalized)					
	Weights	Bench:	Family	ΔDispl	Weights	Bench:	Family	ΔCost	ΔTotal	Feasibility
Pump 1	0.02632	1	1.25313	0.25313	0.02771	1	1.33192	0.33192	0.58505	N
Pump 5	0.01099	1	1	1.1E-07	0.01314	1	1.03898	0.03898	0.03898	Y

Table 3.17: Combined Results for Platforms 1, 2, and 3

	Pump 1	Pump 2	Pump 3	Pump 4	Pump 5
X1	9.00	9.01	11.42	13.04	13.90
X2	27.30	28.52	28.52	28.52	30.00
X3	5.00	6.00	6.00	6.00	6.00
X4	23.79	23.79	23.79	23.79	23.79
Displ	37.94	51.00	65.00	74.49	90.97
Cost	38.98	49.95	61.42	69.34	79.04
Stress	103.00	131.48	115.04	115.04	93.32
Geom	5.49	4.73	4.73	4.73	6.21

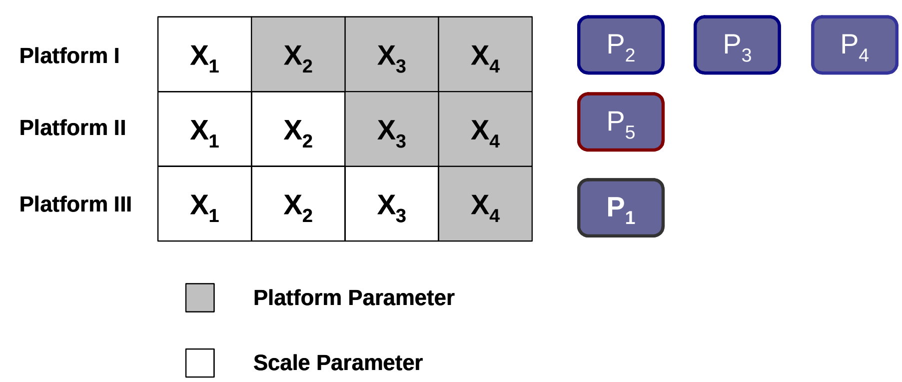


Figure 3.11: Cascading Strategy for Axial Pump Family Design

The cascading stage of PCM addresses questions related to the Multi-platform design. The research question 2, presented in Chapter2, deals with extending the formulation for single platform design to the case of a multi-platform design. In PCM, the modeling approach in case of cascading formulation is similar to that of single platform formulation. Both initially convert the MINLP to a continuous problem and then constraints the solution to discrete spaces. In case of the cascading formulation, cascading constraints are simultaneously used to select the platform parameters and also to constraint it to accept only binary values.

The research questions that are answered in this section are:

RQ2) *How do we extend the mathematical model to design product families supported by multiple platforms?*

RQ2.1) How do we extend the single platform representation as sub-problem for deciding configuration of multiple the platforms?

RQ2.2) How do we extend the mathematical model to evaluate the optimum number of platforms?

RQ2.3) How do we maintain a relationship between the different platforms so that commonality between the different platforms can be established?

PCM does not arrive at a specific value for the optimum number of platforms (Objective O7). The number of platforms depends on the value of acceptable loss of performance of the family members and also the choices made by the designer (case 2 and case 3) at the platform evaluation stage.

In PCM, through cascading, relationship is maintained between the different product platforms. This strategy helps to establish commonality even between platforms, thus increasing cost saving. Objective O6 and O7 were partly achieved through the evaluation stage and partly through the cascading stage.

3.5 Summary

In this chapter, the general steps involved in PCM were presented. Several optimization formulations developed to perform trade-offs between commonality and loss of performance to arrive a platform and family were developed. The different research questions posed in Chapter I were reintroduced to explain how PCM addresses these questions. The different objectives that are achieved through PCM were explained in the corresponding sections.

CHAPTER 4

UNIVERSAL ELECTRIC MOTOR PRODUCT FAMILY DESIGN

In this chapter, the capability of PCM in designing product families based on common platform(s) is demonstrated further by its application to the design of a ten-motor Universal Electric Motor (UEM) product family. In the electric motor example introduced in Section 4.1, a family of ten electric motors, each having a different torque capacity, are to be designed based on common platform(s). The UEM case study was introduced by Simpson *et al*. (2001) and has since been used by many researchers to demonstrate their product family design methods. The application of PCM to the case study is shown in Section 4.2. The results obtained from PCM are compared to those of existing methods in Section 4.3

4.1 Universal Electric Motor Case Study (Adapted from Simpson *et al*., 2001)

Universal Electric Motors are capable of operating on alternating current (AC) and direct current (DC). They deliver more torque for a given current than any other type of AC capable motor (Chapman, 1991). The high performance characteristics of the universal motor, coupled with their flexibility, have led to a wide variety of household products, such as electric drills and saws, blenders, vacuum cleaners, and sewing machines (Veinott and Martin, 2006).

Meyer and Lehnerd (1997) reported that Black and Decker developed a family of Universal Electrical Motors for its power tools in response to a new safety regulation: double insulation. Prior to that, Black and Decker used different motors in each of their 122 basic tools with hundreds of variations, from jigsaws and grinders to edgers and hedge hammers. Through redesign and standardization, Black and Decker was able to produce all their tools using a line of motors that varied in stack lengths and the amount of copper wrapped within the motor. As a result, all of the motors could be manufactured on a single machine with stack lengths varying from 0.8 in to 1.75 inches and power outputs varying from 60 to 650 watts. Through standardization and platform scaling around the motor stack length they were able to reduce material cost from $0.77 to $0.42 per motor and labor costs from $0.248 to $0.045 per motor, yielding an annual savings of $ 1.82 million per year. Tool costs were reported to be reduced by as much as 62%.

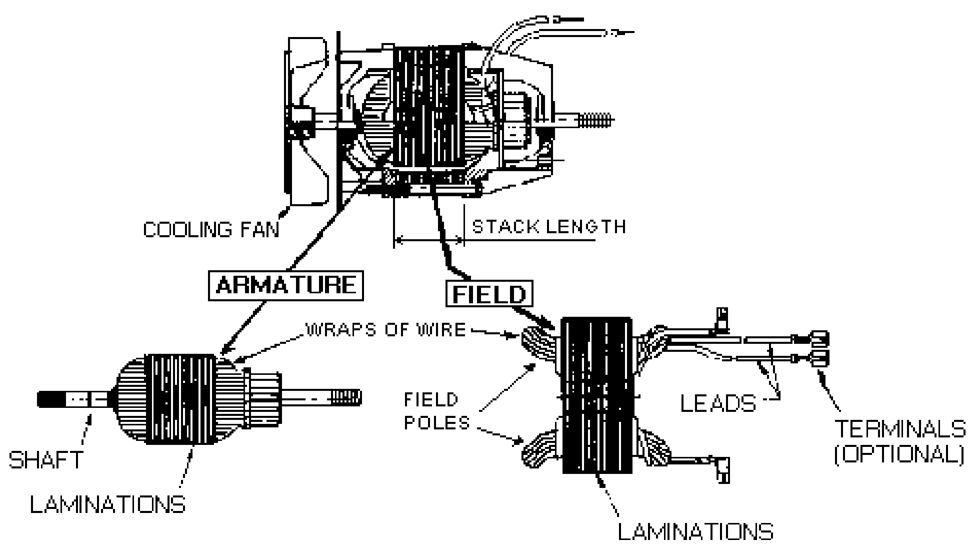
As shown in Table 4.1, a Universal Electrical Motor is composed of an armature and a field, which are also referred to as the motor and stator, respectively. The armature consists of a metal shaft and slats (armature poles) around which wire is wrapped longitudinally as many as a thousand times. The field consists of a hollow cylinder within which the armature rotates. The field also has wire wrapped as many as a hundred times longitudinally around interior metal slats (field poles). In order to reduce cost, size and weight, it is most desirable for the motor to satisfy the performance requirements with the least overall mass and highest efficiency. (Simpson *et al.*, 2001)

The design objective is to design a family of ten Universal Electrical Motors that satisfy a variety of torque and power requirements that utilize a suitable platform, with different varieties scaled from the platform to meet specific requirements. The product parameters for the electric motors are

1. Number of turns in the armature
2. Number of turns in the field
3. Area of the armature
4. Area of the field wire
5. Radius of the motor
6. Thickness of the stator
7. Current drawn by the motor
8. Stack length

There is no manufacturing advantage to be gained by holding current as a platform. Moreover, varying the current can help to achieve different power requirements without having to vary other parameters that affect the manufacturing process. Torque requirements for an individual electric motor are T = {0.05, 0.10, 0.125, 0.15, 0.30, 0.25, 0.30, 0.35, 0.40, and 0.5). The constraint on magnetizing intensity ensures that the magnetic flux intensity within each motor does not exceed the physical flux carrying capacity of the steel. The constraint on feasible geometry ensures that the thickness of

Table 4.1: Requirements for the Universal Electric Motor Product Family (Adapted from Simpson *et al.*, 2001)



Name	***Requirement***
Torque	T= {0.05,0.10,0.125, 0.15, 0.25, 0.30, 0.40, 0.50}
Power	= 300 W
Magnetizing Intensity, H	$\leq$ 5000 A turns/m
Feasible geometry	Radius of motor > thickness of stator
Efficiency of each motor	$\geq$ 0.70
Mass of motor	$\leq$ 2.0 Kg

the stator does not exceed the radius of the stator since the thickness is measured from the outside of the motor inward. The required output power is taken as 300 W and the ten torque values range from 0.05 to 0.5.

There are two goals for each motor, efficiency and mass, with targets of 70% and 0.5kg, respectively. A lower bound of 15% for efficiency and an upper bound of 2.0 kg for mass are imposed for each product within the product family. The design requirements, range of possible values for product parameters, and the constraints related to the product family as introduced by Simpson are shown in Table 4.1.

Several researchers have used the Universal Electrical Motor example as a benchmark for testing their product design methodology. The simplicity and completeness of the mathematical model has made the universal electric model problem a de facto benchmark problem for the different approaches in scale-based product family design developed over the years (Simpson *et al*., 2001; Nayak *et al*., 2002; Messac, *et al*., 2002; Dai and Scott, 2005). The relation between the design parameters and performances in the Universal Electrical Motor model are non-linear. Therefore, the Universal Electrical Motor example will present us with the same challenges that we encounter in a mechanical engineering design scenario. The complete model for the Universal Electrical Motor and the underlying equations as reported by Simpson *et al*. (2006) are shown in Appendix B.

The design objective of Simpson's PPCEM was to design a family of ten Universal Electrical Motors that satisfy a variety of torque and power requirements by scaling a common motor platform around the stack length of the motor. The capability of the Platform Cascading Method for designing scale based product families based on multiple platforms is demonstrated by its application to the Universal Electrical Motor design problem. In this dissertation, the Universal Electrical Motor family design problem will be treated as a non-platform specified design problem. The PCM returns the configuration of the platform(s) from which each motor is leveraged, the value of platform parameters, the value of scale parameters for each motor, and the performance of each motor.

4.2 Application of PCM to the Universal Electric Motor Case Study

As explained in the Section 3.4, PCM is a three-stage design method. The general steps introduced in Section 3.4 will be followed for the design of the electric motor product family. The three stages are explained in the following sub-sections. The general steps in the PCM method are shown in Table 3.5.

The product family *PF* consists of ten electric motors $\{P_1, P_2, ..P_{10}\}$ with torque requirements of $\{$0.05, 0.10, 0.125, 0.15, 0.30, 0.25, 0.30, 0.35, 0.40, and 0.5$\}$. There are eight design parameters that describe each product in the family; hence, there are eight platform commonality parameters in the set *Y*. These parameters are $y_1, y_2, .. y_8$, corresponding to the product parameters $x_1, x_2, .. x_8$. The design objective is to find the optimum value of X that results in minimum performance loss due to commonality and maximum commonality. Since there are eight parameters that describe the motors and ten motors in the family, vectoring of product parameters X can be represented as

$$X = \begin{pmatrix} x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{110} \\ x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{210} \\ x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36}, x_{37}, x_{38}, x_{39}, x_{310} \\ x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}, x_{47}, x_{48}, x_{49}, x_{410} \\ x_{51}, x_{52}, x_{53}, x_{54}, x_{55}, x_{56}, x_{57}, x_{58}, x_{59}, x_{510} \\ x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}, x_{47}, x_{48}, x_{49}, x_{410} \\ x_{71}, x_{72}, x_{73}, x_{74}, x_{75}, x_{76}, x_{77}, x_{78}, x_{79}, x_{710} \\ x_{81}, x_{82}, x_{83}, x_{84}, x_{85}, x_{86}, x_{87}, x_{88}, x_{89}, x_{810} \end{pmatrix}$$

4.2.1 Stage 1: Single Platform Stage

The first step in the single platform stage is the individual optimization of product instances for the purpose of establishing benchmarks. The general optimization formulation for designing the products individually, subject to the requirements and considering no commonality between them, is shown in Table 3.6. This formulation for the universal electric problem is repeated 10 times for each of the products in the family. The general formulation application to the case study is shown in Table 4.2.

The formulation uses a goal-programming model to tackle the multi-objective (target efficiency, target mass) nature of the problem. The positive and negative deviations of the actual efficiency and mass of the motors are captured using deviation variables $d_{Eff}^{+/-}$ and $d_{Mass}^{+/-}$, respectively. In the objective function, the undesirable negative deviation of efficiency and positive deviation of mass (d_{Eff}^{-} and d_{Mass}^{+}) is minimized. The relation between design variables and performance are obtained by simplifying the corresponding equations shown in Appendix B.

Table 4.2: Individual Optimization Formulation Applied to a Universal Electric Motor Family

Indices

$i =$ Design paremeters $\quad \mathrm{I} = \{1,2,...8\}$

$l =$ System Goals $\quad \mathrm{L} = \{1,2\}$

Parameters

$\lg a$ length of air gap $= 0.007 cm$

r_{cu} is the resistivity of copper wire

dsteel density of steel

μ_0 permeablitity of air

μ_r permeablitity of steel given by the following relation where 'h' is the magnetic instensity

$$
|\mu_\mathrm{r}| = \begin{cases} -0.22791h^2 + 52.4111h + 3115.8 & h \leq 220 \\ 11633.5 - 1486.33\ln(h) & 220 \leq h \leq 1000 \\ 1000 & h \geq 1000 \end{cases}
$$

Variables

$x_i =$ Product parameters

$d_{Eff}^{-} =$ Negative deviation of goal 1 (Efficiency > 0.70) from the target

$d_{Eff}^{+} =$ Positive deviation of goal 1 (Efficiency > 0.70) from the target

$d_{Mass}^{-} =$ Negative deviation of goal 2 (Mass < 0.5) from the target

$d_{Mass}^{+} =$ Positive deviation of goal 2 (Mass < 0.5) from the target

$E =$ Efficiency of motor

$M =$ Mass of motor

Objective

$d_{Eff}^{-} + d_{Mass}^{+}$

Subject to:

(1) Bounds on the design variable

$100 \leq x_1 \leq 1500\ turns$

$1 \leq x_2 \leq 500\ turns$

$0.01 \leq x_3 \leq 1\ mm^2$

$0.01 \leq x_4 \leq 1\ mm^2$

Table 4.2: continued…..

$1 \le x_5 \le 1\, cm$

$0.5 \le x_6 \le 10\, mm$

$1 \le x_7 \le 6\, amps$

$0 \le x_8 \le 10\, cm$

(2) Magnetic Intensity of each motor less than 5000A.turns/m

$$\frac{(2x_2 x_7)}{\pi(2x_5 + x_6)/2 + (2(x_5 - x_6 - 0.007) + 2 * 0.007))} \le 5000$$

(3) Feasible geometry (thickness < radius)

$x_6 < x_5$

(4) Mass of motor (M) < 2.0 Kg

$$\pi\, dsteel\ x_7(x_5^2 - (x_5 - x_6)^2) +$$
$$\pi\, dsteel\ x_7(x_5 - (x_5 - x_6 - l\ gap)^2)(2x_7 + 4(x_5 - x_6 - l\ gap))x_1 x_3 +$$
$$((2x_7 + 4(x_5 - x_6)(2x_2 x_4))dcopper \quad \le 2.0 \quad kg$$

(5) Efficiency (E) > 0.15

$$\frac{1}{115x_7}(113x_7 - (\frac{\mu_r . x_1(2x_7 + 4(x_5 - x_6 - l\ gap))}{x_3} + \frac{2\mu_r . x_2(2x_7 + 4(x_5 - x_6))}{x_4})x_7^2) \ge 0.15$$

(6) Torque requirement for individual motors

$$\frac{(x_{1j} x_{2j} x_{7j}^2)}{\pi(\frac{0.000175}{x_7 \mu_0 (x_5 - x_6 - l\ gap}) + \frac{1}{x_7 \mu_0 \mu_r} + \frac{\pi(2x_5 + x_6)}{4\mu_0 \mu_r x_6 x_7}} = t \in T$$

T = {0.05, 0.10, 0.125, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.5}

(7) Deviation of actual efficiency from target efficiency (70%)

$E / 0.7 + d_{Eff}^{-} - d_{Eff}^{+} = 1.0$

(8) Deviation of actual mass from target mass (0.5 kg)

$M / 0.5 + d_{Mass}^{-} - d_{Mass}^{+} = 1.0$

Here, $x_1, x_2, .. x_8$ corresponds to the design parameters - number of turns in the armature, number of turns in the field, area of the armature, area of the field wire, radius of the motor, thickness of the stator, current drawn by the motor, and stack length, respectively. The formulation is repeated for each product instance. This individual optimum corresponds to the best performance that can be achieved for each product in the family. The performances of individually optimized motors are used as benchmarks while designing motors using the product family approach. Table 4.3 shows the results obtained after the individual optimization.

Table 4.3 Results from Individual Optimization

	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6	Motor 7	Motor 8	Motor 9	Motor 10
X1	1019	1020	1021	1021	1029	1011	1024	1021	1020	1022
X2	57	65	69	75	66	57.4	61	54	58	54
X3	0.256	0.215	0.214	0.225	0.218	0.201	0.229	0.218	0.239	0.248
X4	0.272	0.258	0.255	0.251	0.217	0.201	0.232	0.238	0.234	0.243
X5	2.06	2.24	2.24	2.22	2.16	5.49	2.23	2.29	2.37	2.49
X6	5.94	5.72	5.71	5.69	5.56	4.84	5.43	5.55	5.56	5.52
X7	3.19	3.62	3.72	3.73	4.1	2.38	5.62	5.36	5.13	5.82
X8	1.2	1.47	1.65	1.84	2.32	2.3	2.5	2.8	3.12	3.1
Mag: Intensity	3543	3160	4817	4981	5000	5000	5000	5000	5000	5000
Efficiency	0.817	0.72	0.705	0.7	0.635	0.59	0.564	0.548	0.508	0.454
Mass	0.33	0.39	0.415	0.45	0.5	0.56	0.63	0.694	0.733	0.78

The benchmark efficiencies and mass obtained after individual optimization for the product instances are 81.7, 72, 70.5, 70, 63.5, 59.0, 56.4, 54.8, 50.8, and 45.4 % and 0.33, 0.39, 0.415, 0.45, 0.5, 0.56, 0.63, 0.694, 0.733, and 0.78 kg, respectively. The magnetizing intensity for all the motors is within the allowable limit of 5000 A. turns/m. After establishing the benchmarks, the number of platforms counter 'k' is initiated.

Since this being the first platform developed, it is given an initial value of 1. In the single platform stage, all the products are considered for leveraging, hence

$$P_{c1} = PF = \{P_1, P_2, ...P_{10}\}$$

Now, the single platform optimization formulation is used to arrive at a platform that can be used to leverage all the products in the family. The application of the general single platform formulation is shown in Table 4.4.

Table 4.4: Single Platform Formulation Applied to Universal Electric Motor Family

Indices
Same as individual optimization formulation, Table 3.2

Parameters
Same as individual optimization formulation, Table 3.2

Variables

$x_{i\,j}$ = Product parameters i for each family member j

y_i = Platform commonality variables

$d^-_{Eff\ j}$ = Negative deviation of goal 1 (Efficiency > 0.70) from the target for product j

$d^+_{Eff\ j}$ = Positive deviation of goal 1 (Efficiency > 0.70) from the target for product j

$d^-_{Mass\ j}$ = Negative deviation of goal 2 (Mass < 0.5) from the target for product j

$d^+_{Mass\ j}$ = Positive deviation of goal 2 (Mass < 0.5) from the target for product j

E_j = Efficiency of motor j

M_j = Mass of motor j

Objective

$$z = \sum_{j=1}^{10} d^-_{Eff\ j} + \sum_{j=1}^{10} d^+_{Mass\ j} - \sum y_i$$

Subject to:

(1) Bounds on the design variable

$100 \le x_{1\,j} \le 1500\ turns \quad \forall j \in J$

$1 \le x_{2\,j} \le 500\ turns \quad \forall j \in J$

$0.01 \le x_{3\,j} \le 1\ mm^2 \quad \forall j \in J$

$0.01 \le x_{4\,j} \le 1\ mm^2 \quad \forall j \in J$

$1 \le x_{5\,j} \le 1\ cm \quad \forall j \in J$

$0.5 \le x_{6\,j} \le 10\ mm \quad \forall j \in J$

$1 \le x_{7\,j} \le 6\ amps \quad \forall j \in J$

$0 \le x_{8\,j} \le 10\ cm \quad \forall j \in J$

$0 \le y_i \le 1 \quad \forall i \in I$

Table 4.4 Continued …

3. Magnetic Intensity of each motor less than 5000A.turns/m

$$\frac{(2x_{2j}x_{7j})}{\pi(2x_{5j}+x_{6j})/2+(2(x_{5j}-x_{6j}-0.007)+2*0.007))} \leq 5000 \quad \forall j \in J$$

4. Feasible geometry

$$x_{6j} < x_{5j} \quad \forall j \in J$$

5. Mass of motor (M) < 2.0 Kg

$$\pi\ dsteel\ x_{7j}(x_{5j}^2-(x_{5j}-x_{6j})^2)+$$
$$\pi\ dsteel\ x_{7j}(x_{5j}-(x_{5j}-x_{6j}-l\ gap)^2)(2x_{7j}+4(x_{5j}-x_{6j}-l\ gap))x_{1j}x_{3j}+$$
$$((2x_{7j}+4(x_{5j}-x_{6j})(2x_{2j}x_{4j}))dcopper \quad \leq 2.0 \quad kg \quad \forall j \in J$$

6. Efficiency (E) > 0.15

$$\frac{1}{115x_{7j}}(113x_{7j}-(\frac{\mu_r.x_{1j}(2x_{7j}+4(x_{5j}-x_{6j}-l\ gap))}{x_{3j}}+\frac{2\mu_r.x_{2j}(2x_{7j}+4(x_{5j}-x_{6j}))}{x_{4j}})x_{7j}^2)$$
$$\geq 0.15$$

7. Torque requirement for individual motors

$$\frac{(x_{1j}x_{2j}x_{7j}^2)}{\pi(\frac{0.000175}{x_{7j}\mu_0(x_{5j}-x_{6j}-l\ gap})+\frac{1}{x_{7j}\mu_0\mu_r}+\frac{\pi(2x_{5j}+x_{6j})}{4\mu_0\mu_r x_{6j}x_{7j}}} = t \in T$$

T = {0.05, 0.10, 0.125, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.5}

8. Platform commonality constraints

$$(x_{ij}-x_{ij+1})y_i = 0 \quad \forall i \in I\ \&\ \forall j \in J,\ j \neq m$$

9. Integerizing constraints

$$y_i^2 - y_i = 0 \quad \forall i \in I$$

10. Deviation of actual efficiency from target efficiency (70%)

$$E_j/0.7 + d_{Eff\ j}^- - d_{Eff\ j}^+ = 1.0 \quad \forall j \in J$$

11. Deviation of actual mass from target mass (0.5 kg)

$$M_j/0.5 + d_{Mass\ j}^- - d_{Mass\ j}^+ = 1.0 \quad \forall j \in J$$

12. Deviation variables

$$d_{Eff\ j}^-, d_{Eff\ j}^+ \geq 0, \quad d_{Mass\ j}^+, d_{Mass\ j}^- \geq 0 \quad j \in J$$

The Universal Electrical Motor case study is treated in this dissertation as a non-platform specified case. Hence, the formulation should determine the platform parameters for each platform and the value of scale parameters corresponding to each product instance. In single platform optimization, a holistic view of the entire product family is adopted. A suitable platform is arrived at while simultaneously optimizing the platform and the product instances for maximum commonality and loss of performance due to commonality. The objective function consists of minimizing the undesirable negative deviation of efficiency of each motor, the positive deviation of mass of each motor, and the sum of platform commonality parameters.

The platform commonality parameters are initially introduced as continuous variables ($0 \leq y_i \leq 1$). Integerizing constraints are then used to force the formulation to accept only the values of 0 or 1 (binary) for the y_i parameters. This allows the formulation to evaluate the model for values in between 0 and 1. This is required for the formulation to be implemented in gradient-based optimization methods. The commonality constraints are used to force the commonality of platform parameters for all product instances. The constraint ensures that the platform parameters take the same value while scale variables take different values for different products in the family.

A detailed explanation of the mathematical background for the approach was presented in Section 3.4. The constraints for magnetic feasibility, mass, efficiency, geometric feasibility, and torque were introduced for each product instance. The model consisted of 128 design variables and 180 constraints. The formulation was implemented in VRAND® Visual DOC®, a commercially available non-linear optimization tool. Figure 4.1 (a)-(h) shows the variation of design parameters for different design iterations. It can be seen that the platform parameters are forced to take the same values for the different products in the family and the scale parameters have different values. Figures 4.2 (a) and (b) show the variation of y_i parameters. All the y_i parameters are initially assigned a value of 0.5 so that model is unbiased and does not favor any parameter. The formulation tries several values for different y_i values before arriving at the platform and scale parameters.

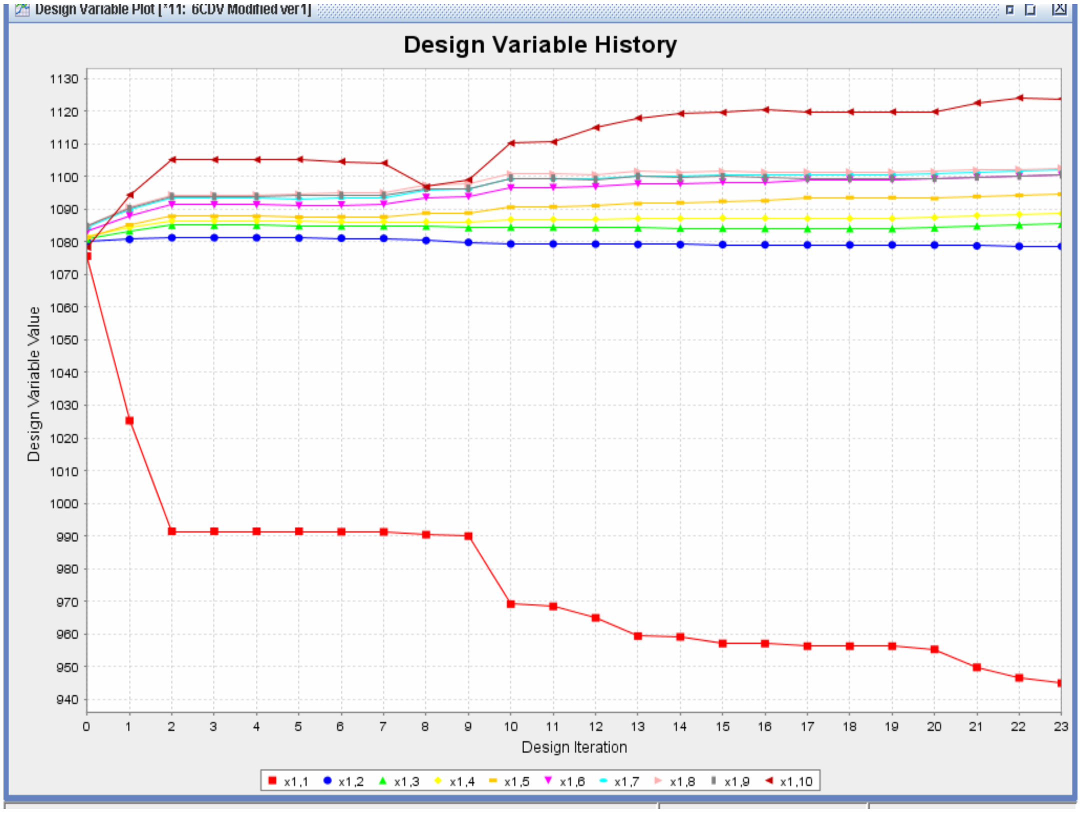


Figure 4.1 (a): Variation of x_{1j} for Different Design Iterations

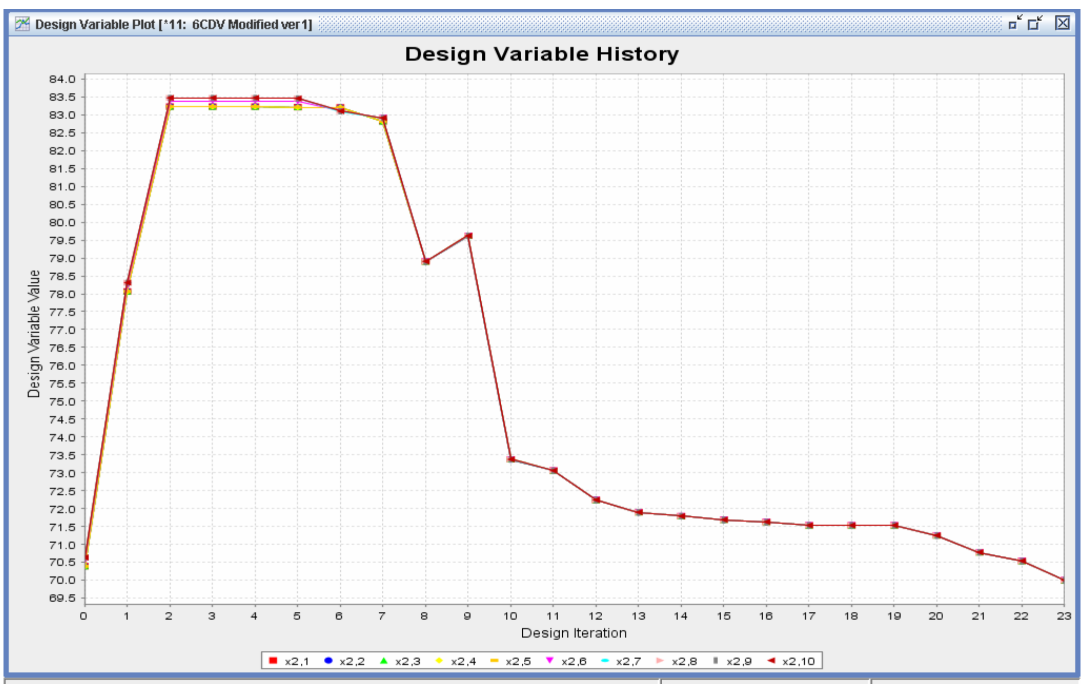


Figure 4.1 (b): Variation of x_{2j} Parameters for Different Design Iterations

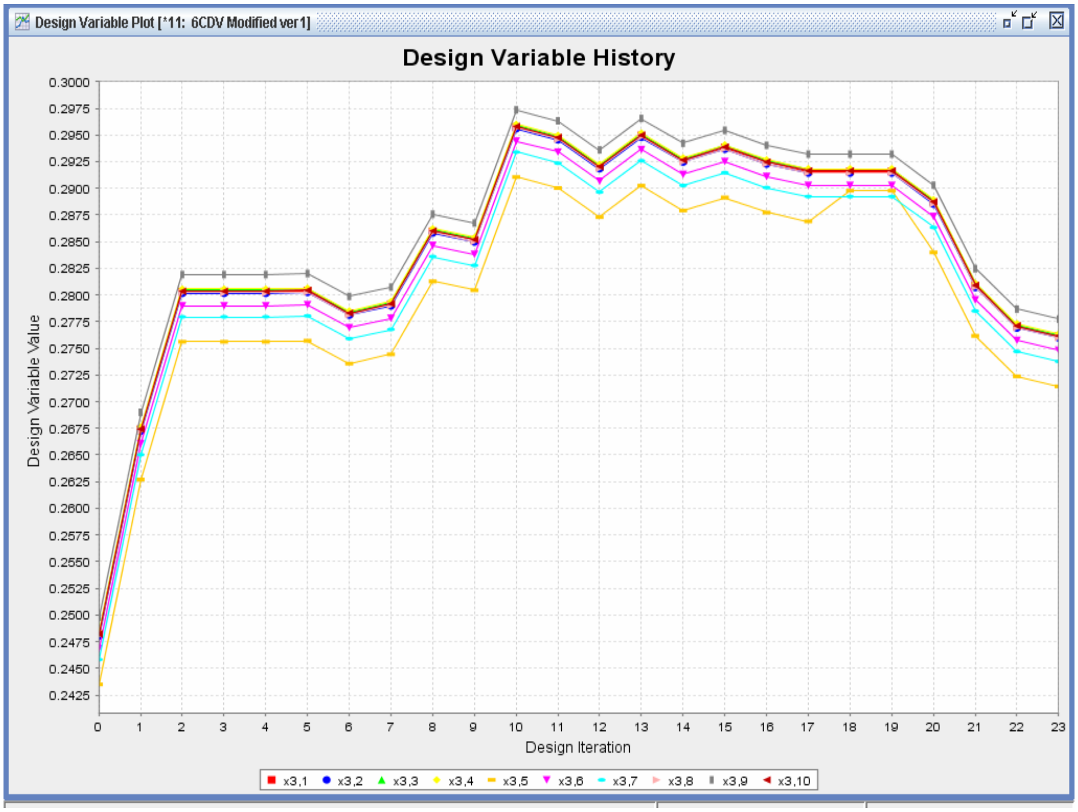


Figure 4.1 (c): Variation of x_{3j} Parameters for Different Design Iterations

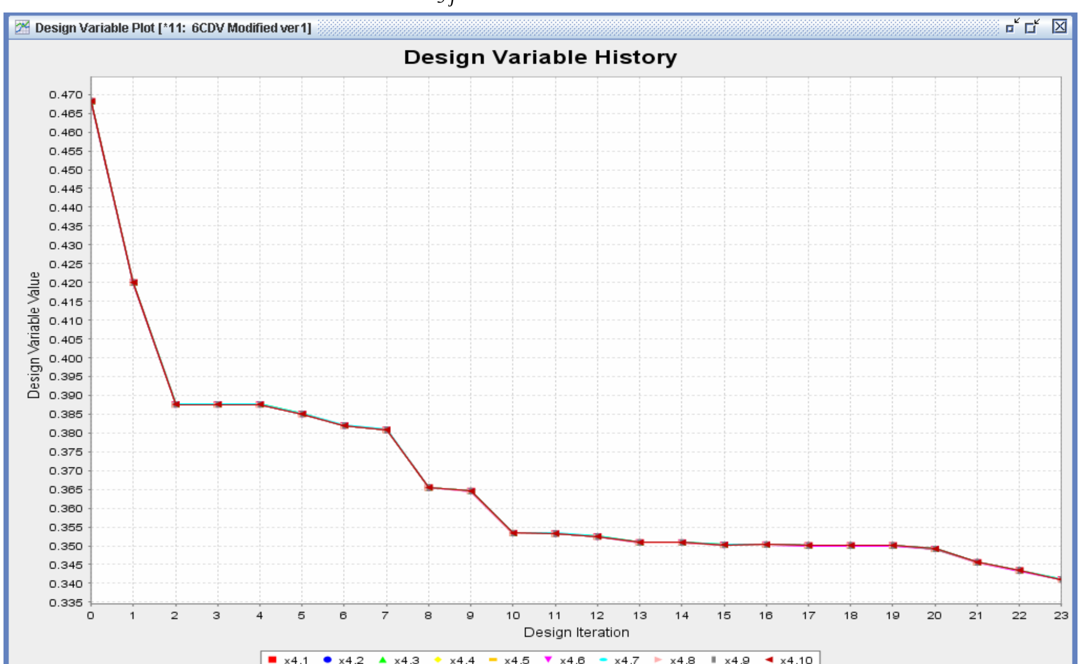


Figure 4.1 (d): Variation of x_{4j} Parameters for Different Design Iterations

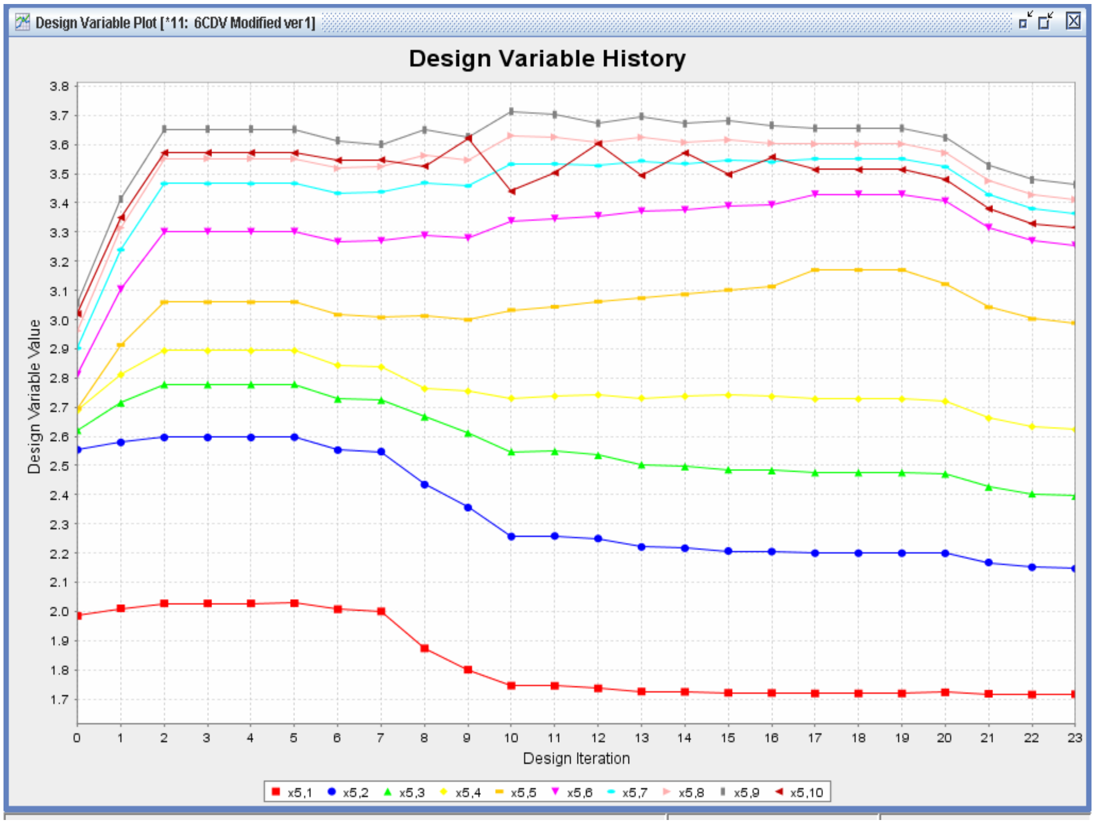


Figure 4.1 (e): Variation of x_{5j} Parameters for Different Design Iterations

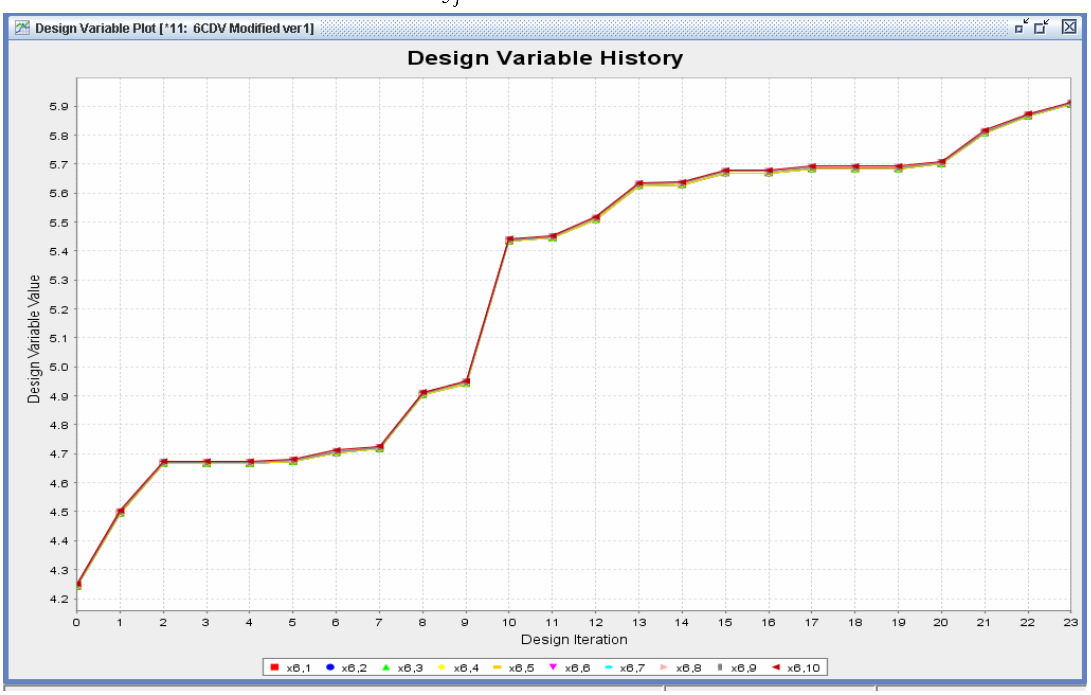


Figure 4.1 (f): Variation of x_{6j} Parameters for Different Design Iterations

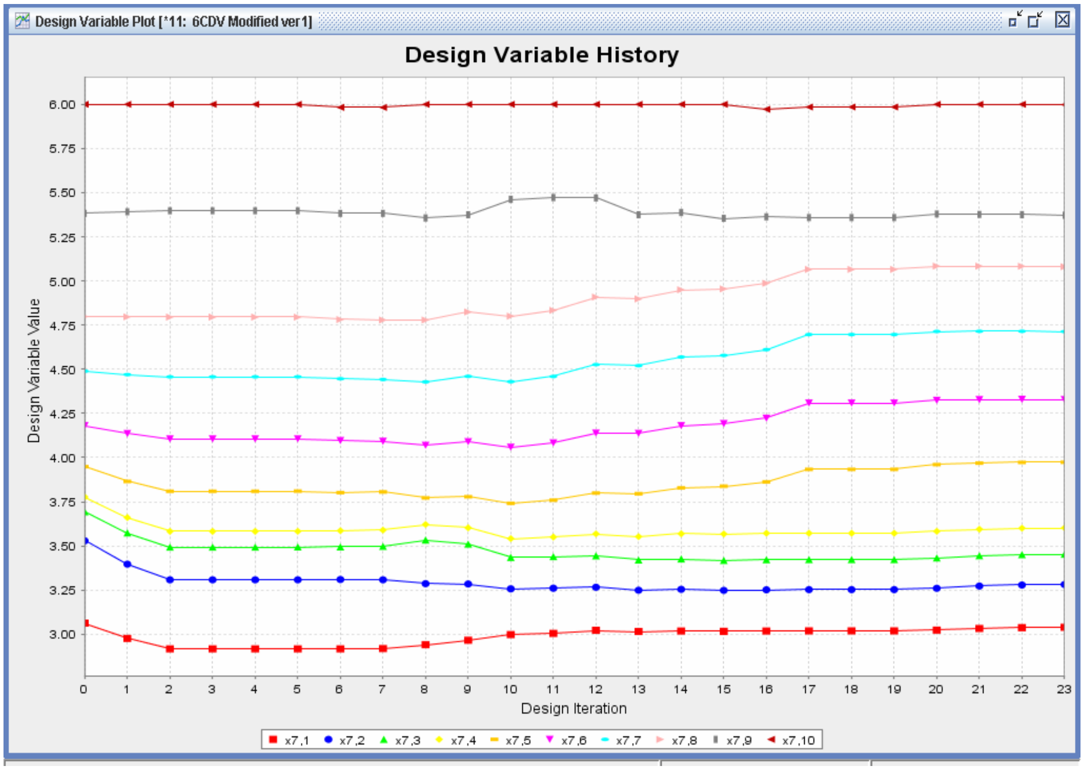


Figure 4.1 (g): Variation of x_{7j} Parameters for Different Design Iterations

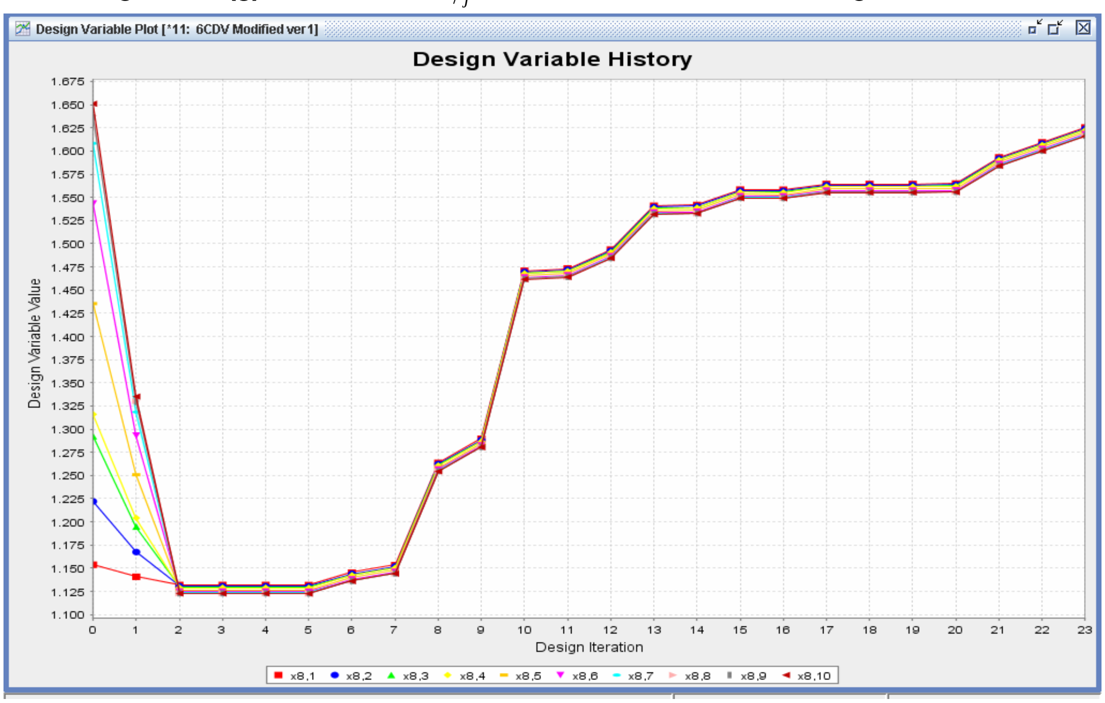


Figure 4.1 (h): Variation of x_{8j} Parameters for Different Design Iterations

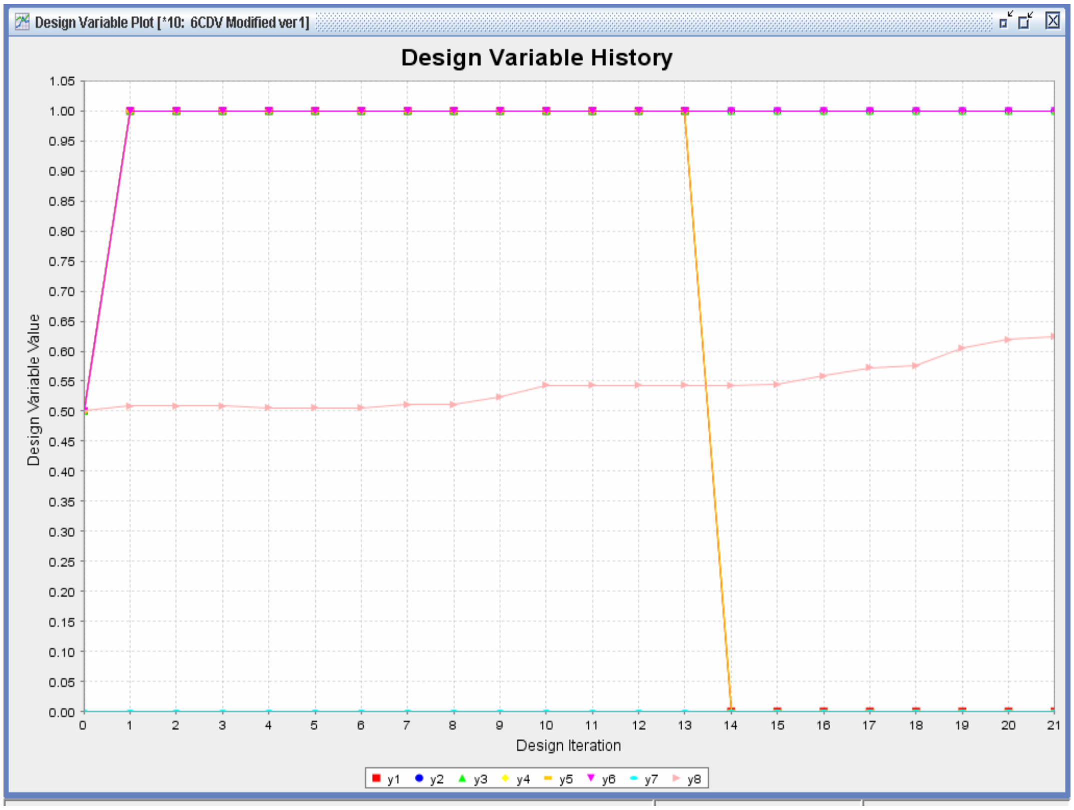


Figure 4.2 (a): Variation of y_i Parameters before Arriving at a Platform

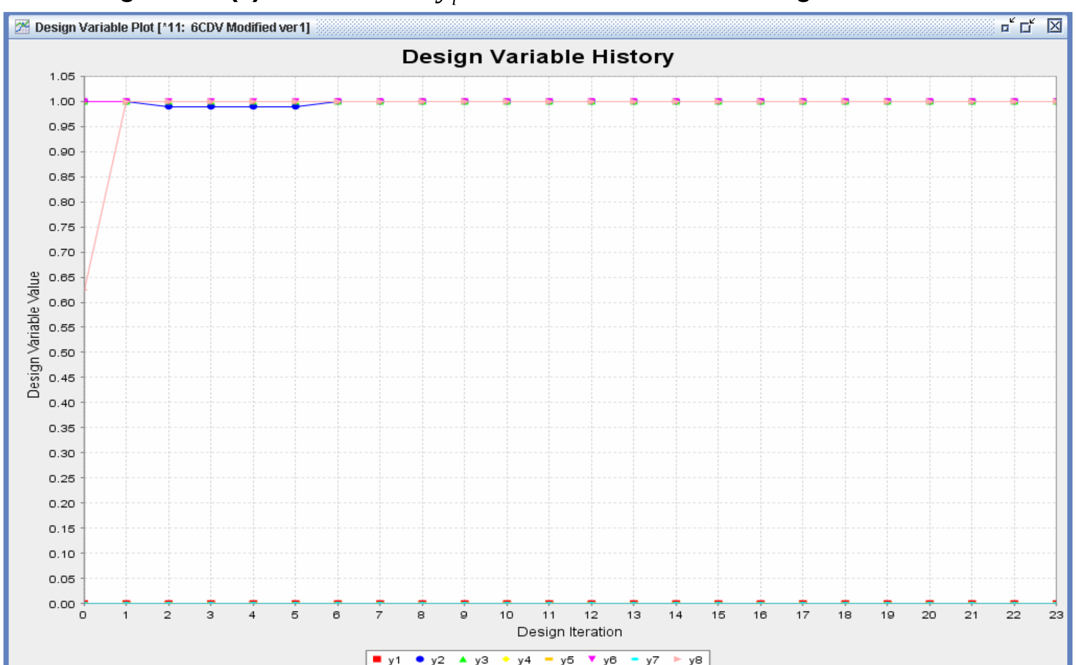


Figure 4.2 (b): Variation of y_i Parameters before Arriving at a Platform (Rerun)

Visual DOC requires several restarts in certain cases when it fails to reach an optimum solution within a certain number of iterations. In this case, the optimum was reached in two runs. Hence, Figures 4.2 (a) and (b) are included to show the complete variation of y_i parameters before arriving at the platform. Table 4.5 shows the results obtained from the single platform optimization formulation.

Table 4.5: Results from the Single Platform Optimization Formulation

	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6	Motor 7	Motor 8	Motor 9	Motor 10
X1	944.96	1078.61	1085.41	1088.65	1094.58	1100.44	1101.90	1102.57	1100.56	1123.78
X2	70	70	70	70	70	70	70	70	70	70
X3	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
X4	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
X5	1.72	2.15	2.40	2.62	2.99	3.25	3.36	3.41	3.46	3.32
X6	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91
X7	3.04	3.28	3.45	3.60	3.97	4.33	4.71	5.08	5.37	6.00
X8	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
Efficiency	0.81	0.72	0.70	0.69	0.66	0.63	0.58	0.55	0.49	0.43
Mass	0.35	0.51	0.54	0.57	0.59	0.66	0.70	0.74	0.76	0.77

The formulation returned a platform consisting of parameters x_2, x_3, x_4, x_6, x_8, with values of 70, 0, 38, 0.34, 5.91, and 1.62, respectively. The number of platforms counter, 'k', is incremented by 1 before the evaluation of products is performed on the platform evaluation stage.

4.2.2 Stage 2: Platform Evaluation Stage

The evaluation function used to evaluate the performance of the products leveraged from the platform is given by Equation 4.1

$$\Delta_j = \pm(N_j^1 \times \mathrm{E}fficiency_j^* - N_j^1 \times \mathrm{E}fficiency_j) \pm (N_j^2 \times \mathrm{M}ass_j^* - N_j^2 \times \mathrm{M}ass_j) \quad \ldots(4.1)$$

Here, N_j^1 and N_j^2 are the corresponding normalizing factors used in the equation N_j^1 and N_j^2 are the scaling factors that can be used to scale the corresponding benchmark performances to 1. The sign conventions introduced in Section 3.4 are used to assign positive or negative signs to the value of Δ_j obtained from the equation $\mathrm{E}\mathit{fficiency}_{\mathrm{j}}^{*}$ and $\mathrm{M}\mathit{ass}_{\mathrm{j}}^{*}$ are the normalized efficiency and mass of the benchmark motors, and $\mathrm{E}\mathit{fficiency}_{\mathrm{j}}$ and $\mathrm{M}\mathit{ass}_{\mathrm{j}}$ are the efficiency and mass of the motors leveraged using the platform. Table 4.6 shows the evaluation of products leveraged from platform 1. The limiting Δ_j value

Table 4.6: Evaluation of Products Leveraged from Platform 1

	Efficiency (Normalized)				Mass (Normalized)					
	Weights	Bench:	Family	Δ Efficiency	Weights	Bench:	Family	Δ Mass	Δ Total	Feasibility
Motor 1	1.2240	1.0000	0.9914	0.0086	3.0303	1.0000	1.0727	0.0727	0.0813	Y
Motor 2	1.3889	1.0000	1.0056	-0.0056	2.5641	1.0000	1.3051	0.3051	0.2996	N
Motor 3	1.4184	1.0000	0.9929	0.0071	2.4096	1.0000	1.3084	0.3084	0.3155	N
Motor 4	1.4286	1.0000	0.9814	0.0186	2.2222	1.0000	1.2622	0.2622	0.2808	N
Motor 5	1.5748	1.0000	1.0409	-0.0409	2.0000	1.0000	1.1840	0.1840	0.1431	Y
Motor 6	1.6949	1.0000	1.0610	-0.0610	1.7857	1.0000	1.1768	0.1768	0.1158	Y
Motor 7	1.7730	1.0000	1.0337	-0.0337	1.5873	1.0000	1.1111	0.1111	0.0774	Y
Motor 8	1.8248	1.0000	1.0018	-0.0018	1.4409	1.0000	1.0620	0.0620	0.0601	Y
Motor 9	1.9685	1.0000	0.9547	0.0453	1.3643	1.0000	1.0355	0.0355	0.0807	Y
Motor 10	2.2026	1.0000	0.9537	0.0463	1.2821	1.0000	0.9910	-0.0090	0.0373	Y

was decided as 0.2. Motors 1, 5, 6, 7, 8, 9, and 10 show performance losses within acceptable limits. The motors with performance loss due to commonality higher than 0.2 (Motors 2, 3, and 4) were separated out to be leveraged from the second platform.

4.2.3 Stage 3: Platform Cascading Stage

In this stage, only the nonconforming products from the previous platform evaluation are considered. The general platform cascading formulation presented in Section 3.4 is applied to motors 2, 3, and 4 as shown in Table 4.7. The objective function in this case consists of minimization of the positive deviation in mass from the target and the negative deviation of efficiency. The bounds on the design variables are the same as that of the single platform formulation. All the y_i parameters that were scale parameters in the previous platform are forced to a value of 0 to hold them as a scale parameter. All the platform parameters in the previous platform are initiated as platform parameters and held to the value obtained from the previous platform (Constraints 2 and 3).

There were five platform parameters in platform 1. The cascading formulation selects a platform parameter from these five platform parameters and converts it to a scale parameter so that motors with acceptable performance are derived. The remaining four platform parameters will have the same value as platform 1. This is achieved by using constraint 4. The constraint can only be satisfied if 4 out of the 5 y_i parameters have a value of 1 and the remaining parameter 0. This constraint also restricts the continuous y_i parameters to accept only binary values. All the remaining constraints are the same as the single platform formulation, except that they are only applied to the concerned motors 2, 3, and 4.

Table 4.7: Platform Cascading Formulation Applied to Universal Electric Motor Family

Minimize

$$\sum_{j=2,3,4} f(d^{+}_{Mass,j}) + \sum_{j=2,3,4} f(d^{-}_{Eff,j})$$

Subject to

(1) Bounds on the design variables

$100 \le x_{1j} \le 1500\ turns$

$1 \le x_{2j} \le 500\ turns$

$0.01 \le x_{3j} \le 1\ mm^2$

$0.01 \le x_{4j} \le 1\ mm^2$

$1 \le x_{5j} \le 1\ cm$

$0.5 \le x_{6j} \le 10\ mm$

$1 \le x_{7j} \le 6\ amps$

$0 \le x_{8j} \le 10\ cm \quad j = 2,3,4$

(2) Platform commonality decision variables

$y_1, y_5, y_7 = 0, \quad 0 \le y_2, y_3, y_4, y_6, y_8 \le 1$

(3) Platform commonality constraints (Cascading)

$(x_{2j} - 70)y_2 = 0 \quad j = 2,3,4$

$(x_{3j} - 0.28)y_3 = 0 \quad j = 2,3,4$

$(x_{4j} - 0.34)y_4 = 0 \quad j = 2,3,4$

$(x_{6j} - 5.91)y_5 = 0 \quad j = 2,3,4$

$(x_{8j} - 1.62)y_8 = 0 \quad j = 2,3,4$

(4) Cascading constraints

$\sum y_i^3 \ge 4 \quad i = 2, 3, 4, 6, 8$

$\sum y_i \le 4 \quad i = 2, 3, 4, 6, 8$

(5) Magnetic Intensity of each motor less than 5000A.turns/m

$$\frac{(2x_{2j}x_{7j})}{\pi(2x_{5j} + x_{6j})/2 + (2(x_{5j} - x_{6j} - 0.007) + 2*0.007))} \le 5000 \quad j = 2,3,4$$

(6) Feasible geometry

$x_{6j} < x_{5j} \qquad j = 2,3,4$

Table 4.7 Continued..

(7) Mass of motor (M) < 2.0 Kg

$$\pi\ dsteel\ x_{7j}(x_{5j}^2 - (x_{5j} - x_{6j})^2) +$$
$$\pi\ dsteel\ x_{7j}(x_{5j} - (x_{5j} - x_{6j} - l\ gap)^2)(2x_{7j} + 4(x_{5j} - x_{6j} - l\ gap))x_{1j}x_{3j} +$$
$$((2x_{7j} + 4(x_{5j} - x_{6j})(2x_{2j}x_{4j}))dcopper \quad \leq 2.0 \quad kg \quad j = 2,3,4$$

(8) Efficiency (E) > 0.15

$$\frac{1}{115x_{7j}}(113x_{7j} - (\frac{\mu_r.x_{1j}(2x_{7j} + 4(x_{5j} - x_{6j} - l\ gap))}{x_{3j}} + \frac{2\mu_r.x_{2j}(2x_{7j} + 4(x_{5j} - x_{6j}))}{x_{4j}})x_{7j}^2)$$
$$\geq 0.15 \quad j = 2,3,4$$

(9) Torque requirement for individual motors

$$\frac{(x_{1j}x_{2j}x_{7j}^2)}{\pi(\frac{0.000175}{x_{7j}\mu_0(x_{5j} - x_{6j} - l\ gap}) + \frac{1}{x_{7j}\mu_0\mu_r} + \frac{\pi(2x_{5j} + x_{6j})}{4\mu_0\mu_r x_{6j}x_{7j}}}$$
$$= \{\ 0.10,\ 0.125,\ 0.15,\ 0.20\}$$

(10) Deviation of actual efficiency from target efficiency (70%)

$$E_j / 0.7 + d_{Eff,j}^- - d_{Eff,j}^+ = 1.0 \quad j = 2,3,4$$

(11)Deviation of actual mass from target mass (0.5 kg)

$$M_j / 0.5 + d_{Mass,j}^- - d_{Mass,j}^+ = 1.0 \quad j = 2,3,4$$

Table 4.8: Optimum Design Variables and Performances Obtained from Cascading Formulation

	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6	Motor 7	Motor 8	Motor 9	Motor 10
X1	---	1018.00	1021.00	1500.00	---	---	---	---	---	---
X2	---	78	86	69	---	---	---	---	---	---
X3	---	0.28	0.28	0.28	---	---	---	---	---	---
X4	---	0.34	0.34	0.34	---	---	---	---	---	---
X5	---	2.15	2.29	2.00	---	---	---	---	---	---
X6	---	5.91	5.91	5.91	---	---	---	---	---	---
X7	---	3.27	3.32	3.64	---	---	---	---	---	---
X8	---	1.62	1.62	1.62	---	---	---	---	---	---
Efficiency	---	0.80	0.78	0.72	---	---	---	---	---	---
Mass	---	0.46	0.50	0.51	---	---	---	---	---	---

Table 4.8 shows the values of product parameters and product performances obtained from the platform cascading formulation. Parameter x_2 was converted from a platform parameter to a scale parameter. Significant improvement can be seen in efficiency and mass of the motors compared to the single platform design. The evaluation of the resulting motors using Equation 4.1 is shown in Table 4.9.

Table 4.9: Evaluation of Products Leveraged from Platform 2

	Efficiency (Normalized)				Mass (Normalized)					
	Weights	Bench:	Family	Δ Efficiency	Weights	Bench:	Family	Δ Mass	Δ Total	Feasibility
Motor 2	1.3889	1.0000	1.1111	-0.1111	2.5641	1.0000	1.1692	0.1692	0.0581	Y
Motor 3	1.4184	1.0000	1.1064	-0.1064	2.4096	1.0000	1.2048	0.2048	0.0984	Y
Motor 4	1.4286	1.0000	1.0229	-0.0229	2.2222	1.0000	1.1333	0.1333	0.1105	Y

The efficiencies of motors 2, 3, and 4 are higher than the benchmark motors. Since the efficiency is higher than target (positive valued in this case), a negative sign is assigned to the difference between the normalized benchmark and the efficiency of the motors. The masses of the motors are higher than the benchmark, which is undesirable; hence, a positive sign is assigned. The combined values, $\Delta_{Total,}$ for the three motors are 0.0581, 0.0941, and 0.1105, which are less than the allowed value of 0.2. Hence, further cascading is not necessary. Table 4.10 shows the combined parameter values and performance of the motors derived from platforms 1 and 2.

Table 4.10: Combined Results for Platforms 1 and 2

	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6	Motor 7	Motor 8	Motor 9	Motor 10
X1	944.96	1018.00	1021.00	1500.00	1094.58	1100.44	1101.90	1102.57	1100.56	1123.78
X2	70	78	86	69	70	70	70	70	70	70
X3	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
X4	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
X5	1.72	2.15	2.29	2.00	2.99	3.25	3.36	3.41	3.46	3.32
X6	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91
X7	3.04	3.27	3.32	3.64	3.97	4.33	4.71	5.08	5.37	6.00
X8	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
Efficiency	0.81	0.80	0.78	0.72	0.66	0.63	0.58	0.55	0.49	0.43
Mass	0.35	0.46	0.50	0.51	0.59	0.66	0.70	0.74	0.76	0.77

The platform leveraging and cascading strategy for the Universal Electrical Motor family is shown in Figure 4.3. Figure 4.3 relates the platform from which each product family member is leveraged and the configuration of each platform in terms of platform parameters and scale parameters.

The various optimization models were executed in an Intel Xeon 2 MHz processor CPU running on Windows XP operating system. The individual optimization models took 20-120 seconds to arrive at a solution. The single platform optimization

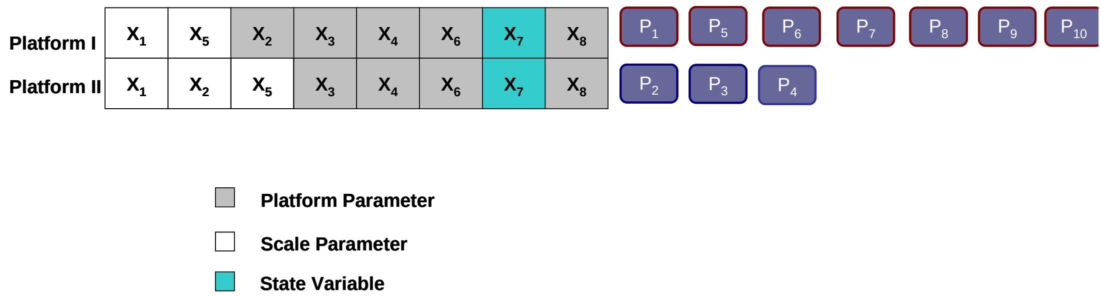
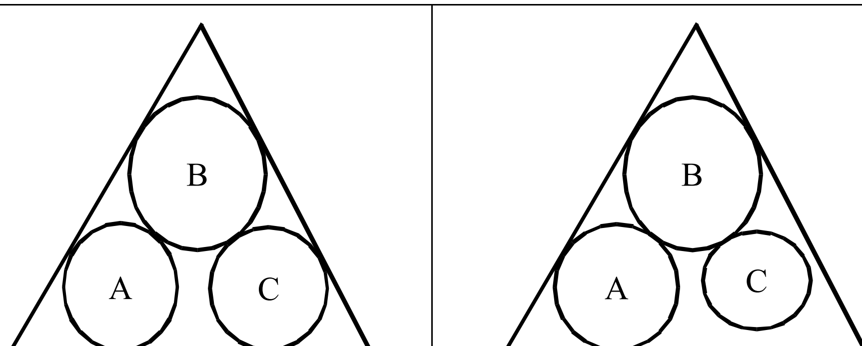


Figure 4.3: Platform Cascading Strategy for Universal Electric Motor Family

model required a run time of about 15-20 minutes depending on the starting point. The cascading formulation converged to a solution in 8-10 minutes.

4.3 Comparison of Results

If the objectives in a multi-objective problem are conflicting, no single point will optimize all the conflicting objectives simultaneously. The different solutions are a trade-off between the different objectives. Therefore, the concept of Pareto Optimal is



(a) Pareto Optimal (b) Not Pareto Optimal

Figure 4.4: Pareto Optimal (a) and Non-Pareto (b) Optimal Sets

used in a multi objective optimization problem. Figures 4.4 (a) and (b) can be used to understand the concept of Pareto optimality.

In these figures, the circles represent objectives that are satisfied best when the area of the circle is maximized. The constraints are that the circles may not overlap and must fit within the triangle. Figure 4.4 (a) shows a Pareto optimal solution; Figure 4.4 (b) is not a Pareto optimal solution, as the area of circle C can be increased without decreasing the area of the other two circles, thereby violating the constraints (Petrie and Webster, 1995). A vector of design variables X^* is said to be Edgeworth Pareto optimal if, for any other vector X, either the values of all objective functions remain the same or at least one of them worsens compared to its value at X^* (Hafta and Gurdal, 1991).

There can be more than one Pareto Optimal solution to a multi-objective problem, as in case of the Universal Electrical Motor problem. This makes it difficult to compare the solutions obtained by using different methods. One method might produce motors with higher efficiency but at the expense of higher mass and vice versa. Moreover, different methods resulted in different platform configurations and number of platforms, making it further difficult to compare the methods. The methods that were applied to the universal electric case study are PPCEM (Simpson *et al.*, 2001), VBPDM (Nayak *et al.*, 2002), PFPF (Messac *et al.*, 2002) and sensitivity based methods (Dai and Scott, 2005). PPCEM is a platform specified method. The other three methods treated the Universal Electrical Motor problem as a non-platform specified problem. The resulting motor family from these three methods compared to the PCM motors in the following sub-sections.

4.3.1 Comparison of VBPDM Motor Family with PCM Motors

As mentioned in Chapter II, VBPDM (Nayak *et al.*, 2002) is a non-platform specified method for product family design. VBPDM is a single platform method. Hence, the results from VBPDM are first compared to motors leveraged using the first platform. The authors reported that VBPDM resulted in a family with four platform parameters namely $x_2, x_3, x_4, and\ x_8$ as opposed to five platform parameters in PCM. Table 4.11 shows the comparisons. VBPDM motors show higher average efficiency of 12.12 % over PCM motors for a very slight increase in average mass (1.238 %).

Table 4.11: Comparison of VBPDM Motor Performances with PCM Motors (Single Platform)

	Efficiency			Mass		
	PCM	VBPDM	%Diff	PCM	VBPDM	%Diff
Motor 1	81.00	89	9.877	0.35	0.5	41.243
Motor 2	72.40	82	13.260	0.51	0.5	-1.768
Motor 3	70.00	79	12.857	0.54	0.5	-7.919
Motor 4	68.70	76	10.626	0.57	0.5	-11.972
Motor 5	66.10	71	7.413	0.59	0.57	-3.716
Motor 6	62.60	67	7.029	0.66	0.63	-4.401
Motor 7	58.30	64	9.777	0.70	0.67	-4.286
Motor 8	54.90	60	9.290	0.74	0.72	-2.307
Motor 9	48.50	58	19.588	0.76	0.76	0.132
Motor 10	43.30	53	22.402	0.77	0.83	7.374
		Avg	12.212		Avg	1.238

Table 4.12 shows the comparison between VBPDM motors leveraged using both platforms in PCM. As shown in Section 4.2, the second platform consists of 4 platform parameters. It can be seen that the difference in average performance is less than in the case of the two-platform PCM family with VBPDM motors having 4.077 % higher average mass. The results indicate that average performances improve as a parameter is relaxed to a scale parameter by cascading. They also help to prove the fact that increased commonality leads to increased loss of performance.

Table 4.12: Comparison of VBPDM Motor Performances with PCM Motors (Multi-Platform)

	Efficiency			Mass		
	PCM	VBPDM	%Diff	PCM	VBPDM	%Diff
Motor 1	81.00	89	9.877	0.35	0.5	41.243
Motor 2	80.00	82	2.500	0.46	0.5	8.696
Motor 3	78.00	79	1.282	0.50	0.5	0.000
Motor 4	72.00	76	5.556	0.51	0.5	-1.961
Motor 5	66.10	71	7.413	0.59	0.57	-3.716
Motor 6	62.60	67	7.029	0.66	0.63	-4.401
Motor 7	58.30	64	9.777	0.70	0.67	-4.286
Motor 8	54.90	60	9.290	0.74	0.72	-2.307
Motor 9	48.50	58	19.588	0.76	0.76	0.132
Motor 10	43.30	53	22.402	0.77	0.83	7.374
		Avg	9.471		Avg	4.077

4.3.2 Comparison of PFPF motor family with PCM motors:

PFPF (Messac *et al.*, 2002) method is also a single platform method for product family design. PFPF motors are comprised of six platform parameters x_1, x_2, x_3, x_4, x_5 *and* x_8.

Table 4.13: Comparison of PFPF Motor Performances with PCM Motors (Single Platform)

	Efficiency			Mass		
	PCM	PFPF	%Diff	PCM	PFPF	%Diff
Motor 1	81.00	76	-6.173	0.35	0.395	11.582
Motor 2	72.40	72.1	-0.414	0.51	0.513	0.786
Motor 3	70.00	70.3	0.429	0.54	0.562	3.499
Motor 4	68.70	68.5	-0.291	0.57	0.606	6.690
Motor 5	66.10	65.1	-1.513	0.59	0.678	14.527
Motor 6	62.60	61.8	-1.278	0.66	0.734	11.381
Motor 7	58.30	58.8	0.858	0.70	0.775	10.714
Motor 8	54.90	55.9	1.821	0.74	0.803	8.955
Motor 9	48.50	53.1	9.485	0.76	0.821	8.169
Motor 10	43.30	47.9	10.624	0.77	0.83	7.374
		Avg	1.355		Avg	8.368

Even though the PFPF motors show an average increase in efficiency of 1.355%, their average mass exceeds that of PCM motors by 8.368%. Higher loss of performance for PFPF motors may be attributed to higher commonality.

4.3.3 Comparison of Sensitivity Based Method with PFPF motors

Dai and Scott (2005) presented a multi-platform product family design method using sensitivity analysis and cluster analysis. Their method, when applied to the Universal Electric Motor case study, resulted in a product family comprising of three platforms. Parameter x_1 had three different values across the platform, x_2 two, and $x_4, and\ x_6$ had one value across the family. The method resulted in slightly higher average efficiency and lower average mass at the expense of a third platform (Table 4.14).

Table 4.14: Comparison of Sensitivity Motor Performances with PCM Motors (Multi-Platform)

	Efficiency			Mass		
	PCM	Sensitivity	%Diff	PCM	Sensitivity	%Diff
Motor 1	81.00	86.2	6.420	0.35	0.347	-1.977
Motor 2	80.00	71.3	-10.875	0.46	0.338	-26.522
Motor 3	78.00	70	-10.256	0.50	0.425	-15.000
Motor 4	72.00	67.1	-6.806	0.51	0.478	-6.275
Motor 5	66.10	66	-0.151	0.59	0.534	-9.797
Motor 6	62.60	64.8	3.514	0.66	0.637	-3.338
Motor 7	58.30	62.6	7.376	0.70	0.717	2.429
Motor 8	54.90	63	14.754	0.74	0.826	12.076
Motor 9	48.50	50.3	3.711	0.76	0.879	15.810
Motor 10	43.30	56	29.330	0.77	0.988	27.814
		Avg	3.702		**Avg**	-0.478

Due to the difficulties in comparing results from a multi-objective design problem explained at the beginning of this section, it is impossible to quantify exactly the merits and demerits of each product family method. These comparisons help to establish the fact that improved commonality comes at the expense of product performance. Moreover it can be seen that the multi-platform approach helps to improve the performance of the family when compared to the single platform approach. None of the methods reported so far have modeled the additional cost burden of having multiple platforms. This is a potential area for future research activities.

4.4 Summary

In this chapter, applicability of PCM was further demonstrated by its application to the Universal Electrical Motor case study. The motors were leveraged from two cascading platforms with acceptable loss of performance due to commonality. The first platform consisted of five platform parameters, whereas the second platform consisted of four platform parameters. The motors showed significant performance improvement when multi-platform leveraging strategy was employed. The platform cascading ensured commonality between different platforms. The methods were compared to existing methods that were applied to the Universal Electrical Motor example problem.

CHAPTER 5

CLOSURE

This is the concluding chapter of this dissertation. It is organized into four sections. Section 5.1 discusses the approach and how the research questions were answered. Section 5.2 presents the contributions made through this dissertation. Section 5.3 discusses some of the limitations of PCM. The last section, 5.4, identifies some of the future research areas.

5.1 Discussion

In this dissertation, the Platform Cascading Method (PCM) for scale-based product family design was presented. The method is capable of designing scale-based product families based on multiple platforms. The research questions that are addressed through this dissertation were presented in Chapter 1 along with the sub-questions.

In Chapter 2, background information relevant to product family concepts and product family optimization was presented. Existing methods for scale-based product family design were presented in Chapter 2, followed by a matrix of comparison between the existing methods. The matrix helped to differentiate the existing work based on the approach, modeling assumptions, number of supported platforms, and solution technique employed. The matrix also helped to establish the uniqueness of the work presented in this dissertation.

The general steps in PCM were presented in Chapter 3. The method starts with designing the entire family of products based on a single platform. Then evaluation of the resulting products is performed to identify the products whose losses of performance due to commonality are higher than the acceptable limits. Those identified products are then considered for leveraging from a new platform formed by cascading the previous platform. This stage is called the cascading stage of the design process.

Cascading involves selecting one of the platform parameters from the previous platform and relaxing it to a scale parameter. The resulting products from cascading are evaluated and the platform is cascaded further if necessary until all products with acceptable performance are leveraged.

Product family design is a trade-off between commonality (platform) and individual product performance. Different optimization formulations were developed to perform these trade-offs at different stages of PCM. The nature and challenges of the scale-based product platform design optimization problem were presented in Chapter 1. The optimization formulations presented in PCM are capable of tackling these difficulties and arriving at an optimum product family design quickly. The formulations are generic and may be implemented in several optimization algorithms, although gradient-based methods were chosen for implementation. The general steps in the design process applicable to all scale-based product families were presented in Chapter 3. These general steps were then illustrated using an axial pump family design problem.

In Chapter 4, PCM was used to design a family of ten Universal Electric Motors. A Universal Electric Motor design problem is considered a de facto product family design problem and has been implemented in several existing works. The Universal Electric Motor family obtained from PCM is compared to those obtained using other methods. Even though exact quantification of the effectiveness of different methods was not possible, the comparisons help to establish the overall effectiveness of PCM. The PCM method is unique in the approach to modeling the product family design problem and also in establishing the relation between different platforms used to leverage the family. The following are the research questions posed in Chapter 1 and information on how PCM provides answer to these questions:

RQ1) *How do we represent a family of products supported by a single platform using a mathematical programming model and identify a solution technique, so that tradeoffs between commonality and performance can be performed to support product family design?*

PCM takes a holistic view of the entire product family design process. The mathematical model developed for single platform design is capable of representing both the product platform and the product variants. During the single platform stage of the design process, both the platform and the product variants are simultaneously optimized. Trade-off is performed between the number of platform parameters and the loss of performance due to commonality to arrive at the optimum platform and the optimum product instances. PCM converts binary platform commonality parameters to continuous parameters to enable the formulation to be implemented in a gradient-based optimization method. The model is constrained mathematically to accept only binary values in the end for the platform commonality parameters. The formulation developed is easy to implement in gradient-based optimization methods and can arrive at optimum solutions quickly.

Research question RQ2 introduced in Chapter 2 is as follows:

RQ2) *How do we extend the mathematical model to design product families supported by multiple platforms?*

In PCM, a cascading approach is used to leverage the family when multiple platforms are required. During cascading, one of the platform parameters is relaxed to a scale parameter so that products with lesser loss of performance can be leveraged. This reduces the number of platform parameters from the previous platform, which in turn can lead to products with better performance.

The sub-questions that related to research question 2 are:

RQ2.1) How do we extend the single platform representation as a sub- problem for deciding configuration of multiple the platforms?

In PCM, the modeling approach in case of cascading formulation is similar to that of single platform formulation. The platform, product instances, and platform commonality are modeled in the cascading formulation, as in case of single platform formulation. Both formulations initially convert the MINLP to a continuous problem and then constrain the solution to discrete spaces. In the case of the cascading formulation, cascading constraints are simultaneously used to select the platform parameters and also to constrain the model to accept only binary values for commonality parameters.

RQ2.2) How do we extend the mathematical model to evaluate the optimum number of platforms?

In PCM, the number of platforms required to support the platform is not modeled as part of the different formulations. Instead, the initial platform is cascaded until all the products with acceptable loss of performance are leveraged. The number of platforms required to support the family depends on the threshold value of the acceptable loss of performance due to commonality and the path chosen by the designer after the evaluation of products [Case 2 (a) or 2 (b)].

RQ2.3) How do we maintain a relationship between the different platforms so that commonality between the different platforms can be established?

PCM uses the single platform and cascades it to generate the subsequent platforms. This approach maintains a relation between the product platforms which can lead to commonality within the different platforms.

RQ2.4) What are the ideal scenarios that determine when a multi-platform approach should be used?

PCM uses an evaluation function to determine the loss of performance of the product family members due to commonality. If the loss of performance due to commonality for any of the products in the family is greater than a user specified value, a multi-platform approach is used.

5.2 Contributions

Some of the key contributions made towards the area of the scale-based product family design through this dissertation are:

(1) A mathematical programming model that represents a scale-based product family in terms of decision variables, constraints and objectives was developed. The mathematical programming model is capable of capturing the commonality of the platform components/parameters and the parametric description of the product instances.

(2) The model is capable in arriving at a suitable platform and the derived product instances simultaneously in a single stage in case of single platform design. The formulation explores different possible platform configurations and identifies the best platform configuration for given set of requirements. The designer has the flexibility of

specifying the platform parameters, in which case the formulation returns the values of platform parameters and scale parameters.

(3) The difficulties encountered in adopting a solution method due to the inherent nature of the model is tackled by converting the problem to a continuous design variable problem and then constraining it mathematically to produce discrete results. This enables the model to be implemented in gradient-based solution algorithms.

(4) To evaluate the performance of the product family members and also to determine whether a multi-platform approach is necessary, a product family evaluation function is introduced. This function is capable of comparing the product family members to that of benchmark products.

(5) PCM is capable of moving from a single platform design to a multi-platform design when necessary by cascading the initial platform. Cascading maintains commonality between subsequent platforms, thereby increasing cost savings. The formulation selects the platform parameter that reduces most the loss of performance due to commonality upon conversion to a scale parameter.

5.3 Limitations of PCM

(1) As explained in Chapter 1, there may be more than one optimum solution to the problem. Gradient-based methods have a tendency of converging to the nearest local optimum. Moreover, results obtained from the solution vary while employing different weights for different components in the objective functions. Both starting the problem at different design points and employing different weights help to arrive at the global optimum solution. In this dissertation, the design points obtained from individual optimization are used as the starting point for the single platform formulation. This helps

the optimization model in converging quickly, as these points are feasible points when considering all constraints except the commonality constraints.

(2) The optimization formulations introduced in PCM arrive at a suitable platform, assuming equal priority to all the candidate parameters. In reality, certain parameters would be preferred over the others due to the manufacturing operations involved. Even though PCM does not address this issue, providing different weights in the objective function corresponding to the priorities can help to model the preferences.

(3) In cases when no suitable platform can be arrived at during the single platform stage, PCM does not provide options to the designer to group the product variants into sub-groups and generate suitable platforms for the sub-groups. It is assumed that a single platform can be arrived for the given set of products.

5.4 Future Work

Some of the future areas for extending this work are:

(1) In PCM, after each platform-leveraging step, the evaluation of the resulting products is done to select the products that perform within acceptable limits. A natural extension to the formulation would be including the evaluation and selection process as part of the optimization formulation. The same logic for selecting the platform parameters can be used to select the products that can be leveraged from the current platform. Table 5.1 shows the modifications that were made to the single platform formulation of PCM to accomplish this. A new binary decision variable, 'z_{kj}', was introduced in the form-ulation

$$z_{kj} = \begin{cases} 1 & \text{when product 'j' is scaled from platform 'k'} \\ 0 & \text{otherwise} \end{cases}$$

Parameter $z_{kj} = 1$, when the family member 'j' is leveraged from platform 'k', and '0' otherwise. The objective function has to be modified to maximize the number of products that are leveraged from the current platform. This is accomplished by the inclusion of the term, '$\sum_{j=1}^{m} z_{kj}$', to the objective function.

$$\sum_{l=1}^{p}\sum_{j=1}^{m} f(d_{lj}^{-}, d_{lj}^{+}) - \sum_{i=1}^{n}{}_{i}\, y_i - \sum_{j=1}^{m} z_{kj}$$

In PCM, the products that will be leveraged from a particular platform were known prior. The commonality of platform parameters were modeled for all products considered for leveraging, using the commonality constraint shown in Table 3.9

$$(x_{ij} - x_{ij+1})y_i = 0 \quad \forall \quad i \in I \quad and \quad j \in J, \quad j \neq m \qquad \text{... (5.1)}$$

But in the present case, commonality has to be modeled for all the possible combinations of products that can be leveraged from the current platform 'k'. Equation 5.1 can be extended as

$$(x_{ij} - x_{ij+1})y_{ki} * z_{kj} = 0 \quad \forall \quad i \in I \quad and \quad \forall\ j \in J \qquad \text{... (5.2)}$$

This will lead to '$\frac{j(j-1)}{2}$', commonality constraints corresponding to each product parameter. Here, 'y_{ki}' is equal to '1' when y_i is a platform parameter in platform 'k' and 0 otherwise. To enable execution in a gradient-based optimizer, the commonality parameters and platform inclusion parameters are treated initially as continuous parameters between 0 and 1 and then constrained to accept only binary values

$$z_{kj}^{\ 2} - z_{kj} = 0 \quad 0 \leq z_{kj} \leq 1 \qquad \text{... (5.3a)}$$

$$y_{ki}^{\ 2} - y_{ki} = 0 \quad 0 \leq y_{ki} \leq 1 \qquad \text{... (5.3b)}$$

Table 5.1: Modifications to Single Platform Formulation to Included Product Selection

Objective

$$\sum_{l=1}^{p}\sum_{j=1}^{m} f(d_{lj}^{+/-}) - \sum_{i=1}^{n}{}_{i}\, y_i - \sum_{j=1}^{m} z_{kj}$$

Subject to

(1) Bounds on the design variable

$$x_{ij}^{lower} \le x_{ij} \le x_{ij}^{upper}, \quad \forall \quad i \in I \text{ and } \forall\ j \in J$$

$$0 \le z_{kj} \le 1$$

$$0 \le y_{ki} \le 1$$

(2) Commonality constraints

$$(x_{ij} - x_{ij+1})y_{ki} * z_{kj} = 0 \quad \forall i \in \text{ and } \forall\ j \in J$$

(3) 'Integerizing' constraints

$$y_{ki}{}^{2} - y_{ki} = 0 \quad \forall i \in I \text{ and } \forall\ j \in J$$

$$z_{kj}{}^{2} - z_{kj} = 0 \quad \forall i \in I \text{ and } \forall\ j \in J$$

(4) Product constraints

$$z_{kj} * g_t(x) = 0 \qquad t = 1, 2, \ldots s$$

(5) Limits on loss of performance due to commonality

$$0 < d_{lj}^{+/-} < \lambda$$

The individual constraints relating to each product variant need to activated or deactivated depending on the value of 'z_{kj}' if the product is considered for leveraging from the current platform or not. Hence, all the product constraints are modified as shown in constraint 4.

The deviation variables capturing the deviation of product performances are modeled as in single platform formulation of PCM. To select only the products with loss of performance within accepted deviation values are constrained to be within specific limits as shown in constraint 5.

This formulation was extended for the case of a Universal Electric Motor problem and implemented in VDOC. The formulation failed to execute due to very drastic increase in the number of equality constraints required to model the platform commonality. When formulation was applied to the UEM problem, there are 438 equality constraints and only 137 design variables. The high number of equality constraints than design variables over-constraints the model and does not allow the gradient-based optimizer to move in the design space. The equality constraints are always active, unlike inequalities which are active only at the optimum. One way to tackle this problem is to convert the equality constraints to inequalities and arrive at the results over several steps consecutively by tightening the limits. The limitation is that it requires several runs and might have very slow convergence towards to the solution. Alternate ways to model the problem need to be investigated.

Another probable research direction is to apply a heuristic solution method instead of gradient-based methods. Heuristic methods selects a set of random design points, evaluates the quality of the solutions obtained, selects the best design point, and moves to next set of points. This approach requires investigating several heuristic methods available and selecting a method capable of solving problems of this magnitude.

(2) PCM utilizes a platform evaluation function (Equation 3.12) to select the performance of the resulting products whose limiting values, 'η', is provided by the designer. As the value of 'η' changes, so does the platform leveraging strategy for the

given family of products. As the value of 'η' increases, products with higher loss of performance are deemed acceptable for leveraging from the current platform and vice versa. The method relies on the designers ability to provide a reasonable value of 'η'. A systematic method needs to be developed to arrive at a reasonable value of 'η' for the case of different product family design problems.

(3) The different methods currently available for scale-based product family design were presented in Chapter 2. The results obtained from PCM were compared to that obtained from other methods implementing the UEM design example. Due to the inherent nature of the multi-objective problems, comparison of these methods is not exact. Moreover, product platforms that resulted from the methods were different. Some methods assumed commonality throughout the products in the family or for groups of products, while others assumed commonality only between product pairs. This makes it difficult to quantify and compare the performance of the resulting product family and platform commonality for different cases. Different indexes have been developed to measure commonality of product families (Thevenot and Simpson, 2005). These indexes need to be extended to include resulting product performances and multi-platform leveraging so that effectiveness of the different methods available can be quantified.

(4) In PCM, the manufacturing costs related to having multiple platforms were not considered. Instead, PCM was based on the assumption that maintaining commonalty between the different platforms can lead to increased savings in cost. The cost burden of having multiple platforms needs to be investigated and modeled to arrive at the number of platforms.

REFERENCES

Allada, V., and Jiang, L., 2002, New modules launch planning for evolving modular product families, ASME Design Engineering Technical Conferences - Design for Manufacturing Conference, Montreal, Quebec, Canada, ASME, Paper No. DETC2002/DFM-34190.

Allada, V., Choudhury A. K., Pakala P. K., Valliyappan,S., Scott, M. J, Simpson, T.W., 2006. Product Platform Problem Taxonomy: Classification And Identification Of Benchmark Problems, Proceedings of DETC 2006 ASME Design Engineering Technical Conferences Philadelphia, PA, USA, DETC2006/DAC-99569

Beale, E. M. L., 1977. *Integer Programming, in the State of the Art in Numerical Analysis* (D. Jacobs, ed.) Academic Press: London, 409-448.

Bhandare, S., and Allada, V., 2006, Scalable Product Family Design: Case Study of Axial Piston Pumps, International Journal of Production Research (#TPRS-2006-IJPR-0351, Under Review).

Blackenfelt, M., 2000, Profit maximization while considering uncertainty by balancing commonality and variety using robust design - The redesign of a family of lift tables, ASME Design Engineering Technical Conferences - Design for Manufacturing, Baltimore, MD, ASME, Paper No. DETC2000/DFM-14013.

Blackenfelt, M., and Sellgren, U., 2000, Design of robust interfaces in modular products, ASME Design Engineering Technical Conferences - Design Automation Conference, Baltimore, MD, ASME, Paper No. DETC00/DAC-14486.

Cetin, O.L., and Saitou, K., 2004, Decomposition-based assembly synthesis for structural modularity, *ASME Journal of Mechanical Design*, 126(2): 234-243.

Chang, T.-S., and Ward, A.C., 1995, Design-in-modularity with conceptual robustness, Advances in Design Automation, Azarm, S., et al., (Eds.), Boston, MA, ASME, 82(1): 493-500.

Chapman, S.J., 1991, *Electric Machinery Fundamentals*, McGraw-Hill, New York, NY.

Chen, W., R. D., 1994. Modularity and the independence of functional requirements in designing complex systems". *Concurrent Product Design*, 74, pp. 31-38.

Dai, Z., and Scott, M. J., 2005. Product platform design through sensitivity analysis and cluster analysis". Journal of Intelligent Manufacturing, Special ssue on Product Family Design and Development.

Duran, M.A. and Grossmann, I.E., 1986. An Outer Approximation Algorithm for a Class of Mixed Integer Nonlinear Programs," *Mathematical Programming* 36, 307-339.

Farrell, R., and Simpson, T.W., 2003, Product platform design to improve commonality in custom products, *Journal of Intelligent Manufacturing*, 14(6): 541-556.

Feitzinger, E., and Lee, H.L., 1997, Mass customization at Hewlett-Packard: the power of postponement, *Harvard Business Review*, 75(1): 116-121.

Fellini, R., Kokkolaras, M., Papalambros, P., and Perez-Duarte, A., 2002b, Platform selection under performance loss constraints in optimal design of product families, ASME Design Engineering Technical Conferences - Design Automation Conference, Fadel, G. (Ed.), Montreal, Quebec, Canada, ASME, Paper No. DETC2002/DAC-34099.

Fellini, R.; Kokkolaras M.;Papalambros, P., 2006, Commonality decisions in product family design". In *Product Platform and Product Family Design*, Simpson, T.; Siddique, ed. Springer, Ch. 9, pp. 157-185

Fujita, K., 2002, Product variety optimization under modular architecture, *Computer-Aided Design*, 34(12): 953-965.

Fujita, K., Akagi, S., and Hirokawa, N., 1993, Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm, Proceedings of the 1993 ASME Design Automation Conference, DE-65(1), pp. 477-484.

Fujita, K., Akagi, S., Yoneda, T., and Ishikawa, M., 1998, Simultaneous optimization of product family sharing system structure and configuration, ASME Design Engineering Technical Conferences - Design for Manufacturing, Atlanta, GA, ASME, Paper No. DETC98/DFM-5722.

Fujita, K., Akagi, S., Yoshida, K., and Hirokawa, N., 1996, Genetic algorithm based optimal planning method of energy plant configurations, Proceedings of the 1996 ASME Design Engineering Technical Conferences, Paper No. 96-DETC/DAC-1464.

Fujita, K., and Ishii, K., 1997, Task structuring toward computational approaches to product variety design, Proceedings of the 1997 ASME Design Engineering Technical Conferences, Paper No. 97DETC/DAC-3766.

Fujita, K., and Yoshida, H., 2001, Product variety optimization: simultaneous optimization of module combination and module attributes, ASME Design Engineering Technical Conferences - Design Automation Conference, Diaz, A. (Ed.), Pittsburgh, PA, ASME, Paper No. DETC2001/DAC-21058.

Fujita, K., and Yoshida, H., 2004, Product variety optimization simultaneously designing module combination and module attributes, *Concurrent Engineering: Research and Applications*, 12(2): 105-118.

Fujita, K., and Yoshioka, S., 2003, Optimal design methodology of common components for a class of products: its foundations and promise, ASME Design Engineering Technical Conferences - Design Automation Conference, Shimada, K. (Ed.), Chicago, IL, ASME, Paper No. DETC2003/DAC-48718.

Fujita, K., Sakaguchi, H., and Akagi, S., 1999, Product variety deployment and its optimization under modular architecture and module commonalization, ASME Design Engineering Technical Conferences - Design for Manufacturing, Las Vegas, NV, ASME, Paper No. DETC99/DFM-8923.

Fujita, K., Takagi, H., and Nakayama, T., 2003, Assessment method of value distribution for product family deployment, Proceedings of 14th International Conference on Engineering Design (ICED 03), Paper No. 1484.

Geoffrion, A.M., 1972. Generalized Benders Decomposition, *Journal of Optimization Theory and Applications* 10(4) 237-260.

Glover F., and Laguna M., (1993), Tabu Search, in *Modern Heuristic Techniques for Combinatorial Problems*, C.R. Reeves, editor, John Wiley & Sons, Inc,

Goldberg, D., 1999. Genetic Algorithms in Search, Optimization, and Machine Learning. Kluwer Academic Publishers, Addison-Wessley.

Gonzalez-Zugasti, J.P., and Otto, K.N., 2000, Modular Platform-Based Product Family Design, ASME Design Engineering Technical Conferences - Design Automation Conference, Renaud, J.E. (Ed.), Baltimore, MD, ASME, Paper No. DETC-2000/DAC-14238.

Grossmann, I. ,1990. "Mixed-integer nonlinear programming techniques for the synthesis of engineering systems." Research in Engineering Design 1.

Hafta, T., and Gurdal, Z., 1991. Elements of Structural Optimization. Kluwer Academic Publishers, New York, NY.

Hafta, T., and Gurdal, Z., 1991. Elements of Structural Optimization. Kluwer Academic Publishers, New York, NY.

Hernandez, G., Simpson, T.W., Allen, J.K., Bascaran, E., Avila, L.F., and Salinas, F., 2001, Robust design of families of products with production modeling and evaluation, *ASME Journal of Mechanical Design*, 123(2): 183-190.

Horst, R., P. Pardalos, et al. (2001). Introduction to Global optimization, Kluwer Academic Publishers.

Jiang, L., and Allada, V., 2001, Design for robustness of modular product families for current and future markets, ASME Design Engineering Technical Conferences - Design for Manufacturing Conference, Pittsburgh, PA, ASME, Paper No. DETC2001/DFM-21177.

Kirkpatrick, S., Gelatt Jr., C. D. and Vecchi, M. P., 1983, “Optimization by Simulated Annealing,” Science, Vol. 220, No. 4598, pp. 671-680.

Kobe, G., 1997. Platforms - gm's seven platform global strategy". *Automotive Industries*, 177(10), October 1997, p. 50.

Kokkolaras, M., Louca, L.S., Delagrammatikas, G.J., Michelena, N.F., Filipi, Z.S., Papalambros, P.Y., Stein, J.L., and Assanis, D.N., 2004, Simulation-based optimal design of heavy trucks by model-based decomposition: An extensive analytical target cascading case study, *International Journal of Heavy Vehicle Systems*, 11(3-4): 402-432.

Lehnerd, A.P., 1987, Revitalizing the manufacture and design of mature global products, in *Technology and Global Industry: Companies and Nations in the World Economy*, Guile, B.R., Brooks, H. (Eds.), National Academy Press, Washington, D.C., pp. 49-64.

Li, H., and Azarm, S., 2002, An approach for product line design selection under uncertainty and competition, *ASME Journal of Mechanical Design*, 124(3): 385-392.

MacDuffie, J.P., Sethuraman, K., and Fisher, M.L., 1996, Product Variety and manufacturing performance: evidence from the international automotive assembly plant study, *Management Science*, 42(3): 350-369.

Martin, M.V., and Ishii, K., 2002, Design for variety: Developing standardized and modularized product platform architectures, *Research in Engineering Design,* 13(4): 213-235.

McDermott, C., and Stock, G., 1994. The use of common parts and designs in high-tech industries: a strategic approach". *Production and Inventory Management Journal,* 35(3), pp. 65-68.

McGrath, M. E., 1995. Product Strategy for High-Technology Companies. Irwin Professional Publishing, New York.

Messac, A., 1996, Physical programming: Effective optimization for computational design, *AIAA Journal,* 34(1): 149-158.

Messac, A., Martinez, M.P., and Simpson, T.W., 2002a, Effective product family design using physical programming, *Engineering Optimization*, 34(3): 245-261.

Messac, A., Martinez, M.P., and Simpson, T.W., 2002b, A penalty function for product family design using physical programming, ASME Journal of Mechanical Design, 124(2): 164-172.

Meyer, M., and Lehnerd, A., 1997. *The Power of Product Platforms: Building Value and Cost Leadership*. Free Press, New York, NY.

Meyer, M.H., 1997, Revitalize your product lines through continuous platform renewal, *Research Technology Management*, 40(2): 17-28.

N.V. Sahinidis and I.E. Grossmann, 1991. Convergence properties of generalized Benders decomposition, Computers and Chemical Engineering, vol. 15, pp. 481-491

N.V. Sahinidis, 1996 "BARON: A general purpose global optimization software package," Journal of Global Optimization, vol. 8, pp. 201-205.

Naughton, K., Thornton, E., Kerwin, K., and Dawley, H., 1997, Can Honda build a world car?, *Business Week*, September, 8: 100-107

Nayak, R. U., W. Chen, et al., 2002. A Variation-Based Method for Product Family Design, *Engineering Optimization* 34(1): 65-81.

Nayak, R., Chen, W., and Simpson, T., 2002. A variation-based method for product family design". *Engineering Optimization*, 34(1), pp. 65-81.

Nelson, S.A., II, Parkinson, M.B., and Papalambros, P.Y., 2001, Multicriteria optimization in product platform design, *ASME Journal of Mechanical Design*, 123(2): 199-204.

Nelson, S.A., II, Parkinson, M.B., and Papalambros, P.Y., 2001, Multicriteria optimization in product platform design, *ASME Journal of Mechanical Design*, 123(2): 199-20

Ortega, R., Kalyan-Seshu, U., and Bras, B., 1999, A decision support model for the life-cycle design of a family of oil filters, ASME Design Engineering Technical Conferences

- Design Automation Conference, Las Vegas, NV, ASME, Paper No. DETC99/DAC-8612.

Petrie C J.,Webster T A., C. R. M., 1995. Using pareto optimality to coordinate distributed agents". AIEDAM special issue on conflict management, 9, pp. 269-281.

Petrie C J.,Webster T A., C. R. M., 1995. Using pareto optimality to coordinate distributed agents". AIEDAM special issue on con°ict management, 9, pp. 269-281.

Pettersson, F., Westerlund, T., 1995 , An extended cutting plane method for solving convex MINLP problems, *Computers and Chemical Engineering*, vol. 19(Suppl.), pp. S131-S136, 1995

Pine, B.J., 1993a, *Mass Customization: The New Frontier in Business Competition*, Harvard Business School Press, Boston, MA.

Pine, J.B., 1993b, Standard modules allow mass customization at Bally Engineering Structures, *Planning Review,* 21(4): 20-22.

Quesada.I, and I.E. Grossmann, 1992. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems, *Computers and Chemical Engineering,* vol. 16, pp. 937-947

Rai, R., and Allada, V., 2003, Modular product family design: Agent-based pareto-optimization and quality loss function-based post-optimal analysis, *International Journal of Production Research,* 41(17): 4075-4098.

Rothwell, R., and Gardiner, P., 1990, Robustness and product design families, in *Design Management: A Handbook of Issues and Methods*, Oakley, M. (Ed.), Basil Blackwell Inc., Cambridge, MA, pp. 279-292.

Sabbagh, K., 1996, *Twenty First Century Jet — The Making and Marketing of The Boeing 777*, Scribner, New York, NY.

Sanderson, S.W., and Uzumeri, M., 1997, *Managing Product Families*, Irwin, Chicago, IL.

Scott, M. J, Arenillas, J. C. G, Valliyappan,S., Simpson, T.W., Allada, V., 2006. Towards A Suite Of Problems For Comparison Of Product Platform Design Methods: A Proposed Classification, Proceedings of DETC 2006 ASME Design Engineering Technical Conferences Philadelphia, PA, USA, DETC2006-99289

Simpson, T. W., 2004. Product Platform Design and Customization: Status and Promise, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 18(1), 3-20.

Simpson, T.W., 1998, A Concept Exploration Method for Product Family Design, *Ph.D. Dissertation*, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Simpson, T.W., 1998, A Concept Exploration Method for Product Family Design, *Ph.D. Dissertation*, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Simpson, T.W., 2004, Product platform design and customization: Status and promise, *AIEDAM,* 18(1): 3-20.

Simpson, T.W., and D'Souza, B., 2004, Assessing variable levels of platform commonality within a product family using a multi objective genetic algorithm, *Concurrent Engineering: Research and Applications,* 12(2): 119-130.

Stadzisz, P., and Henrioud, J., 1995. Integrated design of product families and assembly systems". In Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2, IEEE, pp. 1290-1295.

Stone, R.B., Wood, K.L., and Crawford, R.H., 2000a, A heuristic method to identify modules from a functional description of a product, *Design Studies,* 21(1): 5-31.

Tawarmalani, M. and N. Sahinidis , 2002. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming:Theory, Algorithms, Software and Applications*, Kluwer Academic Publishers.

Thevenot, H.J., and Simpson, T.W., 2006, Commonality indices for product family design: A detailed comparison, *Journal of Engineering Design*, in press

Ulrich, K., and Tung, K., 1991. Fundamentals of product modularity. In Proceedings of ASME Winter Annual Meeting Conference, Vol. DE Vol. 39, pp. 73-80.

Ulrich, K.T., and Eppinger, S.D., 2004, *Product Design and Development*, McGraw-Hill/Irwin, New York.

Veinott C.C., Martin L, E. (1986) Fractional and sub fractional horsepower electric motors, McGraw-Hill, New York

Wheelwright, S., and Clark, K., 1992. Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency and Quality. The Free Press, New York, NY.

Wilhelm, B., 1997. Platform and modular concepts at Volkswagen - their effect on the assembly process. In Transforming Automobile Assembly: Experience in Automation and Work Organization, K. Shimokawa, U. Jrgens, and T. Fujimoto, eds. Springer, New York, NY, pp. 146-156.

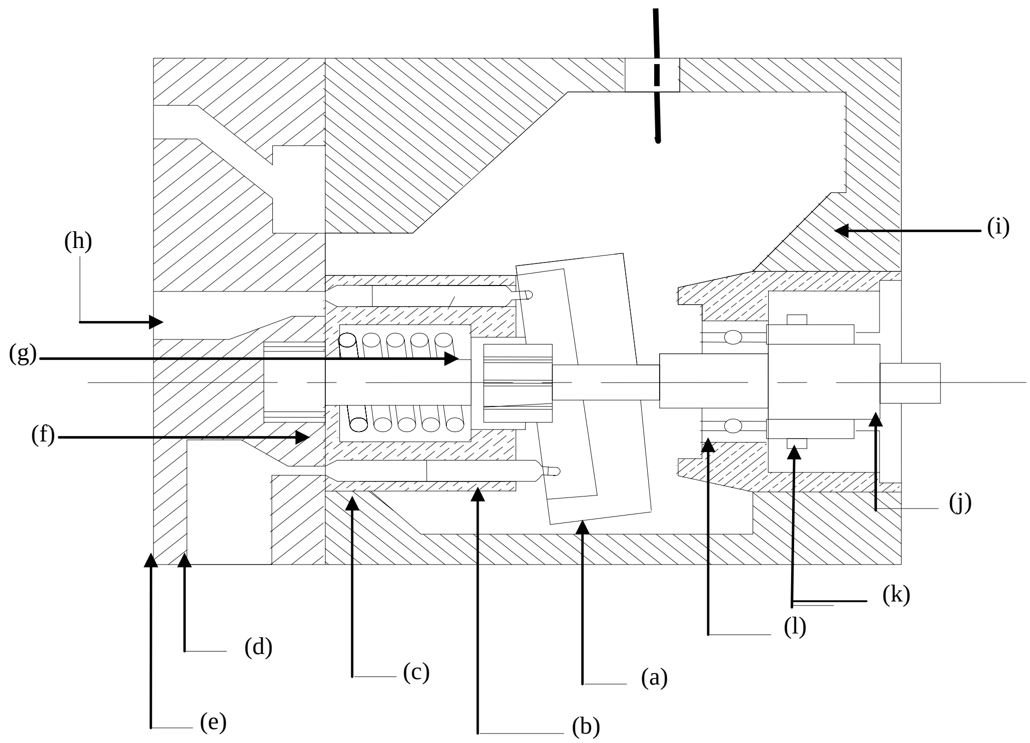
Winston, W. L., 1994, Introduction to Mathematical Programming: Applications and Algorithms. Duxbury Press.

APPENDIX A

AXIAL PISTON PUMPS
(Adapted from Bhandare and Allada, 2006)

Overview:

In the present problem, product platforms are formed for a family of five axial piston pumps. Pumps are devices that transfer mechanical energy into fluid power. They are classified primarily on the type of motion that causes a transfer of energy. The axial piston pump uses reciprocating motion to transfer energy. It is a positive displacement pump with designs available to obtain fixed and variable displacements. Fixed displacement-type pumps have been considered for the present case study. In the present example, various displacement requirements for the individual axial piston pumps have been considered. Demand data (non-uniform) for each pump is assumed to be given *a priori*. There has been no explicit market segmentation based on the displacement of the axial piston pump. The problem considers the manufacturing cost of the axial piston pumps and aims to minimize it by commonalizing the values of the design variables. The major components of a typical fixed-displacement axial piston pump are shown in Figure A1. A valve plate contains an inlet and an outlet port and functions as the back cover. A rotating group consists of a cylindrical block splined to a drive shaft, splined spherical washer, spring, pistons with shoes, swash plate, and shoe plate. The spring forces the cylindrical block against the valve plate, while the spherical washer pushes against the shoe plate. This action holds the piston shoes against the swash plate, ensuring that the pistons reciprocate as the cylinder turns. The swash plate



(a) Swash plate assembly (including shoe plate)
(b) Piston (Plungers) with shoes
(c) Cylinder block
(d) Inlet
(e) Valve plate
(f) Spring
(g) Splined washer
(h) Outlet
(i) Housing
(j) Drive shaft
(k) Shaft seal
(l) Shaft bearing

Figure A1: Schematic of an Axial Piston Pump (Adapted from Bhandare and Allada, 2006)

is stationary in a fixed-displacement design. For every rotation of the shaft there is a change in the angle of the swash plate that leads to a fixed amount of suction and discharge of the fluid. This discharge is controlled by the design parameters affecting the *displacement* (swash plate angle, number of plungers, diameter of the plunger). The displacement of an axial piston pump is dependent on design parameters, such as diameter of the plunger, swash plate angle perpendicular to the axis of rotation, number of plungers used, and pitch circle diameter for the imaginary circle encompassing the plungers.

MATHEMATICAL DESCRIPTION

Nomenclature:

Δ_o	Acceptance tolerance limit by the customer (assumed to be 10%)
A_o	Cost to the manufacturer due to performance at Δ_o in $
A_i	Scaling coefficient for the demand estimation
C_{bi}	Cost for shaft bearing for variant *i* in $
C_{hi}	Cost of housings for variant *i* in $
C_{hi_mat}	Material cost for housing for variant *i* ($ 0.45/Kg)
C_{hi_mch}	Manufacturing cost for housing for variant *i* in $
C_{mi}	Total manufacturing cost for variant *i* in $
$C_{p\_mat\_i}$	Material cost for plunger C_{pni} ($ 0.70/Kg)
$C_{p\_mch\_i}$	Manufacturing cost for plunger C_{pni} $
C_{pni}	Cost of n_i plungers for variant *i* in $
C_{pi}	Individual cost of product *i* in $
C_{pli}	Cost of performance loss for variant *i* due to platforming in $
C_{pni}	Cost of individual plunger for variant *i* in $
C_{sg}	Cost of spring in $0.25 per piece
$C_{sp\_mat\_i}$	Material cost for swash plate assembly for variant *i* $ 0.75/Kg
$C_{sp\_mch\_i}$	Manufacturing cost for swash plate assembly for variant *i* $
C_{spi}	Cost of swash plate assembly for variant *i* in $
d_{bi}	Shaft bearing diameter for variant *i* in mm
D_{hi_i}	Inner diameter for housing for variant *i* in mm
D_{ho_i}	Outer diameter for housing for variant *i* in mm
D_i	Demand for variant *i* in units per year
d_i	Diameter of plunger for variant *i* in mm (14 to 30 mm; incremented in steps of 0.2 mm in this study).
d_{i_p}	Inner diameter for plunger with diameter d_i in mm
d_{max}	Minimum diameter of the plunger in mm
d_{min}	Maximum diameter of the plunger in mm
D_{s_i}	Outer diameter for swash plate assembly in mm
d_{si}	Calculated shaft bearing diameter for variant *i* in mm
fos_{hi}	Factor for safety for housing (Assumed to be 4)
k_c	Unit manufacturing cost for operation *c* in $
k_i	Quality coefficient
m	Number of variants in a product family (Assumed to be 5)
n_i	Number of plungers for variant *i* (Assumed to be 5, 6, or 7)
n_{maz}	Maximum number of plungers
n_{min}	Minimum number of plungers

P_{di}	Pitch circle diameter for variant i in mm.
P_i	Pressure for variant i in bar
Q_i	Displacement for pump variant i in cc/rev (38 to 90 cc/rev)
SL_i	Stroke length for variant i in mm
t_i	Wall thickness for housing of variant i in mm
t_p	Wall thickness for plunger in mm
T_s	Torque developed due to pressure and displacement in N-m
u_{mc_h}	Unit cost of material used for housing in $
u_{mc_p}	Unit cost of material used for plunger in $
w_i	Width of shaft bearing used for variant i in mm (25.40 to 38.10 mm)
Y	Desired target value of displacement
α_i	Swash plate angle for variant i in degrees (9-21 to degrees; incremented in steps of 0.5 degree in this study).
ρ_h	Density of material used for housing in kg/cu.mm (0.00070kg/cu.mm)
ρ_p	Density of material used for plunger in kg/cu.mm (0.00078kg/cu.mm)
ρ_{sp}	Density of material used for swash plate assembly in kg/cu.mm
τ_{sp}	Shear stress for material used for swash plate in MPa (200 Mpa)

Relationship between the parameters presented above to that used in Chapter 3.

$x_1 = \alpha_i$	Swash plate angle for variant i
$x_2 = d_i - t$	Inside diameter of the plunger
$x_3 = n_i$	Number of plungers in the motor
$x_4 = d_i + t$	Outside diameter of the plunger

Cost Modeling:

The cost of providing a product family is defined as the sum of the cost of the product variants and the cost associated with performance loss due to platforming. The objective of the platform problem is to minimize the total cost of providing the product family. Considering the non-uniform demand associated, individual product cost variant, and the performance-loss cost for each variant due to platforming the variables.

$$Min\ \ Z = \sum_{i=1}^{m} D_i \times (C_{pi} + C_{pli}) \quad \text{… (A1)}$$

Where, D_i is the non-uniform demand associated with each product variant i, the cost of the product 'C_{pi} ' of the individual product variant i, and the cost of performance loss 'C_{pli}' for variant i due to platforming. The demand is calculated using the equation:

$$D_i = A_i \times \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2} \qquad \text{… (A2)}$$

Here, the scaling coefficient A_i is assumed to be 0.86.

Table A1 shows the demand data corresponding to the three-demand scenarios (1, 2, and 3) for the five variants of the axial piston pumps.

Table A1: Demand Data for Product Variants (Adapted from Bhandare and Allada, 2006)

Product	Annual Demand		
	Demand Scenario- 1	Demand Scenario- 2	Demand Scenario- 3
V_1	482	800	51
V_2	939	648	224
V_3	1200	322	687
V_4	1059	145	1132
V_5	549	27	1500

Table A2 lists the different technical parameters pertaining to the five variants of the axial piston pumps. These variants differ in terms of the displacement of the pump. Further, the acceptable loss in Q_i for each product introduced through platforming is also given in Table A2.

The performance measure of each pump is assumed to be solely dependent on:

Displacement of the pump(Q_i)

The design variables influencing the performance characteristic are as follows:

Table A2: Specification for Product Variants of the Axial Piston Pump (Adapted from Bhandare and Allada, 2006)

Product Variant	Pressure P_i (bar)	Driver Speed (rpm)	Displacement Q_i (cc/rev)	Acceptable Loss % $\pm r_i$
V_1	350	2650	38	10
V_2	400	2700	51	10
V_3	350	2500	65	10
V_4	350	2400	75	10
V_5	350	2200	91	10

Swash plate angle(α_i) (9-21 degrees)

Diameter of the plunger (d_i)(14-30 mm)

The number of plungers (n_i) (5, 6, or7)

For axial piston pumps, the performance characteristic of importance during platforming is the displacement of the pump (Q_i). Hence, in the proposed methodology, only parameters influencing this performance characteristic during platform design are being considered. The primary formulae to obtain the displacement are shown in Equations (A3) through (A6).

$$Q_i = \frac{SL_i \times A_i \times n_i}{1000} \quad \text{… (A3)}$$

$$SL_i = \tan(\alpha_i) \times P_{di} \quad \text{… (A4)}$$

$$A_i = \frac{\pi \times d_i^2}{4} \quad \text{… (A5)}$$

$$P_{di} = \frac{n_i \times (d_i + 2 \times t_i)}{\pi} \quad \text{… (A6)}$$

Cost of each product variant *i*:

The cost of product variant i is considered as a function of only critical components, such as the plunger, plunger spring, pump housing, shaft bearing, and swash plate assembly. The cost of these elements would account for around 70% of the total cost of the product and hence is a good estimate of the cost of the product.

The manufacturing cost and material cost for variant i are given as follows:

$$C_{mi} = C_{hi} + C_{pi} + C_{spi} + C_{bi} \qquad \text{… (A7)}$$

Prior to introducing equations for calculating cost of the components, the details of the operations are given in Table A3. The following sub sections detail the cost calculations:

Cost of housing:

The cost of housing is sum of the material and manufacturing costs and is given as follows:

$$C_{hi} = C_{hi\_mat} + C_{hi\_mch} \qquad \text{… (A8)}$$

An axial piston pump consists of two housings. Housing 1 encloses the swash plate assembly and the bearing, whereas Housing 2 encloses the plungers. The material cost of the housing is the product of the mass of the material and the unit cost of material per unit mass. The mass of the material is calculated using the formula:

Mass = Volume x density of the material … (A9)

For each product variant i, the material cost for Housing 1 is as follows:

Table A3: Manufacturing Operations for Various Components (Adapted from Bhandare and Allada, 2006)

1) Plunger	**2) Housing 1**	**3) Housing 2**	**4) Swash plate**
1.1) Rough turning on outer diameter	2.1) Casting of the housing	3.1) Casting of the housing	4.1) Face milling
1.2) Drilling for oil flow	2.2) ID turning	3.2) Plunger hole drilling	4.2) Swash angle milling
1.3) Reaming of oil hole	2.3) Step turnings for snap ring and seals	3.3) Plunger hole reaming	4.3) Step turning for the shaft
1.4) Tuff riding on the inner surface	2.4) Bearing slot turning	3.4) Tuff riding	4.4) Spline milling on end of shaft
1.5) Grinding on the outer diameter	2.5) Port drilling for inlet and outlet ports	3.5) Oil hole drilling	4.5) Grinding of shaft
1.6) Finish grinding/buffing on the outer	2.6) Port reaming of the plunger bores	3.6) Check valve port drillings	4.6) Phosphating of the housing bores
	2.7) Port threading for inlet and outlet ports	3.7) Outlet port drilling	
		3.8) Phosphating of the Housing	
		3.9) Port threading	

$$C_{hi\_1\_mat} = [(\frac{\pi \times D_{ho\_i}^{2} \times 3.5 \times SL_i}{4}) - (\frac{\pi \times D_{hi\_i}^{2} \times (3.5 \times SL_i - w_i)}{4})] \times \rho_h \times u_{mc\_h}$$

… (A10)

Table A4: Manufacturing Operation Details for Housing-1 (Adapted from Bhandare and Allada, 2006)

Manufacturing Operation	**Cost Formula**	**Cost/unit length K_c in $**
ID boring	$(3.5 \times SL_i - w_i) \times K_c$	**0.02**
Shaft hole drilling	$w_i \times K_c$	**0.015**

Where, $D_{hi\_i} = D_{s\_i} + 8$ and $D_{ho\_i} = D_{hi\_i} + 4 \times t_i$

The material cost for housing 2 is given as follows:

$$C_{hi\_2\_mat} = [(\frac{\pi \times D_{ho\_i}^2 \times 6 \times SL_i}{4}) - (\frac{n_i \times \pi \times d_i^2 \times 3 \times SL_i}{4})] \times \rho_h \times u_{mc\_h} \quad \text{… (A11)}$$

The cost of the manufacturing is the sum of the manufacturing and material costs for the operations performed. Table A5 gives the manufacturing operations and cost equations for housing 2

Table A5: Manufacturing Operation Details for Housing-2 (Adapted from Bhandare and Allada, 2006)

Manufacturing Operation	**Formula**	**Cost/unit length K_c in $**
Plunger hole drilling	$3 \times SL_i \times K_c$	**0.015**
Plunger hole reaming	$3 \times SL_i \times K_c$	**0.02**
Plunger hole lapping	$3 \times SL_i \times K_c$	**0.05**

Table A6 gives the list of materials used for components of the pumps with unit material cost. The total manufacturing cost for housing is given by

$$C_{hi\_mch} = C_{hi\_1\_mch} + C_{hi\_2\_mch} \qquad \text{... (A12)}$$

Table A6: List of Material Details (Adapted from Bhandare and Allada, 2006)

Component	**Material**	**Unit cost of material per kg in $**
Plunger	Medium carbon steel	**2.25**
Housing 1	Cast iron	**1.75**
Housing 2	Cast iron	**1.75**
Swash plate Assembly	Medium carbon steel(alloy)	**2.25**

Cost of plunger:

The cost of each plunger is the sum of material cost and the manufacturing cost. This is given by

$$C_{pni} = C_{p\_mat\_i} + C_{p\_mch\_i} \qquad \text{… (A13)}$$

The material cost is product of the mass of material and the cost per unit mass of the material used for the plunger. The cost of the spring used for each plunger is also added in the material cost equation. The material cost is given by the equation

$$C_{p\_mat\_i} = ((\frac{\pi \times d_i^2}{4} \times 4 \times SL_i) - (\frac{\pi \times d_{i\_p}^2}{4} \times 3 \times SL_i)) \times \rho_p \times u_{mc\_p} + C_{sg} \qquad \text{… (A14)}$$

Where, d_{i_p} is calculated using the formula,

$$t_p = \frac{d_i - d_{i-p}}{2} \quad Also \quad t_p = \frac{P_i \times d_i \times fos_p}{20 \times \sigma_{p\_mat}} \qquad \text{… (A15)}$$

The cost of the spring C_{sg} is assumed to be \$ 0.25 per piece. This is irrespective of the size of the plunger. Table A7 gives the manufacturing operations and cost equations associated for the plunger.

For the given variant *i*, the total cost of the plungers is the product of the cost of each plunger and the number of plungers (n_i) used for the variant is as follows:

$$C_{pi} = n_i \times C_{pni} \qquad \text{… (A16)}$$

Cost of swash plate:

The cost of swash plate assembly is the sum of the material cost and the manufacturing cost. This is given by the following equation:

$$C_{spi} = C_{sp\_mat\_i} + C_{sp\_mch\_i} \qquad \text{… (A17)}$$

Table A7: Manufacturing Operation Details for Plunger (Adapted from Bhandare and Allada, 2006)

Manufacturing Operation	Formula	Cost/unit length K_c in \$
Turning	$4 \times SL_i \times K_c$	**0.02**
Grinding	$4 \times SL_i \times K_c$	**0.04**
Reaming	$3 \times SL_i \times K_c$	**0.02**
Drilling	$SL_i \times K_c$	**0.015**

The material cost is the product of the mass of material and the cost per unit mass of the material used for the swash plate as be given by equations 14 to 16.

$$C_{sp\_mat\_i} = \left(\frac{\pi \times D_{si}^2}{4} \times SL_i + 3 \times SL_i \times \frac{\pi \times d_{b_i}^2}{4}\right) \times \rho_{sp} \times u_{mc\_sp} \qquad \text{… (A18)}$$

Where, $D_{si} = P_{di} + d_i + 3 \times t_i$

$$d_{si} = \sqrt[3]{\frac{16 \times T_s}{\pi \times \tau_{sp}}} * fos_s \qquad \text{… (A19)}$$

$$T_s = \frac{Q_i \times P_i}{20 \times \pi \times \eta_m} \qquad \text{… (A20)}$$

Here, the mechanical efficiency (η_m) of the pump is taken as 96%.The manufacturing cost for the swash plate assembly is sum of the costs for the manufacturing operations performed. Table A8 shows the manufacturing opera-tions and the corresponding cost equations.

Table A8: Manufacturing Operation Details for Integral Swash Plate-Shaft (Adapted from Bhandare and Allada, 2006)

Manufacturing Operation	**Formula**	**Cost/unit length K_c in $**
Turning	$4 \times SL_i \times k_c$	**0.02**
Grinding	$3 \times SL_i \times k_c$	**0.04**
Milling	$\alpha_i \times k_c$	**0.06**

Bearing Cost:

The bearings in the drive are important in cost estimation. Needle bearings are usually used for motors and pumps. This is a bought-out component. The cost of the bearing would depend on the diameter of the shaft and the speed of application. Since the bearings are available for standard diameters, the selection of bearing is carried as follows:

For a calculated diameter d_{si} of the shaft, standard bearing with diameter d_{bi} is selected. This is under the constraint $d_{bi} >= d_{si}$ and d_{bi} is the nearest value available to d_{si}. Table A9 gives the standard needle bearings available for the product range considered. The standard bearings listed are for maximum speed of desired for the product family. Table A9 also gives the corresponding cost of the standard bearings.

Table A9: Standard Bearing Sizes and Costs (Adapted from Bhandare and Allada, 2006)

Standard size of bearing d_{bi} in mm	Outer diameter (D_0) in mm	Width (w) in mm	Cost in $
28.58	41.28	25.40	3.0
30.16	42.86	25.40	3.0
31.75	44.45	25.40	4.35
33.34	46.04	25.40	4.35
34.93	47.63	25.40	4.35
36.51	49.21	25.40	5.65
38.10	52.39	25.40	5.65
39.69	53.98	25.40	6.05
41.28	55.56	25.40	6.05
42.86	57.15	31.75	6.05
44.45	58.74	31.75	7.45
46.04	60.33	31.75	7.45
47.63	61.91	25.40	7.45
49.21	63.50	25.40	9.0
50.80	65.09	25.40	9.0
57.15	76.20	38.10	10.80
63.50	82.55	38.10	10.80

Cost of Quality loss due to platforming:

For the axial piston pump, we assume that the performance deviation on either side of the target value would result in customer dissatisfaction and, hence, we adopt the nominal the better scenario. To establish the value of the quality co-efficient *k*, we assign the values for the customer dissatisfaction limit and also the cost at these limits. For the family of axial piston pumps, Δ_o, considered as the customer dissatisfaction limit, is assumed to be at 10% deviation from the desired or specified product target value. For example, a product variant with specified displacement value of 40cc/rev would have customer dissatisfaction limits of 36 cc/rev and 44 cc/rev.

Technical Constraints:

The technical constraints are defined by the range limits for the design variables and other design requirements. For the axial piston pumps, the range values for the primary parameters, influencing the performance characteristic and other design considerations are listed in Table A10.

1. The range value for the secondary parameter pitch circle diameter is derived using Equations (A21) and (A22)

$$P_{d\,\min} = \frac{d_{\min} \times n_{\min} + 2 \times n_{\min} \times t_{\min}}{\pi} \qquad \text{… (A21)}$$

$$P_{d\,\max} = \frac{d_{\max} \times n_{\max} + 2 \times n_{\max} \times t_{\max}}{\pi} \qquad \text{… (A22)}$$

2. The pumps must satisfy the pressure requirements for safety. The pump derived through platforming should be able to withstand the pressure rating assigned for the pump through individual design

$$t_i^{'} \geq t_i \qquad \text{… (A23)}$$

$$P_{di}^{'} \geq P_{di} \qquad \text{... (A24)}$$

$$\text{Where, } t_i = \frac{P_i \times d_i}{20 \times \sigma_c} \times fos_{hi} \qquad \text{… (A25)}$$

$$t_i' = \frac{\pi \times P_{di}' - n_i' \times d_i'}{2 \times n_i'} \qquad \text{… (A26)}$$

t_i' is the wall thickness for housing for variant i after platforming

P_{di}' is the pitch circle diameter for variant i after platforming

d_i' is the diameter of plunger for variant i after platforming

n_i' is the number of plungers for variant i after platforming

Table A10: Different Parameters Used in the Mathematical Description of the Problem (Adapted from Bhandare and Allada, 2006)

Parameter	**Range or discrete values**
Diameter of piston (d_i)	14 mm to 30 mm
No. of pistons (n_i)	5, 6, or 7
Swash plate angle (α_i)	9 to 21 degrees
Density of cast iron (ρ_c)	0.00070 kg/cu.mm
Density of steel (ρ_s)	0.00078 kg/cu.mm
Design factor of safety *(fos)* Housing, Plunger, and swash plate	4, 2, and 2.5 for Housing, Plunger, and swash plate, respectively
Yield strength of cast iron (σ_c)	250 MPa
Yield strength of steel (σ_s)	350 MPa

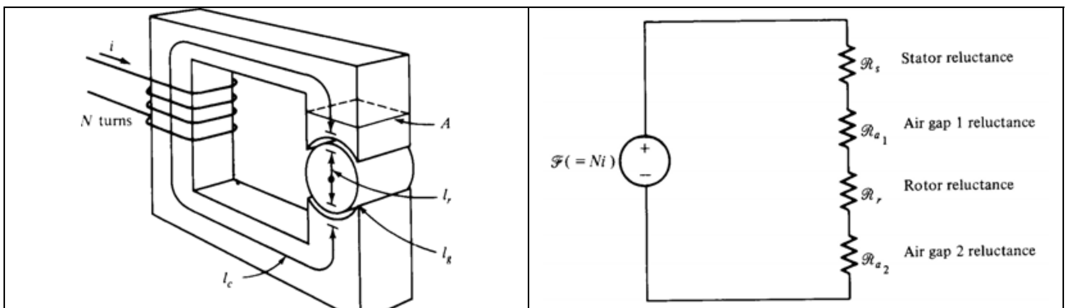
APPENDIX: B

MATHEMATICAL MODEL FOR A UNIVERSAL ELECTRIC MOTOR
(Adapted from Simpson *et al*., 2001)

A universal motor is essentially the same as a Direct Current (DC) series motor. In a universal motor, wire is wrapped around the armature and the field in series, which means that the same current is applied to both sets of wire. As current passes through the field windings, a large magnetic field is generated. This field passes through the metal of the field windings, across an air gap between the field and the armature, through the armature windings, through the shaft of the armature, across another air gap, and back into the metal of the field windings, thus completing a magnetic circuit. When the magnetic field passes though the armature windings, the magnetic field exerts a force on the current carrying wires. Because of the geometry of the windings, current on one side of the armature is always passing in the opposite direction to the current on the other side of the armature. Thus the force exerted by the magnetic field on one side of the armature is opposite to the force exerted on the other side of the armature. Thereby a net *torque* is exerted on the armature, causing the armature to spin.

The model takes as input the design variables {N_c, N_s, A_{wa}, A_{wf}, r_o, t, 1_{gap}, I, V_t, L} and returns as output the power (P), torque (T), mass (M), and efficiency (η) of the motor. To formulate the model, it is necessary to derive equations for P, T, M, and η as functions of the design variables. The equations are derived from Chapman 1991 and Cogdell 1996 for DC electric motors unless otherwise noted. Following are the relationships between parameters used here to that used in Chapter 4.

$x_1 = \text{Nc}$	Number of turns in the armature
$x_2 = \text{Ns}$	Number of turns in the field
$x_3 = \text{Awa}$	Area of the armature
$x_4 = \text{Awf}$	Area of the field wire
$x_5 = \text{r}_0$	Radius of the stator
$x_6 = \text{t}$	Thickness of the stator
$x_7 = \text{I}$	Current drawn by the motor
$x_8 = \text{L}$	Stack length



(a) Physical Model

(b) Magnetic circuit

Figure B1: An Idealized DC Motor (Adapted from Simpson *et al.*, 2001)

Power

The basic equation for power output of a motor is the input power minus losses, where the input power is the product of the voltage (V) and current (I).

$$P = P_{in} - P_{losses} = VI - P_{losses} \quad \text{(B1)}$$

For a universal motor, power is lost:

In the copper wires as they heat-up (copper losses)

At the interface between the brushes and the armature (brush losses)

In the core, due to hysteresis and eddy currents (core losses)

In mechanical friction in the bearings supporting the rotor (mechanical losses)

In heating up the core and copper, which adversely affects the magnetic properties of the core and the current-carrying ability of the wires (thermal losses)

Due to stray losses

Simple analytic expressions only exist for the copper losses and the brush losses. Stray losses are usually assumed to be no more than 1%, and thus can be neglected. Mechanical losses can be minimized by an appropriate choice of the bearing and housing arrangement; however, these variables are beyond the scope of the model. Hence, mechanical losses are neglected. Core losses, especially those incurred by eddy currents, can be minimized by the use of thin laminations in the stator and rotor; assuming this is done, the core losses can be assumed to be small and thus can be neglected. Thermal losses are in general non-negligible, but are highly dependent upon the external cooling scheme (e.g., cooling fan, fins on the housing, etc.} applied to the motor. Since an effective cooling scheme can keep the motor from running too hot, and since the setup of the cooling configuration is beyond the scope of this model, thermal losses are neglected.

The combined effects of all the aforementioned neglected losses will, however, decrease the output power and efficiency from the predicted value from the model. Nevertheless, the following equations serve as a sufficiently accurate model for the DC operation of a universal motor. Consequently, the general equation for power losses reduces from to a more manageable

$$P_{losses} = P_{copper} + P_{brush} + P_{thermal} + P_{core} + P_{mechanical} + P_{stray} \tag{B2}$$

$$P_{lasses} = P_{copper} + P_{brush} \tag{B3}$$

where

$$P_{copper} = I^2.(R_a + R_s) \tag{B4}$$

and

$$P_{brush} = 2\ .\ I \tag{B5}$$

However, R_a and R_s, the resistances of the armature and field windings, can be specified further as functions of the design (input) variables. The resistances, R_a and R_s, can be computed directly from the general equation that the resistance of any wire is given by the resistivity of the wire times the wire length divided by the cross-sectional area of the wire. We assume that each wrap (i.e., turn) of wire on the armature is approximately the shape of a rectangle with length L (the stack length of the motor) and width I_r (the diameter of the armature). In terms of the physical dimensions of the motor, I_r can be expressed as two times the radius of the armature, which is just the outer radius of the stator minus the thickness of the stator minus the air gap length, so that the length of one wrap of wire on the armature is:

$$Length_{\substack{one\\wrap}} = 2.L + 2.lr = 2.L + 4.(r_0 - t - l_{gap}) \tag{B6}$$

The total length of wire on the armature is the stack length, L, times the total number of wraps on the armature, N_c, so that the resistance of the armature, R_a, is

$$R_s = \frac{\rho.P_{field}(2.L + 4.(r_o - t)).N_c}{A_{wire}} \tag{B7}$$

where ρ is the resistivity of the copper wire. Similarly, assuming that each wrap of wire on the field is approximately the shape of a rectangle with the length L (the stack length of the motor) and the width double the inner radius of the stator (r_o-t), the resistance of the stator, R_s, is

$$R_s = \frac{\rho.P_{field}(2.L + 4.(r_o - t)).N_s}{A_{wire}} \tag{B8}$$

However, the purpose of the field windings is to create a magnetic field across the armature, thus requiring two field poles: one for the "North" end of the magnetic field and the other for the "South" end; thus, P_{field} is taken as 2, which completes the derivation of the power equation.

Efficiency

The equation for efficiency can be computed directly from the equation for power. The basic equation for efficiency, expressed as a decimal and not a percentage, is given by:

$$\eta = \frac{P}{P_{in}} \tag{B9}$$

where P and P_{in} are given by Equation B1

Mass

To estimate the mass of the motor, the motor is modeled as a solid steel cylinder, with length L and radius $l_r/2$ for the armature, and a hollow steel cylinder with length L, outer radius r_o, and inner radius (r_o-t) for the stator. The mass of the windings on both the armature and the field are also included, where the length of each winding is the same as those assumed in the derivation of power losses. Thus, the equation for mass is

$$Mass = M_{stator} + M_{armature} + M_{windings} \tag{B10}$$

where

$$M_{stator} = \pi.({r^2}_0 - (r_0 - t)^2).L.\rho_{steel} \tag{B11}$$

$$M_{armature} = \pi.(r_0 - t - l_{gap})^2.L.\rho_{steel} \tag{B12}$$

$$M_{windings} = ((.2.L + 4.(r_0 - t - l_{gap})).N_c.A_{wa} + (2.L + 4.(r_0 - t)).2.N_s.A_{wf}).\rho_{copper} \tag{B13}$$

Now with expressions for M_{stator}, $M_{armature}$, and $M_{winding}$ in terms of the design variables, the mass of the motor, Equation B10, also can be estimated from the design (input) variables.

Torque

The last equation to derive is an equation for torque. In general, the torque of a DC motor is given by the product of a motor constant, K, the magnetic flux, φ, and the current. I:

$$T = K\phi I \tag{B14}$$

For a DC motor, K is computed as:

$$K = \frac{Z.P_{armature}}{2.\pi.a} \tag{B15}$$

where Z (the number of conductors on the armature) is just twice the number of windings on the armature and *a* (the number of current paths on the armature) is just two times the plex of the winding on the armature. Assuming a simplex (m = l) wave winding on the armature, *a* is equal to two. Since the number of armature poles on a universal motor is almost invariably two (see Veinott and Martin, 1986), or

$$P_{armature} = 2 \tag{B16}$$

K can be reduced to

$$K = \frac{2.N_c.2}{2.\pi.2} = \frac{N_c}{\pi} \tag{B17}$$

The derivation of the flux term is much more complicated. Consider the idealized DC motor shown in Figure B3a with its corresponding magnetic circuit shown in Figure B3b. As shown in the figure, N is the number of turns on the stator (which is equal to $2N_s$ for the model being derived), I equals the current, A is the cross-sectional area of the stator, l_r equals the diameter of the armature, 1_g is the gap length, and 1_c is the mean magnetic path length in the stator.

In general the equation for flux through a magnetic circuit is simply the magnetomotive force, , divided by the total reluctance of the circuit,

$$\phi = \frac{\Im}{\Re} \tag{B18}$$

where the magnetomotive force, ~, is simply the number of turns around one pole of the field times the current:

$$\Im = N.I \tag{B19}$$

The total reluctance, , is calculated from the magnetic circuit shown in Figure B2. For a magnetic circuit, reluctances in series add like resistors in series in an electric circuit; therefore, the total reluctance in the idealized DC motor is the sum of the reluctances of the stator, rotor, and two air gaps:

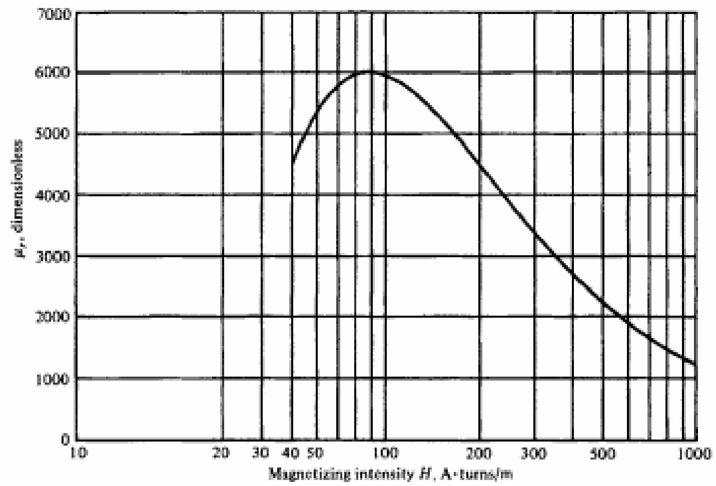


Figure B2: Relative Permeability Versus Magnetizing Intensity for a Typical Piece of Steel (Adapted from Simpson *et al*., 2001)

$$\Re = \Re_s + \Re_r + 2.\Re_a \tag{B20}$$

where, in general, reluctance is calculated as the length of the material divided by the product of the permeability of the material and the cross-sectional area of the material. When permeability, μ, is expressed as the relative permeability of the material times the permeability of free space, μ_o, the reluctance of the stator, rotor, and air gaps are:

$$\Re_s = \frac{l_c}{\mu_{steel}.\mu_0.A_s}, \Re_r = \frac{l_r}{\mu_{steel}.\mu_0.A_r}, \Re_a = \frac{l_g}{\mu_{steel}.\mu_0.A_a} \tag{B21}$$

For a closer approximation to the universal motor for this example, the idealized DC motor geometry shown in Figure B1 is modified to be more representative of a real universal motor. The resulting model geometry is shown in Figure B3a and is described by the outer radius of the stator, r_o, the thickness of the stator, t, the diameter of the armature, l_r, the length of the air gap, 1_g, and the stack length, L. The resulting magnetic circuit is shown in Figure B3b; notice that the magnetic circuit for the idealized DC motor and the magnetic circuit for a universal motor are different, because in a universal motor there are two paths which the magnetic flux can take around the stator, i.e., clockwise and counter-clockwise. These two paths are in parallel and thus are included in the magnetic circuit as two parallel flux paths. Reluctances in parallel in a magnetic circuit act like resistors in parallel in an electric circuit, so that the combined reluctance of two identical reluctances in parallel is simply one-half the reluctance of either path. Therefore, for a universal motor so that Equation B20 for the total reluctance, R, still holds.

$$\Re_s = \frac{l_c}{2.\mu_{steel}.\mu_0.A_s}, \Re_r = \frac{l_r}{\mu_{steel}.\mu_0.A_r}, \Re_a = \frac{l_g}{\mu_{air}.\mu_0.A_a} \tag{B22}$$

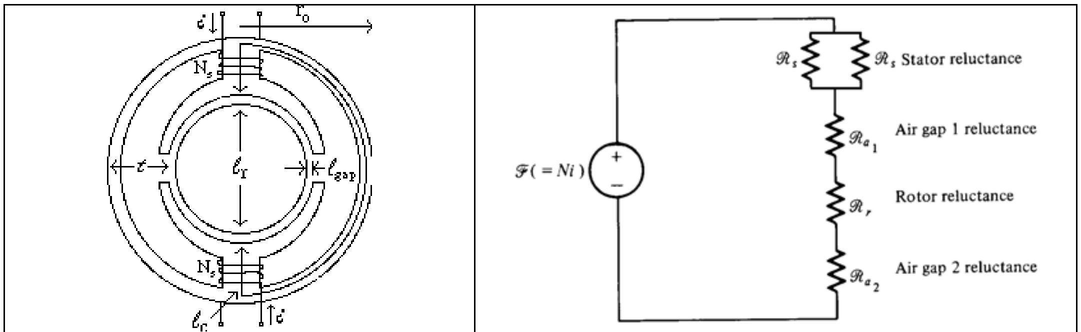
In Equation B22, the mean magnetic path length in the stator, l_c, is taken to be one-half the mean circumference of the hollow stator cylinder

$$l_c = \pi.\frac{(2.r_0 + t)}{2} \tag{B23}$$

The cross-sectional area of the stator, A_s, is taken to be the thickness of the stator times the stack length, and the cross-sectional area of the armature is approximated as the product of the diameter of the armature and the stack length. The cross-sectional area of the air gap is taken to be the length of the air gap times the stack length.

The last expression needed for the calculation of reluctance is the relative permeability of the stator and the armature. For the purposes of this model, both the stator and the armature are assumed to be made of steel with the relative permeability versus magnetizing intensity curve for typical steel shown in Figure B2.

The curve is divided into three sections, and each section is fit with an appropriate numerical expression in order to approximate the permeability in the model.



(a) Physical Model (b) Magnetic circuit

Figure B3: Model Geometry of a Universal Electric Motor (Adapted from Simpson *et al.*, 2001)

The curve fits are:

$$\mu_r = -0.22791.H^2 + 52.411.H + 3115.8 \qquad H \leq 220$$

$$\mu_r = 11633.5 - 1486.33.In(H) \qquad 220 < H \leq 1000$$

$$\mu_r = 1000 \qquad H > 1000 \tag{B24}$$

Where, from ampere's Law, the magnetizing intensity, H, is given by

$$H = \frac{N_c.I}{l_c + l_r + 2.l_{gap}} \tag{B25}$$

The relative permeability of air, $\mu_{air,}$ is taken as unity, and the permeability of free space is a constant, $\mu_0 = 4\pi 10^{-7}$. Now, with expressions for $K, \phi, \Im, \Re_s, \Re_r, \Re_a, 1_c, 1_r, A_s, A_r, A_a$, and μ_{steel} in terms of the design (input) variables, the torque equation is complete.