SELF-RESILIENT PRODUCTION SYSTEMS:
FRAMEWORK FOR DESIGN SYNTHESIS
OF MULTI-STATION ASSEMBLY SYSTEMS

By

Tirawat Phoomboplab

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Engineering

School of Engineering

University of Warwick

July 2012

ABSTRACT

Product design changes are inevitable in the current trend of time-based competition where product models such as automotive bodies and aircraft fuselages are frequently upgraded and cause assembly process design changes. In recent years, several studies in engineering change management and reconfigurable systems have been conducted to address the challenges of frequent product and process design changes. However, the results of these studies are limited in their applications due to shortcomings in three aspects which are: (i) They rely heavily on past records which might only be a few relevant cases and insufficient to perform a reliable analysis; (ii) They focus mainly on managing design changes in product architecture instead of both product and process architecture; and (iii) They consider design changes at a station-level instead of a multi-station level.

To address the aforementioned challenges, this thesis proposes three interrelated research areas to simulate the design adjustments of the existing process architecture. These research areas involve: (i) the methodologies to model the existing process architecture design in order to use the developed models as assembly response functions for assessing Key Performance Indices (KPIs); (ii) the KPIs to assess quality, cost, and design complexity of the existing process architecture design which are used when making decisions to change the existing process architecture design; and (iii) the methodology to change the process architecture design to new optimal design solutions at a multi-station level.

In the first research area, the methodology in modeling the functional dependence of process variables within the process architecture design are presented as well as the relations from process variables and product architecture design. To understand the engineering change propagation chain among process variables within the process architecture design, a functional dependence model is introduced to represent the design dependency among process variables by cascading relationships from customer requirements, product architecture, process architecture, and design tasks to optimise process variable design. This model is used to estimate the level of process variable design change propagation in the existing process architecture design

Next, process yield, cost, and complexity indices are introduced and used as KPIs in this thesis to measure product quality, cost in changing the current process design, and dependency of process variables (i.e, change propagation), respectively. The process yield and complexity indices are obtained by using the Stream-of-Variation (SOVA) model and functional dependence model, respectively. The costing KPI is obtained by determining the cost in optimizing tolerances of process variables. The implication of the costing KPI on the overall cost in changing process architecture design is also discussed. These three comprehensive indices are used to support decision-making when redesigning the existing process architecture.

Finally, the framework driven by functional optimisation is proposed to adjust the existing process architecture to meet the engineering change requirements. The framework provides a platform to integrate and analyze several individual design synthesis tasks which are necessary to optimise the multi-stage assembly processes such as tolerance of process variables, fixture layouts, or part-to-part joints. The developed framework based on transversal of hypergraph and task connectivity matrix which lead to the optimal sequence of these design tasks. In order to enhance visibility on the dependencies and hierarchy of design tasks, Design Structure Matrix and Task Flow Chain are also adopted. Three scenarios of engineering changes in industrial automotive design are used to illustrate the application of the proposed redesign methodology. The thesis concludes that it is not necessary to optimise all functional designs of process variables to accommodate the engineering changes. The selection of only relevant functional designs is sufficient, but the design optimisation of the process variables has to be conducted at the system level with consideration of dependency between selected functional designs.

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor Professor Darek Ceglarek for his kindly help and support in advising and reviewing this thesis. His encouragement greatly helps me to improve the research in many aspects. Also, I would like to thank Dr. Eduardo Izquierdo, Dr. Kevin Neailey, Professor Svetan Ratchev, Dr. Jeffrey Jones, and Professor Vinesh Raja for their valuable suggestions.

My gratitude is also contributed to all of my friends at The Digital Laboratory at the University of Warwick; Sylvester Arnab, Mitan Solanki, Prakash, Nagesh Shukla, Sudi Ceglarek, Duangthida Rotkanok and many others whom I did not mentioned. Their encouragements and suggestions brought me a lot of enjoyments during my study. Last but not least, I also owe gratitude to my family for their love, understanding, and unconditional support which always cheer me up and enhance my confidence to overcome many difficulties. They have a greatly contribution to make this research worthwhile.

DECLARATION

This thesis is presented in accordance with the regulations for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself, except where otherwise stated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AD	Axiomatic Design
ARF	Assembly Response Function
CA	Customer Attribute
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CM-EC	Collaborative Management of Engineering Change
CPA	Change Propagation Analysis
CPM	Change Prediction Method
DP	Design Parameter
DSM	Design Structure Matrix
ECC	Engineering Change Complexity
FR	Functional Requirement
KPI	Key Performance Index
KPC	Key Product Characteristic
KCC	Key Control Characteristic
KCC-HG	Key Control Characteristic - Hierarchical Group
PLM	Product Life-cycle Management
PV	Process Variables
RMS	Reconfigurable Manufacturing System
SOVA	Stream-of-Variation Model
TFC	Task Flow Chain

CHAPTER 1

INTRODUCTION: ENGINEERING CHANGES AND ASSEMBLY PROCESS REDESIGN FOR DIMENSIONAL MANAGEMENT

1.1 Introduction

Frequent and unpredictable market changes combined with ever greater customer expectations have led to enormous increases in design complexities of product and production systems. Additionally, manufacturers must address the simultaneous challenges of enhancing product quality and variety while minimizing investment cost and lead time in developing a new product. Achieving these expectations require advances in the development of fundamental methods and simulation approaches which can efficiently and effectively integrate product and process design with manufacturing process data and service engineering information. The development of such fundamental methods demands interdisciplinary focus towards the integrations of product design model (CAD), manufacturing/production model (CAM), control engineering models (state-space model), statistical models (multivariate statistic analysis), and system engineering models (e.g., product/process architecture) to formulate fundamental system performance optimisation methods. In this thesis, a methodology to optimise system performance is introduced which allows updating and adjusting process architecture design based on constantly changing product architecture design. The proposed methodology aims to achieve the new optimum which is robust to variations and satisfies the unforeseen requirements that emerge in the product life-cycle.

Current state-of-the-art in product and process design focuses on improving robustness. This robustness can be represented by quality indices such as process capability indices (C_p and C_{pk}), sensitivity to noise ratio, or ability of a process to produce outputs within specification limits. Other design criteria such as cost and technical feasibility are also taken into consideration while improving design robustness. Nevertheless, the unforeseen events or additional functional requirements which emerge during production or operation/service phases can affect the robustness of current product and process design. The unforeseen events and emerging information can be referred to as the uncertainty of design information which increases as a function of time as shown in Figure 1.1.

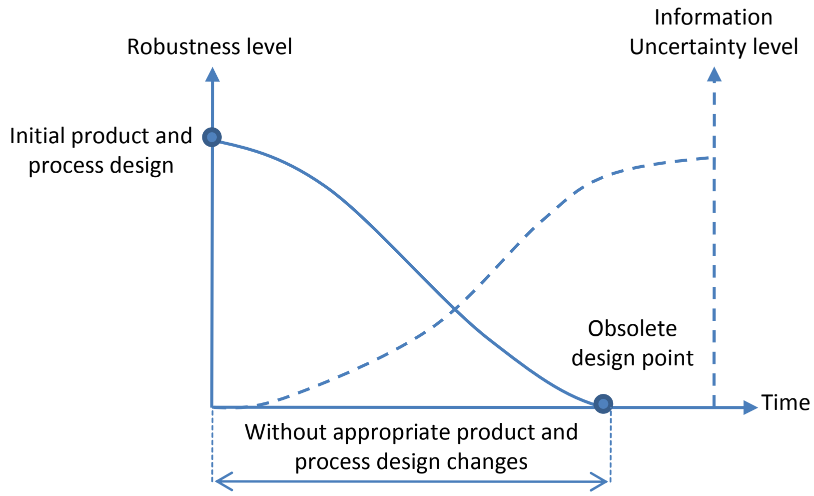


Figure 1.1: Relations between product/process design robustness and the information uncertainty.

For example, customer preferences about product functionalities can constantly change over a period of time. Consequently, the robustness of the initial product and process design declines as the uncertainty of the functional requirements increases. The initial product and process design becomes obsolete if the necessary product and process design adjustments or engineering changes are not implemented.

This thesis proposes a design synthesis methodology for optimizing production system design, which contributes to the development of so called *self-resilient production systems*, a closed-loop lifecycle modeling of production system. The self-resilient production systems aim to address frequent changes of market in term of customer preference, standard/regulation, and technologies as well as unexpected challenges during production and field service. The self-resilient production systems propose a comprehensive methodology to deal with faults (abnormal situations) and unexpected changes that emerge throughout the product life-cycle. These unexpected changes and faults cause several challenges as shown in Figure 1.2.

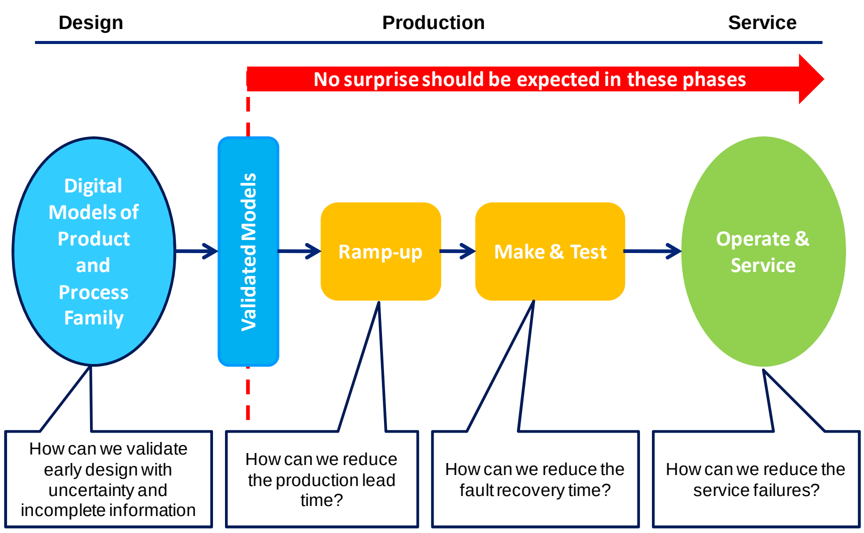


Figure 1.2: Challenges towards developing zero-defect products and processes in Product Life-cycle Management (PLM).

The self-resilient production system consists of three interconnected loops as shown in Figure 1.3 which are: (i) design synthesis; (ii) production ramp-up synthesis; and (iii) service synthesis. The information flow within as well as between PLM phases is one of the key concepts in development of closed-loop lifecycle modeling of self-resilient production system, which involves *intra-loops* and *inter-loops* information flows. The *intra-loop* information flow is related to modeling, transferring and analysis of data, information and engineering models needed in a single phase of PLM in order to satisfy key performance indicators of a single PLM phase. On the other hand, the *inter-loop* information flow involves modeling, transferring and analysis of data, information and engineering models between interconnected PLM phases in order to satisfy the overall key system performance indicators. A more detailed representation of the self-resilient production system can be found in Prakash *et al.* (2009).

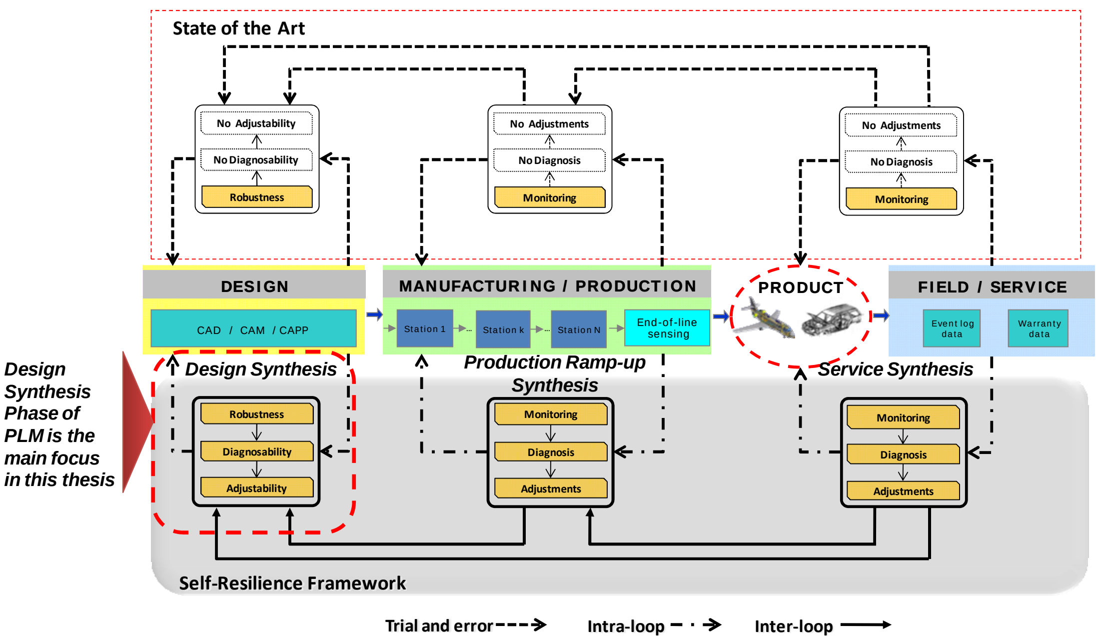


Figure 1.3: Self-resilient production system: Closed-loop lifecycle modeling of production system.

The inter-loop information flows from (i) production ramp-up synthesis; and (ii) service synthesis provide input data and information for the design synthesis phase. This is necessary for the analysis of potential engineering changes that must be conducted due to 6-sigma process and product failures that are non-compliant with key system performance indicators. For example, the required engineering changes might be caused by product faults that occur during the production phase or functional product faults that occur during service phase. The requirements for engineering changes are used as input for design synthesis to assess need for a new design configuration of process architecture. In the next section, the research motivation is demonstrated which the current industrial design practices have challenges to change the product and process design to meet new requirements.

1.2 Research Motivation

The limitations of the current design practice towards improving the assembly system are illustrated by using a case study of an automotive front-end assembly. The presented limitations of the design practice serves as the motivation to develop a proposed methodology described in this thesis. The automotive front-end assembly design model provided by a major automotive company is illustrated in Figure 1.4.

The front-end model consists of 215 *KCCs* including all part locating features and part-part mating surface dimensions. There are 61 measurement points, *Key Product Characteristics* (*KPCs*), to describe the front-end subassembly design parameters and functional requirements. For illustration purposes, all initial tolerance design of critical process variables, Key Control Characteristics (*KCCs)*, were assigned as +/**-** 1.00 mm according to current industrial best-practices. All locator positions, part-to-part joints, and assembly sequence are determined and designed based on best practice experience. The specifications of all 61 measurements were assigned as +/- 0.75 mm. Finally, the *KPC* variations are simulated by using variation simulation analysis software (3DCS Analyst). The initial analysis showed a very low process yield ≈0.00, (process yield defined as the joint probability that all *KPCs* are within the allowable specifications simultaneously) because the average process capability index, C_p, of all *KPCs* are lower than 0.45. It is obvious that if all *KPCs* are independent with the individual yield = 0.45, the simultaneous conformance rate or the yield will be very low for the overall assembly system $(0.45)^{61} \approx 0$.

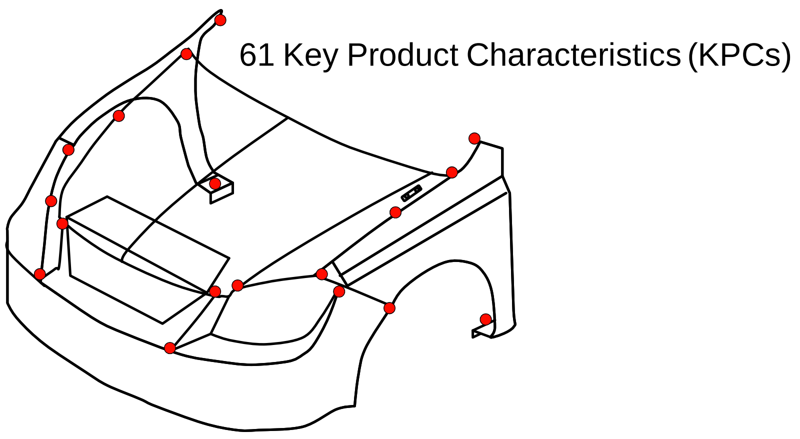


Figure 1.4: The Key Product Characteristics on Front-End assembly.

To determine the maximum feasible process yield of the Front End Assembly, all *KCC* tolerances are tightened to the smallest technically feasible though not necessary economically justified tolerance window that current technology can achieve. Assuming that current technology allows to set the smallest feasible *KCC* tolerance windows for all *KCCs* at +/- 0.10 mm., the maximum potential process yield is 59.16 % as shown in Figures 1.5 (a) and 1.5 (b) (graph marked as current system).

In the case that all *KCC* tolerances are set below +/- 0.10 mm., process yield can approach 100%. However, we assume that the tolerances of *KCCs* below +/- 0.10 mm not only cannot be economically justified but currently are also technically infeasible, and it cannot be obtained with current design approaches. Thus, to enhance the process yield of the front-end assembly, the design nominal of the product and process architecture such as part geometry and fixture position have to be redesigned. This leads to the suggested development of a design synthesis methodology which is able to enhance the robustness of the system within the context of both complete and incomplete design information. The complete and incomplete information in this thesis is classified by the source of information used in design simulation. For example, the incomplete information is referred to the design scenario that the information is gained from intra-loop flow within a design synthesis phase shown in Figure 1.3. On the other hand, the complete information indicates that the design information is received from both intra-loop and inter-loop flow, and can be utilized during design synthesis phase.

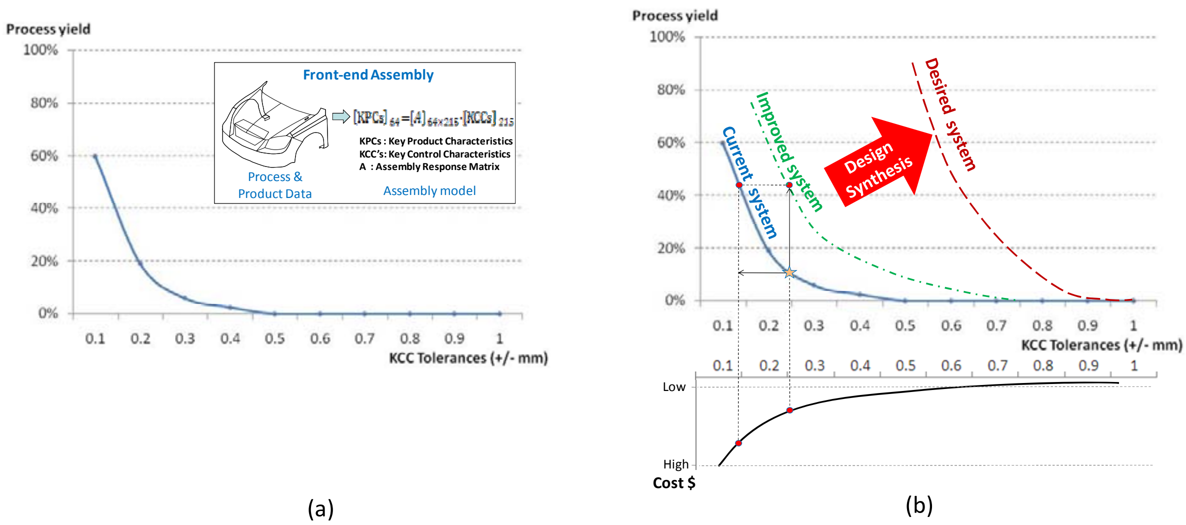


Figure 1.5: (a) Limitation of the current design approach in improving process yield of the front-end assembly; (b) the target of system performance.

1.3 Research Scope, Challenges, and Objectives

The research scope of this thesis is presented first in this section to define the general relation between product and process architecture, and impacts of engineering changes on the process architecture design. The challenges in adjusting the process architecture design are discussed. Then, the research objectives to address the challenges in designing the process architecture according to engineering changes are presented. The details of the research scope, research challenges, and the research objectives are presented in Sections 1.3.1, 1.3.2, and 1.3.3, respectively.

1.3.1 Scope of Research

In this thesis, the methodology in adjusting process architecture is the main focus. However, it is necessary to understand the relationship between product and process architecture design since it is usually interconnected. The development of the methodology in designing the process architecture has to consider the impacts on product architecture design since it can cause the design changes to each other.

The relation between product architecture and process architecture can be explained by using Axiomatic Design approach (Suh (1990, 1995)). The Axiomatic Design classifies the design of product and process architecture into four domains which are: (i) Customer Attribute (CA); (ii) Functional Requirements (FR); (iii) Design Parameters (DP); and (iv) Process Variables (PV). The product is designed to serve customer requirements defined in CA domain which become the criteria or guideline to design of product functionalities in FR domain. For example, dimensional quality of automotive body is defined as one customer requirements in CA domain. Then, the functional requirements to describe dimensional quality may be identified as, for example, parallel of gap between automotive body and doors or position/orientation of front-end. Next, the detail design of each functional requirement is elaborated in terms of design parameters in the DP domain. For instance, to characterize the parallel of gap between automotive body and door, the nominal positions of measurement points on the side frame and door are assigned as well as their tolerances or allowable specifications. Last, the process variables in the PV domain are designed in order to assemble parts and to ensure that the design parameters are met. For example, fixture positions and their tolerances designed to assemble doors to side frame are process variables, and gap between door and side frame is the design paramater. The *Key Product Characteristics (KPCs)* and *Key Control Characteristics* (*KCCs)* defined in this thesis can be mapped into Design Parameter domain and Process Variable domain, respectively. The Key Performance Indices (*KPIs)* to evaluate the performance of the process architecture design is equivalent to function requirement defined in the FR domain. These relations with Axiomatic Design can be structured as follow. In general, the product architecture consists of three components which are:

(i) Classification of *Key Product Characteristic* (*KPC*): The *KPCs* define design nominal of key design features which are crucial in assessing product functional performance. For example, *KPCs* may be referred to design featured measured on automotive body side frame and/or door assemblies which characterize the door fit gap variation (i.e., functional requirements of vehicle represented by dimensional quality of door fit process)

(ii) Hierarchical groups of *KPCs*: The *KPCs* are hierarchically distributed into the final product and all sequential subassemblies and parts. The models representing distribution of *KPCs* in product architecture is important in assessing impact of each *KPC* on the *KPI*.

(iii) *KPCs* design specifications (tolerances)

Similarly, the process architecture can be defined according to Ceglarek *et al.* (1994) which represents the process architecture as:

(i) Classification of *Key Control Characteristic* (*KCC*) points: The *KCCs* define design nominals of key design features of the product/subassemblies or parts which are used to control the process. For example, *KCCs* may be referred to design featured measured on automotive body side frame and/or door assemblies which characterize positions and type of fixture locators (locating layout), clamps and part-to-part joints which are used in each assembly station.

(ii) Hierarchical groups of *KCCs*: The *KCC* points are hierarchical distributed in all assembly stations. The models representing distribution of *KCCs* in process architecture is important in assessing impact of each *KCC* on the *KPCs*.

(iii) *KCCs* design specifications (tolerances).

The relationship between product and process architecture design is shown in Figure 1.6. The process architecture design change can be resulted from: (i) the engineering change on product architecture such as functional design or customer requirements changes; or (ii) the engineering design change on process architecture itself such as rearranging of the assembly line. First, the design changes can exert impacts on the current *KPC* design (e.g., changes in customer preferences require new *KPC* design) which consequently affect the existing *KCC* design and changes of process architecture. Thus, the process architecture design has to be changed to meet the new product architecture requirements. Second, the engineering changes can directly impact the current process architecture (e.g., tooling faults during manufacturing and production cause the need for adjustmen/redesign of *KCCs*) which can result in the deterioration of the process performance. The process architecture has to be adjusted in order to enhance the performance of the system. In this research, the design changes of the product architecture are assumed to be given (i.e., *KPC* nominal and *KPC* allowable specifications are given).

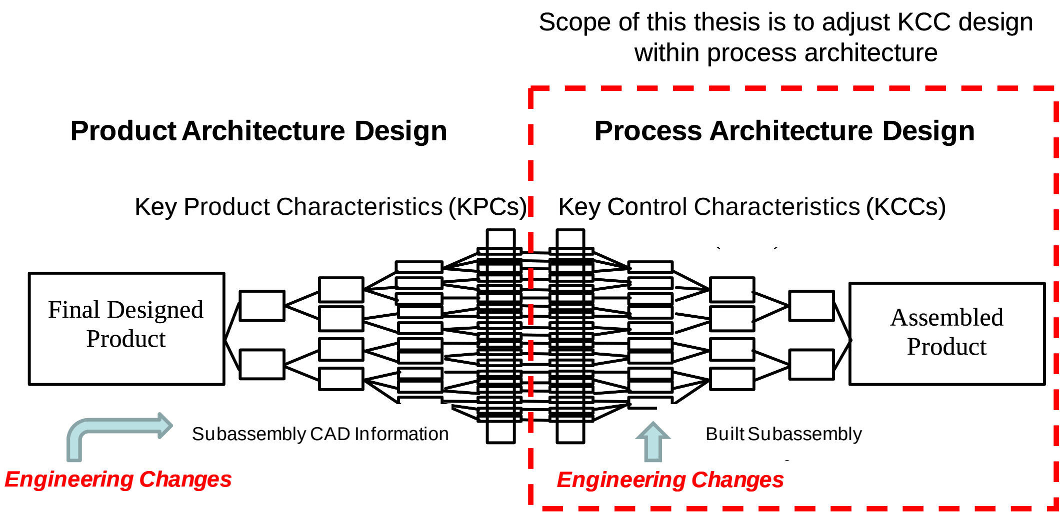


Figure 1.6: The relationship between product and process architecture design (Whitney 2004) and research scope of this thesis.

The scope of the proposed design methodology in this thesis focuses on the design of the process architecture only. The design changes regarding to product architecture, design constraints, functional requirements, and allowable specifications of functional requirements are given. The information of these design changes will be used by the proposed methodology to conducts design changes of the existing process architecture.

1.3.2 Research Challenges

As such optimizing the design of process architecture is challenging because of the complex interrelations that exist between the large number of parts and tooling elements within the product and process architecture. For example, Ceglarek and Shi (1995) reported that automotive body assembly involves around 60-100 assembly stations where 200-250 sheet metal panels are assembled, and these assembly stations consist of around 1,800-2,100 locators. The variations of these parts and locators affect the final dimensional quality of the automotive body which is characterized by 100-200 *KPCs*. Moreover, the nonlinear relations between *KCCs* and final product dimensional quality characterized by *KPCs* pose difficulties in determining the new optimal *KCC* design, e.g., fixture layout design and *KPC* variations. This poses challenges in terms of adjusting the process architecture design in single assembly station level and system level (i.e., multiple assembly stations).

In single assembly station level, the design optimization techniques such as tolerance optimization, fixture layout optimization, or assembly sequence optimization have to be developed in order to adjust process architecture effectively. On the other hand, when focusing on system level, the research challenges emerge in terms of integrating these design optimization techniques to ensure that design adjustments meet new requirements. The sequence on implementing design optimization techniques or iteration between design tasks can be time-consuming process or can be trapped in local optimal design solutions.

1.3.3 Research Objectives

There are three major research objectives presented in this thesis to address the aforementioned challenges in redesigning the process architecture. First, methodologies in developing the process architecture models are needed in order to assess the dependency of *KCCs* in the existing process architecture. Second, the development of *KPI*s is required to assess the impact of new requirements on the current process architecture design. The *KPIs* will provide the decision support whether to adjust the process architecture design. The *KPIs* must be covered robustness assessment, assembly cost, and difficulty in adjusting the current process architecture design. The *KPI* model will provide as-is functional performance of the existing process architecture and predict the to-be performance after process architecture adjustment. Third, the framework for integrating multiple design tasks to adjust the design of existing process architecture is required to achieve the new optimal process architecture design. The optimal design task sequence is determined to accommodate the engineering changes while the obtained design solutions meet the requirements of the selected *KPI* models and using the minimum computational time. The details of these objectives are presented as follows:

Objective 1: Process Architecture Modeling

The process architecture model is necessary in order to help in understanding the relationship or dependency among *KCCs* within process architecture as well as relationship with product architecture design. The dependency of *KCCs* significantly poses challenges in controlling design change propagation within the current process architecture design. The challenge of this research objective involves the difficulty in analyzing the design dependency among *KCCs*. The model leads to the development of methodology which is able to adjust process architecture design effectively. Therefore, this objective entails the development of a strategy to systematically organize the *KCC* in process architecture which allows determining the *KCC* dependency efficiently and is able to consider the impact of design changes from product architecture and design constraints.

Objective 2: Key Performance Index Development and Assessment

In order to make decisions whether to change the design of current process architecture, it is important to assess the impacts of engineering changes on the current process architecture. This objective involves the formulations of *KPIs* which consists of indices to assess the impact of engineering changes in terms of: (i) final product quality described by the *KPCs* variations; (ii) cost related to the planned adjustments; and (iii) the change propagation complexity in changing the process architecture design.

Quality *KPI* involves the development of index which is able to assess multiple *KPCs* simultaneously as well as correlation between *KPCs*. For the cost KPI development, it can be extremely challenging since cost can occur at any stage of a supply chain after conducting design change of process architecture. Thus, only cost which are relevant to design change of process architecture is selected in this thesis. The cost index of the *KPI* model in this thesis is the cost in changing *KCC* tolerances in order to maintain final product at the same quality level. Cost in adjusting *KCC* tolerances is usually lower than cost in changing *KCC* nominal design and *KCC* distribution in assembly system. The *KCC* nominal and *KCC* distribution design change usually require determining the new suitable *KCC* tolerances which become additional cost of *KCC* nominal and *KCC* distribution design change. Therefore, the cost index of the *KPI* model involves retaining the nominal *KCC* design and adjusting *KCC* tolerances in order to maintain the product dimensional quality. Thus, the cost index of the proposed *KPI* model entails the development of a methodology to determine the minimal cost in maintaining the final product quality at the same level if the engineering changes are implemented.

Finally, the challenges in measuring the process architecture complexity involves: (i) a large number of dependency relations among *KCCs*; and (ii) difficulty in defining quantitatively the strength level of *KCC* dependency. The complexity index of the proposed *KPI* model entails the development of index to measure the level of engineering change propagation in process architecture. The index formulation must be able to consider design dependency in two levels which are: (i) dependency between design tasks is resulted from sharing the same design objectives/constraints; and (ii) dependency between individual *KCCs* is caused by functional dependence.

Objective 3: Methodology Development for Adjusting the Existing Process Architecture Design

To change the design of process architecture, it can be very challenging since the design adjustment by one design task can be dependent or affect others design tasks. This poses difficulty in adjusting the process architecture by several design tasks. Therefore, this objective entails the development of methodology to integrate multiple design tasks while minimizing the number of iterations among design tasks.

1.4 Organization of Thesis

In order to lead to the proposed design methodology addressing the aforementioned research objectives as well as differentiating the research contributions, the related research studies are reviewed and their challenges to address the aforementioned research objectives are presented in Chapter 2. The proposed methodology is presented in Chapter 3, 4, and 5. Figure 1.7 illustrates the thesis structure.

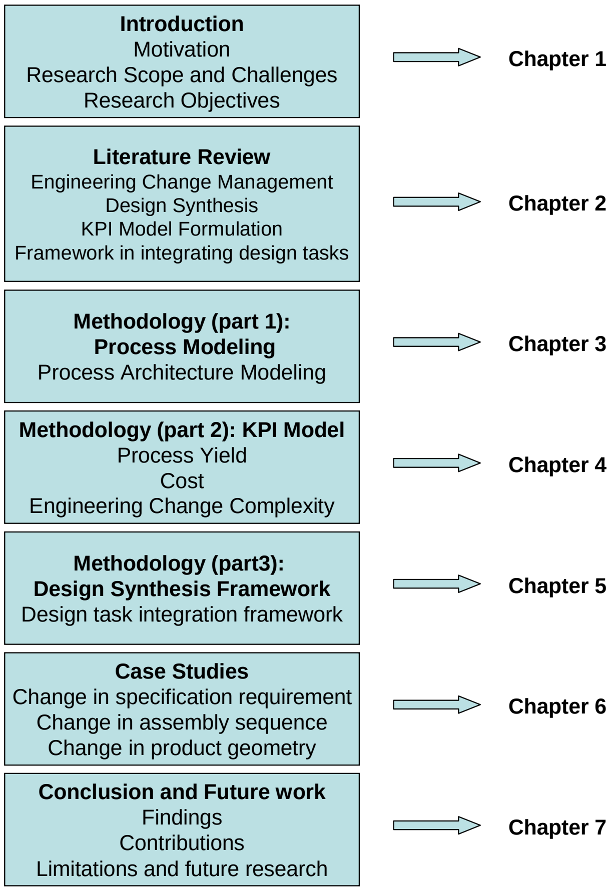


Figure 1.7: A brief outline of the thesis

Chapter 3 presents the methodology to describe the dependency within the process architecture (i.e., design task and *KCC* functional dependency). The methodology involves the process architecture partitioning based on functionality criteria to formulate the *Key Characteristic Groups* (*KCC-HGs*). Each *KCC-HG* consists of the *KCCs* from all assembly stations which have the same functionality such as fixture group or part-to-part joint group. Then, the design task or the optimisation algorithm for each *KCC-HG* is developed to optimise *KCC-HG* design. The dependency of between design tasks, called *Task Connectivity Matrix*, is analyzed by using hypergraph technique and sensitivity analysis approach. Finally, the dependency among individual *KCCs* can be modeled by using hypergraph techniques and functional design dependency analysis.

Chapter 4 introduces the *KPI* model to assess the impacts of engineering changes on the current process architecture. The first index of the proposed *KPI* model is process yield which is used for quantifying dimensional quality of produced product. The process yield is the joint probability that all *KPCs* are being within allowable specifications simultaneously. The impacts of engineering changes can be illustrated by the change of the process yield. The second index of the proposed *KPI* model is cost in maintaining the process architecture performance to produce product at the same dimensional quality. The cost is obtained from conducting *KCC* tolerance optimisation. Next, the third measure is the engineering change complexity index which is the indicator presenting degree of *KCC* dependency and number of *KCC* involving in dependency chain in the current process architecture. By based on the process yield, cost, and engineering change complexity indices, engineers can make the decision on changing the design of the current process architecture.

Chapter 5 introduces the design synthesis framework to integrate multiple design tasks and to generate the design task sequences which avoid the iteration among design tasks. The *transversal of hypergraph* and *task connectivity matrix* presented in Chapter 2 are used to determine the dependency between design tasks. Also, the Design Structure Matrix and Task Flow Chain proposed by Phoomboplab and Ceglarek (2007) are adopted in order to enhance visibility on the dependencies of multiple design tasks simultaneously and to generate the task sequences represented by design task hierarchy.

Chapter 6 presents the examples in adjusting the design of automotive underbody in three different engineering change scenarios. These engineering change scenarios involve: (i) *KPC* specification changes; (ii) assembly sequence changes; and (iii) part geometry changes. The optimal design task sequences are identified in each scenario.

Finally, in Chapter 7 the research contributions of the thesis are concluded and the future research is discussed for opportunities for improvement.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In general, the engineering changes can be classified into four categories, which are (i) production change; (ii) adjustment of production system; (iii) design change of product; and (iv) design change of process architecture. The production change is related to the modification of production line or equipment to produce different parts within a given product family. The adjustment of production system involves the error or fault correction during PLM production phase. The design change of product is related to the changes of product design in any PLM phase. Last, the design change of process architecture is the changes of *KCC* configurations, distributions of hierarchical groups of *KCCs*, and *KCC* tolerances during PLM design phase.

Several studies have been conducted to address the challenges in the adjustment of production system and production change in order to maintain production performance in an uncertain environment. For example, reconfigurable manufacturing system (RMS) has been introduced by many studies to address the uncertainty of capacity demand, product variety, and technology change (Koren and Ulsoy (1997); Koren *et al.* (1999); Mehrabi *et al.* (2000); ElMaraghy (2005). RMS aims to (i) reduce lead time in launching the new production system; and, (ii) upgrade and integrate the new process technology responsively. This increases the flexibility to produce the variety of parts as well as towards adjusting the production system.

The flexibility in RMS is established by creating and using the basic process modules both in terms of hardware and software. Thus, RMS becomes the open-end system which can be upgraded by integrating the new process technologies or can be reconfigured as a way to respond to the changes to future products. There are five key performance indicators (ElMaraghy (2005)) that are involved in developing RMS, namely, (i) modularity, (ii) integrability, (iii) convertibility, (iv) diagnosability, and (v) customization. These indices tend to provide the advantages towards the ramp-up phase of production system rather than the design phase of the product and process architecture of which the robustness of the system is one of the primary key performance indicators.

On the other hand, there are also several studies which are related to the design changes of product and process architecture under uncertain environment (e.g., Eckert *et al.* (2004), Rutka *et al.* (2004), Jaratt *et al*. (2002), Giffin *et al.* (2007), Clarkson *et al.* (2004), Riviere *et al.* (2003), Jin and Shi (1999), Mantripragada and Whitney (1999), Ding *et al.* (2000), Huang *et al.* (2009), Phoomboplab and Ceglarek (2008), Phoomboplab *et al.* (2009), and Lee and Saitou (2003)). These studies can be categorized into two major groups which are: (i) engineering change management; and, (ii) design synthesis. In general, the studies in engineering change management aim to formulate a model which represents the interrelation between product and process architecture. By using the formulated model, the adjustment of product and process architecture can be performed efficiently and the engineering change propagation can be controlled. However, the studies in engineering change management require intensive information about product and process architecture as well as historical engineering change data in order to predict the impact and create the proper approach for adjusting the design. This poses limitations towards applying the model of engineering change management on industrial applications because (i) it requires significant time to build a database while the technologies in product and process design are being constantly updated; (ii) there might be only a few relevant cases which are insufficient to formulate the model of interest; and, (iii) it requires significant investment to manage the change records effectively. In addition, current studies in modeling engineering change propagation focus mainly on managing the changes in a product architecture where the impacts of engineering changes on assembly process architecture are often overseen. In turn, this leads to insufficient information for decision-making involving making the necessary engineering changes since the impact of process architecture on product quality, cost in modifying the process architecture and technical difficulty in adjusting the process architecture are not taken into consideration. Moreover, most of the current studies in adjusting an assembly process can be considered as station-level adjustment which the system-level adjustment to ensure optimal design is rare. Therefore, the integration of individual assembly stations cannot guarantee the quality of the final product since the interrelations of process architecture between assembly stations are ignored. The related studies in engineering change management are elaborated in Section 2.2.1.

In the recent years, several studies in *design synthesis* have been introduced to optimise the design of product and process architecture (e.g., Jin and Shi (1999), Mantripragada and Whitney (1999), Ding *et al.* (2000), Huang *et al.* (2009), Phoomboplab and Ceglarek (2008), Phoomboplab *et al.* (2009), and Lee and Saitou (2003)). Instead of depending on historical change records, the design synthesis is based on simulation approach and formulated assembly response function model to determine the impacts of product and process architecture design changes on the final product functional requirements. Specifically, there are many studies on design synthesis that aim to enhance dimensional quality of a product produced on a multi-stage assembly system (e.g., Shiu *et al.* (2003), Ding *et al.* (2005), Kim and Ding (2004), Camelio *et al.* (2004), Ceglarek and Shi (1998), Phoomboplab *et al.* (2009), Lee and Saitou (2003), Wang and Ceglarek (2008)). These studies focus on optimizing various functional requirements of the multi-station assembly system architecture such as *KCC* tolerancing (e.g., Shiu *et al.* (2003), Ding *et al.* (2005), and Huang *et al.* (2009)), fixture layouts (e.g., Kim and Ding (2004), Camelio *et al.* (2004), Izquierdo *et al.* (2006), and Phoomboplab and Ceglarek (2008)), part-to-part joints (e.g., Ceglarek and Shi (1998), and Phoomboplab *et al.* (2009)), and/or assembly sequences (e.g., Lee and Saitou (2003), and Wang and Ceglarek (2008)). The aforementioned design synthesis methodologies which separately optimise various functional requirements can also be classified as *individual design tasks*. However, it usually involves several design tasks to adjust process architecture design. It still lacks of an approach which will allow integrating and simultaneously selecting the sequence of optimizing individual design tasks. The *design synthesis framework* for integration of individual design tasks and then selection of tasks sequence is necessary due to the fact that each individual design tasks present different level of "coupling" with other design tasks what directly affects the propagation of design changes in process architecture redesign and might lead to local optimal solutions.

In the next section, the related studies of engineering change management and design synthesis are elaborated in section 2.2.1 and 2.2.2, respectively. In section 2.2.3, the related work in integrating multiple design tasks to adjust process architecture design is discussed. Finally, the literature review for *KPI* model is presented in section 2.2.4.

2.2 Literature Review

2.2.1 Related Work in Engineering Changes Management

In this section, the related studies in managing engineering changes in product and process design are discussed. Eckert *et al.* (2004) classified engineering changes into two types according to the causes of changes: (i) *initiated changes*; and (ii) *emergent changes*. The *initiated changes* are caused by external factors of product and assembly process design. The examples of the initiated changes are new customer requirements resulting in new *KPC* specifications or manufacturing standard changes. On the other hand, *emergent changes* have resulted from design problems that occur during product and assembly process design or the product failures reported by customers.

To analyze and predict the impact of engineering changes, the relations between all parts and assembly process components are modeled to describe the engineering change propagation characteristics within the product and process architecture. The relations between *KCCs* and their effects on *KPC* variations can be described both qualitatively and quantitatively (Rutka *et al.*, 2004). Most of the current research studies in engineering change managements are based on a qualitative approach since the impacts can be assessed faster than by applying a quantitative approach in a complex product such as an aircraft assembly. Although the qualitative approach helps in visualizing the dependencies, the impacts of changes are difficult to quantify. Thus, it poses a challenge in evaluating the improvement of functional requirements which require numerical assessment indices representing for example six-sigma dimensional quality or cost. The current research studies based on the qualitative approach includes: (i) Change Propagation Analysis (CPA); (ii) Change Prediction Method (CPM); (iii) Collaborative Management of Engineering

Changes (CM-EC); and (iv) Design Structure Matrix (DSM). On the other hand, quantitative approaches are based on using the physical parametric information such as part geometry and coordinates of tooling elements in an assembly process to describe the relation between *KCCs* and *KPCs.* Therefore, the impacts of engineering changes on *KCCs* and *KPCs* can be simulated and quantified. However, the current engineering management methods within the quantitative area require comprehensive information about the product which can be computationally intensive and time-consuming. The methodologies categorized in the quantitative approach consist of: (i) *quantitative requirement traceability*; (ii) *TIES: Technology Identification, Evaluation, and Selection*; (iii) *C-FAR: Change Favorable Representation*; and (iv) *CAD: Computer-Aided Design software*. The description of the aforementioned research studies in engineering change management is summarized below.

Change Propagation Analysis, CPA, has been introduced to predict and simulate the impacts of engineering changes in order to improve quality and reduce time and cost (Jaratt *et al*., 2002). Eckert *et al.* (2004) presented a comprehensive description of CPA involving the characteristics of engineering changes in complex product design. In a similar vein, Giffin *et al.* (2007) presented the change propagation analysis in electronic equipment design based on engineering change records. Rutka *et al.* (2006) presented the CPA approach by defining the types of changes as well as the levels of changes which represent the amount of rework required from the current design setting. To predict the impacts of engineering change accurately, a type of change as well as a level of change have to be defined precisely which is usually based on past experience or engineering change records reported in the past. In addition, the CPA approach tends to focus mainly on product design instead of both product and assembly process designs.

Clarkson *et al.* (2004) presented the *Change Prediction Method*, CPM, which is based on the integration of Design Structure Matrix (DSM) and risk management techniques. The dependencies between components in the system are described by DSM and the scale of change propagation between components is expressed in terms of the likelihood of changes and the impact of the changes, i.e., risk level. CPM has a limitation in that the level of risk is determined based only on the amount of rework and probability of impacts on cost. Moreover, in order to minimize the complexity of the model in representing the product, Clarkson *et al.* (2004) suggested limiting the number of elements in the model to fewer than 50 components. This limits the applicability of CPM for redesign of industrial assembly systems which usually involves hundreds of *KCCs*.

Riviere *et al.* (2003) presented *Collaborative Management of Engineering Changes*, CM-EC, based on six types of dependencies between elements in the system, namely; association, composition, dimension, functional, location, and organizational dependencies. CM-EC offers a visual representation of the change propagation and can help designers to understand the relation of elements in the system with different types of dependencies. However, the CM-EC approach relies on the designers experience and expertise of a particular system in defining the dependencies.

Steward (1981) introduced *Design Structure Matrix*, DSM, which can be used to represent the relationship between design requirements for product design and assembly process design. The heuristic procedures to reduce the complexity resulted from sub-system interdependencies are also proposed. The DSM approach provides a fundamental technique in modeling product and assembly process which allow studying engineering change propagation. Similarly, Eppinger *et al.* (1994) and Browning (2001) presented the DSM hybrid model where the relations of design activities and parameters are defined in the same matrix. This is a viable model as it integrates design activities and can identify the parameters that are coupled within each sub-system. However, the DSM approach does not indicate the scale of system redesign that result from engineering changes.

On the other hand, the quantitative approaches are used to analyze the impacts of engineering changes. Sutinen *et al.* (2002) introduced *quantitative requirement traceability* to quantify impacts of modifications of product-parts on functional requirements during the early design stage of a product development. The model representing the relation between functional requirements and product definition such as part geometry and life cycle of product is formulated based on the response surface equations. Thus, the impacts of engineering changes on functional requirements of interest can be quantified by using the developed response surface function. Based on the response surface model, several impacts of engineering changes can be simulated and the trade-off analysis among the engineering changes options can be conducted. However, this approach has proven to be time-consuming and very expensive particularly when developing and testing prototypes in order to formulate the response surface equations. Additionally, Kirby and Mavris (1999) introduced the *Technology Identification, Evaluation, and Selection, TIES,* to assess the trade-off among design options during the preliminary design stage of a complex product such as an aircraft design. *TIES* involves a mapping of the customer requirements and quantitative evaluation criteria which are used for selecting the list of technologies which potentially fulfills the requirements. Next, the technical feasibility of the technology is investigated based on the Response Surface Methodology and Fast Probability Integration techniques. If the list of the initial selected technologies cannot achieve customer requirements, the alternative design has to be designed and the analysis is repeated. However, the dependencies among the selected technologies defined as binary (i.e., existing or non-existing relations) is insufficient to predict the impacts of engineering changes in the detail.

Engineering change management can also be found in the area of computer-aided design. Cohen *et al.* (2000) presented *Change Favourable Representation, C-FAR,* to trace and predict change propagation. *C-FAR* represents the dependencies among individual components in the system in three levels (i.e., high, medium, and low) which is suitable for simple product design. Solid modeller in CAD commercial software tools such as CATIA or Siemens NX can be used to visualize the engineering change propagation on product design. In addition, software tools such as 3DCS Analyst and Vis-VSA embedded in CATIA and Siemens NX allow analyzing the dimensional variations of a given assembly process design. However, the CAD software tools are only limited to check the geometric compliance to GD&T of implemented engineering changes. Moreover, these software tools are incapable of capturing the interrelations of individual design tasks such as tolerance optimisation, fixture layout optimisation, part-to-part joint selection, and assembly sequence analysis and selection. The summary of the aforementioned research studies in engineering change management is shown in Table 2.1.

The methodology for redesigning an assembly system introduced in this thesis can be classified as a quantitative approach. The synthesis of individual design tasks for a given change requirements is proposed by taking into consideration both product and process information such as (i) geometric and dimensional information about parts and subassemblies represented by a set of *KPCs*, (ii) geometric and dimensional information about control points in all assembly stations represented by a set of *KCCs*, and (iii) tolerances of both *KPCs* and *KCCs*. The sequence in implementing individual design task to adjust and optimise the design of process architecture is performed (e.g., optimisation of fixture locator layout positions, part-to-part joint types and assembly sequences). The design tasks used in adjusting design are developed based on stochastic optimization and statistical analysis such as Monte Carlo simulation. The related researches classified as quantitative approach and applicable for both product and process architecture design are elaborated in the next section.

Table 2.1: Related researches in engineering change management.

		Design consideration domains	
		Product design	Product and Assembly process design
Analysis and prediction of engineering change impacts	Qualitative approach	Jaratt *et al.* (2002) Eckert *et al.* (2004) Giffin *et al.* (2007) Rutka *et al.* (2006) Clarkson *et al.* (2004) Riviere *et al.* (2003)	Steward (1981) Eppinger *et al.* (1994) Browning (2001)
	Quantitative approach	Sutinen *et al.* (2002) Kirby and Mavris (1999) Cohen *et al.* (2000)	Design Synthesis (Section 2.2.2) ***Proposed in this thesis***

2.2.2 Related Work in Design Synthesis for Dimensional Management

In recent years, several studies have endeavored the design of product and process architecture to optimise dimensional quality management (e.g., Jin and Shi (1999), Shiu *et al.* (2003), Kim and Ding (2004), Ceglarek and Shi (1998), Wang and Ceglarek (2008)). These studies address challenges involving a large number of *KPCs* and *KCCs* in a multistage assembly system can be classified as quantitative approach.

Each individual design task focuses on optimizing the design of process architecture such as *KCC* tolerancing, fixture layouts, part-to-part joints, and assembly sequences. However, the current design approaches have two limitations. First, design synthesis is limited to optimizing independently each individual design tasks what does not guarantees to reach the global optimum of functional requirements as shown in Section 1.2. Second, the design changes suggested by optimizing one design task can affect the other functional requirements which can impact the design of other functions. As a result, the improvement of the overall system cannot be ensured. In this section, the related work in the area of design synthesis for dimensional management is reviewed as discussed below.

The current design approaches in improving the final product dimensional variation during early design phase can be classified into two groups namely: (i) Design Analysis; and (ii) Design Synthesis. The descriptions of these two design approaches are elaborated below.

(i) *Design Analysis*: individual *KCCs* in assembly processes are designed and their impacts on *KPC* variations are assessed using the vector loop technique or variation analysis software tools such as 3DCS Analyst from 3DCS Company, Vis-VSA from Siemens PLM Software, or CETOL 6 Sigma from Sigmetrix. If the *KPC* variations are out-of-specification requirements, some of *KCCs* such as fixture locator layouts or part-to-part joints as well as *KCC* tolerances are required to be reconsidered. These adjustments are usually conducted by experienced designers. In complex products such as an automotive body or an aircraft fuselage, the process adjustment can be time-consuming and involve significant cost of engineering changes. Figure 2.1 illustrates the Design Analysis approach.

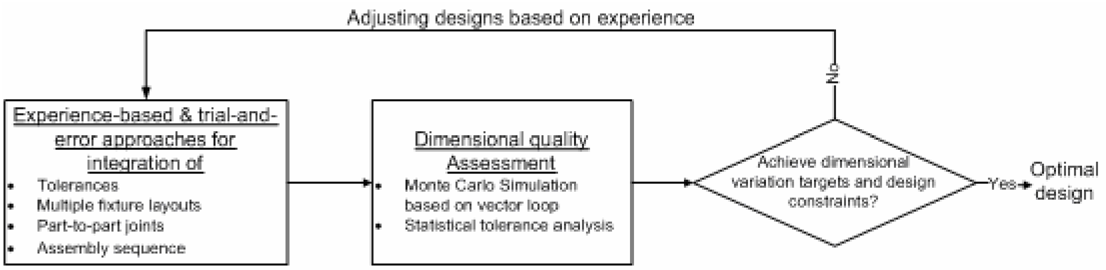


Figure 2.1: Design Analysis approach in current product and process design.

(ii) *Design Synthesis*: Instead of analyzing the *KPC* variations based on given *KCCs*, the design synthesis is to determine the optimal *KCCs* which allow achieving *KPC* Six-sigma requirements subject to given design constraints. Design synthesis can be extremely helpful in terms of time and cost in designing a new assembly process and is less dependent on experience. In general, design synthesis consists of three crucial elements which are: (i) automatic generation of assembly response function (a variation prediction model); (ii) design tasks for optimizing product and process design such as optimisation of tolerances, fixtures, and part-to-part joints; and (iii) the framework for multiple design task integration. The automatic generation of a variation prediction model is imperative for a design task development since all design candidates selected by a design task have to be evaluated for their impacts on final product dimensional quality during iteration search. Figure 2.2 illustrates the Design Synthesis approach.

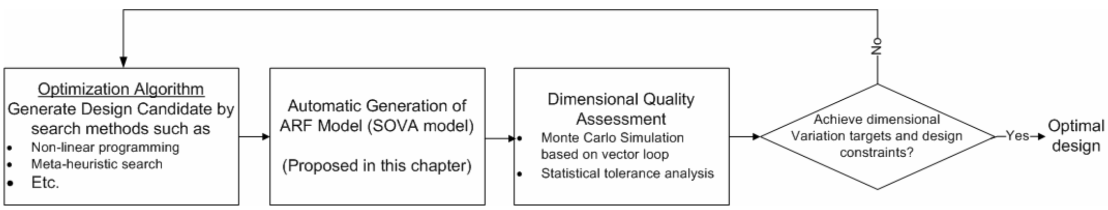


Figure 2.2: Design synthesis for product and process optimisations.

The individual design task consists of three components which are: (i) assembly response function (*ARF*) model; (ii) *KPI* model selected as a specific objective function of an individual design task; and (iii) optimisation algorithms to minimize/maximize the selected *KPI* model. The current design synthesis tasks present two challenges which lack of algorithms for: (i) automatic generation of *assembly response function (ARF)* model. The automatic generation of the ARF model is necessary to support design task optimisation which require automatically running multiple iterations each with different *ARF*; and (ii) design synthesis framework which will allow integrating and simultaneously selecting the sequence of optimizing individual design tasks.

In the design synthesis for dimensional management, the development of variation prediction model serves as the assembly response function model to evaluate dimensional quality. The design tasks in optimizing the process architecture design such as *KCC* tolerancing, fixture layouts, part-to-part joints, and assembly sequences are using different *KPI* models in assessing the impact of process architecture design on the dimensional quality as well as optimisation algorithm. The current research in design synthesis for dimensional management is presented as follow.

Several studies exist in developing the model to predict *KPCs* variation in a multi-stage assembly process. Jin and Shi (1999), and Mantripragada and Whitney (1999) adopted the State Transition Model approach from control theory to integrate homogeneous transformations in order to describe the dimensional variation propagation in an assembly process. Furthermore, Ding *et al.* (2000) developed a 2D state-space model for modeling variation propagation in a multi-stage sheet metal assembly process where the variations of fixtures are approximated into linear explicit functions. However, the variation propagation model proposed by Ding *et al.* (2000)

is limited to fixture layouts for a 2-D prismatic workpiece. Recently, Huang *et al.* (2007a, b) introduced a Stream-of-Variation (SOVA) model to predict dimensional variation in 3D multi-stage assembly processes. The SOVA model is formulated based on point-based geometric constraint models which consider both part-to-part joints and fixture locators. Next, the variation propagation model is developed by defining the relationships among virtual mating points and fixture locating points on the variation of *KPCs*. The SOVA model is represented in the explicit math-based model as shown in Eq. (2-1).

$$\begin{bmatrix} KPC_1 \\ KPC_2 \\ \vdots \\ KPC_m \end{bmatrix}_{m\times 1} = \overbrace{\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1,n} \\ c_{21} & c_{22} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{bmatrix}_{m\times n}}^{\text{SOVA Matrix}} \begin{bmatrix} KCC_1 \\ KCC_2 \\ \vdots \\ KCC_n \end{bmatrix}_{n\times 1} \quad \text{or } \mathbf{y} = \mathbf{A}\mathbf{u} \qquad (2\text{-}1)$$

where $c_{i,j}$ is a constant value based on the nominal design of assembly processes.

Nevertheless, the SOVA methodology (Huang *et al.*, 2007a, b) has two limitations: (i) modeling an assembly process which involves only fixture layouts for prismatic workpiece (3-2-1 fixture layout); and (ii) modeling variation propagation through specific types of part-to-part joints. To address the challenges of the SOVA methodology proposed by Huang *et al.* (2007a, b), Phoomboplab and Ceglarek (2009) presented the generalized SOVA model approach which allows modeling an assembly process consisting of both 3-2-1 and non 3-2-1 fixture layouts. The generalized SOVA model adapted the methodologies in predicting variations of machining process proposed by Cai *et al.* (1997) and Loose *et al.* (2007) together with homogenous transformation techniques to formulate a generalized SOVA model. The methodology is based on determining workpiece constraint condition and location by using a Jacobian matrix to analyze infinitesimal errors of contact points between locators and part geometry. The developed model is able to address the challenges of the generalized 3D fixture layouts which are not limited to a prismatic workpiece in an assembly system.

However, the development of the individual design task requires the automatic generation of the *ARF* model. Specifically for the developments of design tasks such as fixture layout, part-to-part joint, or assembly sequence optimisations, it is necessary to automatically generate the variation prediction model when the design candidates are changed during the search iteration. This thesis addressed this need by proposing a software tool, called AutoSOVA Model Generator, to automatically formulate a SOVA model. The AutoSOVA Model Generator is integrated with an optimisation algorithm of a design task. Then, the SOVA model associated to each process architecture design candidate can be obtained and is used to evaluate for selected *KPI* model.

Several design tasks have been proposed to improve the final product dimensional quality by designing a multi-stage assembly process to be robust to variation sources and satisfying design constraints. The design tasks for dimensional management can be categorized into four groups which are: (i) process-oriented tolerance optimisation; (ii) multi-fixture layout optimisation; (iii) part-to-part joint selection; and (iv) assembly sequence analysis and optimisation.

(i) Process-oriented tolerance optimisation is used to determine the optimal *KCC* tolerances of both part critical geometry and tooling element in an assembly process while the traditional tolerance optimisation focus on determining optimal tolerances on product architecture only. In addition, the process-oriented tolerance optimisations can optimise *KCC* tolerances at multi-station levels. Therefore, process-

oriented tolerance optimisation that considers both *KPC* and *KCC* tolerances can provide the optimal tolerances which are able to achieve final product dimensional quality requirement and satisfy cost requirements. The current research studies in process-oriented tolerance optimisation can be found in Shiu *et al.* (2003), Ding *et al.* (2005), and Huang *et al.* (2009).

(ii) Multi-fixture layout optimisation is to determine a set of fixture layouts which are insensitive to variation sources. The multi-fixture layout optimisation is used to optimise the fixture positions in all assembly stations simultaneously. This is different from the current practice wherein a fixture layout is optimised in each assembly station separately. The integration of individually optimised fixture layouts cannot guarantee the optimal fixture layouts at the system level. Kim and Ding (2004) presented the methodology in optimizing fixture layouts for rigid part assembly. Camelio *et al.* (2004) proposed the fixture layout optimisation for compliant part assembly. Izquierdo *et al.* (2006) introduced the fixture layout optimisation for a product family. However, these multi-fixture layout methodologies are limited to 2D assembly processes. Recently, Phoomboplab and Ceglarek (2008) presented the multi-fixture layout optimisation for 3D rigid part assembly taking into consideration part stability provided a fixture layout design.

(iii) Part-to-part joint selection is generally used to determine a set of part-to-part joint which can minimize the variation propagation in the assembly process. The challenge of part-to-part joint selection involves the interdependency between part-to-part joint and fixture planning in constraining degrees of freedom of parts in an assembly process. Therefore, part-to-part joint selection can be considered as the combinatorial optimisation with multi-fixture layout optimisation. Phoomboplab *et al.* (2009) presented the methodology to formulate the geometric constraint model which

can help in allocating the constrained degrees of freedom between a part-to-part joint and a fixture layout as well as modeling variation propagation through different types of part-to-part joints. The geometric constraint model can lead to the determination of optimal part-to-part joints and their associated optimal fixture layouts.

(iv) *Assembly sequence analysis and optimisation* helps to determine the assembly sequence which can minimize the *KPC* variations on the final product. To determine the assembly sequence, it involves various aspects of product architecture design which impact the subsequent decisions of an assembly process design. For instance, the assembly sequence has to be feasible and parts are assembled without the interferences among parts. Here, the accessibility of tooling elements has to be considered. Design of assembly sequence also poses the constraint in designing fixture layouts. Several efforts have been conducted to address the aforementioned challenges of assembly sequence analysis and optimisation. These efforts, for example, can be found in Lee and Saitou (2003) and Wang and Ceglarek (2005).

Although the aforementioned studies in design synthesis for dimensional management in a multistage assembly system can provide optimal design solution for particular areas, there still lacks a framework which can help to integrate multiple design tasks to optimise the design of process architecture in order to accommodate the engineering changes and achieve design solution in system level. Phoomboplab and Ceglarek (2007) proposed the methodology to integrate design tasks by using Hybrid-Design Structure Matrix and Task Flow Chain to represent hierarchy of design tasks diagram. However, it still lacks of systematic approach in analyzing the dependency between design tasks. To address this challenge, this thesis proposes a comprehensive methodology based on transversal of hypergraph and task connectivity matrix in analyzing the dependency between design tasks. The DSM and Task Flow

Chain are used only for graphical representation in generating design task sequences. In the next section, the related studies on multiple design task integration are discussed.

Table 2.2: Assembly response function model and individual design task for dimensional management in multi-stage assembly processes.

Research area	Methodology	Research in design for dimensional management in multi-stage assembly processes
Assembly Response Function	Variation prediction model	Jin and Shi (1999) Mantripragada and Whitney (1999) Ding *et al.* (2000) Zhou *et al.*(2003) Camelio *et al.* (2003) Shi (2006) Huang *et al.* (2007a,b)
	Automatically generation of variation prediction model	Phoomboplab and Ceglarek (2009)
Individual design task developments	Process-oriented tolerance optimisation	Shiu *et al.* (2003) Ding *et al.* (2005) Huang *et al.* (2009)
	Multi-fixture layout optimisation	Kim and Ding (2004) Camelio *et al.* (2004) Izquierdo *et al.* (2006) Phoomboplab and Ceglarek (2008)
	Part-to-part joint selection	Ceglarek and Shi (1998) Phoomboplab *et al.* (2009)
	Assembly sequence analysis and optimisation	Lee and Saitou (2003) Wang and Ceglarek (2008)
Framework for integrating multiple design tasks	Experience-based design task dependency analysis	Phoomboplab and Ceglarek (2007)
	Transversal of hypergraph in analyzing design task dependency	**Proposed in this thesis** **(The related literature review on design task integration is described in Section 2.2.3)**

2.2.3 Related Work in Integration of Multiple Design Tasks

In this section, the related studies in area of design task integration or multidisciplinary design activities are discussed. Design-for-Manufacturing-and-Assembly (DFMA) approaches (Boothroyd, G. (1994), Selvaraj *et al.* (2009), and Cutkosky and Tenembaum (1990)) are introduced for reducing time and cost during product development. DFMA-based approaches are used in many industries such as automotive and aerospace industries wherein variety of parts and system involve. Basically, DFMA focuses on minimizing assembly parts by providing heuristic guidelines that can reduce a number of potentially unnecessary *KPC*s and *KCC*s. DFMA also considers tradeoffs between additional criteria to determine the best set of *KCC*s such as manufacturability, reliability, and maintainability. DFMA is a qualitative-based design synthesis approach and relies heavily on past experience and therefore, posses limited capability to develop new design (Cutkosky and Tenembaum (1990)).

Computer Aided Design/Manufacturing/Engineering (CAD/CAM/CAE) software have been introduced to mitigate some of the challenges involving time and cost expenses in an effort to visualize dimensional and geometric relations and to simulate manufacturing processes before building and testing the prototypes. CAD/CAM/CAE allows the multiple design disciplines can collaborate in product and process design. Although designers can perform *what-if* scenarios in CAD/CAM/CAE environment, the adjustments of design configurations are still based on design experience. On the other hand, statistical approaches have also been introduced into the prototyping and testing stages in order to improve product design.

On the other hand, Quality Function Deployment (QFD) (El-Haik 2005)) and Axiomatic Design (AD) (Suh (1990, 1995, 1997)) are widely used to integrate design activities and information flow in a product design development. The product functional requirements are analyzed by using QFD technique, and a set of *KPCs* are extrapolated that can satisfy all predefined functionality as described by customer needs. However, the applications of QFD are limited on the early stage of conceptual design where the identified set of *KPCs* is then used in subsequent design stages. In a similar vein, Suh (1990, 1995, 1997) introduced the concept of Axiomatic Design, AD, for product and process design during conceptual design stage. AD provides a systematic approach to generate and refine a set of *KCCs* that can respond to predetermined *KPCs*. The AD approach is based on a top-down design process wherein design solutions are generated by mapping the relationships between four design domains: Customer Attributes (*CAs*); Functional Requirements (*FRs*); Design Parameters (*DPs*); and, Process Variables (*PVs*). The design solutions are then evaluated and selected based on two design criteria defined as independence axiom and minimum information axiom. The extension of AD with quality engineering can be found in El-Haik (2005). However, AD represents mainly on the existence of the interrelationship between parameters in four domains rather than the quantified level of the interdependencies. Therefore, this poses the difficulty in conducting the detail design where the functional requirements have to be evaluated by numerical indices.

Design Structure Matrix (DSM) is another research area which is widely used for organizing multidisciplinary design activities (Steward (1981), Eppinger *et al*. (1994), and Browning (2001)). Steward (1981) proposed the DSM to address these challenges by using a matrix to represent the interdependencies of design tasks as well as heuristic procedures to reduce the complexity caused by these interdependencies. In similar vein, Eppinger *et al*. (1994) and Browning (2001) presented the DSM hybrid model where the relations of activities and parameters are defined in the same matrix. This is a viable model as it integrates design tasks and can identify the parameters that are coupled within each task. However, the design tasks represented in the DSM approach are developed based on experience of the design team and best design practices in the past.

To address the need of quantitative evaluation, Design of Experiment (DOE) – based approaches, including Design-for-Six-Sigma (DFSS) and Taguchi's methods, are used to statistically model and analyze a specific functional requirement (e.g., Fowlkes and Creveling (1995), Box *et al*. (2005), and Montgomery (2009)). DOE-based approaches have also been introduced into the prototyping and testing stages in order to capture the responses of the system at different setting conditions. The responses are used to formulate the explicit model between functional requirements of interest and *KCCs*. The DOE-based approaches focus on determining *KCCs* that have the most significant effects on product design objectives and then formulate the analytical model to optimise their configurations. However, conducting DOE-based approaches can be very expensive and time-consuming for complex products represented by large number of *KCC*s (Box *et al*. (2005)). Therefore, DOE approach can function effectively in the design problem which involves limited *KCCs*. In addition, there lacks of the methodology to integrate multiple DOE models to characterize the system functionalities.

Phoomboplab and Ceglarek (2007) proposed the methodology to integrate design tasks (e.g., the design task for dimensional management discussed in Section 2.2.2) by using Hybrid-Design Structure Matrix and Task Flow Chain. The DSM and Task Flow Chain introduced by Phoomboplab and Ceglarek (2007) are very useful for graphical representation of design task hierarchy when several design tasks are involved. However, it still lacks of clear approach in analyzing the dependency between design tasks.

To address this challenge, this thesis proposes a comprehensive methodology based on transversal of hypergraph and task connectivity matrix in integrating several design tasks. The details of hypergraph technique and task connectivity matrix are introduced in process architecture modeling presented in Chapter 3 of this thesis. The DSM and Task Flow Chain introduced by Phoomboplab and Ceglarek (2007) are also adopted for graphical representation of design task hierarchy. The proposed framework can help to address the challenges of the current research studies by providing the quantitative design solution in the system level. The design synthesis framework functions as the high level management in considering the dependency between design tasks and analyzing the optimum design task sequence. The individual design task is autonomous in determining its optimum design solutions in the system level. This allows each design task to have the unique optimisation algorithm which can be customized to respond to the individual needs. A review of the related research studies in design activities integration is shown in Table 2.3.

Table 2.3: Related research in multiple design task integration

	Applicable for limited number of *KCCs*	Applicable for design in system level
Qualitative approach	**DFMA** (Boothroyd, G. (1994), Selvaraj *et al.* (2009), and Cutkosky and Tenembaum (1990))	**Axiomatic Design** Suh (1990, 1995, 1997) **QFD** El-Haik (2005) **DSM** (Steward (1981), Eppinger et al. (1994), and Browning (2001))
Quantitative approach	**Design of Experiment (DOE)** (Fowlkes and Creveling (1995), Box et al. (2005), and Montgomery (2009)	**Proposed in this thesis**

2.2.4 Related Work in KPI Development and Assessment

2.2.4.1 Related Work in Quality KPI Development

There are several studies that have aimed to develop the indices to measure the quality of product. Process capability index is one of the traditional approaches in measuring the variation of final product quality compared with allowable specifications. However, the process capability indices have a challenge in measuring the final product which consists of multiple *KPCs*. On the other hand, several studies have also contributed to the developments of quality measure in term of evaluating the robustness of product and process architecture design. The studies in this approach include the Taguchi's loss function and sensitivity index. Nevertheless, the measures in evaluating the robustness are difficult to relate with the number or percentage of conformance products. The details of aforementioned research studies in quality assessment are elaborated below.

The percentage of non-conforming items can be used to evaluate the final product dimensional quality. In general, the process performance is measured by Process Capability Indices, $\mathrm{C_p}$ or $\mathrm{C_{pk}}$ where $\mathrm{C_p}$ can be defined as $(USL - LSL)/6\sigma$; and USL and LSL are the upper and lower specification limits, respectively, and σ is the standard variation of a single *KPC* variable. In multivariate cases, the *KPC* tolerance/specification region in multivariate m dimensional space is the volume of the hyper-rectangular cube (Taam *et al*., 1993) which can be defined as:

$$\prod_{i=1}^{m}(USL_i - LSL_i) \qquad (2\text{-}2)$$

The *KPC* variations of multivariate processes can be assessed by using Chi-square distance defined as:

$$\chi_0^2 = (\mathbf{y}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu}) \tag{2-3}$$

However, C_p in evaluating multivariate normal *KPC* variations cannot be obtained directly by dividing the volume of the *KPC* hyper-rectangular cube specification as shown in Eq. (2-2) with actual process Chi-square distance expressed in Eq. (2-3) because *KPC* tolerances/specifications are hypercube while the Chi-square distance has elliptical probability region. Thus, to determine C_p, it is necessary to estimate the *KPC* tolerance region into an ellipsoid shape. As a result, when the process is centered at the target and C_p=1, this implies that 99.73% of the process variations are inside the estimated *KPC* tolerance ellipsoid. Taam *et al.* (1993) proposed an approach to calculate C_p by approximating the *KPC* tolerance hypercube with the largest ellipsoid that can fit inside the *KPC* tolerance hypercube. However, estimating the largest ellipsoid volume is difficult in the case where $m > 3$.

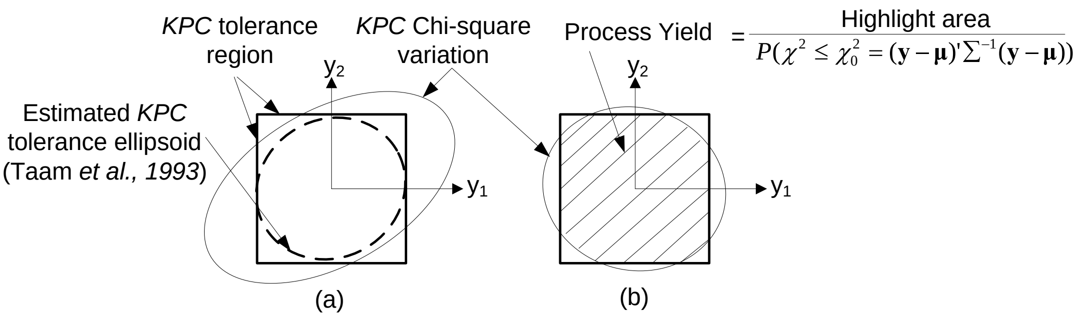


Figure 2.3: *KPC* variations compared with *KPC* tolerance region (a) before optimizing fixture layouts, (b) after optimizing fixture layouts.

In the sensitivity index approach, the product dimensional quality is measured by the variations of $\mathbf{y}^{\mathbf{T}}\mathbf{y} = \mathbf{u}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}}\mathbf{A}\mathbf{u}$ (Phoomboplab and Ceglarek, 2009). To minimize the variations of $\mathbf{y}^{\mathrm{T}}\mathbf{y}$, the robustness of assembly response function, $\mathbf{A}$, has to be improved in order to be insensitive to the *KCC* variation inputs, $\mathbf{u}$. The sensitivity index can be defined as the variations of output signals to input noise which can be expressed as:

$$S = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{y}}{\mathbf{u}^{\mathrm{T}}\mathbf{u}} = \frac{\mathbf{u}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{u}}{\mathbf{u}^{\mathrm{T}}\mathbf{u}} \tag{2-4}$$

The sensitivity index, S, has to be minimized such that the significant variations of $\mathbf{u}^{\mathrm{T}}\mathbf{u}$ contribute to minor variations of $\mathbf{y}^{\mathrm{T}}\mathbf{y}$. If the *KCC* variations of vector $\mathbf{u}$ are constant, the *KPC* variations depend on the assembly response function $\mathbf{A}$. The challenge is to select the design index to assess $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. Several measures are proposed based on optimality criteria in experimental design. Kim and Ding (2004) provided the analysis of three optimality criteria in system design which are: (i) D-optimality (min det($\mathbf{A}^{\mathrm{T}}\mathbf{A}$)); (ii) A-optimality (min tr($\mathbf{A}^{\mathrm{T}}\mathbf{A}$)); and (iii) E-optimality (min $\lambda_{\max}(\mathbf{A}^{\mathrm{T}}\mathbf{A})$; $\lambda_{\max}$ is the extreme eigenvalue). The advantages and disadvantages of these three optimality criteria are discussed below:

D-optimality is to minimize the determinant of a matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, (min det($\mathbf{A}^{\mathrm{T}}\mathbf{A}$)). The advantage of D-optimality is that it minimizes both the variances and the covariences of matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. It is equivalent to minimizing the overall process variations; min det($\mathbf{A}^{\mathrm{T}}\mathbf{A}$) = min $\prod_{i=1}^{m} \lambda_i$ where λ_i is an eigenvalue. D-optimality is very effective to evaluating the design problems which inherent highly non-linear relationships such as fixture layout design. However, the singularity of matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is a major obstacle to the use of D-optimality in multistage fixture layout design.

A-optimality is to minimize the trace of matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, min tr($\mathbf{A}^{\mathrm{T}}\mathbf{A}$) which is the summation of sensitivities of all *KCC*-*KPC* pairs in the assembly processes.

Nevertheless, A-optimality does not consider the dimensional variation impact from covariances within matrix $\mathbf{A}^{\mathbf{T}}\mathbf{A}$. Thus, A-optimality does not imply that the percentage of non-conforming items will be reduced since the covariances among *KCC* nominal design on *KPC* variations are high.

E-optimality is to minimize the extreme eigenvalue of matrix $\mathbf{A}^{\mathbf{T}}\mathbf{A}$, min $\lambda_{\max}(\mathbf{A}^{\mathbf{T}}\mathbf{A})$. E-optimality is similar to D-optimality which considers both variances and covariances of all pairs of *KCC-KPC*, but E-optimality considers only $\lambda_{\max}(\mathbf{A}^{\mathbf{T}}\mathbf{A})$. Thus, E-optimality can avoid the singularity of matrix $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ during computation, and it is aligned with the pareto principle in quality engineering. However, minimizing only the maximum eigenvalu, $\lambda_{\max}(\mathbf{A}^{\mathbf{T}}\mathbf{A})$ cannot guarantee that overall variations, $\prod_{i=1}^{m} \lambda_i$, of the new set of fixture layouts design are decreased. It leaves the possibility that several principle components dominate the overall variations of matrix $\mathbf{A}^{\mathbf{T}}\mathbf{A}$, and the summations of these eigenvalues can contribute to larger variations even though its extreme eigenvalue is lower than the previous fixture layout design. Therefore, it is difficult to decide that process increases its robustness by assessing only the extreme eigenvalue.

2.2.4.2 Related Work in Developing Model to Assess Quality Indices

A model of the assembly system is necessary for optimizing the design of process architecture since it can be used as the *assembly response function* to evaluate product dimensional quality. The development of a variation prediction model can be classified according to the kinds of assembly processes which involve different types of elements in constraining part degrees of freedom of parts. There are three types of assemblies. The first two types of assembly processes are defined by Mantripragada and Whitney (1998). These are:

(i) Type-1 assemblies in which all degrees of freedom of a mating part are fully constrained by part-to-part joints of neighboring parts

(ii) Type-2 assemblies in which all degrees of freedom of a mating part are fully constrained by fixtures. Part-to-part joint might be involved in Type-2 assemblies in order to provide the support, but it does not have an effect on position and orientation accuracy of a part in the assembly. Part-to-part joints of two parts are assembled together by assembly operations such as riveting, welding, or hemming.

Phoomboplab and Ceglarek (2009) proposed Type-3 assemblies wherein all degrees of freedom of a mating part are constrained by part-to-part joints and locators concurrently. The dependency between part-to-part joint and fixture layout in constraining degrees of freedom poses a challenge towards the development of a variation propagation model for Type-3 assemblies. The allocation of degrees of freedom that part-to-part joint and fixture locator constrain affects the fixture planning which is related to the number of locators and their orientation in constraining part degrees of freedom. The research on formulating variation prediction models for each type of assembly are elaborated below.

Related Work in Variation Prediction Model Development for Type-1 Assemblies

Several variation propagation models have been introduced to address the challenges that are inherent in Type-1 assemblies and which are related to assemblability. Assemblability involves analyzing the contact conditions between part-to-part joints or part-to-locators which can be classified as gap, contact or fit, and interference. Several studies have proposed a graph-based representation to identify part-to-part contact conditions (i.e., Mullins and Anderson, 1998; Zou and Morse, 2003; 2004). The graph-based representations aim to provide an efficient simulation approach to analyze part-to-part contact conditions when part dimensions and tolerances in the assembly are changed. Turner (1990) proposed an approach to determine the relative positions of parts in assembly based on the geometric relationships between part features and mating features of its neighboring parts. Similarly, Inui and Kimura (1991) proposed an algebraic method to approximate the actual contacts of part-to-part joints under shape variation. Nevertheless, the approach proposed by Inui and Kimura (1991) is limited to planar and cylindrical part-to-part joints. To simplify the difficulty in determining feature contact conditions, a point-based variation chain is introduced for modeling variation propagation in Type-1 assembly. In this case, a point-based variation chain is developed by establishing points of local reference frames on the part-to-part joints of assembled parts. The differentiation of two local reference frames of assembled part-to-part joints in terms of positions and orientations determines the actual contact conditions. For example, Lee and Andrews (1985) introduced concepts of spatial relationships defining "*fit*" and "*against*" conditions in two planar and hole-cylinder part-to-part joints, respectively. Whitney *et al.* (1994) proposed the use of homogeneous transformations to model variation propagation in multiple assembly stations. The positions and orientations of successive parts can be determined by chaining sets of homogeneous transforms from a preceding part through its part-to-part joint.

Related Work in Variation Prediction Model Development for Type-2 Assemblies

Recent studies have introduced variation prediction models for machining and assembly applications which involve only fixtures in constraining parts. For machining application, Cai *et al.* (1997) proposed a variation model based on differentiating the contact conditions between part surfaces and locators under an infinitesimal error assumption. The approach proposed by Cai *et al.* (1997) is a generic approach in developing a variation prediction model for fixturing layouts of both prismatic and non-prismatic workpieces. However, it has limited application in predicting variation in multi-stage assembly processes. To predict variation in multi-stage assembly processes, Jin and Shi (1999), and Mantripragada and Whitney (1999) adopted the State Transition Model from control theory which integrates homogeneous transforms to describe the dimensional variation propagation in an assembly process. In a similar vein, Ding *et al.* (2000) developed a 2D state-space model for modeling variation propagation in multi-stage sheet metal assembly processes where the variations of fixtures are approximated into linear explicit functions. However, the variation propagation model proposed by Ding *et al.* (2000) is limited to fixture layouts for a 2-D prismatic workpiece.

Related Work in Variation Prediction Model Development for Type-3 Assemblies

There is limited research on the development of a variation propagation model for Type-3 assemblies because of the challenge in consolidating variation propagation model through part-to-part joint and variation model induced by fixture locators. The actual contact points on part-to-part joints are difficult to determine since part-to-part joints are usually form features. This challenge is simplified by modeling a part-to-part joint into a set of points called *virtual mating points*. Based on this assumption, Huang *et al.* (2007a, b) introduced a Stream-of-Variation (SOVA) model to predict dimensional variation in 3D multi-stage assembly processes. The SOVA model is formulated based on point-based geometric constraint models which take into account both part-to-part joints and fixture locators. Then, the variation propagation model is developed by defining the relationships among virtual mating points and fixture locating points on the variation of *KPCs*. The methodology in formulating the SOVA model proposed by Huang *et al.* (2007a, b) is limited to a 3-2-1 fixture layout. The 3-2-1 fixture layout is the basic fixture layout design where six controlling points of locators are assigned their positions in three orthogonal planes as shown in Figure 2.4.

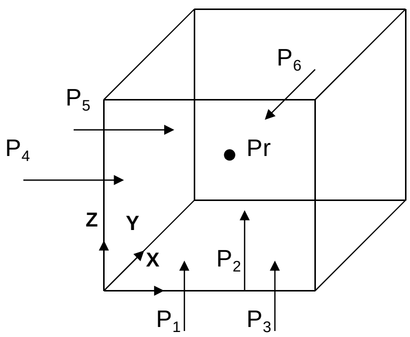


Figure 2.4: 3-2-1 fixture layout.

Thereby, the methodology proposed by Huang *et al.* (2007a, b) has a limited capability in industry applications. Instead of using three orthogonal planes to constrain a part as described by Huang *et al.* (2007a, b), Cai *et al.* (1997) and Loose *et al.* (2007) proposed to use a Jacobian matrix to express first-order and second-order contact constraints of the workpiece surface/locator errors on the overall position and orientation variation of the parts. Therefore, locator positions and their constrained directions are not limited to prismatic parts fixture layouts or 3-2-1 fixture layouts. Then, the variation propagation model for a multi-stage assembly station can be developed by using homogenous transformations to integrate all single part variation models in the process. Phoomboplab and Ceglarek (2009) proposed methodology in formulating Type-3 variation propagation model by adopting concept of virtual mating point proposed by Huang *et al.* (2007a, b) and approach proposed by Loose *et al.* (2007) to obtain generalized model. However, it is very difficult to develop the model manually from CAD data which pose limitation on industrial application. In this thesis, the variation propagation model formulation proposed by Phoomboplab and Ceglarek (2009) is adopted to develop software to support in generating the variation prediction model. The summary of related researches in variation propagation model development is shown in Table 2.4.

Table 2.4: Related researches in variation propagation model developments.

	Types of assembly processes		
	Part-to-part joint only (Type-1 assemblies)	Fixture only (Type-2 assemblies)	Part-to-part joint and fixture concurrently (Type-3 assemblies)
Single station modeling	Mullins and Anderson (1998) Zou and Morse (2003,2004) Tunner (1990) Inui and Kimura (1991) Lee and Andrews (1985)	Cai *et al.* (1997)	Huang *et al.* (2007a)
Multi-station modeling	Whitney *et al.* (1994)	Jin and Shi (1999) Mantripragada and Whitney (1999) Ding *et al.*(2000)	Huang *et al.* (2007b) Phoomboplab and Ceglarek (2009) – **(The model proposed by Phoomboplab and Ceglarek (2009) is adopted in this thesis to develop software which is able to generate assembly response function directly from CAD information)**

2.2.4.3 Related Work in Developing KPI for Assessing Design Complexity

Several complexity measures were proposed in various research disciplines to address specific needs in each area. Among the first attempts in measuring complexity, Hartley (1928) proposed the use of a logarithm function to measure the physical of information instead of a psychological consideration. Based on Hartley (1928), Shannon (1948) proposed information entropy to quantify the number of bits which is necessary to describe information in a telegraph message. The Shannon Information Entropy provided the foundation in subsequent researches by using a probability in measuring complexity. The entropy can be perceived as the average uncertainty or the average reduction in uncertainty of a receiver. The entropy of *y* can be expressed as:

$$H(y) = -\sum_{i,j} p(i,j)\log p(i,j) \tag{2-5}$$

where $p(i,j)$ is the probability of a discrete Markov process which describe the probability of *j* given *i*. That means the probability of the next state only depends on the current state.

By applying the Shannon entropy on the assembly process design to describe the complexity of the system, let us assume that p_i is the probability that KPC_i meets its specification requirement, and it is independent from other *KPCs*. The entropy or the average uncertainty of the assembly system is zero when all *KPCs* are either within or out-of-specification. This means that the product produced from the system can be identified definitely whether the product is conformed to specifications or out-of-specification. The uncertainty or complexity increases when it is not confident that a product produced from the stochastic process is in or out-of-specification. Therefore, the complexity measure based on Shannon entropy information has the limited use for designing the system in term of system performance since most of the system are not completely in- or out-of-specifications.

Similarly, Gell-Mann and Lloyd (1996) proposed probability-based measures to describe the information and complexity. These are so-called "*effective complexity*" and "*total information*" measures. Effective complexity involves measuring the complexity where the entities or elements in the system are considered to be non-random and predictable. On the other hand, the total information is the integration between effective complexity and Shannon information which describes the behavior of the overall system.

The complexity measures are also found in software engineering to evaluate the complexity of a software algorithm. MaCabe (1976) introduced the "*cyclomatic"* measure to evaluate the complexity of the software algorithm instead of the traditional measure based on the number of lines in the software which is subjective to software languages. The cyclomatic measure is meant to be used as a guideline for software engineers to evaluate the simplicity of the developed software which can lead to shorter time in testing and debugging the errors. The cyclomatic measure is developed based on a graph theory, and the measure represents the number of unique independent paths in the algorithm. The combination of these paths can generate all paths in the algorithm. Therefore, testing and debugging can be performed only on these independent paths. The shortcomings of the cyclomatic measure were also described by Shepperd (1988). Other complexity metric used in the software engineering can be found in Carver (1986).

In the areas of engineering and manufacturing, the development of complexity measures has been widely used. Suh (2005) provided a comprehensive explanation of complexity in designing the product and process described by Axiomatic Design Theory. Suh (2005) measured complexity of product design in terms of the uncertainty of product performance in achieving the specified functional requirements. The complexity measure proposed by Suh (2005) is similar to the concept of a probability of non-conformance product produced from a production system. However, the measure proposed by Suh (2005) does not take into consideration the complexity emerging from the dependency of *KCCs* within the process architecture.

Specifically, Suh (2005) defined the complexity of a system as the Information Content which is the probability of the design parameters in Physical Domain (e.g., product architecture) which can satisfy all functional requirements in Functional Domain described in Axiomatic Design Theory. The probability of the Physical Domain performance satisfying the functional requirement can be illustrated by the area of common range as shown in Figure 2.5. The Information Content for a system with *m* functional requirements can be expressed as:

$$I_{sys} = -\log_2 P_{\{m\}} \qquad (2\text{-}6)$$

where $P_{\{m\}}$ is the joint probability that all functional requirements are satisfied. In general, performance of design parameters is statistically independent, $P_{\{m\}}$ and thus can be defined as:

$$P_{\{m\}} = \prod_{i=1}^{m} P_{i|\{j\}} \qquad \text{for } \{j\} = \{1, \ldots, i-1\} \qquad (2\text{-}7)$$

where $P_{i|\{j\}}$ is the conditional probability of satisfying FR_i and other correlated FR_j.

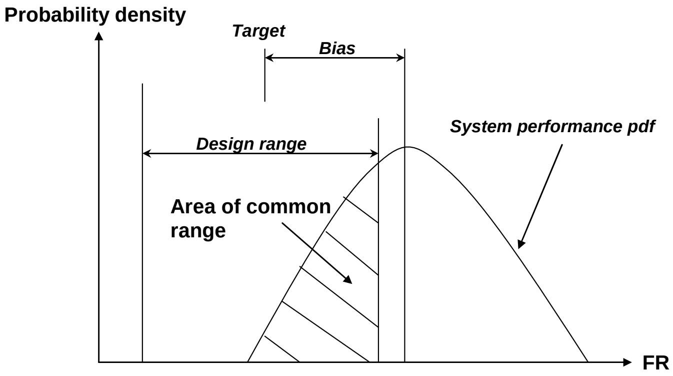


Figure 2.5: Design range, common range, and system pdf for an FR (Suh 2005).

Therefore, the Information Content of the system is defined as:

$$I_{sys} = -\sum_{i=1}^{m} \log_2 P_{i|\{j\}} \tag{2-8}$$

The Information Content introduced by Suh (2005) provides the physical meaning in assessing the complexity between the Physical and Functional Domains in terms of performance needed to satisfy the requirements. For instance, if probability that all functional requirements are satisfied is equal to 1.0, then the required information content is zero. Conversely, when the probability is zero, the information required is infinite. There are two scenarios that information content is infinite. First, the design of process variables and design parameters in Process and Physical Domains cannot deliver the functional performances as defined by the requirements (Process and Physical Domain in Axiomatic Design Theory is equivalent to product architecture and process architecture in this thesis, respectively). Second, the functional requirements cannot represent the actual requirements of customers. Nevertheless, the Information Content cannot provide the clear linkage to improve product and process architecture design.

Furthermore, MacDuffie *et al.* (1996) presented the relationship between various levels of product variety and final product quality performance to enhance the robustness of production system. Although the complexity study of MacDuffie *et al*. (1996) can help in developing robust and lean assembly system, it is based on intensive data collection from production line which requires significant time and investment. Recently, the applications of a complexity measure in assessing the manufacturing system involving multiple product models have been the subject of several studies. The complexity measures in evaluating the uncertainty in assembly operations such as positioning, fixturing, and inserting parts based on an entropy information concept are presented by Fujimoto and Ahmed (2001), and Fujimoto *et al.* (2003). This aims to be applied in reduced complexity in assembly sequence planning. Zhu *et al.* (2008) presented the complexity measures which incorporate product variety (e.g., part selection) and process information (e.g., fixture, tool, and assembly procedure) to enhance production planning performance.

The challenges of the aforementioned research studies can be summarized into two aspects. First, there is a lack of complexity measure which is able to address the dependency among *KCCs* in the process architecture. This complexity measure is necessary for evaluating the scale of engineering change propagation in the system. Second, the physical meaning of complexity measure has to be interpretable and provide the guideline for users to understand and manage the system. In addition, the easiness to formulate the model is crucial in order for the complexity measure to be relevant for industrial applications.

2.3 Conclusions

The related studies discussed in this chapter present the limitation in adjusting the process architecture to meet the new requirements. The studies related to engineering change management have limitations because they require significant time and investment in building and managing database. On the other hand, the related studies on design synthesis tasks present challenges in automatic generation of the assembly response function model to support iteration process and in integration of multiple design tasks. Moreover, it still lacks of *KPI* model to support decision making in adjusting process architecture. Therefore, this thesis proposes methodology to address aforementioned challenges. In the next chapter, the process architecture model is presented in order to understand the dependency between *KCCs* within the process architecture. The *KPIs* are used to evaluate product quality, cost, and design complexity are introduced in Chapter 4. Finally, the framework for integrating the multiple design tasks is presented in Chapter 5. The application of the proposed methodology in this thesis is illustrated in Chapter 6 on industrial process architecture design.

CHAPTER 3

PROCESS ARCHITECTURE MODELING FOR ENGINEERING CHANGE MANAGEMENT

In this chapter, the design dependency among *KCCs* is presented which is important in controlling the engineering change propagation in process architecture design. This is crucial for changing the design of the existing process architecture to meet new requirements. Additionally, the model representing the dependency among *KCCs* is required in order to analyze the potential chain of design change propagation. The challenges to formulate the *KCC* dependency model include: (i) lack of process architecture mathematical model which incorporates information about *KCCs* and process configuration layout; (ii) a large number of *KCCs* in many assembly systems such as in automotive and aerospace industries; and (iii) *KCCs* dependency that can occur either from the *KCC* design itself or from design objective or design constraint used in a design task which intends to optimise one or more *KCCs*. To address these challenges, this chapter introduces a new approach to model process architecture which is then used to represent and analyze *KCC* dependency in multi-stage assembly systems. The proposed methodology consists of four steps which are: (i) process architecture model based on *Key Control Characteristic- Hierarchical Groups* (*KCC-HGs*) using hypergraph data structure representation, and; (ii) formulation of *design task* to optimise *KCCs* of the developed *KCC-HG*; (iii) design task dependency analysis based on the hypergraph techniques and the proposed sensitivity analysis. The dependencies of all design tasks are expressed by the *Task Connectivity Matrix,* T_{CM}, and (iv) the dependency of individual *KCC* in the process architecture is analyzed based on functional dependence criteria and described by using hypergraph technique.

In an assembly process, *KCCs* are distributed in all assembly stations which provide various functions such as locating part position and orientation, holding part against external force, or joining parts together. These *KCCs* can be described as shown in Figure 3.1. In an assembly station, a part or a subassembly from the previous assembly station is placed in the position and hold by fixture locator and clamps. Then, other parts are assembled and fastened to the part which is hold in the assembly station. Finally, the subassembly is released and transferred to the next assembly station.

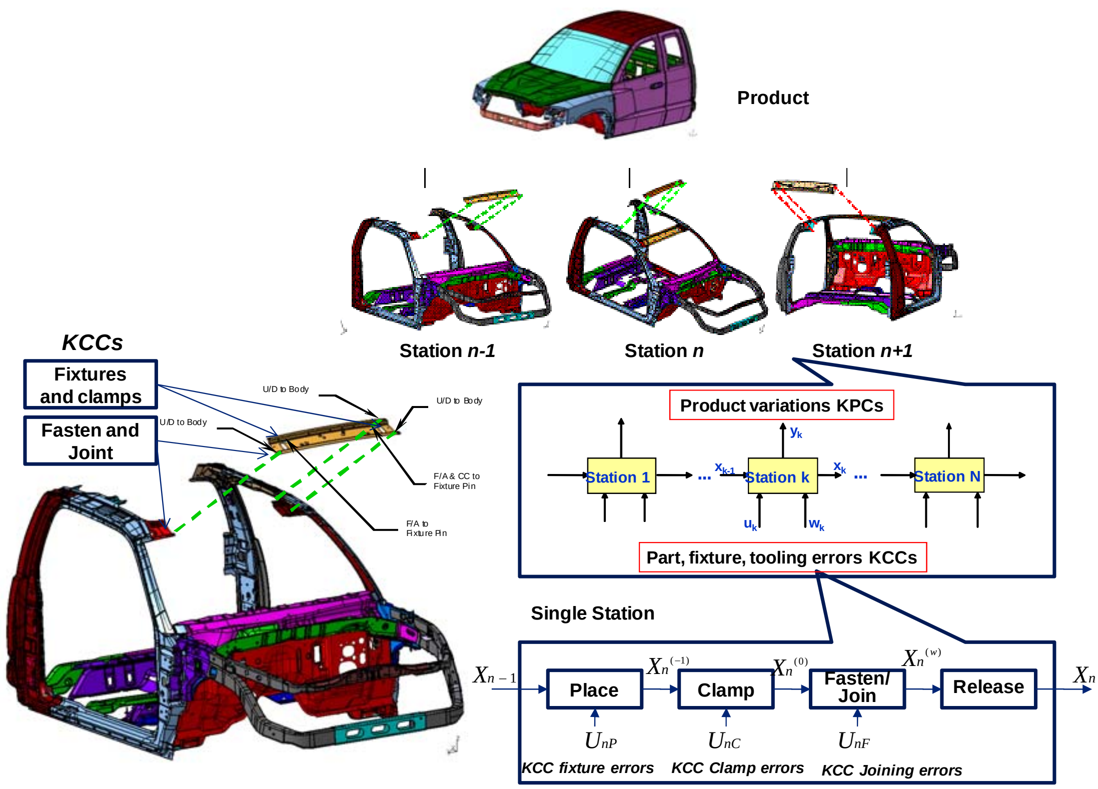


Figure 3.1: The example of *KCCs* in assembly process.

The variations of *KCCs* such as fixture locators, clamps, and fastening process directly impact on the quality of the subassembly. The variations are propagated and accumulated in the final product. To adjust the *KCC* design, it is necessary to consider the functional dependence among *KCCs* since the design adjustment of one *KCC* can affect design of other *KCCs* in the assembly process. Therefore, in this chapter, the concept of *Key Control Characteristic-Hierarchical Groups* (*KCC-HGs*) is introduced to group *KCCs* which have the similar functionality from all assembly stations and then can be optimised by conducting a design task. For example, the fixture locators from all assembly stations are grouped into fixture locator *KCC-HG*. The multi-fixture layout optimisation design task is conducted to determine the optimal fixture locator positions in all assembly stations. The dependency in process architecture can be viewed into two levels which are (i) functional dependence of individual *KCCs*; and (ii) design task level. The dependence of individual *KCCs* is resulted from that *KCCs* have common functional requirement in process architecture. The changes of one *KCC* can affect the design of other *KCCs* in performing the functional requirement. The dependency between design task design tasks is resulted from sharing the same design objectives or design constraints. The dependency between design tasks will lead to difficulty in determining the design task sequence. The overview of this chapter is illustrated in Figure 3.2.

The chapter begins with a discussion on the product and process architecture modeling based on the concept of hypergraph data structure and *KCC-HG* as provided in Section 3.2. The overview of design task development for each *KCC-HG* is described in Section 3.3, and finally the formulation of *KCC* functional dependency Model and *Task Connectivity Matrix* is presented in Section 3.4.

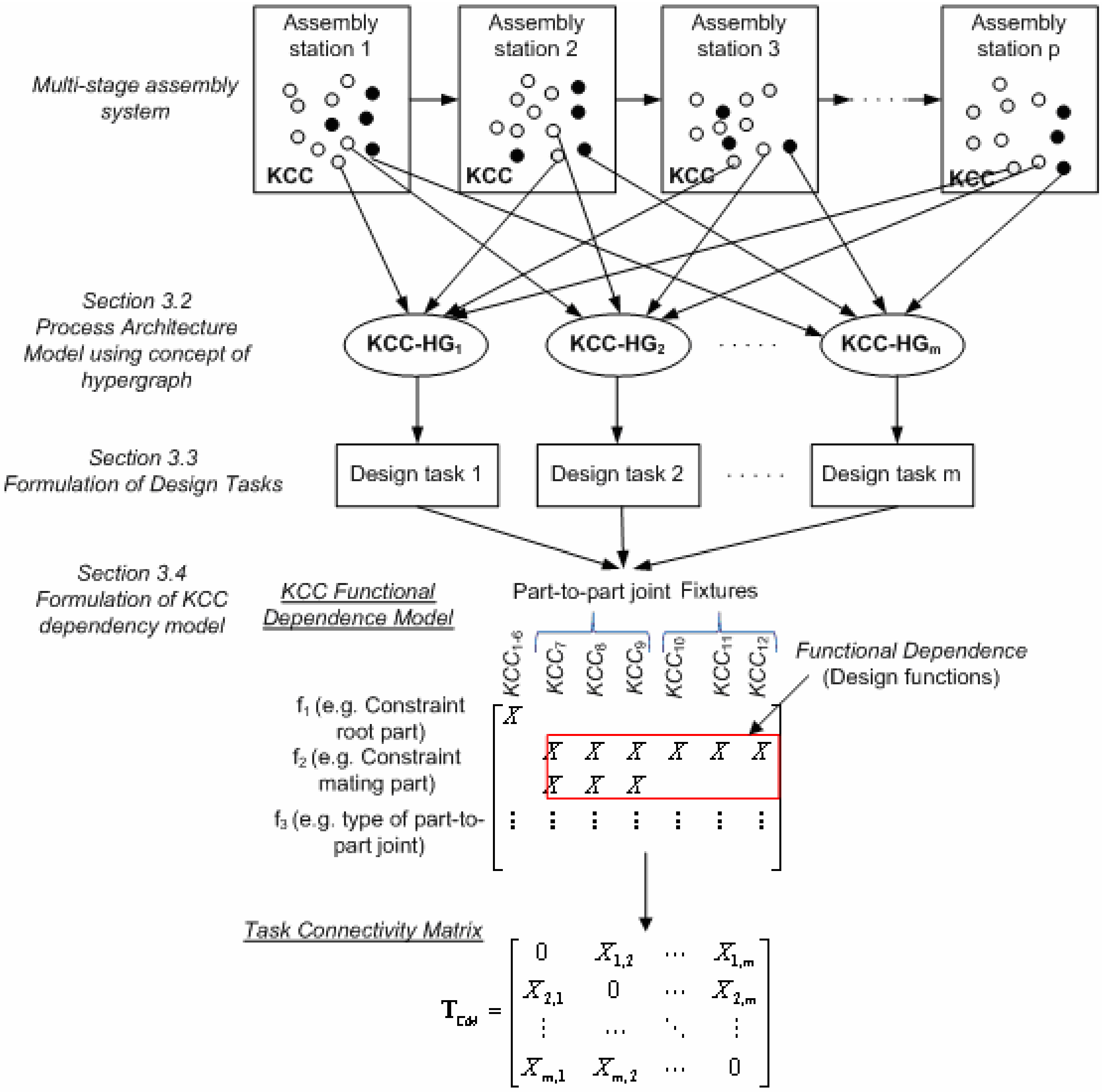


Figure 3.2: The overview of procedure in modeling the dependency of *KCCs*.

3.1 Process Architecture Model using Concept of Hypergraph Data Structure and Key Control Characteristic-Hierarchical Group (KCC-HG)

In this section, the product and process architecture model are represented by using the hypergraph data structure. First, the product architecture is modeled into beam-based representation and is described through beam-based tables. Then, the impacts of process architecture design on the product architecture are discussed. Next, the process architecture is represented by cascading the relationship between *KPCs*, *KCCs*, *KCC-HGs*, and functional dependence of *KCCs*. The hypergraph data structure is also adopted to describe the cascading relationship of the process architecture. The overall cascading relationship in product and process architecture is shown in Figure 3.3. The symbol "X" in Figure 3.3 represents the numerical value which indicates the coefficient between two parameters in the matrix.

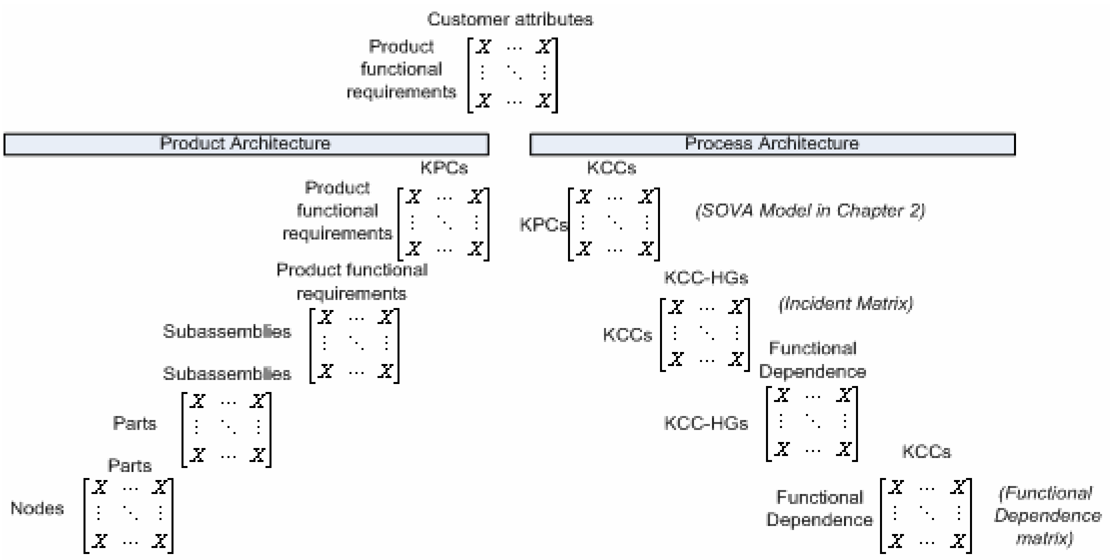
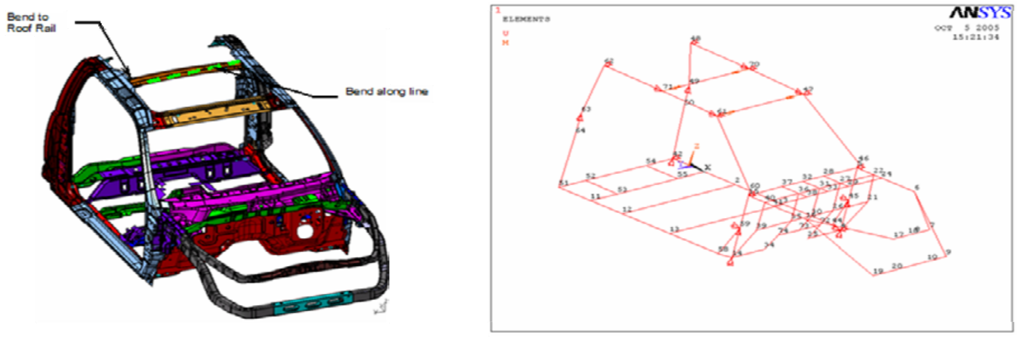


Figure 3.3: The cascade relationship of product and process architecture.

3.1.1 Capture of Product Architecture by Using Beam-based Model and Beam Tables

The product architecture is consisted of several parts assembled into subassemblies in various assembly stations, and become the final product at the end of the production line. The final product must have the capability as design to serve the customer requirements or Customer Attributes in Axiomatic Design Theory. The product architecture is usually modeled by CAD software. However, the basic CAD data structure in describing product consists of vertex table, edge table, and surface table. This basic CAD data structure is similar to the data structure of beam-based model. Moreover, the beam based model allows predicting the deformation of vehicle body structure up to 95 percent of accuracy (Chon, 1986). The example of CAD model and its beam model is shown in Figure 3.4.



(a) CAD Model (b) Beam-based Model

Figure 3.4: The example of product CAD model and its beam-based model.

By adopted beam-based model, a given part is represented as a beam element which consists of two nodes; (B_{vi} and B_{vj}) as shown in Figure 3.5. The beam element is also characterized by its stiffness, *b*, which is a function of (i) Young's modulus; (ii) moment of inertia; and (iii) cross-section area. Then, a beam element is defined as:

$$\mathrm{P_E} = \left((B_{vi}, B_{vj}), b\right) \tag{3-1}$$

The beam nodes can be viewed as the part-to-part joint mating features which is defined as $B_{vi} = \left((X, Y, Z), c\right)$ where $\left(X, Y, Z\right)$ is a node position in global coordinate system, and attribute *c* is defined as translational degrees of freedom(DOFs) along x-, y-, and z-axes and rotational DOFs around x-, y-, and z-axes, respectively. The attribute c is consisted of a 6-tuple, as presented in equation (1), with an entry value of "1" indicating the constraint applied, and "0" otherwise.

$$c = (\text{Tran-x, Tran-y, Tran-z,Rot-x, Rot-y,Rot-z}) \tag{3-2}$$

Part and part deformation represented by beam-based model are shown in Figure 3.5. The deviation of a beam node from its nominal position (B_{v2} to B_{v2}') can be the results of part fabrication errors or the errors during the assembly process in placing, clamping, fastening/joining, and releasing the part. The variation from placing a part to its fixture locator causes part rigid body movement. Clamping can cause the part deformation due to the extra force exerting by fixture locators. Fastening and joining part can also deform a part such as welding gun closing gaps between joints. Releasing subassembly also leads to spring-back causing part deformation.

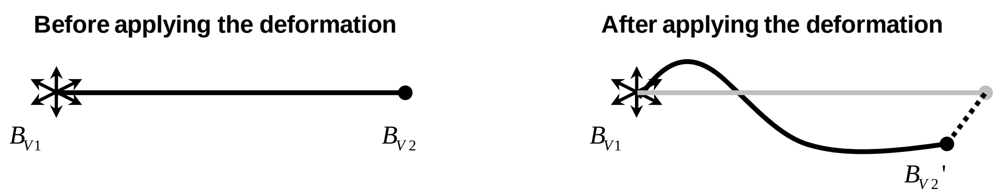


Figure 3.5: The Beam-based representation and deformation of beam-based model.

To represent product architecture by using beam-based model is similar to describing CAD model by using vertex table, edge table, and surface table. The beam-based product architecture consists of node table, beam table, and subassembly table as shown in Figure 3.6

Node Table	Beam Table	Subassembly Table
$B_{v1}:((x_1,y_1,z_1),c_1)$	$P_{E1}:(B_{v1},B_{v2}),b_1)$	$S_{E1}:(P_{E1},P_{E2},P_{E3})$
$B_{v2}:((x_2,y_2,z_2),c_2)$	$P_{E2}:(B_{v2},B_{v3}),b_2)$	$S_{E1}:(P_{E4},P_{E5})$
…………………….	…………………….	…………………….
$B_{vi}:((x_i,y_i,z_i),c_i)$	$P_{Ei}:(B_{vi},B_{vj}),b_i)$ with $i \neq j$	$S_{Ei}:(P_{Ei},P_{Ej})$ with $i \neq j$

Figure 3.6: Node, beam, and subassembly tables to represent product architecture.

3.1.2 Hypergraph-based Definition of the Assembly Model

The process architecture consists of hundreds of *KCCs* located in all assembly stations. These *KCCs* have various functions such as locating, clamping, fastening, and joining which support in assembling parts into subassemblies and final product. These *KCCs* have to be designed in such a way that the *KPC* variations on the final product are minimized as well as the other pre-defined design objectives are satisfied. However, these *KCCs* are usually dependent on each other which resulted in the chains of change propagation when design change is applied on one *KCC*. Therefore, it is very challenging to predict and control the *KPC* variations based on given engineering change conditions. In addition, the non-linear relationships between *KCCs* and *KPCs* pose additional difficulty in optimizing the design of *KCCs* to meet the functional requirements or engineering change initiations. This section describes the process architecture through the cascading matrices from *KPC-KCC* relations to the functional dependence of *KCCs* as shown in Figure 3.3.

This thesis adopts generalized SOVA model proposed by Phoomboplab and Ceglarek (2009) to describe the relation between *KPCs* and *KCCs*. The SOVA model allows predicting the *KPC* variations based on given *KCC* design. In order to optimise the design of *KCCs* in the assembly process, the process architecture is necessary to be systematically modeled. The hypergraph data structure is adopted in this chapter to describe the process architecture and to partition the process architecture into hierarchical groups for the subsequent analysis. In general, the hypergraph is a pair **PA**= (**X**, **E**) where **X** is a set of elements, called vertices, and **E** is a non-empty set of **X** called hyperedges. The elements are equivalent to *KCCs* and the hyperedge is equivalent to the assembly station which consists of *KCCs* in that station. For example, the floor pan assembly process shown in Figure 3.7 involves three assembly stations and has *KCCs* in locating and joining parts. Thus, the hypergraph representing the process architecture of floor pan assembly is:

$$\mathbf{PA} = (\mathbf{X}, \mathbf{E})$$

where $\mathbf{X} = \{P_{1,1}, P_{1,2}, \ldots, P_{3,10}\}$; and

$$\mathbf{E} = \{e_1, e_2, e_3\} = \{\{P_{1,1}, \ldots P_{1,11}\} \quad \{P_{2,1}, \ldots P_{2,10}\} \quad \{P_{3,1}, \ldots P_{3,10}\}\}.$$

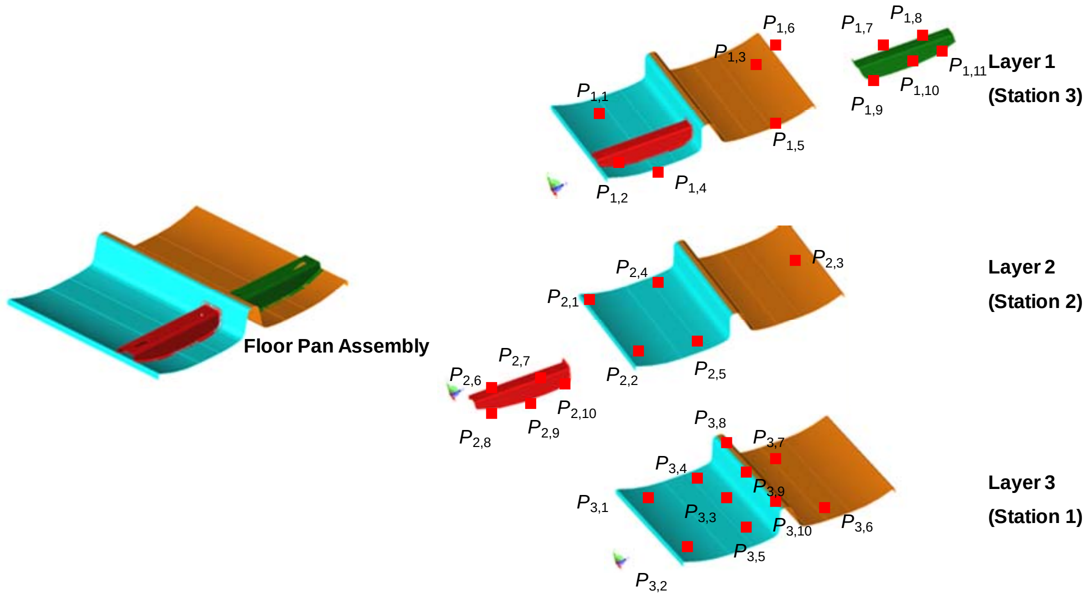


Figure 3.7: *KCCs* in Floor Pan Assembly Process.

However, optimizing the *KCCs* in each assembly station cannot guarantee that the optimal process architecture is obtained. This chapter proposes to partition the process architecture into *Key Control Characteristic-Hierarchical Groups, KCC-HGs*.

Then, the appropriate optimisation technique can be applied on each *KCC-HG*. The concept of *hierarchical group* is adopted from Ceglarek *et al.* (1994) to represent *KCCs* in process architecture. The concept of *Key Control Characteristic-Hierarchical Groups* is to consolidate the *KCCs* in all assembly stations which have similar functionality together. For example, by using the floor pan assembly shown in Figure 3.7, the *KCC-HG* of all fixtures and part-to-part joints located in three layers can be expressed as:

$$\mathbf{KCC-HG}_{fixture} = \{\{P_{1,1}, P_{1,2}, \ldots, P_{1,8}\} \quad \{P_{2,1}, P_{2,2}, \ldots, P_{2,7}\} \quad \{P_{3,1}, P_{3,2}, \ldots, P_{3,7}\}\}$$

$$\mathbf{KCC-HG}_{joint} = \{\{P_{1,9}, P_{1,10}, P_{1,11}\} \quad \{P_{2,8}, P_{2,9}, P_{2,10}\} \quad \{P_{3,8}, P_{3,9}, P_{3,10}\}\}$$

Then, the hypergraph representing the process architecture of floor pan assembly based on *KCC-HG* partitioning can be expressed as:

$$\mathbf{PA} = (\mathbf{X}, \mathbf{E}_{KCC-HG})$$

where $\mathbf{X} = \{P_{1,1}, P_{1,2}, \ldots, P_{3,10}\}$; and

$$\mathbf{E}_{KCC-HG} = \{\mathbf{KCC-HG}_{fixture}, \mathbf{KCC-HG}_{joint}\}$$

For each *KCC-HG*, the design task can be applied to determine the optimal design. The formulation of a design task is presented in the next section. In the complex assembly product such as the automotive body shown in Figure 3.8, it can be partitioned the assembly process into several *KCC-HGs* such as fixture layouts, part-to-part joint, and *KCC* tolerances. *KCCs* in each *KCC-HG* are consolidated from all of assembly stations in the process. Thus, the hypergraph based on *KCC-HG* partitioning can represent the process architecture as:

$$\mathbf{PA} = (\mathbf{X}, \mathbf{E}) \tag{3-3}$$

where $\mathbf{X} = \{P_1, P_2, \ldots, P_n\}$; and

$$\mathbf{E} = \{\mathbf{KCC} - \mathbf{HG}_1, \ldots, \mathbf{KCC} - \mathbf{HG}_m\}$$

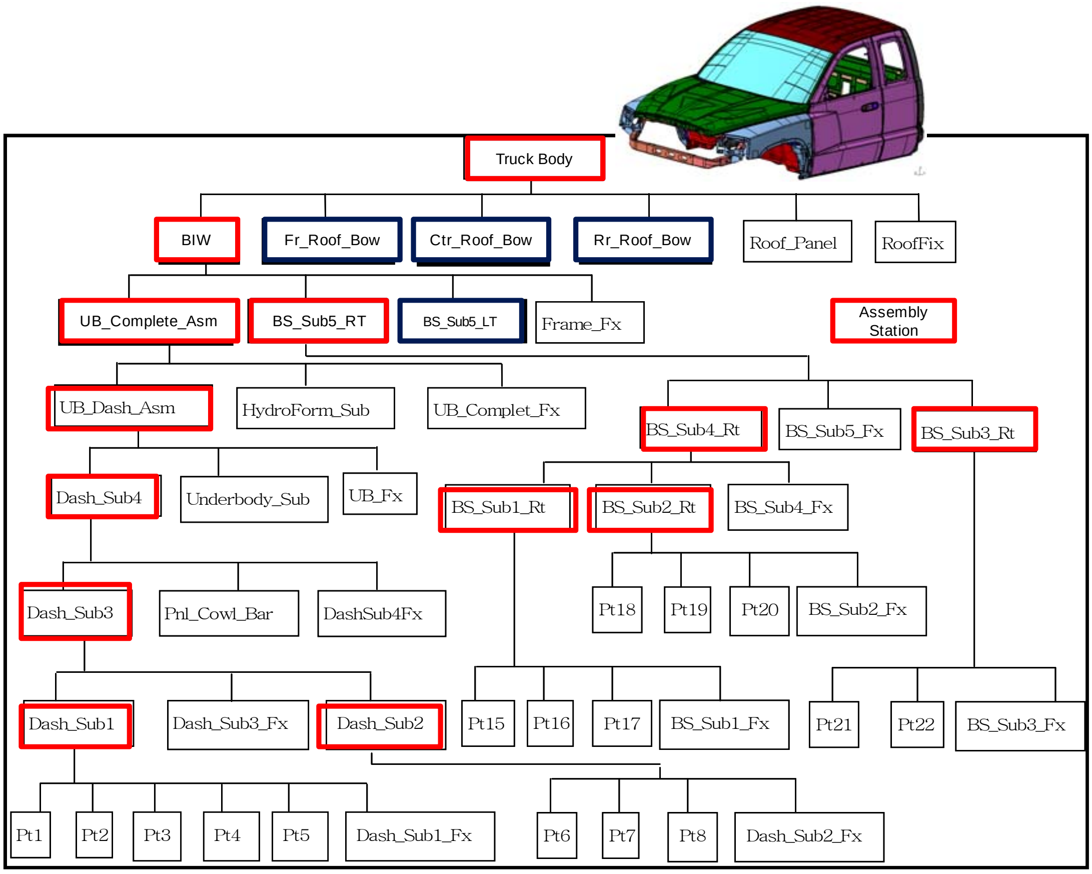


Figure 3.8: Assembly tree of truck body assembly process.

The partitioning of process architecture based on *KCC-HG* can be shown as incident matrix. The incident matrix can help to visualize the *KCCs* in each *KCC-HG* and it will lead to determine the functional dependence between *KCCs*. The rows of incident matrix shown in Figure 3.9 are all of *KCCs* in the process architecture and the columns are the process architecture partitioned into *KCC-HGs*.

$$
\begin{array}{cc}
 & \begin{array}{cccc} KCC\text{-}HG_1 & KCC\text{-}HG_2 & & KCC\text{-}HG_m \end{array} \\
\begin{array}{c} KCC_1 \\ KCC_2 \\ \\ \\ KCC_n \end{array} & \begin{bmatrix} & & \cdots & X \\ X & & & \\ & X & \cdots & \vdots \\ \vdots & & & \\ & & \cdots & X \end{bmatrix}
\end{array}
$$

Figure 3.9: Incident matrix based on *KCC-HG* Partition criteria.

The proposed methodology in partitioning *KCCs* into $KCC\text{-}HG_i$ provides two advantages. First, it reduces the complexity in defining the dependency among the large number of individual *KCCs* in the process architecture. The dependency is defined between *KCCs* within a $KCC\text{-}HG_i$ in controlling the final product dimensional quality. Each individual design task is developed to optimise the design of specific $KCC\text{-}HG_i$. Second, the *KCC-HG* arrangement also leads to efficient *KCC* design management since the customized optimisation algorithm of an individual design task can be developed for different needs of each $KCC\text{-}HG_i$. A design task to optimise the *KCC-HG* design is different from current design practices which aim to optimise the design of all types of *KCCs* in each assembly station. As a result, current practices cannot guarantee that the integrated assembly station provides the global system optimisation since the dependency of *KCC* design between assembly stations is not taken into consideration. The relations of functional requirements, *KPCs*, *KCCs*, *KCC-HGs*, and design tasks are shown in Figure 3.10.

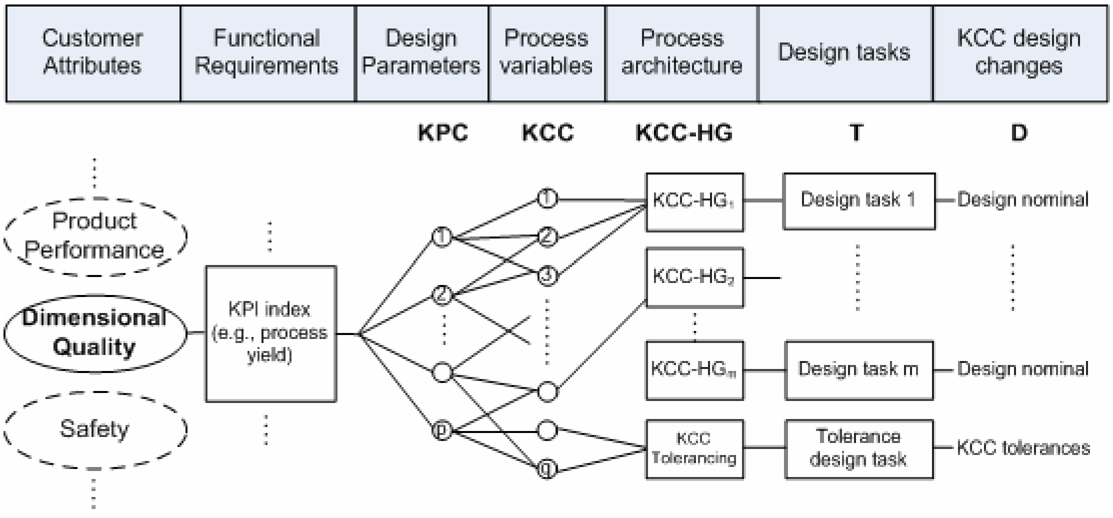


Figure 3.10: Modeling framework for engineering change management driven by design synthesis.

An example of the relations among dimensional quality, *KPCs*, *KCCs*, *KCC-HGs*, and design tasks by using the automotive door fitting process is presented as follow. The customer perception on dimensional quality problems of the door fitting process can be poor fit, wind noise, closing effort, or water leakage. These dimensional quality problems are resulted from the variations of gap/flush, seal gap, or closing effort which are functional requirements or *KPCs*. For example, the flushness and gap between the door and side frames are shown in Figure 3.11. The variations of these *KPCs* are directly affected by product and process variations. The product variations, for example, are the dimensional variations in producing hinges, doors, and fasteners. The process variations are position variations of fixture locators or hinge. The fixture locator position and hinge variations are considered as *KCC* variations. There might be other sources that can cause the *KPC* variations such as painting process. However, variation from painting process is not considered as *KCC* variations since it does not involve directly in an assembly process. The fixture locators and hinges are grouped into a single so-called *KCC-HG fixturing*. The *design* *configurations*, *D*, required for *KCC-HG fixturing* in all assembly stations are: (i) nominal locator positions; and, (ii) tolerances for all locator positions. Then, the design task, T, is developed to optimise each design configuration. For example, fixture layout design task is applied to determine the locator positions for all locators in a *KCC-HG* fixturing group. Table 3.1 provides an example of the relations in the door fitting process.

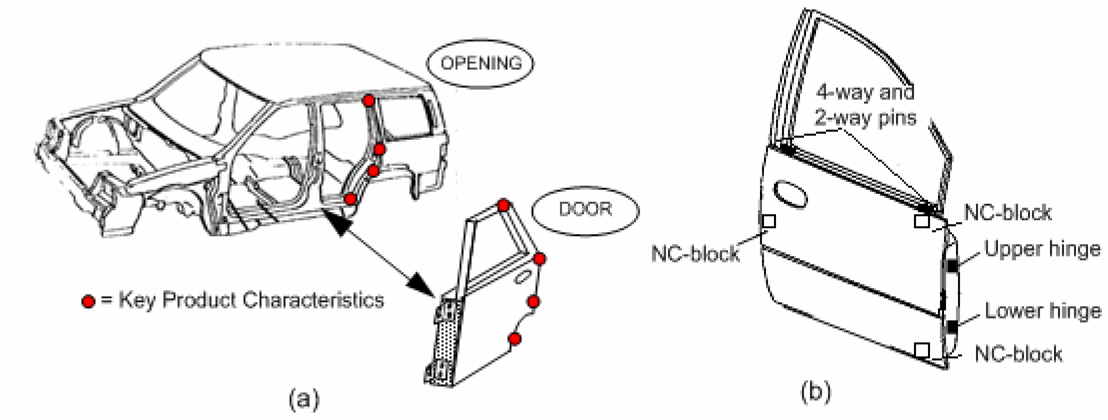


Figure 3.11: Door fitting assembly (a) *KPC* location on side frame and door to measure gap and flushness; (b) *KCC* fixtures in assembling door.

Table 3.1: An example of mapping process for door fitting in automotive body assembly.

Customer Attributes	***Functional Requirement***	**KPC**	**KCC**	***KCC-HG***	**T**	**D**
poor fit, wind noise, closing effort, or water leakage	variations of **gap/flush**, and seal gap	*KPCs* on door and side frame (Figure 3.3 (a))	1) 4-way pin 2) 2-way pin 3) NC block #1 4) NC block #2 5) NC block #3 6) Upper hinge 7) Lower hinge 8) Etc.	**KCC-HG Fixture & KCC tolerances**	1) Fixture layout design task 2) Tolerance optimisation design task	1) Fixture layouts 2) *KCC* Tolerances

3.2 Formulation of Design Tasks

After the *KCCs* are grouped into *KCC-HGs*, a design task for each *KCC-HG* is formulated. A design task, T, is a methodology to optimise a specific design configuration, D; which can be either optimised *KCC* nominal designs or *KCC* tolerances which is subjected to design constraints, C_T. The design task, T, can be defined as minimization or maximization of an objective function subject to design constraints which can be expressed as:

$$\text{Min/Max} \quad f(D) \tag{3-4}$$

$$\text{s.t.} \quad C_T(\mathbf{\Phi}, D) \geq 0; \quad D \in \mathbf{S}$$

where $f(\bullet)$ is the objective function; D is variable; $\mathbf{S}$ is design space of D; $C_T(\bullet)$ is design constraint function; and $\mathbf{\Phi}$ is constant parameter.

The objective function, $f(\bullet)$, usually is the Key Performance Index which aims to achieve. The objective functions, for example, are cost, product dimensional quality, or time in assembly. For instance, the *KCC* tolerance optimisation design task proposed by Huang *et al.* (2009) in minimizing cost of *KCC* tolerances subject to process yield higher than threshold can be expressed as:

$$\text{Min}_{\mathbf{t}} \quad \sum_{i=1}^{n} \mathrm{C}_i(t_i) \tag{3-5}$$

$$\text{s.t.} \quad yield(\Lambda, \mathbf{t}) = \Pr\left\{ \bigcap_{k=1}^{m} (L_k \leq y_k \leq U_k) \right\} \geq \Pr_{\text{Theshold}}; \; t_i^L \leq t_i \leq t_i^U,$$

where t_i is a *KCC* tolerance which aims to be optimised, a cost function is $C_i(t_i) = A_i e^{-B_i(t_i)} + G$; A_i, B_i, and G_i are model constants associated with the manufacturing cost for i^{th} *KCC* tolerance; Λ is fixture layouts and part-to-part joints in process architecture which are given (constant); $\mathbf{t}$ is a set of all *KCC* tolerances; L_k and U_k are lower and upper specification limits of *KPC*, y_k; $\Pr_{\text{Threshold}}$ denotes a required threshold yield; and t_i^L, t_i^U are process precision limits on i^{th} *KCC* tolerance, t_i.

The dependency between design task A and B can occur at two levels which are:

- The design tasks in optimizing the designs of two *KCC-HGs* share the same design objectives or design constraints. The objective functional dependence of design tasks in optimizing *KCC-HGs* design can be shown in Figure 3.12.
- The optimal *KCC* design configurations, D_A, which are optimised by design task A become constant parameter, $\mathbf{\Phi}_B$, of design task B which impact the optimal design results of design task B. This type of dependency will be elaborated in the next section.

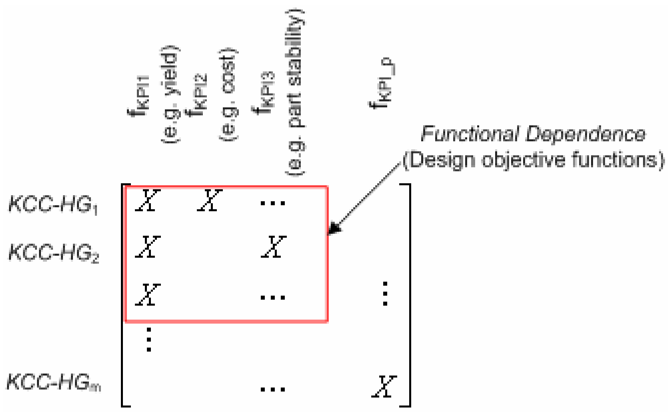


Figure 3.12: Functional dependence of *KCC-HGs* and design task objective functions.

3.3 Formulation of KCC Dependency Model

This section is to describe the dependency among *KCC* design which causes the dependency in *KCC-HG* and design task level. The concept of hypergraph is adopted to analyze the dependency of *KCCs*. As shown in Figure 3.13, nodes/vertices represent *KCCs* in process architecture defined as $V = \{v_1, v_2, v_3, \ldots v_{10}\}$, and edges represent *KCC-HGs* defined as $E = \{e_1, e_2, e_3\}$. The vertices or *KCCs* are grouped into an edge or *KCC-HG* by selecting the *KCC* having the same functionality. For example, all *KCC* fixtures, $\{v_1, v_2, v_3, v_4, v_5\}$, in a process architecture are grouped into fixturing *KCC-HG* or edge $e_1 = \{v_1, v_2, v_3, v_4, v_5\}$. On the other hand, all *KCC* part-to-part joints $\{v_6, v_7, v_8, v_9, v_{10}\}$ are grouped into part-to-part joint *KCC-HG* or edge $e_2 = \{v_6, v_7, v_8, v_9, v_{10}\}$. However, let assume that in one assembly station the *KCC* fixtures v_1 and v_2 have design dependency with *KCC* part-to-part joint v_5 in constraining the degree-of-freedom of a part in that assembly station. Thus, the *KCC* fixtures and part-to-part joint are grouped into edge $e_3 = \{v_2, v_3, v_6\}$. The edge e_3 causes the design dependency between edge e_1 and edge e_2 which can be described as ***transversal*** of hypergraph $H = (V, E)$ is a set $T \subseteq V$ that has nonempty intersection between edges. The *KCCs* $\{v_2, v_3, v_6\}$ have ***edge degree*** equal to two since they are co-exist in two edges. On the other hand, in the case that two edges have an empty set of intersection (i.e. hypergraph has ***clutter*** condition) all *KCC-HGs* are independent and all *KCCs* have edge degree equal to one.

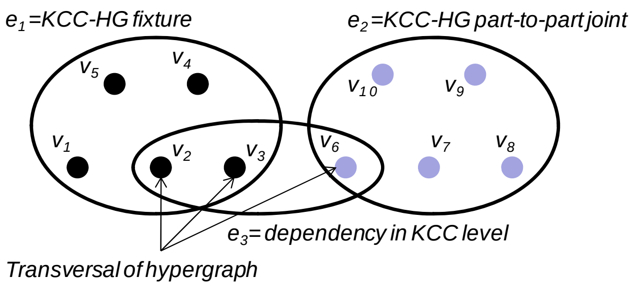


Figure 3.13: Hypergraph to represent dependency in *KCC* level.

The dependency in *KCC* level also can be represented by using functional dependence matrix where rows designated for design functions and the column labeled with *KCCs* as shown in Figure 3.14. The example in formulating the functional dependence matrix is illustrated through the following case study.

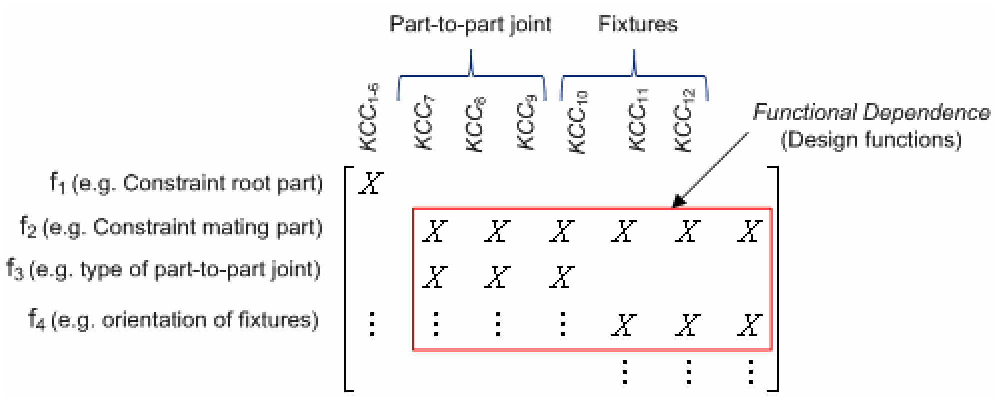


Figure 3.14: Functional dependence of *KCC-HGs* and design task objective functions.

The part-to-part joint and fixture layout can have interdependent relation between each other in constraining the degrees of freedom of a part in an assembly process. The allocation of degrees of freedom which is to be constrained by part-to-part joints and fixtures impact directly on *fixture planning* processes which are related to the number of fixtures needed, the type of fixtures, and the orientation of fixtures corresponding to parts. In a rigid body assembly, there are six degrees of freedom that part-to-part joint and fixture locators have to constrain. The design can begin with the selection of a specific type of part-to-part joint and then design fixture locators as shown in Figure 3.15. On the other hand, fixture layout can be designed first and a part-to-part joint can be realized later. Based on the example provided in Figure 3.15, it can be seen that the dependency among *KCCs* can pose the difficulty in determining the sequence in conducting design task. The dependency of the design tasks can be determined by transversal of hypergraph presented in the next section.

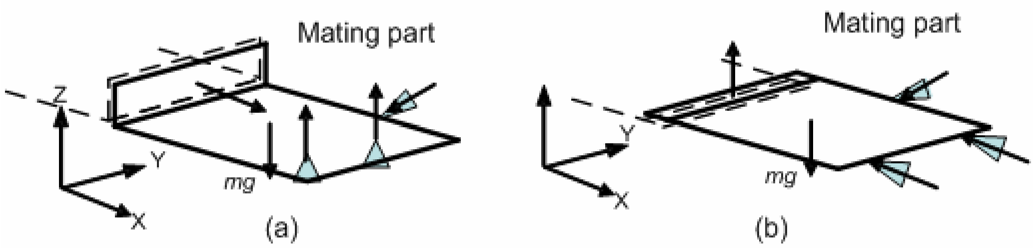


Figure 3.15: The impact of part-to-part joint selection and fixture layout designs, (a) butt joint, and (b) lap joint.

3.3.1 Determine Dependency of Design Tasks by Transversal of Hypergraph

The dependency in the design task level can be determined by using the concept of transversal of hypergraph. The *KCCs* besides *KCC* of interest might be required in order to conduct design optimisation by a design task. For example, to optimise the fixture layout for a mating part, the part-to-part joint type has to be given as shown in Figure 3.15. This given design information is denoted as **Φ** or "$KCC_{passive}$", and the *KCCs* which are optimised by the design task are called "KCC_{active}". Four scenarios of the dependency in the design task level can be explained as follows:

Scenario 1:

Two design tasks are independent if (i) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} = \phi$, and (ii) $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} = \phi$. These two design tasks can be conducted concurrently as shown in Figure 3.16 (a).

Scenario 2:

The design task T_2 is dependent on the design task T_1 if (i) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} \neq \phi$ and (ii) $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} = \phi$. Then the design task T_2 has to be conducted after the design task T_1 as shown in Figure 3.16(b)

Scenario 3:

The design task T_1 is dependent on the design task T_2 if (i) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} = \phi$ and (ii) $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} \neq \phi$. Then the design task T_1 has to be conducted after the design task T_2 as shown in Figure 3.16(c)

Scenario 4:

Two design tasks are interdependent if (i) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} \neq \phi$ and (ii) $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} \neq \phi$. The feedback iteration between both design tasks is required as shown in Figure 3.16(d).

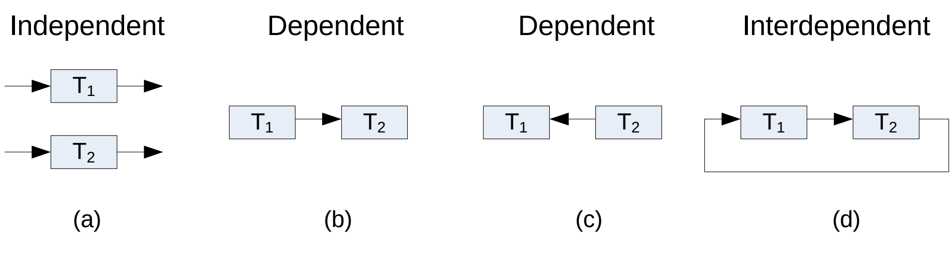


Figure 3.16: Dependency in design task level and design task sequences.

The example of dependency among design tasks for the process architecture design such as tolerance optimisation, fixture layout optimisation, part-to-part joint optimisation, and assembly sequence optimisation design tasks are presented. For example, the optimal *KCC* tolerances are dependent on the nominal designs of assembly sequence, part-to-part joint, and fixture layout since the optimal *KCC* tolerances can be obtained only when the other *KCC* nominal positions are defined. The assembly sequence design can affect the decision of part-to-part joint selection in the case that assembled product have the closed structure. As shown in Figure 3.17, if part-to-part joints of Part 3 and Part 4 have to be assembled together in the last assembly station, the part-to-part joints have to be designed as slip joints instead of butt joints in order to absorb the variation of the distance between Part 1 and Part 3. Meanwhile, the part-to-part joint selection can also pose a constraint on assembly sequence design in terms of interference between the parts as shown in Figure 3.18.

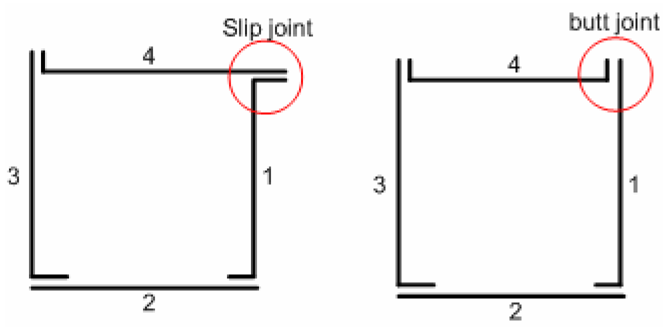


Figure 3.17: The impact of assembly sequence on part-to-part joint selection (Lee and Saitou (2003)).

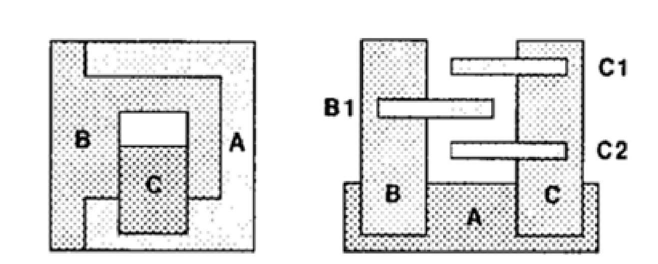


Figure 3.18: The impact of part-to-part joint selection on assembly sequence. (Wang and Ceglarek (2005)).

The assembly sequence design and fixture layout design can have interdependent relations. The assembly sequence in loading part on fixture layouts (i.e., a root part and a mating part) can affect the design of fixture layout. For example, in the rigid body assembly, a root part is fully constrained by six locators while a mating part is constrained by a part-to-part joint of the root part and fixture locators. Therefore, the number of locators and the locations of locators on a part which is assigned as a root part or a mating part are different. On the other hand, accessibility of locators in fixture layout design can affect the assembly sequence design as well.

3.3.2 Formulation of Task Connectivity Matrix

In order to represent the sequence of multiple design tasks, this thesis adopted the concept of Activity-based Design Structure Matrix (Browning (2001)). An activity-based Design Structure Matrix describes the input and output relationship of design activities or design tasks which show the dependency of design tasks based on the requisite information flow or $KCC_{passive}$, $\mathbf{\Phi}$, used in this thesis.

The activity-based design structure matrix used in this thesis is called *Task Connectivity Matrix*, T_{CM}, to represent the design task dependency model in the process architecture as shown Figure 3.19. The relation between design tasks can be classified as either independent or dependent expressed by "0" and "1" in the matrix, respectively.

Therefore, $\mathbf{T}_{\mathrm{CM}}$ of m design tasks can be expressed as:

$$\mathbf{T}_{CM} = \begin{bmatrix} 0 & X_{1,2} & \cdots & X_{1,m} \\ X_{2,1} & 0 & \cdots & X_{2,m} \\ \vdots & \cdots & \ddots & \vdots \\ X_{m,1} & X_{m,2} & \cdots & 0 \end{bmatrix} \tag{3-6}$$

where $X_{i,j}$ is the dependency of a design task T_j against a design task T_i. If the design task T_j is independent from T_i, $X_{i,j}$ is equal to 0. On the other hand, if the design task T_j is dependent on T_i, $X_{i,j}$ is equal to 1.

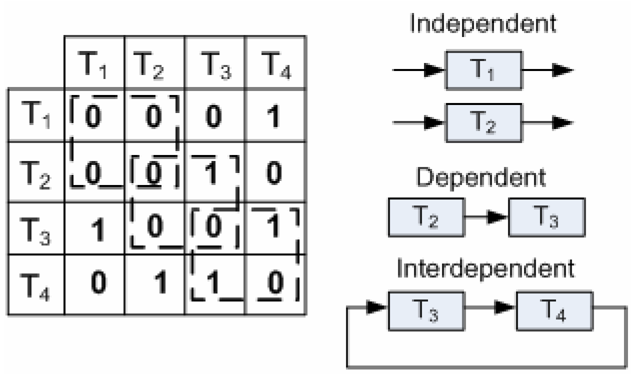


Figure 3.19: Task Connectivity Matrix and design task sequence.

The dependency between design tasks shown in the Task Connectivity Matrix can be determined by either using hypergraph representation. Then, the case study in formulating the Task Connectivity Matrix is presented.

3.3.3 The Case Study in Formulating Task Connectivity Matrix

The concept of transversal of hypergraph and the sensitivity analysis are applied on the floor pan assembly of automotive body in order to analyze the dependency between part-to-part joint design task, T_1, and the multi-fixture layout design task, T_2. In order to conduct dependency of design tasks based on transversal of hypergraph, the hypergraph of the floor pan assembly is formulated as shown in Figure 3.20.

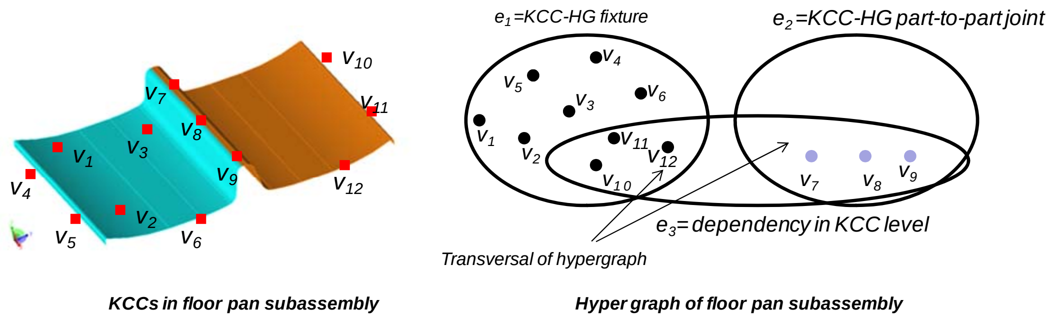


Figure 3.20: Floor pan subassembly and its hypergraph representation.

The hypergraph, **PA**, consists of 12 vertices, $V = \{v_1, v_2, v_3, \ldots v_{12}\}$, (i.e., 12 *KCCs* in the floor pan subassembly processes) which are grouped into 3 hyperedges, $E = \{e_1, e_2, e_3\}$. The hyperedge, e_1, represents fixturing *KCC-HG* which consists of six fixture locators, $\{v_1, v_2, v_3, \ldots v_6\}$, in constraining the root part (Floor pan left) and three locators, $\{v_{10}, v_{11}, v_{12}\}$, in constraining the mating part (Floor pan right). The hyperedge, e_2, represents the part-to-part joint *KCC-HG* between floor pan left and right which are described by three vertices, $\{v_7, v_8, v_9\}$. Last, the hyperedge, e_3, represent the fixture locators, $\{v_{10}, v_{11}, v_{12}\}$, and part-to-part joint, $\{v_7, v_8, v_9\}$, in constraining the degree-of-freedom of the mating part. The fixture locators, $\{v_{10}, v_{11}, v_{12}\}$, and part-to-part joint, $\{v_7, v_8, v_9\}$, are interdependent since the types of part-to-part joints contribute to the different constrained direction of the locator. To illustrate this, two different types of part-to-part joints, for example butt and lap joint can lead to differences in planning two fixture layouts which require locators having different constrained direction as shown in Figure 3.21. Thus, the optimal fixture layout designs of two different types of part-to-part joints are different, (i.e., lap joint and butt joint).

To use the transversal of hypergraph representing the dependency between a part-to-part joint design task (T_1) and fixture layout design task (T_2), the fixture locators are the active *KCCs*, $\{\mathbf{KCC}_{active2}\}$, which are optimised by fixture layout design task (T_2). The fixture layout design task (T_2) requires part-to-part joint *KCCs* as the constant parameters, $\mathbf{\Phi} = \{\mathbf{KCC}_{passive1}\}$, from part-to-part joint *KCC-HG* or hyperedge, e_2. Thus, transversal of hypergraph has non-empty intersection (i.e., $\{\mathbf{KCC}_{passive1}\} \cap \{\mathbf{KCC}_{active2}\} \neq \phi$), then the design task T_2 is dependent on the design task T_1, and both design tasks have to be conducted in sequential order. The task connectivity matrix can be described as:

$$\mathbf{T}_{CM} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

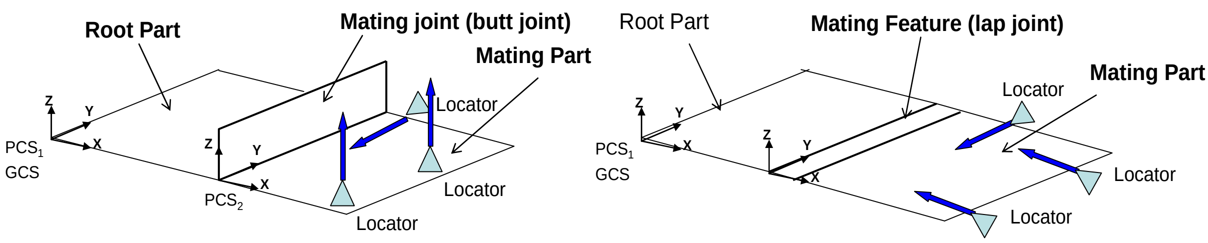


Figure 3.21: Mating part fixture layouts and their constrained directions subjected to part-to-part design.

3.4 Conclusions

The methodology in formulating the *KCC* functional dependency model and the subsequent *Task Connectivity Matrix* to represent the dependency of design tasks and *KCCs* in the process architecture is introduced in this chapter. The *KCC* functional dependency model and Task Connectivity Matrix helps to assess the scale of engineering change propagation in the system if the process architecture design is adjusted. The significance of this chapter involve: (i) the partitioning of assembly system into *KCC-HGs*; and (ii) hypergraph representation and sensitivity analysis to determine the *KCC* and design task dependencies. The model formulation in representing the *KCC* dependency can help to understand the relation of *KCCs* in the process architecture and lead to approach in controlling the engineering change propagation if the process architecture design is changed. Instead of relying on the engineering change records, the proposed approach can simulate the potential propagation chains. The developed model will be used in Chapter 4 in formulating the redesign complexity measure.

CHAPTER 4

KPI MODEL FORMULATION FOR ASSESSMENT OF ENGINEERING CHANGE ON PROCESS ARCHITECTURE

4.1 Introduction

This chapter proposes *Key Performance Index* (*KPI*) model which consists of the multiple assessment indices for analyzing the impacts of the engineering changes on current process architecture. The key performance index can be obtained from the design objective functions of a design task described in Chapter 3. The assessment is performed in three critical areas which involve: (i) impacts of engineering changes on the final product dimensional quality and process yield; (ii) the engineering change complexity (ECC) or scale of engineering change propagation in changing the process architecture design; and (iii) costs in changing process architecture to maintain product dimensional quality (In this thesis, cost is limited to changes of *KCC* tolerances). The quality index or process yield is obtained by adopting assembly response function model proposed by Phoomboplab and Ceglarek (2009). The engineering change complexity index is assessed by using process architecture model presented in Chapter 3. For the cost related to redesigning the process architecture, this thesis proposes to use the cost incur by optimizing *KCC* tolerances in order to maintain the dimensional quality of the final product. Cost in adjusting tolerances in order to maintain product quality usually is lower than cost in adjusting nominal design since changing the nominal design affect both product and process architecture as well as related operations. For example, changing location of pin fixture causes the change of hole position on a part as well as adjusting operation in forming a part. These three indices are used as preliminary information to make decisions regarding to the redesign of the process architecture.

The assessment of the engineering change impact on the final product dimensional quality is also addressed in this chapter by proposing an index based on the change of *process yield* level. Here, process yield is expressed as the joint probability that all *KPCs* of a product simultaneously satisfy the corresponding *KPC* design specifications. The process yield can address the shortcoming of the traditional process capability indices in assessing multivariate *KPCs* which have the ambiguity in comparing the process variation, represented often as a multivariate normal distribution of an ellipsoid shape, with allowable design specifications represented as a hyper-rectangular shape. The process yield obtained from the simulation of engineering changes can provide the crucial production information before making the decision in redesigning the process architecture system.

The engineering change complexity (ECC) index is necessary for changing the process architecture design. Since the process architecture usually consists of parts and tooling elements in multiple assembly stations, the changes on one *KCC* can affect other *KCCs* and the overall final product dimensional quality. The ECC index can indicate the scope of work to implement suggested engineering change with simultaneously taken into consideration all changes resulting from the predicted change propagation. The proposed ECC index represents engineering change of the process architecture design and depends on three major factors which are: (i) dependency between design tasks; (ii) the number of *KCCs* in multi-stage assembly system; and (iii) the number of *KCCs* which have functional dependence. The dependencies between *KCCs* are caused by design functional requirements such as assembly process requirements to fully constrain a part during assembly process by using fixture locators and part-to-part joints as shown in Figure 3.15. In many complex products where all parts and systems are closely linked, changes to one part of the system are highly likely to result in a change to another part, which in turn can propagate further. The greater the connectivity, called in this thesis dependency, between systems, the greater is the chance that a change to one system leads to changes in other systems. Many industries, such as the automotive industry, are working on modular designs, which clearly defined interfaces between sub-systems, to reduce the complexity caused by inter-dependencies within their products and facilitate the reuses of sub-systems across a product range. The dependencies between *KCCs* cause that the process architecture might display several features of complex systems. For example, deterministic chaos is apparent when apparently insignificant changes to a design specifications lead to a considerable variation in the design and associated cost to bring it to market.

To address challenges in defining complexity index, this chapter adopts the *KCC* functional dependence model and the *Task Connectivity Matrix* presented in Chapter 3 to describe the dependency of *KCCs*, and also proposes the *weight matrix* to represent the number of *KCCs* in each *KCC-HG*. The proposed ECC index provides a comparison of the *KCC* dependency between the current process architecture and the hypothetical process architecture where all *KCC-HGs* are fully dependent to each other. The proposed ECC index helps to looks at the potential causes and effects of changes, and analyses the formal and informal processes that are used to handle change. It describes a scale of engineering change propagation in changing the process architecture design. For example, the suggested change of the process architecture become problematic when a change to one system propagates to other systems, because the tolerance margins of individual parameters are exceeded or production yield is reduced.

Finally, the cost assessment is also incorporated into the proposed *KPI* model to assess of engineering change impacts. Since the cost structure of the product and process is very complex, this chapter focuses on evaluating only cost of adjusting *KCC* tolerances in order to maintain the process yield at the same level as prior to implementing engineering changes. Minimizing the cost of *KCC* tolerances can be set as the design objective in the optimisation scheme subject to the desired process yield requirement. The proposed cost analysis can be incorporated into a comprehensive overall cost analysis of redesigning process architecture.

The rest of this chapter is organized as follows. The process yield for assessing the impact of engineering change on product dimensional quality is described in Section 4.2. The engineering change complexity is presented in Section 4.3. The cost in adjusting the *KCC* tolerances based on tolerance optimisation is presented in Section 4.4. Finally, the conclusions are discussed in Section 4.5.

4.2 The Change of Process Yield

The engineering changes of the current product and process architecture design can affect final product quality (e.g., deterioration of process yield as not all products are conforming to quality standard) or deterioration of product quality (e.g., increase of process variation). Therefore, it is necessary to assess or predict the impact of engineering changes on the current product quality. This section proposes to utilize the SOVA model developed in Chapter 2 to predict the product dimensional quality. The dimensional quality represented by a set of *KPCs*, $\mathbf{y}$, can be obtained from:

$$\mathbf{y} = \mathbf{A}\mathbf{u} \tag{4-1}$$

where $\mathbf{A}$ is SOVA model, and $\mathbf{u}$ is the *KCC* variations

Specifically in automotive body or aircraft fuselage and wing, there are multiple *KPCs* which have to be evaluated simultaneously. This poses the difficulty in assessing the final product quality by using the traditional quality indices such as process capabilities indices or sensitivity analysis.

4.2.1 Proposed Process Yield

The process yield is defined as the joint probability of all *KPCs* simultaneously being within their respective specification ranges as shown in Eq. (4-2).

$$\text{Yield}\,(\boldsymbol{\Lambda}) = \Pr\left\{\bigcap_{i=1}^{m} \mathrm{LSL}_\mathrm{i} \le KPC_i \le \mathrm{USL}_\mathrm{i}\right\} \tag{4-2}$$

where $\boldsymbol{\Lambda}$ is the process architecture design such as nominal designs of fixture locator positions or part-to-part joint; and LSL_i and USL_i are the lower and upper specification limits for KPC_i, respectively.

Under normality assumption, the distribution of *KCC* variation, u_i is expressed in Eq. (4-3):

$$f_i(u_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{u_i-\mu_i}{\sigma_i})^2} \text{ and } \sigma_i = \frac{t_i}{3}, \mu_\mathrm{i} = 0 \tag{4-3}$$

The yield is defined as:

$$\Pr\left\{\bigcap_{k=1}^{m}(L_k \le y_k \le U_k)\right\} = \int_{L_m}^{U_m}\dots\int_{L_1}^{U_1} q(y_1, y_2, \dots, y_m)\varphi(y_1, y_2, \dots, y_m)dy_1 dy_2 \dots dy_m \quad (4\text{-}4)$$

The process yield can be estimated by using Monte Carlo technique by simulating k **KCC** vectors, $\mathbf{KCC}_1, \mathbf{KCC}_2, \dots, \mathbf{KCC}_k$, where $\mathbf{KCC}_i = \begin{bmatrix} KCC_1 & \dots & KCC_n \end{bmatrix}_i^T$. A variation of each *KCC* expressed in Cartesian Coordinate, $(\delta x, \delta y, \delta z)$, is randomly generated based on its statistical characterizations. Then, the $\mathbf{KCC}_i; i = 1, \dots, k$, is substituted into Eq. (4-1) to obtain a vector of *KPC* variations, $\mathbf{KPC}_1, \dots, \mathbf{KPC}_k$, where $\mathbf{KPC} = \begin{bmatrix} KPC_1 & \dots & KPC_m \end{bmatrix}^T$.

$\Phi(\mathbf{KPC}_i)$ is a function to provide a response whether all *KPCs* are in-specification windows. If all *KPC* variations are within specification windows; $\mathbf{LSL} \le \mathbf{KPC}_i \le \mathbf{USL}$, then $\Phi(\mathbf{KPC}_i) = 1$, otherwise $\Phi(\mathbf{KPC}_i) = 0$. Thus, yield can be expressed as:

$$\text{Yield } (\mathbf{\Lambda}) = \frac{\sum_{i=1}^{k}\Phi(\mathbf{KPC}_i)}{k} \times 100\% \quad (4\text{-}5)$$

The proposed process yield offers several advantages in the area of measuring the robustness of the process architecture design. The process yield can indicate the percentage of conforming product which people involving in product and process architecture design as well as in manufacturing and assembly can understand. This advantage cannot be found in using sensitivity indices such as optimality indices since the sensitivity indices are relative measures of *KPC* variations against *KCC* variations. Therefore, the sensitivity indices are difficult to be interpreted by manufacturing engineers about the relation of sensitivity indices and level of conforming product.

This also leads to another advantage of process yield in measuring and comparing robustness in different what-if scenarios in process architecture redesign. The process yield can be considered as the absolute measure since it compares *KPC* variations with allowable specifications which are given and unchanged in conducting what-if scenario analysis. On the other hand, the sensitivity analysis is the relative measure between *KPC* variations and *KCC* variations which both *KPC* and *KCC* variations can be changed in what-if analysis. Therefore, it is difficult to compare the design improvement by using sensitivity indices in the process architecture redesign.

The numerical estimation of the process yield based on Monte Carlo simulation can result in high computational efforts. This thesis also proposes an approach to expedite the computational efficiency in estimating the process yield. The impacts of engineering changes on product dimensional quality can be evaluated through the changes of the process yield which can be expressed as:

$$\Delta yield = yield_{\mathrm{eng}} - yield_{\mathrm{intial}} \tag{4-6}$$

Thus, the change of final product dimensional quality represented by the process yield expressed in Eq. (4-6) can be obtained. This thesis also presents the software tool to help in generating model represent the relationship between *KPCs* and *KCCs*. The software tool can supports in evaluating process yield. The detail of developed software is described in the next section.

4.2.2 The AutoSOVA Model Generator Development

To support process yield assessment discussed in the previous section, the software tool, called AutoSOVA Model Generator, is developed based on the variation propagation model for Type-3 assembly proposed by Phoomboplab and

Ceglarek (2009). The AutoSOVA Model Generator is also necessary for the development of the subsequent design tasks since all design tasks require constant assessment of design candidates. The information required to formulate the generalized SOVA model involves:

(i) Number of assembly stations in the system

(ii) Locator and virtual mating point positions

(iii) Constrained direction of locators and virtual mating points; and

(iv) Measurement points coordinates and assembly station where measurement is performed.

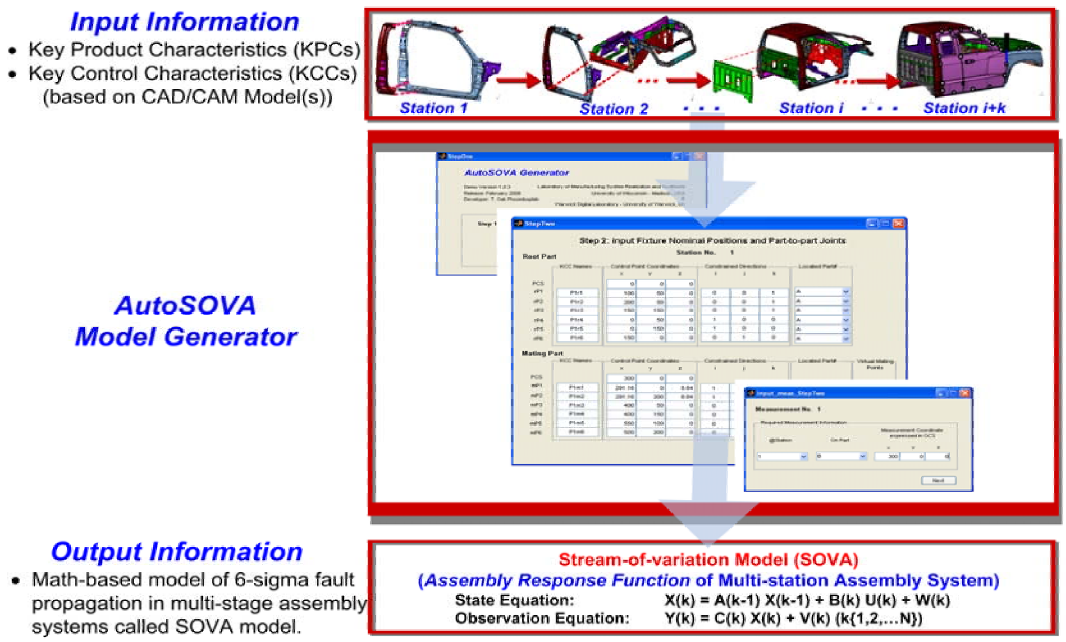


Figure 4.1: Graphic User Interface (GUI) of the AutoSOVA Model Generator.

The information can be acquired by using two approaches. First, the AutoSOVA Model Generator can acquire information from: (i) CAD/CAM design; (ii) variation analysis software such as 3DCS Analyst or Vis-VSA; or (iii) design synthesis tasks. Second, the generalized SOVA model can be generated manually by inputting the required information through the Graphic User Interface as shown in Figure 4.1.

The AutoSOVA Model Generator software is developed in MATLAB software as well as its GUI. The information regarding to process architecture can be acquired in text file format and input in GUI which consists of six steps:

(1) Input the number of assembly stations. The software will calculate the total number of *KCCs* required in the assembly system automatically and prepare template for input *KCC* information in the next step. The GUI of this step is shown in Figure 4.2.

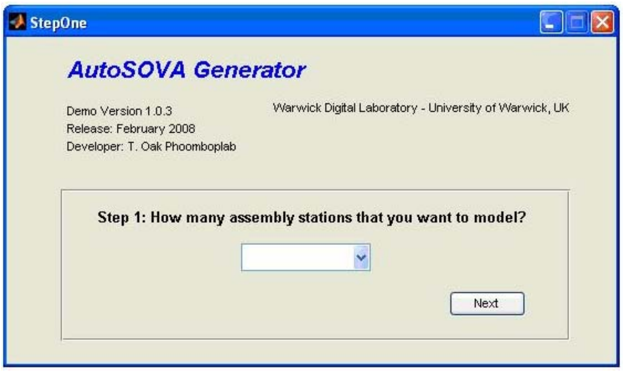


Figure 4.2: Graphic User Interface (GUI) for input number of assembly stations.

(2) Input coordinates of all *KCCs* (fixture locators and part-to-part joints) in each assembly station. In this step, the degree-of-freedom that each *KCC* constrains and Part Reference Point in Global Coordinate System are also input. The software will not allow to continue in the next step, if all fields are incomplete. The GUI of this step is shown in Figure 4.3. If the constraint directions inputted in this step is resulting in under-constraint condition, the warning screen is pop-up as shown in Figure 4.4, and checking information is mandatory.

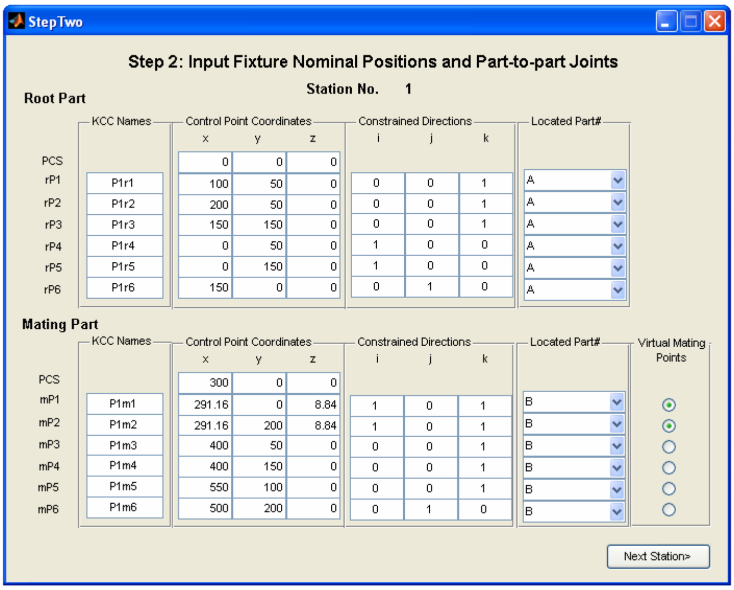


Figure 4.3: Graphic User Interface (GUI) for input *KCC* coordinates and constraint directions.

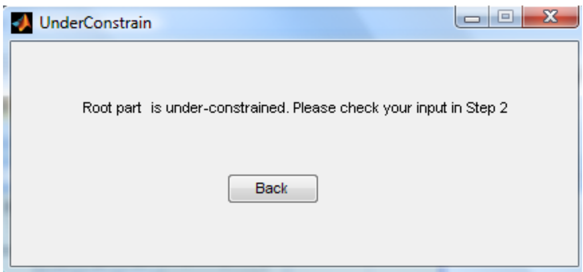


Figure 4.4: Warning screen if a part is in under-constraint condition.

(3) Input the number of *KPCs* and their coordinates. The *KPC* also can be designated the station, part that *KPC* is located on. The GUI of this step is shown in Figure 4.5.

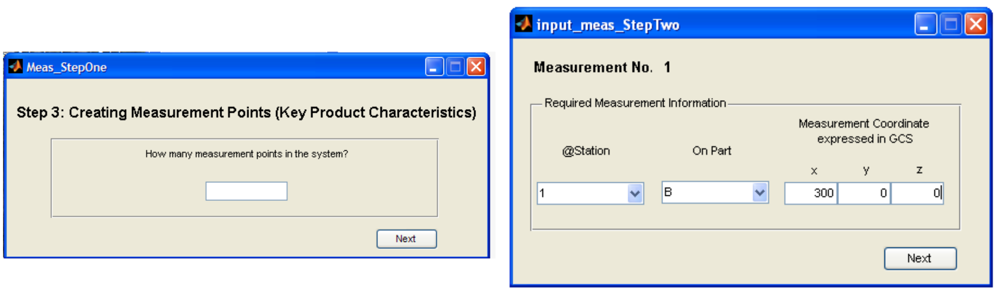


Figure 4.5: Graphic User Interface (GUI) for input *KPC* coordinates.

(4) The software shows the screen that the SOVA mode is successfully formulated. The location of SOVA model in text file format is also shown. The GUI of this step is shown in Figure 4.6.

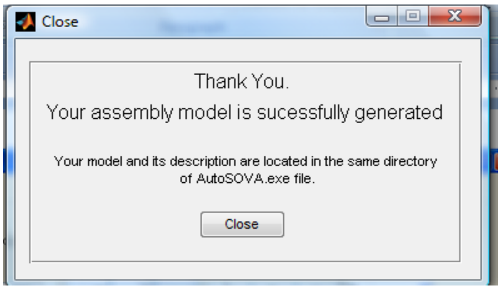


Figure 4.6: Screen shown that the SOVA model is successfully generated.

(5) The software shows the optional step in calculating the variation of KPCs by using Monte Carlo simulation technique. In Monte Carlo simulation, random number is generated based on normal distribution with 6-sigma standard deviation specified by user in tolerance range field. The number of random samples can be specified in this step. The GUI of this step is shown in Figure 4.7.

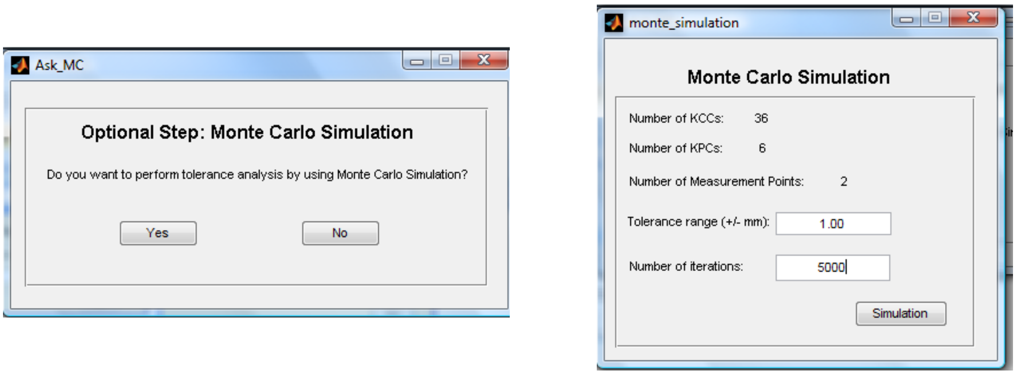


Figure 4.7: Graphic User Interface (GUI) for assessing *KPC* variations (Optional step).

(6) The software shows the result of *KPC* variations assessed by Monte Carlo simulation. The analysis of each *KPC* includes the variation in x,y, and z axes in Global Coordinate System. The GUI of this step is shown in Figure 4.8.

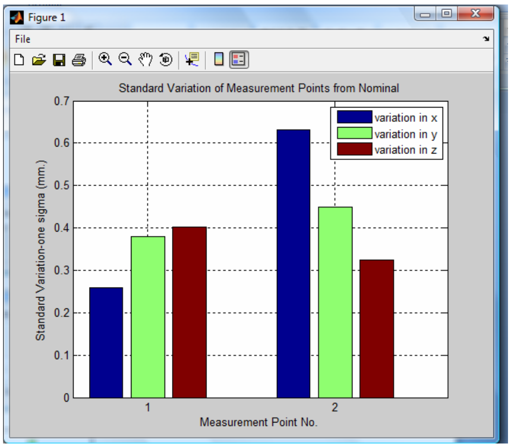


Figure 4.8: The *KPC* variation analysis result obtained from Monte Carlo simulation.

4.3 Engineering Change Complexity (ECC) Index

"*Complexity*" has increasingly received the attention from various scientific fields and endeavors to address the sophisticated challenges emerging in mathematics, physics, biology, chemistry, natural science, social science, and engineering. The study of the complex systems and the phenomena of complexity can be formulated as interdisciplinary researches which function as the framework to integrate and transfer knowledge among multiple disciplines. Flood and Carson (1993) provided a comprehensive description of the complexity which arises from a combination between a human perception and a quality of an object. The complexity in the context of the human perception can be analyzed as social phenomena while the complexity of an object can occur through: (i) the number of elements; and (ii) the diverse relations among the elements. In Sciences of the Artificial, Simon (1996) defines the complexity of a product in terms of the connections between its parts, and calls engineering products “almost decomposable systems” where connections between parts of a system can never be fully avoided.

In keeping with the scope of this study, this section focuses on the complexity of the objects in large engineering systems such as automotive body, aircraft wing and fuselage, and ship hull assemblies. The need of measures to evaluate complexity has been long acknowledged by researchers and engineers for practical uses. Complexity measure is necessary for a complex system because: (i) it can advise system designers on how to monitor and limit their system complexity; and (ii) it also hold implications about the amount of resources and time needed to adjust and redesign the system. Next, the formulation of the proposed redesign complexity measure is presented.

4.3.1 Proposed Engineering Change Complexity Index

The proposed complexity index is to measure the design complexity of the process architecture. The complexity measure is formulated based on: (i) dependency between design tasks; (ii) the number of *KCCs* which each design task has to optimise; and (iii) the number of dependent *KCCs* defined by functional dependence model. The dependency of design task is represented by the *Task Connectivity Matrix*, T_{CM}, as describe in Eq. (3-6) in Chapter 3. The number of *KCCs* that each design task has to optimise the design configurations is described by *Weight Matrix*, and the number of dependent *KCCs* is obtained from *transversal of* hypergraph analysis described in Chapter 3.

Task Connectivity Matrix

The formulation of the Task Connectivity Matrix is demonstrated in Eq. (3-6). The dependency between design tasks shown in Task Connectivity Matrix can be conducted either by transversal of hypergraph or sensitivity analysis.

KCC-HG Weight Matrix

The number of *KCCs* needs to be incorporated into the complexity measure. Although not all of *KCCs* in a particular $KCC\text{-}HG_i$ are dependent on *KCCs* in other *KCC-HGs*, this thesis assumes that a *KCC-HG* is uniform and represents one entity. Then, the number of *KCCs* of all *KCC-HGs* can be represented by *KCC-HG weight matrix* which is expressed as:

$$\mathbf{W}_{KCC-HG} = \begin{bmatrix} w_1 & w_2 & \dots & w_m \end{bmatrix}^T \tag{4-7}$$

where w_i is the number of *KCCs* within i^{th} *KCC-HG*.

KCC Functional Dependence

The *KCCs* which have functional dependence are identified by hypergraph technique. The *KCC* functional dependence causes the chain of engineering changes which result in the difficulty in changing the *KCC* design besides the dependency in design task level (i.e., design objective and constraints cause the dependency between design tasks). The number of *KCCs* which have functional dependence can be represented according to their *KCC-HG* as follow:

$$\mathbf{W}_H = \begin{bmatrix} h_1 & h_2 & \dots & h_m \end{bmatrix}^T \tag{4-8}$$

Engineering Change Complexity Index

The information regarding to dependency described by $\mathbf{T}_{\mathrm{CM}}$ in Eq. (3-6), the number of *KCCs* in each *KCC-HGs* expressed in Eq. (4-7), and the number of functional dependence *KCCs* expressed in Eq. (4-8) is used to formulate the ECC index representing the complexity in changing the process architecture design. Basically, the proposed complexity index, ECC, represents the ratio of design task dependency and number of the *KCCs* having functional dependence in the current system compared with hypothetical system that all design tasks are fully dependent on each other. The redesign complexity index, ECC, can be formulated as:

$$\mathrm{ECC} = \frac{\left\|\mathbf{T}_{CM} \cdot \mathbf{W}_{KCG}\right\|}{2\left\|\mathbf{G} \cdot \mathbf{W}_{KCG}\right\|} + \frac{\left\|\mathbf{W}_{h}\right\|}{2\left\|\mathbf{W}_{KCG}\right\|} \tag{4-9}$$

where $\mathbf{G}$ is $\mathbf{T}_{\mathrm{CM}}$ which all design tasks are dependent. $\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & \ddots & 1 \\ 1 & 1 & 0 \end{bmatrix}_{m\times m}$

If $\mathrm{ECC} = 0$, it indicates that all design tasks can be performed independently and there is no *KCC* which have functional dependence. This condition indicates the least complex in designing the system. On the other hand, if $\mathrm{ECC} = 1$, it indicates that all design tasks are totally dependent on each other, and all *KCCs* have functional dependence. This condition indicates the highest difficulty in design since it creates the closed chain of engineering change propagation.

Therefore, reducing the complexity of the system can be performed by three approaches which are: (i) reduce the dependency between design tasks; (ii) reduce the number of *KCCs* between the dependent design tasks; and (iii) reduce the number of *KCCs* which have functional dependence. Reducing the dependency is similar to the suggestion made by Axiomatic Design Theory which aims to design the sub-system as uncoupled relations. This contributes to the ease in design, control, and adjustment. On the other hand, reducing the number of *KCCs* in the system is also aligned with current practices in Design-for-Manufacturing and Assembly (DFMA). Reducing the number of *KCCs* leads to cost reduction and simplification of design throughout the supply chain. Therefore, the proposed redesign complexity index can be interpretable and aligned with the current practice in engineering design. Next, the cost incurred from adjusting *KCC* tolerances to maintain the process yield at the same level before conducting engineering changes is presented. The cost indicates the least cost that can be potentially incurred if engineering changes are performed.

4.4 Cost from Optimizing KCC Tolerances

In order to make the decision to change the current process architecture design, information regarding investment cost is crucial. However, the estimation of cost in changing the design is very complex since cost can occur in any steps of supply chains in producing and assembling product. Thus, to gain cost information related to changing of product and process architecture design, this thesis proposed to use cost in changing *KCC* tolerances to maintain product quality under the change of product and process architecture design. Cost in adjusting tolerances in order to maintain product quality is adopted since it is usually lower than cost in adjusting nominal design. Changing the nominal design affects both product and process architecture as well as related operations. For example, changing location of pin fixture causes the change of hole position on a part as well as adjusting operation in forming a part. Therefore, cost in adjusting *KCC* tolerances is the preliminary indicator for the investment required in changing the design of product and process architecture according to the requirements. The cost in adjusting tolerance can be incorporated to be a part of total investment cost estimation after the decision on changing the process and product architecture design is made.

The tolerance optimisation scheme can be formulated in order to obtain the minimum cost in adjusting *KCC* tolerance. The tolerance optimisation scheme can be expressed as:

$$\text{Min}_{\mathbf{t}} \quad \sum_{i=1}^{n} \mathrm{C}_i(t_i) \tag{4-10}$$

Subject to Process yield = the initial process yield, $t_i^L \le t_i \le t_i^U$, $t_i > 0$

where $C_i(t_i)$ is a cost function which is varied for each *KCCs* within the design. The cost $C_i(t_i)$ is inversely affected by *KCC* tolerances which can be obtained by model fitting if the cost-tolerance data are available. The type of cost function does not affect the procedure and optimality because of their common monotonic property. The model coefficients give the flexibility for tolerance-cost data fitting or process-cost knowledge inclusion, e.g. different weights can be assigned to processes to represent different cost-contributions. t_i^L, and t_i^U are the process precision limits on i^{th} *KCC* tolerance, t_i. To satisfy the optimisation scheme shown in Eq. (4-10), the tolerance optimisation technique proposed by Huang *et al.* (2009) is adopted in this thesis.

4.5 Conclusions

This chapter presents the *KPI* model which involves the formulations of (i) process yield; (ii) engineering change complexity (*ECC*) index; and (iii) the minimal cost from optimizing *KCC* tolerances to maintain product dimensional quality. The proposed *KPI* model provides the key indicators to help in making a decision on changing the existing process architecture design. The impacts of engineering change on the final product dimensional quality is evaluated by the change of process yield. The engineering change complexity can help to estimate the scale of engineering change propagation in the current system. Lastly, the cost to adjust *KCC* tolerances in order to maintain product quality is obtained to indicate the least cost that potentially occurs if the engineering change is implemented. Based on these three indices, the decision to change the process architecture design can be made. In the next chapter, a systematic approach to change and optimise the process architecture design based on design synthesis framework is presented. This involves a framework to integrate the multiple design tasks that are needed to change the current process architecture design in order to achieve the new optimum and design constraints within the minimum computational time.

CHAPTER 5

DESIGN TASK INTEGRATION BASED ON HYPERGRAPH TECHNIQUE FOR ADJUSTING PROCESS ARCHITECTURE DESIGN

5.1 Introduction

In this chapter, the design synthesis framework is proposed to integrate the individual design tasks mentioned in Chapter 2. The proposed framework can support in determining the sequence of design tasks which allow achieving the optimal *KCC* design within the reasonable computational time. The integration of design tasks to adjust and optimise the process architecture design is necessary due to the interdependencies that exist between design tasks. First, such interdependencies can potentially lead to time-consuming iterations that are needed for simulations to converge to optimal design configurations. Second, these interdependencies create a need for a Pareto of design tasks to simultaneously satisfy dimensional management quality goal (as measured, for example, by Six-Sigma) and all predetermined design constraints.

Current trends in manufacturing design, specifically time-based competition which involves frequent model changes, requires an enormous investment in terms of time and cost. In recent years, these challenges have been addressed through various developments in Design-for-Quality (DFQ) and Design-for-Six-Sigma (DFSS) approaches. For example, in design processes of automotive body, shipbuilding hull, and aircraft fuselage assembly, DFQ- and DFSS-based approaches focus on dimensional variation reduction management which affects a broad range of product characteristics including production cost, downtime, product performance, functionality, and aesthetics. Although the developments of DFQ- and DFSS approaches as well as design synthesis tasks described in Chapter 2 can help in reducing the *KPC* variations by optimizing a particular group of *KCCs*, there still lacks a design synthesis framework which can integrate multiple design synthesis tasks to optimise the multiple groups of *KCCs*. Phoomboplab and Ceglarek (2007) proposed the methodology to integrate design tasks by using Hybrid-Design Structure Matrix and Task Flow Chain. However, it still lacks of clear approach in analyzing the dependency between design tasks. To address this challenge, this thesis proposes a methodology based on transversal of hypergraph and task connectivity matrix in analyzing the dependency between design tasks and generating design task sequences. The transversal of hypergraph and task connectivity matrix are introduced in Chapter 3 of this thesis. The DSM and Task Flow Chain proposed by Phoomboplab and Ceglarek (2007) are used only for graphical representation of design task hierarchy.

The contents of this chapter are presented in the following sections. Section 5.2 describes the proposed methodology for the design synthesis framework. In Section 5.3, a case study illustrates the application of the proposed design synthesis framework. Finally, conclusions are presented in Section 5.4.

5.2 Proposed Methodology

In the case that several design tasks are required to design the process architecture, it is necessary to develop the framework to integrate the relevant design tasks. Then, the implementation sequence of design tasks is determined in order to consider the trade-off among dimensional quality of final product, design constraints, and computational time in designing the assembly process. The sequence of design tasks can be defined as:

$$\boldsymbol{\zeta} = \{\mathrm{T}_i, \ldots, \mathrm{T}_k\} \tag{5-1}$$

where $\boldsymbol{\zeta} = \{\mathrm{T}_i, \ldots, \mathrm{T}_k\} \subset \{\mathrm{T}_1, \ldots, \mathrm{T}_n\}$. There are totally n design tasks for improving dimensional quality, and implementation design task sequence begins from T_i to T_k.

It is necessary to select a design task sequence, $\boldsymbol{\zeta}$, consisting of minimum number of design tasks which can improve dimensional quality to the predetermined level of Six-sigma variation. This can be formulated as:

$$\text{Objective: Select } \boldsymbol{\zeta} = \{\mathrm{T}_i, \ldots, \mathrm{T}_k\} \tag{5-2}$$

Subject to: (1) Dimensional quality < 6-sigma quality threshold

(2) $\mathbf{C}_{Ti}, \ldots \mathbf{C}_{Tk} \leq$ thresholds

where $\mathbf{C}_{Ti}, \ldots, \mathbf{C}_{Tk}$ are design constraints.

To determine the optimal design task sequence, $\boldsymbol{\zeta}$, which can address the functional requirements, this chapter proposes methodology to integrate design tasks and determine their implementation sequences. The detail of the proposed methodology is following.

5.2.1 Step 1: Determining Dependency of Design Task by using Transversal of Hypergraph

The transversal of hypergraph presented in Chapter 3 is used to determine the dependency between design tasks. The dependency between design tasks leads to four scenarios of design task implementation sequences as shown in Figure 5.1.

Classification of design task interdependency	*Uncoupled*	*Decoupled*	*Coupled*	*Optimized the same design configuration*
Design task sequence	Concurrently	Sequentially	Iteration	Algorithm portfolio

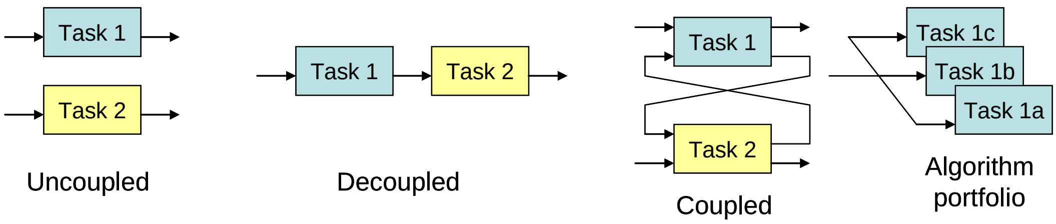


Figure 5.1: Classification of design task interdependency.

(i) Two design tasks are *uncoupled* if (i) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} = \phi$, and (ii) $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} = \phi$. These two design tasks can be conducted concurrently. The task connectivity matrix can be described as $\mathbf{T}_{CM} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

(ii) Two design task are *decoupled* if (i) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} \neq \phi$ and $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} = \phi$; or (ii) $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} = \phi$ and $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} \neq \phi$. These two design tasks can be conducted sequentially. The task connectivity matrix can be described as $\mathbf{T}_{CM} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$

(iii) Two design task are *coupled* if $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{passive2}\} \neq \phi$ and (ii) $\{\mathbf{KCC}_{active2}\} \cap \{\mathbf{KCC}_{passive1}\} \neq \phi$. Then these design tasks are *coupled* and can be conducted *sequentially* with additional *iteration loops*. The task connectivity matrix can be described as $\mathbf{T}_{CM} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

(iv) Two design tasks are described as an *algorithm portfolio* if $\{\mathbf{KCC}_{passive1}\} = \{\mathbf{KCC}_{passive2}\}$ and $\{\mathbf{KCC}_{active1}\} \cap \{\mathbf{KCC}_{active2}\} \neq \phi$. Two design tasks optimise the same design configuration using two different algorithms and can be conducted *concurrently* which the better solution is selected using performance measure. The example of algorithm portfolio is the multi-fixture layout optimisations proposed by Kim and Ding (2004) and Phoomboplab and Ceglarek (2008). Although both methodologies aim to determine the optimal fixture locator positions, the former is based on the optimality criteria while the latter is based on a process yield.

The Design Structure Matrix (DSM) (Phoomboplab and Ceglarek, 2007) can be used to graphically represent the dependency between design tasks as shown in Figure 5.2. However, the dependency between design tasks shown in DSM still needs to be analyzed by hypergraph technique. Nevertheless, DSM can provides additional detail regarding to *KCCs* which are constant design configurations, **Φ** (passive *KCCs* which are necessary for conducting a design task), and design constraints.

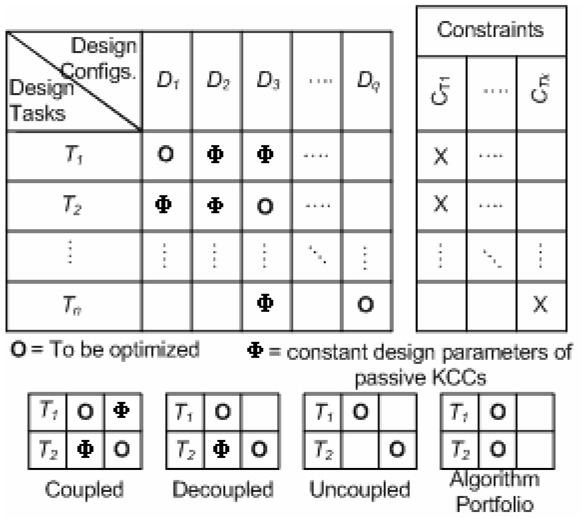


Figure 5.2: Design Structure Matrix proposed by Phoomboplab and Ceglarek (2007).

The columns of the DSM represent the design configurations which resulted from mapping dimensional quality, *KPC*s, *KCC*s, and *KCC-HG*s domains as illustrated in Figure 5.2. The rows of the DSM represent the design tasks, T_i, in synthesizing a design configuration, D_i. The design task constraints are also incorporated into the DSM. To formulate the DSM, design configurations have to be classified into three groups corresponding to each design task. These three groups of design configurations are: (i) design configuration to be optimised shown as "**O**" in Figure 5.2; (ii) design configuration that is a constant design parameters ($\mathbf{\Phi}$) or passive *KCCs*; and, (iii) design configuration which is not considered in the design task represented by empty spaces in the DSM.

The coupled relation between design tasks can be revealed by using the DSM. For example, configurations D_1 and D_3 cause the coupled relations between design tasks T_1 and T_2 as shown in Figure 5.2. This means that the optimised design configuration from one design task can be an input or a constant design parameter for the other design task. Thus, the coupled design tasks cause the iterations in order to converge to optimal design which usually requires significant computational time. To justify the sequence between these two design tasks, the dependency between two design tasks is determined in Step 2 which adopts the concept of sensitivity analysis proposed by Krishnan *et al.* (1997) to approximate the level of dependency between design tasks.

5.2.2 Step 2: Sensitivity Analysis to Approximately Decouple Design Tasks

In this step, the aim is to minimize the computational time needed in the iterations of the coupled design tasks in converging to optimal design solutions. After the coupled design tasks are identified in the previous step, sensitivity analysis is also used as an approach in approximately decoupling the coupled design tasks into sequential design task implementation. The concept of the sensitivity analysis for decoupling the design tasks proposed in this chapter is adopted from Krishnan *et al.* (1997).

One of the most challenging issues is to define the design task sequence for coupled design tasks since coupled design tasks usually require a significant number of iterations to converge to optimal solutions. The coupled relation between two design tasks can be basically described as a design scenario with two design tasks A and B using each other design outputs , $\{\mathbf{KCC}_{active}\}$, as inputs $\{\mathbf{KCC}_{passive}\}$ for conducting the design tasks. By using sensitivity analysis approach, the coupled design task can be expressed in Eq. (5-3).

$$\text{if } \frac{\partial \mathbf{D}_1}{\partial \mathbf{\Phi}_1} \neq 0 \text{ and } \frac{\partial \mathbf{D}_2}{\partial \mathbf{\Phi}_2} \neq 0, \text{ then } \mathrm{T}_1 \text{ and } \mathrm{T}_2 \text{ are coupled or } \mathbf{T}_{CM} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (5\text{-}3)$$

Sensitivity Analysis to Decouple Design Tasks

Conducting iterations between coupled design tasks can take a significant computational effort to achieve the optimal design solution. Thus, iterating between coupled design tasks can be justified as tradeoff between reaching the optimal solution and computation time efforts. The initial sequence between two tasks T_1 and T_2 can be determined using sensitivity analysis.

$$\text{if } \frac{\partial \mathbf{D}_1}{\partial \mathbf{\Phi}_1} > \frac{\partial \mathbf{D}_2}{\partial \mathbf{\Phi}_2}, \text{ then } \mathrm{T}_2 \rightarrow \mathrm{T}_1 \text{ or } \mathbf{T}_{CM} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad (5\text{-}4)$$

Eq. (5-4) can be interpreted as the optimal design configuration D_2 obtained from T_2 is less sensitive to the changes of its constant design parameter ($\mathbf{\Phi}_2$) than optimal design configuration D_1 obtained from T_1. Therefore, T_2 is less dependent on information included in D_1,($D_1 \subseteq \mathbf{\Phi}_2$) and thus all initial conditions for D_1 can be arbitrary set in conducting design task T_2. To evaluate optimal design configurations from design tasks T_1 and T_2, an assembly response functions such as SOVA model (i.e., model generated from AutoSOVA Model Generator) as presented in Chapter 4 can be used subjected to design problems.

5.2.3 Step 3: Design Task Hierarchy Development Based on Task Connectivity Matrix

After the coupled design tasks are decoupled by using sensitivity analysis in the previous step, the sequence of design tasks can be developed. In practical, several designs might be selected in order to adjust the process architecture design. Thus, the hierarchy of these design tasks can be determined based on their dependency level. The combination of any two (or more) selected design tasks can be formulated into design task sequence. For example, task connectivity matrix of three design tasks after decoupled can be obtained as:

$$\mathbf{T}_{CM} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

These three design tasks lead to seven combinations of design task sequences which are $\zeta_1 = \{T_1\}$, $\zeta_2 = \{T_1, T_2\}$, $\zeta_3 = \{T_1, T_2, T_3\}$, $\zeta_4 = \{T_2\}$, $\zeta_5 = \{T_2, T_3\}$, $\zeta_6 = \{T_1, T_3\}$ and $\zeta_7 = \{T_3\}$. However, when several design tasks involve in determining the design task sequence, it is difficult to represent the design task sequences by the task connectivity matrix. Therefore, *Task Flow Chain*, *TFC,* introduced by Phoomboplab and Ceglarek (2007) is adopted to represent the sequences of design tasks as shown in Figure 5.3. The most independent design task is placed on the upper left position and all other design tasks which are less independent are placed in cascading mode to the most dependent design task at the lower right of the diagram. The optimal design configurations from design tasks at higher level of independency can be used as a constant design parameter in the lower level design tasks. There are seven design task sequences generated from TFC shown in Figure 5.3.

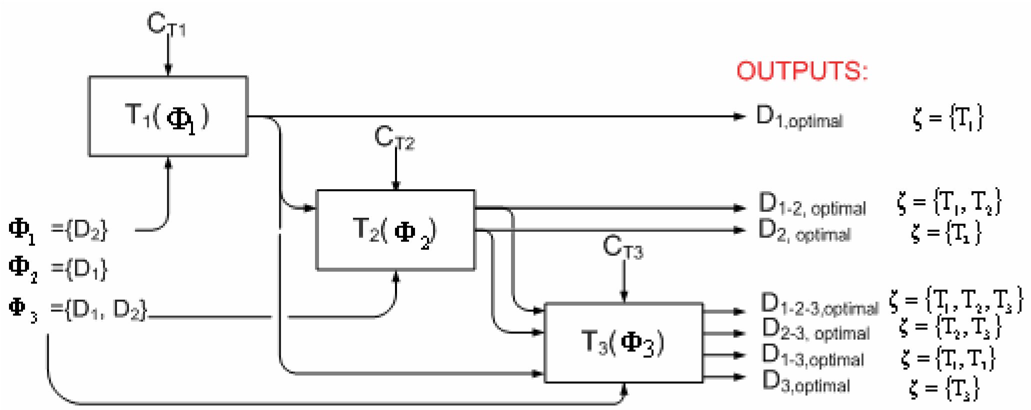


Figure 5.3: Task Flow Chain (TFC) diagram proposed by Phoomboplab and Ceglarek (2007).

The design task is optional for improving the dimensional quality of final product. Therefore, it is not necessary to implement all available design tasks. Only a few design tasks can be selected and used to determine the optimal design solutions within design constraints and reasonable computation time. To minimize the computation time of selected design tasks, the implementation of these design tasks should avoid iterations.

Specifically, the optimization objective shown in Eq. (5-2) is to select a design task sequence, $\zeta = \{T_i, \ldots, T_k\}$, which can improve dimensional quality of the predetermined level of Six-sigma variation and design constraints. To obtain a design task sequence which leads to minimum computational time, the number of design tasks in a design task sequence would be minimized. Thus, the challenge involves the decision on selecting design tasks to adjust the existing process architecture. In this thesis, two guidelines are proposed to apply on industrial design applications. The first guideline is the bottom-up of hierarchical design tasks in the TFC diagram which is illustrated in Figure 5.4. The most dependent design task is recommended to be selected first which is located at the lowest level of TFC diagram. If the most dependent design task is unable to adjust the process architecture design to meet the requirement, a design task in the next level in TFC is incorporated to generate the design task sequence together with the most dependent design task. This iteration can be performed to gradually involve design tasks in adjusting process architecture design until the requirements are met. The criteria in determining order in evaluating generated design task sequences are: (i) most dependent design task, and (ii) number of design tasks in the generated sequence. The proposed guideline aims to limit the impacts of engineering change propagation and keep the design adjustment simple.

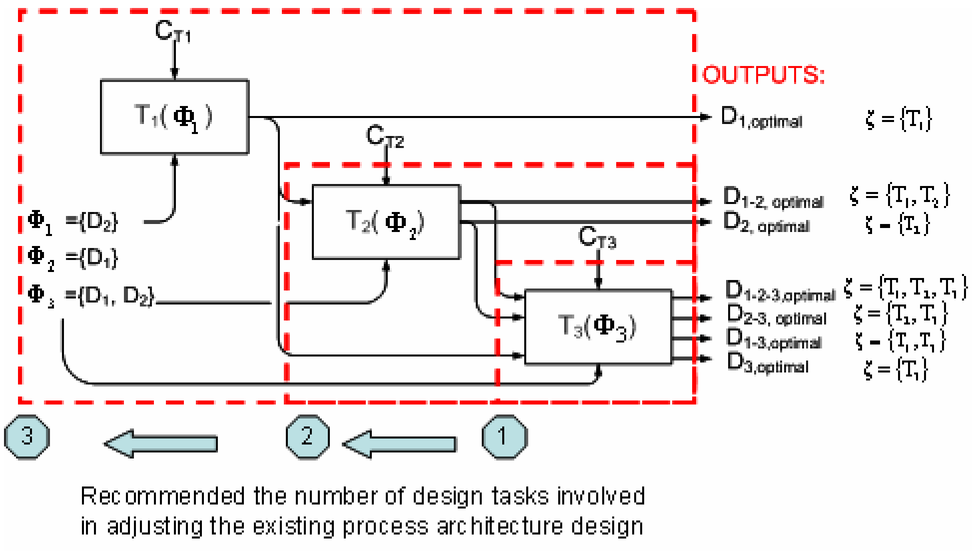


Figure 5.4: Selection of design task sequences based on bottom-up approach.

For example, the suggested priority in implementing design task sequences shown in Figure 5.4 can be determined as follow:

If selected only two design tasks, the priority orders in evaluating generated design task sequences are:

Priority 1: $\zeta_1 = \{T_1\}$

Priority 2: $\zeta_4 = \{T_2\}$

Priority 3: $\zeta_2 = \{T_1, T_2\}$

If selected three design tasks, the priority order in evaluating generated design task sequences are:

Priority 1: $\zeta_1 = \{T_1\}$

Priority 2: $\zeta_4 = \{T_2\}$

Priority 3: $\zeta_6 = \{T_3\}$

Priority 4: $\zeta_2 = \{T_1, T_2\}$

Priority 5: $\zeta_6 = \{T_1, T_3\}$

Priority 6: $\zeta_5 = \{T_2, T_3\}$

Priority 7: $\zeta_3 = \{T_1, T_2, T_3\}$

Based on the examples of design task sequences shown above, the lower number of design tasks involved in a design task sequence will have higher priority. However, if a functional design can be predetermined for adjustment, the aforementioned guideline might not be the most effective approach. This leads to the other guideline in selecting the design task sequence.

The other guideline in selecting design tasks to adjust the existing process architecture design is the selection of one design task together with tolerance optimization. The proposed guideline can be illustrated in Figure 5.5. The concept of this approach is that tolerance optimization can be considered as a superficial adjustment of process architecture design while the other design task focus on adjusting core structure of the design which is predetermined to be adjustment target. However, before using this guideline to select a design task, the careful review of dependency between design tasks must be performed. This will prevent the change propagation which affects to other functional designs unintentionally.

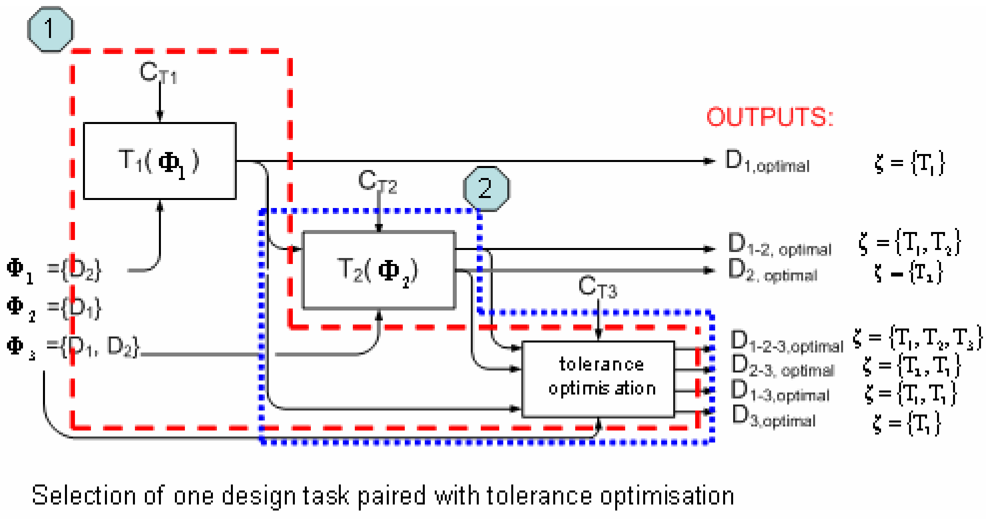


Figure 5.5: Selection of design task sequence based on a design task together with tolerance optimisation.

For example, if only design task T_1 and tolerance optimization, T_3, are selected for formulating Task Flow Chain. The order in evaluating generated design task sequences are:

Priority 1: $\zeta_1 = \{T_1\}$

Priority 2: $\zeta_2 = \{T_1, T_3\}$

Based on this guideline, the predetermined design task is selected which can reduce the computational effort. Nevertheless, if the selected design task sequence cannot meet the design objective and design constraints, another design task which is closely related to the design task T_1 (i.e., it can be identified from Task Connectivity Matrix) has to be introduced to generate design task sequence.

Evaluate and Select Design Task Sequence

Finally, the generated design task sequences from the previous step are evaluated in terms of their dimensional quality and constraints. At this stage, numerical simulation techniques such as Monte Carlo simulations or Computer-aided Engineering (CAE) such as Finite Element Analysis can be used to assess optimal design configurations in terms of impacts on final product dimensional quality. In this thesis, the AutoSOVA Model Generator introduced in Chapter 4 is used to evaluate the final product dimensional quality. In addition, other response functions such as cost and stability of parts can be used as the design objective in selecting the best design task sequence. Those design task sequences that meet the quality and constraint threshold can then be selected and are benchmarked for the best design task sequence option.

5.3 Conclusions

This chapter developed a design synthesis framework for a dimensional management in multistage assembly systems. The proposed methodology takes into consideration the interdependencies between design synthesis tasks to determine the optimal sequence of design tasks to satisfy dimensional management quality requirements. The proposed methodology is based on the process architecture modeling presented in Chapter 3 in analyzing the dependency of design tasks based on transversal of hypergraph and task connectivity matrix. In order to be able to visualize the dependency of several design tasks simultaneously, the Design Structure

Matrix and Task Flow Chain are adopted. Finally, the design task sequences can be generated and evaluate for the best design adjustment options. In the next chapter, the applications of the proposed design synthesis framework and methodology as presented in Chapters 3 to 5 are demonstrated by applying toward the redesign of the automotive body assembly in three different engineering change scenarios.

CHAPTER 6

ASSEMBLY PROCESS REDESIGN BASED ON SELF-RESILIENT DESIGN SYNTHESIS: CASE STUDIES

6.1 Introduction

In this chapter, the proposed methodologies presented in Chapters 2 through 5 are applied on an automotive body design in three scenarios of engineering changes. The engineering changes of these three scenarios involve: (i) changes in *KPC* allowable specifications; (ii) changes in assembly sequence; and (iii) changes in part geometry. The methodology begins with: (i) partitioning the system into *Key Control Characteristic-Hierarchical Groups* and selecting relevant design tasks; (ii) assessing *KPI* model of the changes in terms of process yield, engineering change complexity, and cost; and (iii) integrating the selected design tasks to optimise the process architecture design.

The design tasks which are taken into consideration to design the process architecture include: (i) *KCC* tolerance optimisation; (ii) multiple fixture layout design optimisation; (iii) part-to-part joint selection; and (iv) assembly sequence analysis. The design constraints that exist in the current design can be incorporated into the design synthesis framework such that the feasible design task sequence can be generated. As a result, the design task sequence that meets the final product quality requirement, design constraints, and shortest computational time is selected.

The rest of the chapter is organized as follows. The descriptions of case study and three scenarios of engineering changes are presented in Section 6.2. Section 6.3 presents the methodology applied to analyze the impacts of engineering changes. In Section 6.4, the simulation results of three scenarios are presented. Finally, conclusions are drawn in Section 6.5.

6.2 Case Study Descriptions

The developed methodology is demonstrated by applying it on an automotive underbody assembly process. The Floor Pan Assembly is one of the subassemblies in the automotive underbody which consists of four parts: Floor Pan Left and Right, and Bracket Left and Right, assembled in three stations as shown in Figure 6.1. Dimensional quality of Floor Pan Assembly is characterized by 12 *KPC*s which depends on the variations of 63 critical *KCC*s. In each assembly station, there are 21 *KCCs* which can be categorized into three groups as follows:

- The first nine *KCCs* are on fixtures. The first six *KCCs* constrain a root part and the other three *KCCs* constrain the mating part. The feasible tolerance region of nine fixtures is assumed to be 0.06 ~ 0.15 mm. The cost function is $C(t)=2+10/e^{2t}$.
- The *KCCs* indexed from 13 to 15 and 19 to 21 are defined on linear mating features in which three of them are located on a root part or a subassembly and the other three are located on a mating part. The feasible region of the linear mating features is 0.10 ~ 0.25 mm. The cost function is $C(t) = 1+15/e^{t}$.
- The *KCCs* indexed from 10 to 12 and 16 to 18 are for angular mating features in which three of them are located on a root part or a subassembly

and the other three are located on a mating part. The feasible region of the angular mating features is 0.10~0.25 deg. The cost function is $C(t) = 3+20/e^{3t}$.

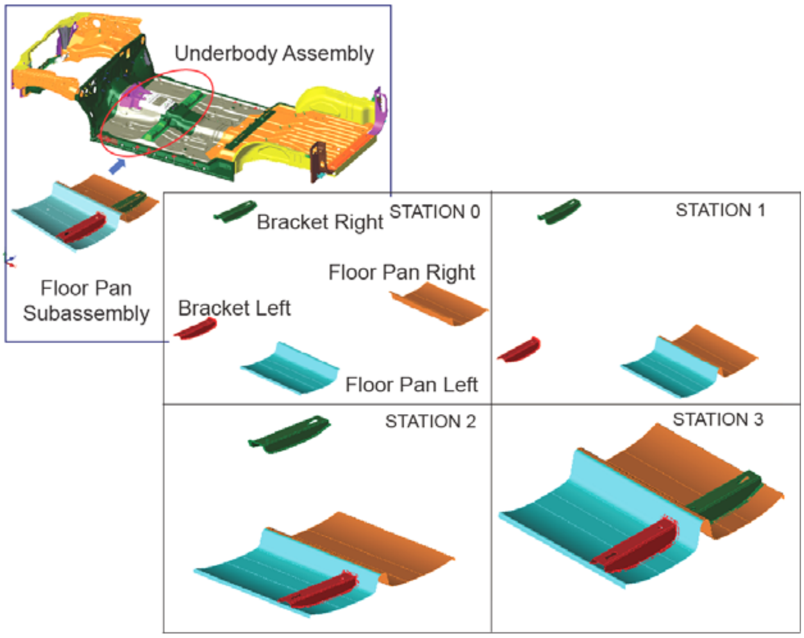


Figure 6.1: Automotive underbody assembly.

The proposed assembly process redesign methodology that is applied to this case study in three different scenarios are: (i) *KPC* specification window changes; (ii) assembly sequence changes; and (iii) the part geometry changes.

In the first scenario, the manufacturer aims to improve the dimensional quality of the assembly by setting higher specification requirements. The allowable specifications ranges of 12 *KPCs* are smaller than the initial allowable specification windows as shown in Table 6.1 by 25 percent. In the second scenario, the assembly sequence is changed to reduce the lead time in assembly the Floor Pan. The left and right sections of the Floor Pan can be performed simultaneously as shown in Figures 6.2 and 6.3 which differ from the previous design that all part components are assembled sequentially. Finally, the engineering changes in the third scenario involve the change in part geometry of the Floor Pan Left and Right to increase the space for the engine compartment as shown in Figure 6.4 where the 12 *KPC* specifications and the assembly sequence are not changed from the initial design. The proposed methodology is applied on these three scenarios as follows:

Table 6.1: Scenario 1: Initial *KPC* specification windows.

KPC No.	1	2	3	4	5	6	7	8	9	10	11	12
Specification (+/- mm.)	0.20	0.30	0.20	0.40	0.20	2.50	0.40	0.40	1.00	0.60	0.20	3.00

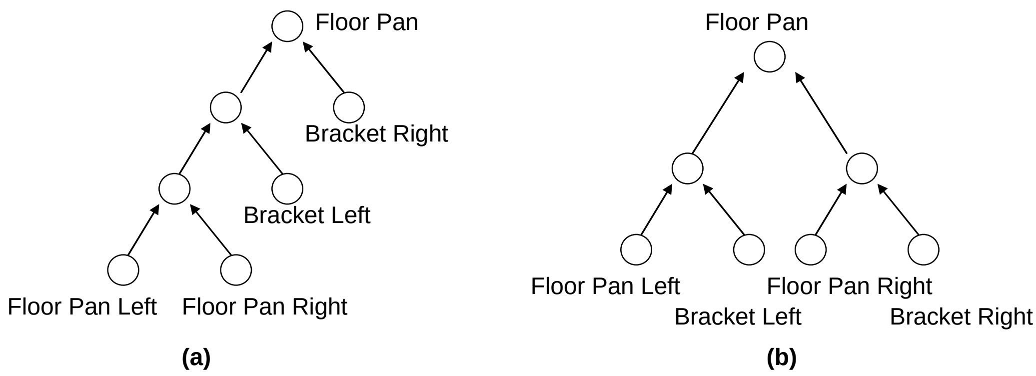


Figure 6.2: Scenario 2: Floor Pan Assembly sequence: (a) initial assembly sequence design; and (b) the new assembly sequence design.

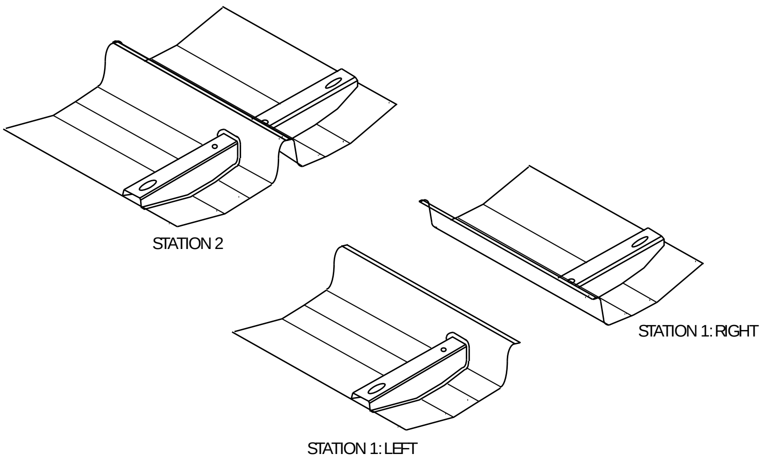


Figure 6.3: Scenario 2: A new assembly sequence of Floor Pan Assembly sequence.

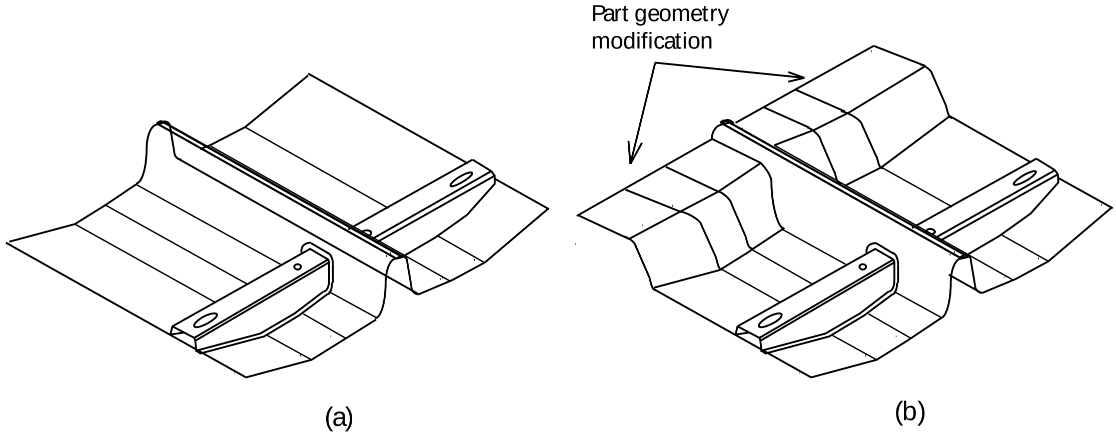


Figure 6.4: Scenario 3: Geometry changes of Floor Pan Left and Right: (a) original design; (b) change in part geometry.

6.3 Methodology

The methodology begins with partitioning the process architecture into *KCC-HGs*. The relevant design tasks can be selected according to *KCC-HGs*, and the formulation of Task Connectivity Matrix to represent the dependencies among *KCC-HGs* can be obtained. Next, the *KPI* model assessment of the current system proposed in Chapter 4 is performed to gain information in terms of: (i) process yield; (ii) engineering change complexity; and (iii) cost. Finally, to change the process architecture design, the design synthesis framework introduced in Chapter 5 is conducted by formulating the Design Structure Matrix and Task Flow Chain by based on transversal of hypergraph and Task Connectivity Matrix, respectively. Therefore, the optimal design solution can be obtained which can achieve the design objectives and design constraints within minimum computational time. The overall procedure in redesigning the assembly system is illustrated in Figure 6.5

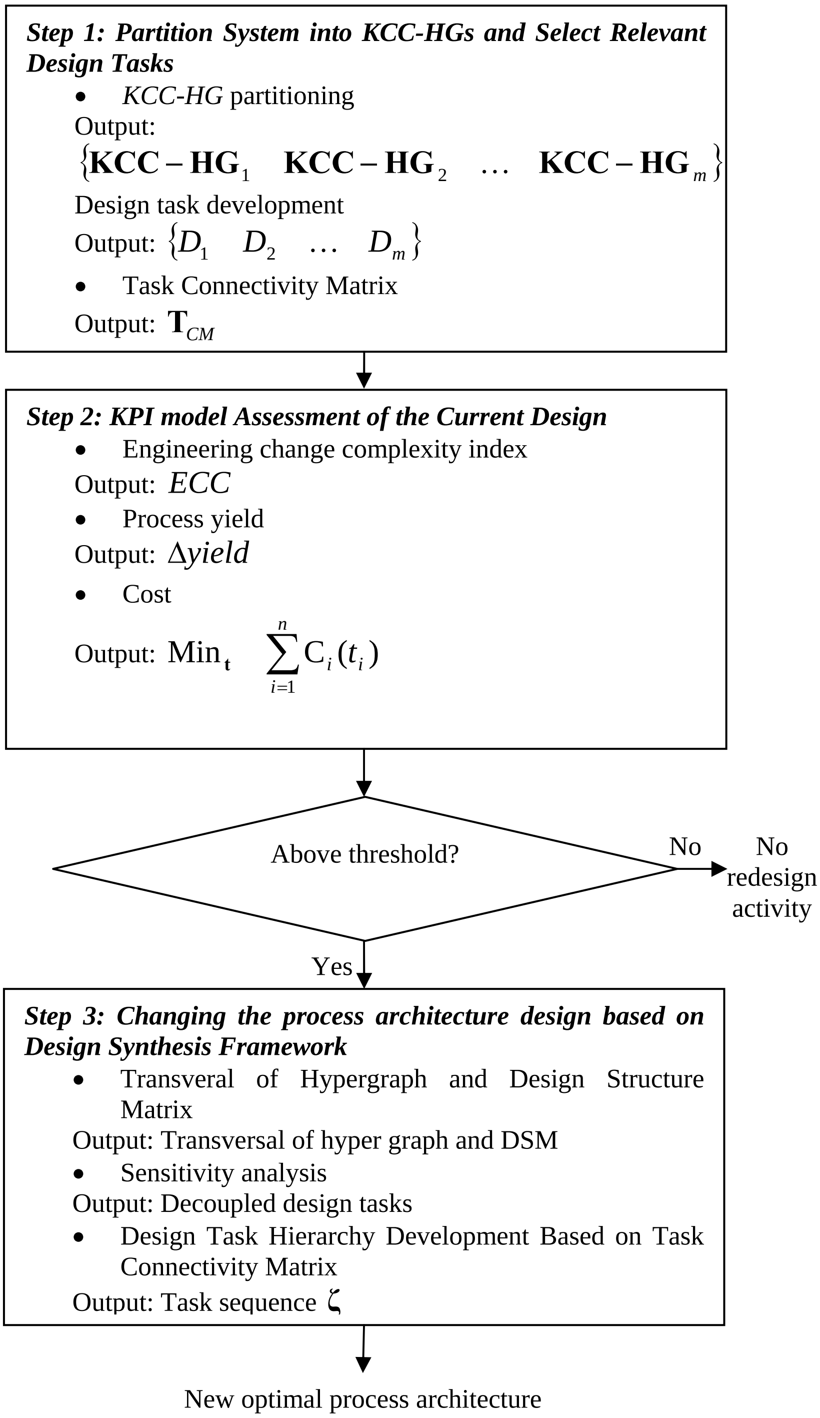


Figure 6.5: The overall procedure in redesigning an assembly system.

Step 1: Partitioning System into KCC-HGs and Selecting Relevant Design Tasks

The *KCCs* in the system are partitioned and grouped into *KCC-HGs*. The hypergraph based on *KCC-HG* partitioning can represent the process architecture as:

$$\mathbf{PA} = (\mathbf{X}, \mathbf{E}) \tag{6-1}$$

where $\mathbf{X} = \{P_1, P_2, \ldots, P_n\}$; and

$$\mathbf{E} = \{\mathbf{KCC-HG}_1, \ldots, \mathbf{KCC-HG}_m\}$$

The design task of each group of *KCC-HG* is developed by customized optimisation algorithm to be suitable with each design group. The design tasks are selected for the system can be defined as:

$$\{\mathrm{T}_1 \quad \mathrm{T}_2 \quad \ldots \quad \mathrm{T}_m\} \tag{6-2}$$

Then, the Task Connectivity Matrix, $\mathbf{T}_{\mathrm{CM}}$, is defined by using sensitivity analysis as presented in Chapter 5. Next, the $\mathbf{T}_{\mathrm{CM}}$ is used in the next step to evaluate the complexity in redesigning the current assembly system.

Step 2: KPI Model Assessment of the Current Design

The impacts of engineering changes on the current system are assessed in terms of the engineering change complexity in the process architecture, product dimensional quality represented by process yield, and cost in adjusting tolerances. These indices in the *KPI* model are formulated as shown in Chapter 4. The engineering change complexity index can be expressed as:

$$\mathrm{ECC} = \frac{\left\|\mathbf{T}_{CM} \cdot \mathbf{W}_{KCG}\right\|}{2\left\|\mathbf{G} \cdot \mathbf{W}_{KCG}\right\|} + \frac{\left\|\mathbf{W}_h\right\|}{2\left\|\mathbf{W}_{KCG}\right\|} \tag{6-3}$$

where $\mathbf{G}$ is $\mathbf{T}_{\mathrm{CM}}$ which all design tasks are dependent. $\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & \ddots & 1 \\ 1 & 1 & 0 \end{bmatrix}_{m \times m}$

The impacts of engineering changes on the product dimensional quality can be evaluated through the changes of the process yield shown in Eq. (6-4) which can be expressed as:

$$\Delta yield = yield_{\text{eng}} - yield_{\text{intial}} \tag{6-4}$$

Finally, the cost in optimizing *KCC* tolerances in order to maintain the product dimensional quality at the same level before applying engineering change can be obtained from Eq. (4-10) which are:

$$\text{Min}_{\mathbf{t}} \quad \sum_{i=1}^{n} \text{C}_i(t_i) \tag{6-5}$$

Subject to Process yield = the initial process yield, $t_i^L \le t_i \le t_i^U$, $t_i > 0$

Based on the *KPI* model assessment in Eqs. (6-3) to (6-5), the decision in redesigning the assembly system can be reached. Let us assume that the design changes of the process architecture are required. Therefore, the third step of the proposed methodology is conducted.

Step 3: Changing the Process Architecture Design based on Design Synthesis Framework

The design tasks are selected and integrated into the framework to optimise the process architecture design as described in Chapter 5. The general guideline in selecting the design tasks can be established based on three criteria as follows:

- Selecting the design tasks wherein the optimisation parameters such as design constraints and design objectives are relevant to the engineering changes.

- Selecting the design tasks which have uncoupled and decoupled relation first in order to avoid the iteration in redesigning process.
- If there are no uncoupled and decoupled design tasks, this chapter suggests selecting the design tasks which are the most dependent design tasks in the system in order to shorten the chain of change propagation created by the selected design tasks.

The design synthesis framework integrates the selected design tasks by: (i) defining the relations between design tasks by using transversal of hypergraph; (ii) decoupling coupled design tasks using deterministic sensitivity analysis; and (iii) generating design task sequence by using task connectivity matrix. In the next section, the case study of the proposed methodology is presented.

6.4 Case Studies

Step 1: Partitioning System into KCC-HGs and Selecting Relevant Design Tasks

The relations between dimensional quality, 12 *KPCs*, and 63 *KCCs* are presented in Figure 6.6. These 63 *KCCs* can be grouped into two *Key Characteristic Groups* (*KCC-HGs*); (i) part-to-part joint group ($KCC\text{-}HG_1$); and (ii) fixturing group ($KCC\text{-}HG_2$). The part-to-part joint group ($KCC\text{-}HG_1$) is required to define two design configurations which are: (i) the direction of degrees of freedom constrained by a part-to-part joint (D_1) and (ii) tolerances of a part-to-part joint (D_3). The fixturing ($KCC\text{-}HG_2$) is also required to define two design configurations which are: (i) the locator positions (D_2); and (ii) positional tolerances of locators in each fixtures (D_3). The process architecture of floor pan assembly represented by hypergraph can be expressed as:

$$\mathbf{PA} = (\mathbf{X}, \mathbf{E}_{KCC-HG})$$

where $\mathbf{X} = \{KCC_1, KCC_2, \ldots, KCC_{63}\}$; and

$$\mathbf{E}_{KCC-HG} = \{\mathbf{KCC-HG}_{fixture}, \mathbf{KCC-HG}_{joint}\}$$

$$\mathbf{KCC-HG}_{fixture} = \{\{KCC_1, \ldots, KCC_9\} \; \{KCC_{22}, \ldots, KCC_{30}\} \; \{KCC_{43}, \ldots, KCC_{51}\}\}$$

$$\mathbf{KCC-HG}_{joint} = \{\{KCC_{10}, \ldots, KCC_{21}\} \; \{KCC_{31}, \ldots KCC_{42}\} \; \{KCC_{52}, \ldots, KCC_{63}\}\}$$

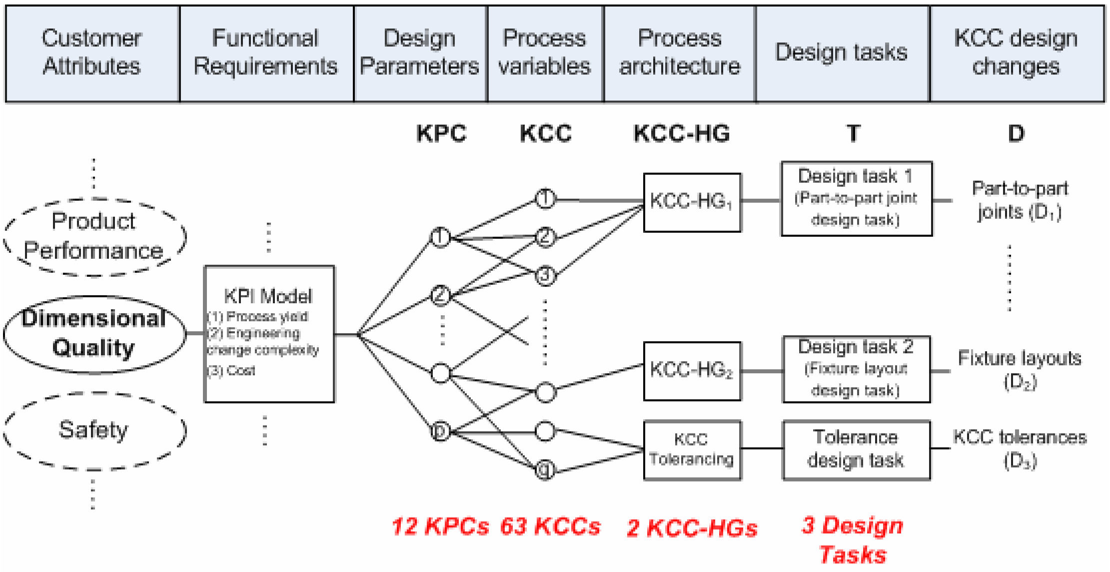


Figure 6.6: Relationships among *KPCs*, *KCCs*, *KCC-HGs*, Design Configurations, and Design Tasks of Floor Pan Assembly.

Three design tasks which are T_1, T_2, and T_3, are formulated to optimise the design configurations of part-to-part joints, fixture layouts, and *KCC* tolerances, respectively. The design tasks allow achieving the optimal design configurations of *KCC-HGs* in all three assembly stations simultaneously. The dependency of three design tasks in Floor Pan Assembly is represented by Task Connectivity Diagram. The sensitivity analysis of all pairs of design tasks is performed to determine their dependency. The Task Connectivity Diagram of the Floor Pan Assembly is expressed as:

$$\mathbf{T}_{CM} = \begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \end{array} \begin{array}{c} \begin{array}{ccc} \scriptstyle T_1 & \scriptstyle T_2 & \scriptstyle T_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \tag{6-6}$$

The part-to-part joint design task, T_1, and fixture layout design task, T_2, are dependent on each other because fixtures and part-to-part joints function together in constraining the degrees of freedom of a workpiece in the assembly system. On the other hand, tolerance optimisation design task is dependent on the part-to-part joint design task, T_1, and fixture layout design task, T_2 since both T_1 and T_2 optimise the nominal designs of *KCCs* which affect the sensitivity of *KPCs* on defining *KCC* tolerances.

The *KCCs* which have functional dependence are *KCCs* fixture locator in constraining degrees of freedom of Floor Pan Right, and *KCC* part-to-part joint between the Floor Pan Left and Right. The functional dependence occur since the selection of part-to-part joint affects the fixture planning and the subsequent fixture layout design as discussed in Figure 3.21. There is no functional dependence of *KCC* fixture locators and part-to-part joints between (i) Floor Pan Left and Bracket Left; and (ii) Floor Pan Right and Bracket Right since the part-to-part joints have to be designed as lap joint only according to part geometry. The functional dependence locators and part-to-part joint between Floor Pan Left and Right are shown as vertices in Figure 6.7 can be expressed as.

$$V = \{v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$$

$$= \{\{KCC_7, KCC_8, \ldots, KCC_{18}\}_{joint} \quad \{KCC_{19}, KCC_{20}, KCC_{21}\}_{fixture}\}$$

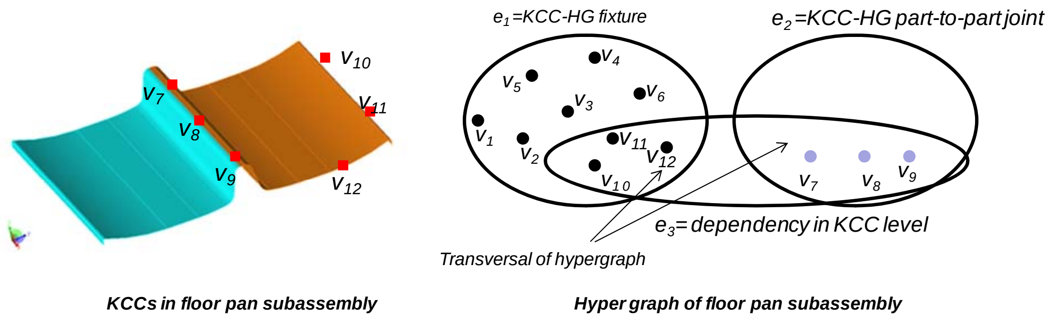


Figure 6.7: Functional dependence vertices in Floor pan subassembly and its hypergraph representation.

Step 2: KPI Model Assessment of the Current Design

Engineering Change Complexity Index

The engineering change complexity index aims to evaluate the scale of engineering change propagation in the current Floor Pan Assembly design. The complexity index is formulated based on: (i) Task Connectivity Matrix; (ii) *KCC* Weight Matrix; and (iii) *KCC* functional dependence. The Task Connectivity Matrix is obtained from Eq. (6-6). The *KCC* Weight Matrix is to indicate the number of *KCCs* in each *KCC-HG*. The *KCC* Weight Matrix can be expressed as:

$$\mathbf{W}_{KCG} = \begin{array}{c} \begin{array}{ccc} KCC-HG_1 & KCC-HG_2 & KCC-HG_3 \end{array} \\ \begin{bmatrix} 36 & \quad 27 \quad & 63 \end{bmatrix}^T \end{array} \tag{6-7}$$

The functional dependence of KCCs in Floor Pan Assembly can be expressed as:

$$\mathbf{W}_H = \begin{array}{c} \begin{array}{ccc} KCC-HG_1 & KCC-HG_2 & KCC-HG_3 \end{array} \\ \begin{bmatrix} 12 & \quad 3 \quad & 63 \end{bmatrix}^T \end{array} \tag{6-8}$$

Therefore, the redesign complexity index of the Floor Pan Assembly is:

$$\text{ECC} = \frac{\left\|\mathbf{T}_{CM} \cdot \mathbf{W}_{KCG}\right\|}{2\left\|\mathbf{G} \cdot \mathbf{W}_{KCG}\right\|} + \frac{\left\|\mathbf{W}_h\right\|}{2\left\|\mathbf{W}_{KCG}\right\|} = 0.8670 \tag{6-9}$$

where $\mathbf{G}$ is T_{CM} which all design tasks are dependent. $\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

The redesign complexity index indicates that design tasks of the current design of the Floor Pan Assembly are highly dependent (e.g., design tasks are fully dependent at $\mathrm{ECC} = 1.00$). Thus, three engineering change scenarios potentially have high impacts on the design of the current system.

To reduce the complexity of the Floor Pan Assembly or a new product development can be conducted using three approaches which are: (i) reducing the dependency between design tasks or *KCC-HGs*; (ii) reducing the number of *KCCs*; and (iii) reducing the number of *KCCs* which have functional dependence. The first approach is similar to the current practices in Axiomatic Design which aim to design the system to be an uncoupled system. On the other hand, reducing the number of *KCCs* is aligned with the practices of DFMA in minimizing the number of components and simplifying the system.

The Change in Process Yield

The engineering change scenarios exert differential impacts on the process yield of the Floor Pan Assembly. The process yield of the Floor Pan Assembly before conducting engineering changes is 86.13 percent. After three scenarios of engineering changes are applied on the current design without redesigning any *KCC*, the SOVA model for each scenario is formulated, and process yield can be evaluated by conducting Monte Carlo simulation. The process yields can be predicted as shown in Table 6.2.

In the first scenario, changing the *KPC* specification windows without redesigning the assembly process causes the increase of non-conformance items as shown in Figure 6.8 (i.e., reduction of the process yield). The specification requirements of 12 *KPCs* can be viewed as a hyper-rectangular cube and the actual variations of 12 *KPCs* on Floor Pan Assembly process can be observed as being of ellipsoid shape. The volume of ellipsoid which lies outside *KPC* specification hyper-rectangular cube is the probability of non-conformance items produced from the Floor Pan Assembly process. The process yield is reduced by 27.36 percent.

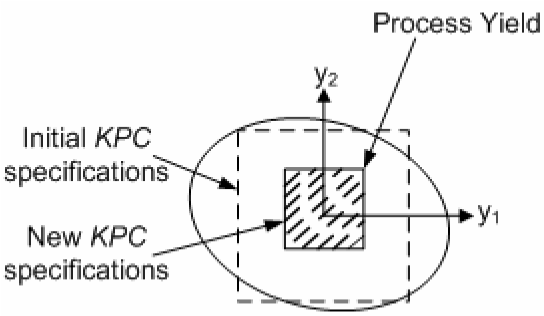


Figure 6.8: The reduction of process yield in the first scenario from changes of *KPC* specification requirements.

Table 6.2: Change of tolerance costs and process yields in the three scenarios.

Scenarios	Cost incurred from adjusting *KCC* tolerances	Process yield after applied engineering changes
Scenario I: Changing allowable *KPC* specifications	20.91%	58.76%
Scenario II: Changing assembly sequence	N/A	0%
Scenario III: Changing part geometry	N/A	0%

Remark: N/A = Not Applicable.

The engineering change of assembly sequence in the second scenario causes the initial design of fixture layout design and *KCC* tolerance assignment to become obsolete. Thus, the change of process yield is not available because both fixture layouts and *KCC* tolerances have to be redesigned first. Similarly, the current designs of fixture layouts and *KCC* tolerances cannot accommodate the part geometry changes in the third scenario. Therefore, the information about the impacts on the engineering changes on the process yield is not available.

The Cost from Optimizing KCC Tolerance

In this step, the cost in adjusting *KCC* tolerances is determined in order to maintain the process yield at the same level before applying the engineering changes. To obtain the cost information, the *KCC* tolerance optimisation scheme as shown in Eq. (6-5) is used in all three scenarios. The objective is to obtain *KCC* tolerances which incur the minimal cost subjected to the process yield at 86.12 percent.

The optimisation algorithm which is used to determine the optimal *KCC* tolerances is adopted from Huang *et al.* (2009). The cost functions of all *KCC* are assumed to be exponential functions. The cost function is $C_i(t_i) = A_i e^{-B_i(t_i)} + G$; where A_i, B_i, and G_i are model constants associated with the manufacturing cost for i^{th} *KCC* tolerance. t_i^L, t_i^U are process precision limits on i^{th} *KCC* tolerance, t_i. The cost function $C_i(t_i)$ is inversely affected by *KCC* tolerances that can be obtained by model fitting if the cost-tolerance data are available. The parameters of cost tolerance functions of three types of *KCCs* in the Floor Pan Assembly process are described at the beginning of the case studies. The type of cost function does not affect the procedure and optimality because of their common monotonic property. The model coefficients give the flexibility for tolerance-cost data fitting or process-cost knowledge inclusion, e.g. different weights can be assigned to processes to represent different cost-contributions.

The cost from *KCC* tolerance optimisation in the first scenario increases by 20.91 percent on average compared with the initial tolerance design. For the second and third scenarios, the *KCC* tolerance optimisation cannot be performed since the fixture layouts have to be redesigned first. This can imply that the cost incurred in the second and third scenarios are potentially higher than the cost incurred in the first scenario because the cost in the second and third scenarios involve changing the design of fixture layout, locating points on parts, and the new optimal tolerance design.

Step 3: Changing the Process Architecture Design based on Design Synthesis Framework

In spite of the high redesign complexity index, the impacts of three engineering change scenarios on the final product dimensional quality are significant. Hence, it is necessary to redesign the current Floor Pan Assembly system. The design synthesis framework is used for integrating the multiple design tasks to change and optimise the process architecture design. To minimize the complexity in computation, only fixture layout design task and tolerance optimisation design task are initially selected. The part-to-part joint design is given and functions as a design constraint for fixture layout optimisation design task. The interdependency between multi-fixture layout design task and *KCC* tolerance optimisation design task are identified by using DSM. Then, the level of interdependency is quantified by sensitivity analysis in order to formulate the *Task Flow Chain*, *TFC*. Finally, the design task sequence options are generated and are evaluated in the subsequent step.

Determining Dependency of Design Task by using Transversal of Hypergraph

The design tasks of multi-fixture layout design and *KCC* tolerance optimisation are analyzed in this step to determine their interdependency. As described in Chapter 3, the tolerance design task is dependent on the fixture layout design information. On the other hand, the fixture layout design task is independent from tolerance design task. Therefore the task connectivity matrix of these two design task can be represented as:

$$\mathbf{T}_{CM} = \begin{array}{c} \\ T_2 \\ T_3 \end{array} \begin{bmatrix} {\scriptstyle T_1} & {\scriptstyle T_2} \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

The design attributes and design constraints of each algorithm are inputted into DSM as shown in Figure 6.9. The interdependency of both design tasks can be described as follow.

Design Config. / Design Tasks	Design Configurations: Locations of fixture (D_2)	Design Configurations: Tolerances (D_3)	Constraints: Assembly stability(C_{T2})	Constraints: Cost of Tolerance(C_{T3})
Multi-Fixture Layout Design (T_2)	o		X	
Tolerance Optimization(T_3)	⧱	o		X

O = To be optimized   ⧱ = constant design parameters of passive KCCs   X= Constraint of design task

Figure 6.9: Design structure matrix of Floor Pan Assembly case study.

The optimal *KCC* tolerances are used to control the random variations of *KCCs*, $\mathbf{u}$, such that the *KPC* variations, $\mathbf{y}$, are within the specification window. However, the impact of *KCC* variations on *KPC* are also controlled by SOVA model, $\mathbf{A}$, which is formulated based on fixture locator positions and part-to-part joint design. To optimise the fixture layout design, it is not necessary to have the information regarding to the optimal *KCC* tolerances (arbitrary *KCC* tolerances can be set in conducting fixture layout optimisation). On the other hand, to conduct tolerance optimisation, it is required to have information regarding to the fixture layout, part-to-part joint, and assembly sequence. Thus, the appropriate *KCC* tolerances can be determined according to given process architecture information.

Sensitivity Analysis to Decoupled Design Tasks

To affirm dependency analysis in Step 2.1, the multi-fixture layout design task can be simulated in two different sets of *KCC* tolerances. The optimal fixture positions obtained from two different sets of *KCC* tolerances are almost identical. Therefore, it can be concluded that the multi-fixture layout design task, T_2, is independent *KCC* tolerance assignment, $\mathbf{\Phi}_2 = D_3$, which can be represented as:

$$\frac{\partial \mathbf{D}_2}{\partial \mathbf{\Phi}_2} = 0$$

On the other hand, the *KCC* tolerance optimisation design task, T_3, is dependent on the design of fixture layouts since the fixture layout affect the structure of SOVA model which represents the sensitivity of *KPCs* on *KCC* tolerances. Thus, the sensitivity analysis of the *KCC* tolerance optimisation design task, T_3, can be expressed as:

$$\frac{\partial \mathbf{D}_3}{\partial \mathbf{\Phi}_3} \neq 0; \text{ where } \mathbf{\Phi}_3 \text{ is fixture locator positions}$$

Thus, these design tasks are *decoupled* and can be conducted *sequentially*.

Design Task Hierarchy Development Based on Task Connectivity Matrix

By based on Task Connectivity Matrix in Eq. (6-6) and sensitivity analysis in the previous section, both design tasks are formulated into *Task Flow Chain, TFC,* in order to generate the design task sequence options as shown in Figure 6.10. There are totally three design task sequence options for each engineering change scenarios. These three design task sequences involve: (i) option I: multi-fixture layout optimisation; (ii) option II: *KCC* tolerance optimisation; and (iii) integration of multi-fixture layout optimisation and *KCC* tolerance optimisation.

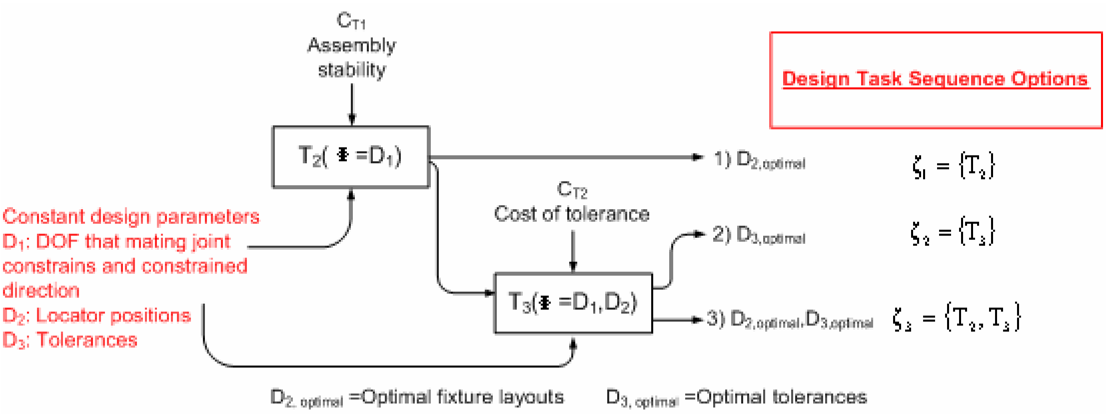


Figure 6.10: Task Flow Chain for Floor Pan Assembly case study in three engineering change scenarios.

Analysis of the Design Task Sequence Options

The improvement of process yield and changes in *KCC* tolerance cost for three design task sequence options in each engineering change scenario are demonstrated in this step. The target process yield after the assembly process is redesigned to accommodate the engineering changes is 95 percent or above. If the process yield after the assembly process is redesigned below threshold, the engineering change initiator is notified about the limitations of assembly process redesign to accommodate the engineering change, and the modifications of engineering change can be performed. The analyses of three engineering change scenarios are presented as follows:

Scenario 1: Change in KPC Allowable Specifications

In this scenario, the initial *KPC* specification windows of 12 *KPCs* are required to be 25 percent smaller than the initial design shown in Table 6.1. The engineering changes on the *KPCs* specification window cause the process yield reduced from 86.12 percent to 58.76 percent. The three design task sequence options as shown in Figure 6.10 are then implemented. The impacts on the process yield and cost of adjusting *KCC* tolerances compared with initial *KCC* tolerance design are analyzed and shown in Table 6.3.

Table 6.3: Process yield and *KCC* tolerance cost increase of three design task sequence options for engineering change in the first scenario.

	Process yield	*KCC tolerance cost increase*
Option 1: *Multi-fixture layout design optimization*	91.13%	0%
Option 2: *KCC tolerance optimization*	93.62%	30.20%
Option 3: *Integration of multi-fixture layout and KCC tolerance optimizations*	94.58%	8.40%

Scenario 2: Changes in Assembly Sequence

The assembly sequence is changed to reduce the lead time in assembly. The left and right sections of the floor pan can be performed simultaneously as shown in Figures 6.2 and 6.3 which differ from the previous design that all part components are assembled sequentially. The engineering change of assembly sequence causes the initial design of fixture layout design and *KCC* tolerance assignment to become obsolete. The process yield and increase of *KCC* tolerance cost are shown in Table 6.4.

Table 6.4: Process yield and *KCC* tolerance cost increase of three design task sequence options for engineering change Scenario 2.

	Process yield	*KCC tolerance cost increase*
Option 1: *Multi-fixture layout design optimization*	94.10%	0%
Option 2: *KCC tolerance optimization*	94.24%	25.87%
Option 3: *Integration of multi-fixture layout and KCC tolerance optimisations*	95.67%	5.70%

Scenario 3: Changes in Part Geometry

The engineering changes are made on Floor Pan Left and Right geometry to increase the space for the engine compartment as shown in Figure 6.4. The 12 *KPC* specifications and the assembly sequence are unchanged from the initial design. The changes of Floor Pan Left and Right require fixture layout redesign in locating the parts during assembly process as well as new *KCC* tolerances. The changes of part geometry create the changes of the design space that locator can be placed, and this also subsequently affects the design of *KCC* tolerances. In option 1, the *KCC* tolerances are maintained as the same while the fixture layouts are optimised. In the design option 2, the fixture positions are assigned based on experience, and the *KCC* tolerances are optimised. For the design option 3, the fixture layouts are optimised first, and then the optimal *KCC* tolerances are determined based on the optimal fixture layouts. The process yield and *KCC* tolerance cost increase of three design options are shown in Table 6.5.

Table 6.5: Process yield and *KCC* tolerance cost increase of three design task sequence options for engineering change Scenario 3.

	Process yield	*KCC tolerance cost increase*
Option 1: *Multi-fixture layout design optimisation*	85.47%	0%
Option 2: *KCC tolerance optimization*	84.50%	21.03%
Option 3: *Integration of multi-fixture layout and KCC tolerance optimisations*	89.55%	13.12%

6.5 Discussion

Based on three engineering change scenarios, this study shows that the integration of multi-fixture layout design tasks and *KCC* tolerance optimisation leads to the highest process yield. The integration of other design tasks such part-to-part joint selection design task and assembly sequence design task into the design task framework can potentially improve the process yield and reduce cost of *KCC* tolerances in the engineering change requirements. However, the time in conducting multiple design task sequence options have to be taken into an account. Moreover, the additional design matrices such as reusability of fixture locator, tooling accessibility, and stability of parts supported by locators can be used as the evaluation indices besides the cost, quality, and design lead time. These matrices can be formulated as design objectives in changing the process architecture design.

Finally, in the case that the assembly process redesign cannot achieve the design objective requirements such as process yield threshold or acceptable cost of *KCC* tolerances, the system for negotiating the engineering changes between product design and assembly process design has to be established. The interactive communication between product design and assembly process architecture design can help in redesigning an assembly process to achieve the new optimal design solutions which can accommodate the engineering changes.

6.6 Conclusions

This chapter presents the applications of the proposed methodologies in Chapters 3 to 5 in redesigning the automotive body subassembly system. Three scenarios of engineering changes are illustrated which are: (i) changes of *KPC* allowable specifications; (ii) changes of assembly sequence; and (iii) changes of part geometry. The procedure in adjusting the process architecture design involves: (i) modeling of the dependency among *Key Characteristic Group* and developing the Task Connectivity Matrix; (ii) assessing the impacts of engineering changes in terms of complexity in redesign, process yield, and cost; and (iii) redesigning the system by using design task sequence generated by Task Flow Chain. The results from the case studies show that the proposed methodology can help in benchmarking design task sequence options and selecting the best design task sequence option to optimise the process architecture design. The future work will incorporate other design constraints which exist in the current assembly system in the redesign process in order to enhance the capability of the current redesign methodology to be more relevant to the industrial applications.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this chapter, the contributions and findings of the thesis summarized. The potential future research is also discussed and outlined.

7.1 Research Contributions

The research proposed in this thesis is motivated by the needs of industries to maintain product quality and to accommodate emerging engineering changes in product life-cycle. The research aims to optimise the design of process architecture to be robust to variation sources in new design requirements. The design changes on both product and process architectures are taken into consideration in changing the process architecture design which allows manufacturers greater control over investment cost and final product quality. The significance of the research proposed in this thesis can be summarized into three major areas which are (i) process architecture modeling; (ii) *KPI* model development and assessment; and (iii) framework in adjusting the process architecture to meet the new requirements.

1. ***Process Architecture Modeling***

The contribution of the proposed process architecture modeling can be summarized into four levels which are:

- Process architecture model based on the proposed concept of the hypergraph data structure and *Key Control Characteristic- Hierarchical Groups* (*KCC-HGs*): The *KCCs* in process architecture is partitioned into a functionality groups instead of traditional approach in grouping the *KCCs* according assembly stations. The functionality group or *KCC-HG* consists of *KCCs* from all assembly stations with having similar functionality such as fixture locator group and part-to-part joint group. Modeling the process architecture into *KCC-HGs* offers two advantages. First, the design optimisation of *KCCs* in *KCC-HGs* can be conducted to obtain the optimal design solution in system level instead of in assembly station level. Second, a design task can be developed according to functional requirements and design constraints for each *KCC-HG*.

- Formulation of *design task* to optimise *KCCs* of the developed *KCC-HG*: the optimisation algorithm of a design task can be customized according to the design objective and design constraint for each *KCC-HG*. The robustness of the algorithm which can avoid local optimal solution and convergence rate to the global optimal solution are the key success in developing a design task. The design objective and design constraint for each design task are used to determine the dependency between design tasks. Moreover, the software development, AutoSOVA Model Generator, is also developed to automatically generate the assembly response function from CAD information. The AutoSOVA Model Generator reduces the difficulty in formulating the assembly response function manually, and allows the development of design tasks by integrating the AutoSOVA Model Generator in the optimisation algorithm of a design tasks
- Design task dependency analysis based on hypergraph technique and the proposed sensitivity analysis: The dependency between two design tasks is caused by: (i) dependence on output from one design task as the input of

the other design task; and (ii) dependence on design objective and design constraints. The hypergraph technique and sensitivity analysis can be used to analyze these dependencies. The dependencies of all design tasks are expressed by *KCC* dependency model called the *Task Connectivity Matrix,* T_{CM}. The Task Connectivity Matrix provides the approach to measure the complexity of an assembly system which consists of a large number of *KCCs* and *KPCs* as well as design constraints.

- *KCC* Functional dependency model: The functional dependence of *KCCs* can be analyzed in every assembly station to understand the relation between individual *KCCs* in delivering functionality. The *KCCs* which have functional dependence can cause the change propagation when the product and process architecture design are changed. The design dependency among *KCCs* can be modeled and used to assess the level of engineering change propagation during changing the design of process architecture.

2. *KPI Model Development and Assessment of the Current Process Architecture Design*

The *KPI* indices in assessing the engineering change impacts on the current assembly system are one of the unique contributions of this thesis. The indices involve: (i) process yield; (ii) cost; and (iii) the engineering change complexity index. The process yield is proposed in this thesis to indicate the percentage of conformance product produced from the assembly system. The process yield can serve as an unambiguous index in evaluating the product consisting of multiple *KPCs*. The change of the process yield as a result of engineering change can have implications about the impact of the changes on the final product quality. The SOVA model is used as the assembly response function in the numerical simulation such as Monte Carlo technique to estimate the changes of process yield.

The evaluation of cost in changing the process architecture design is also incorporated in the *KPI* model assessment. In the case that an engineering change does not affect the existing *KCC* nominal design, cost obtained from the tolerance optimisation can indicate the investment of the simplest approach in changing the existing process architecture design. In addition, the proposed cost estimation can be performed easily within short computational time which is necessary for engineers to make the decision in the case that engineering changes occur frequently.

Finally, the engineering change complexity index can help to indicate potential level of engineering change propagation in the current process architecture if the process architecture has to be optimised in the new requirements. The engineering change complexity index is formulated base on design task dependency described by Task Connectivity Matrix, the number of *KCCs* expressed by *KCC* Weight matrix, and the number of *KCCs* described in functional dependence model. The application of the proposed complexity index is not limited to redesigning the system, but also as a guideline for designing the new product and process architectures.

3. Framework to Integrate Multiple Design Tasks in Redesigning the Process Architecture based on Transversal of Hypergraph and Task Connectivity Matrix

The developments of design tasks can shorten the new product development lead time and reduce number of engineering changes during launching production. However, it lacks of a framework to integrate these design tasks and optimise their implementation sequence for dimensional management. This challenge is addressed by proposing the methodology to formulate the design synthesis framework for dimensional management in multistage assembly systems which are based on: (i) *Transversal of Hypergraph and Design Structure Matrix* to determine the interdependency between design tasks in terms of design configurations and design constraints; (ii) sensitivity analysis for approximately decoupling design tasks to reduce computational efforts; and (iii) *Task connectivity matrix and Task Flow Chain* for modeling design task hierarchy and generating the sequences of design tasks. These generated design task sequences can be used as a guideline to synthesize the optimal design configurations. The proposed methodology is illustrated and validated in the process of designing configurations for automotive underbody subassembly by integrating three design tasks: (i) Tolerance optimisation, (ii) Multi-fixture layout design, and (iii) part-to-part joint design. The proposed methodology is illustrated in redesigning assembly process of automotive underbody subassembly to respond to changes related to: (i) specification requirements; (ii) assembly sequence; and (iii) part geometry.

7.2 Findings and Discussion

1) Discussion of simulation results of three scenarios presented in Chapter 6

Scenario 1: Change in KPC Allowable Specifications

Changing the *KPC* specification windows in this scenario without redesigning the assembly process causes the increase of non-conformance items as shown in Figure 7.1(a-b) (i.e., reduction of process yield). The specification requirement of 12 *KPCs* can be viewed as the hyper-rectangular cube and the actual variations of 12 *KPCs* on Floor Pan Assembly process can be observed as ellipsoid shape. The *KPC* tolerance/specification region in multivariate m (m =12) space is the volume of the hyper-rectangular cube (Tamm *et al*., 1993) which can be defined as:

$$\prod_{i=1}^{12}(USL_i - LSL_i) \tag{7-1}$$

where *USL* and *LSL* are the upper and lower specification limits of *KPC*, respectively.

The *KPC* variations of multivariate processes can be assessed by using Chi-square distance defined as:

$$\chi_0^2 = (\mathbf{y}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu}) \tag{7-2}$$

where $\mathbf{y}$ is a vector of *KPC* variations, $\boldsymbol{\mu}$ is a vector of *KPC* nominal design, and Σ is the covariance matrix of 12 *KPCs*.

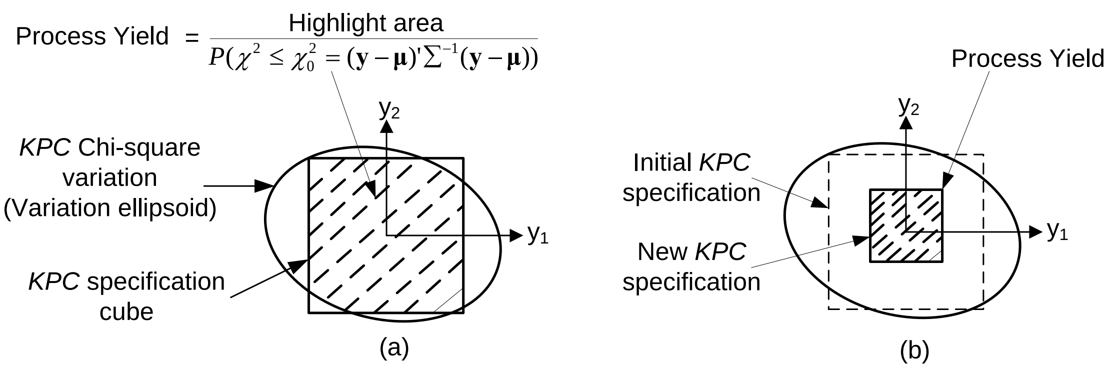


Figure 7.1: *KPC* variations compared with *KPC* tolerance cube: (a) initial process yield; (b) process yield after adjusting *KPC* specification windows.

The volume of ellipsoid which lies outside the *KPC* specification hyper-rectangular cube is the probability of non-conformance items produced from the Floor Pan Assembly process. On the other hand, process yield can be defined as the intersection of volume of *KPC* specification hyper-rectangular cube, and *KPC* ellipsoid divided by total volume of *KPC* ellipsoid as shown in Figure 7.1(a). The curtailment of the *KPC* specification window results in declining of process yield. The adjustments of fixture locator positions by T_2 and T_3 design tasks affect on *KPC* variations as follow.

The design changes of locator positions by multi-fixture layout design task affect the *KPC* variation ellipsoid at two levels. First, changing in locator positions alters the sensitivity of the fixturing system on the process variation sources while all *KCC* tolerances are unchanged. This causes the changes in the direction of eigenvector directions and eigenvalues describing the process variations as shown in Figure 7.2(a). Second, the design changes of locator positions are to shift the process variation mean to meet *KPC* specification windows as shown in Figure 7.2(b). This situation can be found in the case of locating fault such as locator damage. The adjustment of locator positions is to recover the assembly process variations into controlled specification windows.

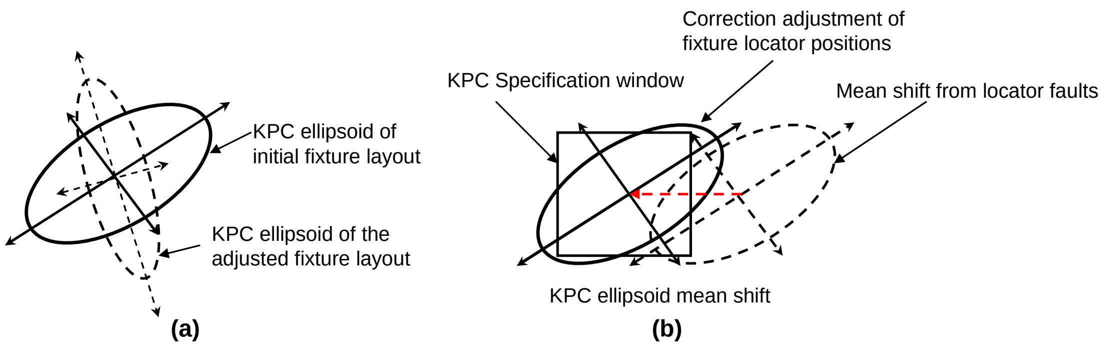


Figure 7.2: The adjustment of fixture locator causing process variation changes in: (a) eigenvectors and eigenvalues, and (b) process variation mean shift adjustment.

Tolerance optimisation affects only the scale of eigenvalue along the eigenvector direction since the eigenvectors of the assembly process are dependent on the fixture locator positions, part-to-part joint selection and assembly sequence. Impacts of *KCC* tolerance adjustments on the assembly process variations can be graphically presented as in Figure 7.3.

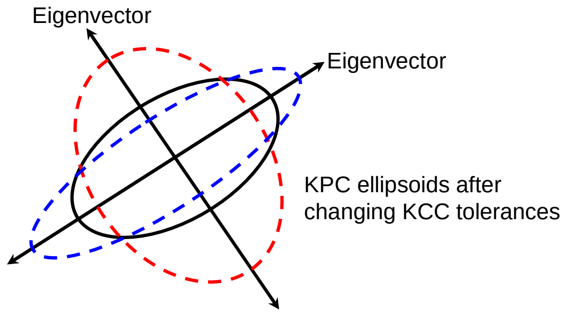


Figure 7.3: *KPC* ellipsoids after changing *KCC* tolerances.

Scenario 2: Changes in Assembly Sequence

The new assembly sequence creates the modularity between the left and right section of Floor Pan. This modularity offers several advantages including: (i) less complexity in fixture layout and *KCC* tolerance optimisations; and (ii) less variation propagation between two modular designs. The complexity of design is reduced since the fixture layout in either for Floor Pan Left or Right can be reused in the last assembly station. Furthermore, the variations from the fixture locators in the last assembly station is eliminated which results in the improvement of process yield with minimum increase of *KCC* tolerance cost after three design task sequence options are implemented.

Scenario 3: Changes in Part Geometry

The changes of part geometry in this scenario result in the dramatic changes of the fixture layouts and *KCC* tolerances. The existing design of fixture layouts and

KCC tolerances cannot be reused which can lead to significant changes in process yield and increase of *KCC* tolerance cost. In this scenario, although the process yield is at the same level of the initial design, the cost of *KCC* tolerance surges considerably in order to increase the process yield. In order to achieve the 95 percent of process yield threshold and reduce the increase of tolerance cost, the design tasks such as assembly sequence and part-to-part joint optimisations can be introduced further in redesigning the assembly process. However, the trade-off between improvement of process yield, and computation time in conducting several design task sequence options have to be considered.

(2) Relationship between reduction of *KCC* dependency and process architecture robustness

The reduction of redesign complexity index implies the easiness in managing the engineering change propagation which can be conducted by either reducing the design task dependency, *KCC* functional dependency, or a number of *KCCs*. Redesign complexity index proposed in this thesis is aligned with concept of various research studies in the product and process architecture design such as Axiomatic Design (AD), Design for Manufacturing and Assembly (DFMA), and modular design. To improve the design of product and process architecture, AD suggests: (i) maintaining design independence of functional requirements (known as Independence Axiom); and (ii) minimizing the information content of the design (known as Information Axiom) (Suh, 1990). Maintaining design independence in AD is similar to reducing the *KCC* dependency. This allows for the adjusting of functional design of the system independently or at least in a sequential manner. On the other hand, the DFMA approach improves the design of product by reducing assembly cost. Reducing the number of parts is also a DFMA approach which is similar to the study proposed in this thesis in minimizing the number of *KCC* to reduce the complexity in process architecture. Minimizing the number of parts usually leads to the reduction of *KCCs* and assembly process configuration. For assembly process architecture design, a reduction of *KCCs* can be performed using several approaches such as reusing the *KCCs* (e.g., reusing fixture in the third scenario in Chapter 6) or using reconfigurable *KCCs* in an assembly process for a product family. In addition, reduction of a number of parts and *KCCs* in product and process architecture can simplify the system and also helps in monitoring and controlling the critical parts and *KCCs*. Finally, modular design also helps in reducing the complexity of the product and process architecture design. For example, some functional designs which have coupled relation can be grouped into module and then maintain the relations of this module with other functional design as uncoupled or decoupled relations. This aims to control the engineering change propagation on a minimal scale. Thus, controlling variation propagation can be performed efficiently.

(3) Process yield as common index for measuring the robustness of product and process design

The process yield presented in this thesis offers several advantages in the area of measuring the robustness of the process architecture design. The process yield can indicate the percentage of conformance product which people involving in product and process architecture design as well as in manufacturing and assembly can understand. This advantage cannot be found in using sensitivity indices such as optimality indices since the sensitivity indices are relative measures of *KPC* variations against *KCC* variations. Therefore, the sensitivity indices are difficult to be interpreted by manufacturing engineers about the relation of sensitivity indices and level of conformance product. This also leads to another advantage of process yield in measuring and comparing robustness in different what-if scenarios in process architecture redesign. The process yield can be considered as the absolute measure since it compares *KPC* variations with allowable specifications which are given and unchanged in conducting what-if scenario analysis. On the other hand, the sensitivity analysis is the relative measure between *KPC* variations and *KCC* variations which both *KPC* and *KCC* variations can be changed in what-if analysis. Therefore, it is difficult to compare the design improvement by using sensitivity indices in the process architecture redesign.

(4) Shortcoming of tolerance optimisation in enhancing the quality and robustness

The limitations of the *KCC* tolerance optimisation approach in improving the robustness of process architecture design is the key finding of this thesis and serves as the research motivation. Since *KCC* tolerances can be adjusted independently without any effect on the nominal design of *KCCs*, it is the easiest approach in reducing the variations of *KPCs*. However, tightening *KCC* tolerances contributes to the requirements of higher precision technologies for production and assembly which can subsequently lead to higher investment. As shown in Chapter 1, sometimes tightening *KCC* tolerances to the highest precision that current technology can offer without considering economical concern cannot guarantee that final product quality will meet the target. Specifically, *KCC* tolerance optimisation or tightening *KCC* tolerances directly does not enhance the robustness of an assembly system. Adjusting *KCC* tolerances is the approach in controlling sources of variations which have the monotonic relation with *KPC* variations (i.e., reduction of *KCC* variations causes the reduction of *KPC* variations). It can be viewed as a superficial improvement of assembly system robustness because the optimal *KCC* tolerances are dependent on given *KCC* nominal design. Thus, the nominal design optimisations such as fixture layout, part-to-part joint, and assembly sequence optimisations are implemented first and then *KCC* tolerance optimisation is conducted to fine tune the assembly system as shown and discussed in the first case study scenario presented in Chapter 6.

7.3 Limitations of the Proposed Methodology

Although the proposed research in this thesis can address several challenges of the current research studies in engineering change management in redesigning the product and process architecture, this thesis still has a few limitations which lead to future research opportunities. First, the proposed variation prediction model and design synthesis tasks used in redesigning the process architecture is limited to rigid body assembly. Based on the current research, the difference of the simulation results of the proposed variation model and FEM software on beam-based model of automotive body is illustrated in Figure 7.4. When the beams in the model behave as rigid body (i.e. each beach is represented by two nodes), the simulation results in predicting the dimensional variations obtained from the proposed variation prediction model and FEM software are almost identical. However, when the number of nodes increases for the FEM software simulation (beam model becomes more flexible), the discrepancy of both models in predicting the variation is significant. This example shows the limitation of the proposed variation prediction model in estimating the variation of non-rigid part assembly. Nevertheless, the proposed variation prediction model is still applicable to industrial practices. According to studies presented by Shiu *et al.* (1996) in automotive body assembly system, around two-third of fixture layouts are 3-2-1 fixture layout scheme. This implies that around two-third of parts assembled in automotive body behave similar to rigid body.

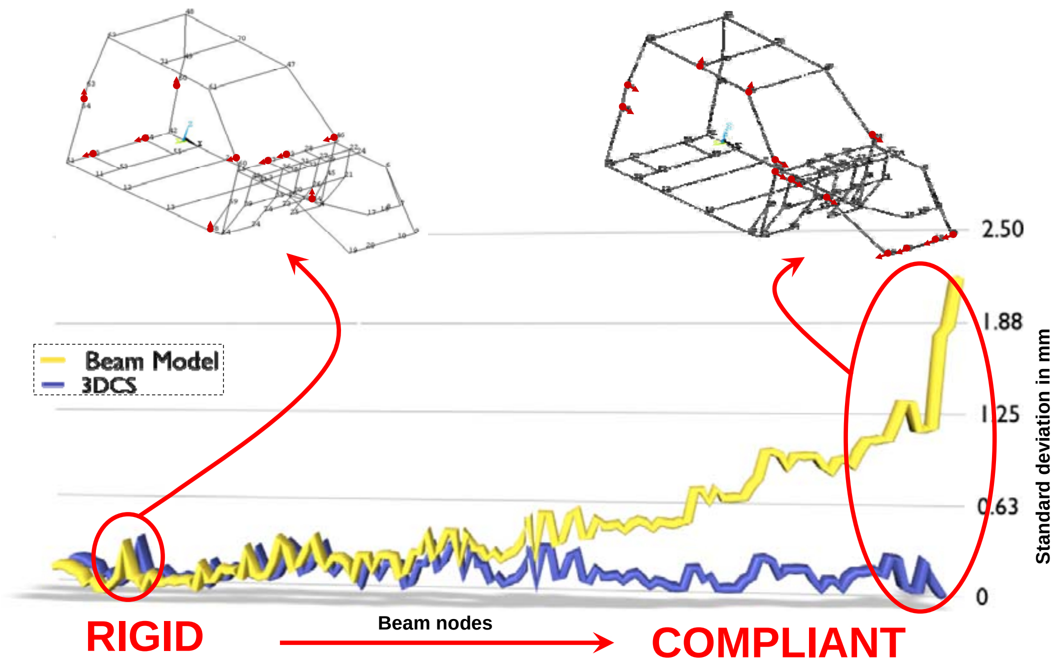


Figure 7.4: Comparison of simulation result obtained from the proposed variation prediction model and FEM software.

Second, it is time-consuming in applying the proposed methodology on the complex assembly process design since all of design information in CAD has to be transformed into point-based information first and then input into AutoSOVA model Generator to create assembly response function. The current approach in transforming the CAD information to point-based information is mainly manual approach. The *KPC* and *KCC* features in CAD information such as cylinder feature of a pin or parallel feature of gap have to be defined by designers and a set of points is used to describe those features. This requires a considerable time and experience of designer to convert geometrical feature into point-based system. The automatic conversion of

KPC and *KCC* described by CAD features into point-based information and then interface with AutoSOVA model Generator to generate the assembly response function would greatly expedite the redesign iterations. Moreover, this would allow engineers can adjust the design directly on CAD environment and be able to monitor the impacts on *KPIs* in the same time.

Last, although the proposed *KPIs* can address the basic criteria in redesign the process architecture, they still pose limitations in order to apply on many industrial design problems. The additional *KPIs* are required to address various needs of the industrial applications. In self-resilient production system, the inter-loops information flow from production ramp-up synthesis and service synthesis phases are crucial to redesign process architecture. Therefore, it is necessary to establish the *KPIs* for the inter-loop information flow such as diagnosbility and reconfigurability. These additional *KPIs* will enhance capability of the current design synthesis. Moreover, the end-to-end cost model should be explored and incorporated into the current *KPIs*. Thus, cost impact analysis can be conducted and be able to address the needs in other areas such as impacts on productivity, manpower, or disassembly cost at the end of product and process life-cycle. These additional *KPIs* would help to eliminate limitation of the current proposed methodology. In the next section, the suggestions for the future work are presented in order to address the aforementioned limitations.

7.4 Suggestions for Future Work

The studies presented in this thesis involve the theoretical formulation of the design synthesis for dimensional management in multi-stage assembly processes. However, the further research and design application developments are necessary to create the relevant impacts on industrial design challenges. The future research can be summarized as follows:

1. *Generalized AutoSOVA Model Generator and CAD Software Applications*

The current limitations of the AutoSOVA Model Generator can be summarized in two folds. First, the methodology is based on a rigid body assembly while an assembly process usually involves both rigid and flexible bodies. This leads to future opportunity to enhance the current AutoSOVA Model Generator to be able to model a compliant part assembly. Second, the interfaces of the AutoSOVA Model Generator with other CAD software applications have to be developed. This will enhance AutoSOVA Model Generator to be more user-friendly and align with the current CAD system. The interface between AutoSOVA Model Generator and CAD software applications can help in increasing the ease in generating the SOVA model for assembly process which involves multiple assembly lines.

2. *Enhancement of the KPI Indices*

The future development of the *KPI* indices can be helpful for providing insight into redesigning the product and process architecture. The relationship between proposed indices and other areas of product and process design should be explored. For example, the relations among the redesign complexity index diagnosability and reconfigurability of the process architecture merits future exploration. On the other hand, the extension of cost information proposed in this thesis with the overall cost structure in product and process design is also necessary in providing the accurate economic evaluation in redesigning the process architecture. The integration of information system to formulate the cost evaluation can be extremely helpful for making the decision.

3. *Development of an Artificial Intelligent System for Design Negotiation between Product and Process Design*

The artificial intelligent system can be developed as a further extension of the design synthesis methodology presented in this thesis. Such a system can help in negotiating the designs between product and assembly process. This system can help to ensure the functionalities and feasibility in producing the final product. The system allows the real-time interaction with change propagation created by product design. The dimensional quality and other desirable functionalities of a final product can be assessed after an assembly process is redesigned.

For example, in aircraft fuselage design, the design synthesis for dimensional management presented in this research has to be integrated with other design disciplines as shown in Figure 7.5. The artificial intelligent system can help in managing the information regarding engineering changes in each discipline and communicate to others. This can lead to multidisciplinary design optimisation which is essential within the context of complex product design such as automotive and aircraft design.

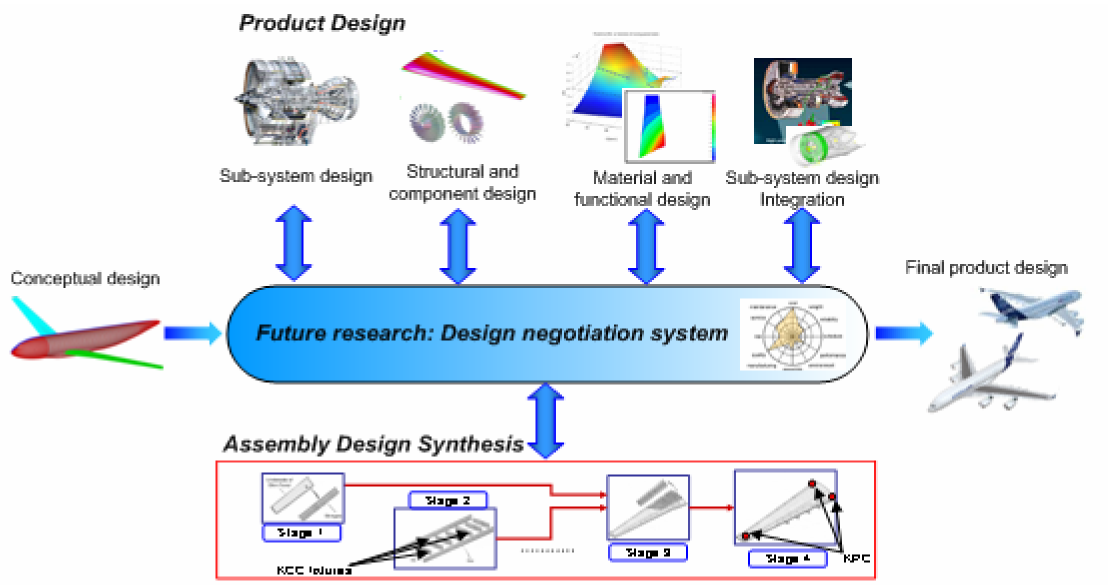


Figure 7.5: Integration of design synthesis toolbox with other design disciplines.

REFERENCES

Boothroyd, G. (1994) Product design for manufacture and assembly, *Computer-Aided Design*, 26(7), 505-520.

Box, George E. P., Hunter, J. S., and Hunter, W. G., (2005) *Statistics for experimenters: design, innovation, and discovery (2nd Edition)*, Wiley-Interscience.

Browning, T. R. (2001) Applying the Design Structure Matrix to System Decomposition and Integration Problems: A review and New Direction. *IEEE Trans. on Engineering Management*, 48(3), 292-306.

Cai, W., Hu, S. J., and Yuan, J. X. (1997) A Variational Method of Robust Fixture Configuration Design for 3-D Workpieces. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 119, 593-602.

Camelio, J., Hu, S. J., and Ceglarek, D. (2003) Modeling Variation Propagation of Multi-station Assembly Systems with Compliant Parts. *Transactions of ASME, Journal of Mechanical Design*, 125(4), 673-681.

Camelio, J. A., Hu, S. J., and Ceglarek, D. (2004) Impact of Fixture Design on Sheet Metal Assembly Variation. *Journal of Manufacturing Systems*, 23(3), 182-193.

Ceglarek, D., Shi, J., and Wu, S.M., (1994) A Knowledge-based Diagnosis Approach for the Launch of the Auto-body Assembly Process. *Transactions of ASME, Journal of Engineering for Industry*, 116(4), 491-499.

Ceglarek, D., and Shi, J. (1995) Dimensional Variation Reduction for Automotive Body Assembly. *Manufacturing Review*, 8(2), 139 - 154.

Ceglarek, D., and Shi, J. (1998) Design Evaluation of Sheet Metal Joints for Dimensional Integrity. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 120(2), 452-460.

Chon, C.T. (1986) Generic 'stick' model of a vehicular structure. SAE Paper No. 860825.

Clarkson, P. J., Simons, C., and Eckert, C. (2004) Predicting Change Propagation in Complex Design. *Transactions of ASME Journal of Mechanical Design*, 126, 788-797.

Cohen, T., Navthe, S., and Fulton, R. E. (2000) C-FAR: Change Favorable Representation. *Computer-aided Design*, 32, 321-328.

Carver, D. L. (1986) Criteria for Estimating Module Complexity. *Journal of Systems Management*, 18-23.

Cutkosky, M. R., and Tenembaum, J. M. (1990) A Methodology and Computational Framework for Concurrent Product and Process Design. *Mechanic Machinery Theory*, 25(3), 365-381.

Ding, Y., Ceglarek, D., and Shi, J. (2000) Modeling and Diagnosis of Multistage Manufacturing Process: Part I State Space Model. *Proceedings of 2000 Japan-USA Symposium of Flexible Automation*, July 23-26, Ann Arbor, MI, 2000JUSFA-13146.

Ding, Y., Ceglarek, D., and Shi, J. (2002) Design Evaluation of Multi-station Manufacturing Processes by Using State Space Approach. *Transactions of ASME*, *Journal of Mechanical Design*, 124(4), 408-418.

Ding, Y., Shi, J., and Ceglarek, D. (2002) Diagnosability Analysis of Multi-station Manufacturing Processes. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 124(1), 1-13.

Ding, Y., Jin, J., Ceglarek, D., and Shi, J. (2005) Process-oriented Tolerancing for Multi-station Assembly Systems. *IIE Transactions on Design & Manufacturing*, 37(6), 493-508.

Eckert, C., Clarkson, P. J., and Zanker, W. (2004) Change and Customisation in Complex Engineering Domains. *Research in Engineering Design*, 15, 1-21.

El-Haik, B. S. (2005) *Axiomatic Quality: Integrating Axiomatic Design with Six-Sigma, Reliability, and Quality Engineering*. A John-Wiley & Sons Inc. Publication.

ElMaraghy, H. A. (2006) Flexible and Reconfigurable Manufacturing Systems Paradigms. *International Journal of Flexible Manufacturing Systems*, 17, 261-276

Eppinger, S. D., Whitney, D. E., Smith, R. P., and Gebala, D. A. (1994) A Model-based Method for Organizing Tasks in Product Development. *Research in Engineering Design*, 6, 1-13.

Flood, R. L., and Carson, E. R. (1993) *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science, 2nd Edition*. Plenum Press. New York.

Fowlkes, W. Y. and Creveling, C. M. (1995) *Engineering Methods for Robust Product Design: Using Taguchi's Methods in technology and Product Development*, Addison-Wesley Pub. Co., Massachusetts.

Fujimoto, H., and Ahmed, A. (2001) Entropic Evaluation of Assemblability in Concurrent Approach to Assembly Planning. *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, 306-311.

Fujimoto, H., Ahmed, A., Iida, Y., and Hanai, M. (2003) Assembly Process Design for Managing Manufacturing Complexities Because of Product Varieties. *International Journal of Flexible Manufacturing Systems*, 15/4: 283-307.

Gell-Mann, M., and Lloyd, S. (1996) Information Measures, Effective Complexity, and Total Information. *Complexity*, 21: 44-52.

Giffin, M., de Weck, O. L., Bounova, G., Keller, R., Eckert, C., and Clarkson, J. (2007) Change Propagation Analysis in Complex Technical Systems. *DETC2007-34652, Proceedings of ASME IDETC/CIE 2007*, September 4-7, Las Vegas, NV.

Hartley, R. V. L. (1928) Transmission of Information. *Bell System Technical Journal*, 35, 535-563.

Huang, W., Lin, J., Bezdecny, M. R., Kong, Z., and Ceglarek, D. (2007a) Stream-of-Variation (SOVA) Modeling I: A Generic 3D Variation Model for Rigid Body Assembly in Single Station Assembly Processes. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 129(4), 821-831.

Huang, W., Lin, J., Kong, Z., and Ceglarek, D. (2007b) Stream-of-Variation (SOVA) Modeling II: A Generic 3D Variation Model for Rigid Body Assembly in Multi Station Assembly Processes. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 129(4): 832-842.

Huang, W., Phoomboplab, T., and Ceglarek, D. (2009) Process Capability Surrogate Model Based Tolerance Synthesis of Multi-station Manufacturing Systems. *IIE Transactions*, Vol. 41, pp. 309-322.

Inui, M., and Kimura, F. (1991) Algebraic Reasoning of Position Uncertainties of Parts in an Assembly. *Proceedings of the first ACM symposium on Solid Modeling Foundations and CAD/CAM Applications*.

Izquierdo, E. L., Du, H., Hu, S. J., Jin, R., Shi, R. J., and Jee, H. (2006) Robust Fixture Layout Design for a Product Family Assembled in a Multistage Reconfigurable Line. *Proceedings of 2006 ASME International Conference on Manufacturing Sciences and Engineering*, October 8-11, 2006, Ypsilanti, Michigan, USA.

Jaratt, T. A. W., Eckert, C., and Clarkson, P. J. (2002) Product Architecture and the Propagation of Engineering Change. *Proceeding of 7th International Design Conference*, Dubrovnik, Croatia, 1, 75-80.

Jin, J. and Shi, J. (1999) State Space Modeling of Sheet Metal Assembly for Dimensional Control. *Transactions of ASME, Journal of Manufacturing Science and Engineering*, 121(4), 756-762.

Kim, P., and Ding, Y. (2004) Optimal Design of Fixture Layout in Multi-station Assembly Processes. *IEEE Transactions on Automation Science and Engineering*, 1(2), 133-145.

Kirby, M. R., and Mavris, D. N. (1999) Forecasting Technology Uncertainty in Preliminary Aircraft Design. *World Association Congress*, San Francisco, USA.

Koren, Y., and Ulsoy, A. G. (1997) Reconfigurable Manufacturing Systems. *Engineering Research Center for Reconfigurable Machining Systems (ERC/RMS) Report 1*, The University of Michigan, Ann Arbor.

Koren, Y., Heisel, U., Jovane, F., Moriwoki, T., Pritschow, G., Ulsoy, A. G., and Van Brussel, H. (1999) Reconfigurable Manufacturing Systems. *Annals of the CIRP*, 2, 1-13.

Krishnan, V., Eppinger, S. D., and Whitney, D. E. (1997) Simplifying Iterations in Cross-Functional Design Decision Making. *Transactions of ASME Journal of Mechanical Design*, 119, 485-493.

Lee, K., and Andrews, G. (1985) Inference of the Positions of Components in an Assembly: Part 2. *Computer-Aided Design*, 17(1), 20 - 24.

Lee, B., and Saitou, K. (2003) Decomposition-based Assembly Synthesis for In-Process Dimensional Adjustability. *Transactions of ASME, Journal of Mechanical Design*, 125, 464-473.

Loose, J. P., Zhou, S., and Ceglarek, D. (2007) Kinematic Analysis of Dimensional Variation Propagation for Multistage Machining Processes with General Fixture Layouts. *IEEE Transactions on Automation Science and Engineering*, 4(2), 141-152.

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.

Manassa, R. J., and Devries, W. R., (1991) Optimization Method Applied to Selecting Support Positionin Fixture Design. *ASME Transactions Journal of Engineering Industries*, 113, 412-418.

Mantripragada, R., and Whitney, D. E. (1998) The Datum Flow Chain: A Systematic Approach to Assembly Design and Modeling. *Research in Engineering*, 10, 150-165.

Mantripragada, R., and Whitney, D. E. (1999) Modeling and Controlling Variation Propagation in Mechanical Assemblies using State Transition Models. *IEEE Transactions on Robotics and Automation*, 15(1), 124-140.

McCabe, T. J. (1976) A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2(4), 308-320.

McCabe, T. J., and Butler, C. W. (1989) Design Complexity Measurement and Testing. *Artificial Intelligence and Language Processing*, 32(12), 1415-1425.

Mehrabi, M. G., Ulsoy, A. G., and Koren, Y. (2000) Reconfigurable Manufacturing Systems: Key to Future Manufacturing. *Journal of Intelligent Manufacturing*, 11(4), 403-419.

Montgomery, D. C., (2009) *Design and Analysis of Experiments (7th Edition)*, John-Wiley & Sons Inc.

Mullins, S. H., and Anderson, D. C. (1998) Automatic Identification of Geometric Constraints in Mechanical Assemblies. *Computer-Aided Design*, 30(9), 715-726.

Phoomboplab, T. and Ceglarek, D. (2007) Design Synthesis Framework for Dimensional Management in Multistage Assembly Processes. *Annals of CIRP*, 57(1), 153-158.

Phoomboplab, T. and Ceglarek, D. (2008) Process Yield Improvement through Optimum Design of Fixture Layouts in 3D Multi-stage Assembly Systems. *Transactions of ASME, Journal of Manufacturing and Science Engineering*, Vol.140, December 2008.

Phoomboplab, T. and Ceglarek, D. (2009) AutoSOVA Model Generator: Automatic Generator of Variation Prediction Model for Multi-stage Assembly Systems with General 3D Fixture Layout, *2nd CIRP Conference on Assembly Technologies and Systems, Toronto, Canada*, 21-23 September, 2008.

Phoomboplab, T., Ceglarek, D., and Uicker, J. Jr. (2009) "Geometric Constraint and Variation Propagation Models in Multi-stage Assembly Systems with Part-to-part Joints and General 3D Fixture Layout", *Design Synthesis CIRP Design Conference, Enschede, The Netherlands,* 7-9 April, 2008.

Prakash, Phoomboplab, T., Izquierdo, L. E., and Ceglarek, D. (2009) Self-Resilient Production Systems: Going Beyond Design Robustness. *Proceedings of CIRP Computer-Aided Tolerance Conference*, Annecy, France.

Riviere, A., Feru, F., and Tollenaere, M. (2003) Controlling Product related Engineering Changes in the Aircraft Industry. *14th International Conference on Engineering Design*.

Roy, U., and Liao, J. (2002) Fixturing Analysis for Stability Consideration in Automated Fixture Design System. *ASME Transactions Journal of Manufacturing Science Engineering*, 118(3), 289-300.

Rutka, A. Guenov, M., Lemmens, Y., Schmidt-Schaffer, T., Coleman, P., and Riviere, A. (2006) Methods for Engineering Change Propagation Analysis. *25th International Congress of the Aeronautical Science*.

Selvaraj, P., Radhakrishnan, P., and Adithan, M. (2009) An integrated approach to design for manufacturing and assembly based on reduction of product development time and cost. *International Journal of Advanced Manufacturing Technology*, 42(1-2), 13-29.

Shannon, C. E. (1948) A Mathematical Theory of Communication. *Bell System Technical Journal*, 27: 379-423.

Shepperd, M. (1988) A Critique of Cyclomatic Complexity as a Software Metric. *Software Engineering Journal*, 30-36.

Shi, J. (2006) *Stream of Variation Modeling and Analysis for Multistage Manufacturing Processes*. Boca Raton, FL: CRC Press.

Shiu, B. W., Ceglarek, D., and Shi, J. (1996) Multi-stations Sheet Metal Assembly Modeling and Diagnostics. *Transactions of NAMRI*, 24, 199-204.

Shiu, B. W., Apley, D. W., Ceglarek, D. and Shi, J. (2003) Tolerance Allocation for Compliant Beam Structure Assemblies. *IIE Transactions Design and Manufacturing*, 35(4), 329-342.

Steward, D. V. (1981) *Systems Analysis and Management: Structure, Strategy, and Design*. Petrocelli Books, New York.

Suh, N. P. (1990) *The Principles of Design*. Oxford University Press, New York.

Suh, N. P. (1995) Design and Operation of Large Systems. *Journal of Manufacturing Systems*, 14(3), 203-213.

Suh, N. P. (1997) Design of Systems. *Annals of CIRP*, 46(1), 75-80.

Suh, N. P. (2005) *Complexity: Theory and Applications.* Oxford University Press.

Sutinen, K., Almefelt, L., and Malmqvist, J. (2002) Supporting Concept Development Using Quantitative Requirements Traceability. *12th Annual International Symposium INCOSE 2002*, Las Vegas, USA.

Taam, W., Subbaiah, P., and Liddy, J. W. (1993) A Note on Multivariate Capability Indices. *Journal of Applied Statistics*, 20(3), 339-351.

Turner, J. U. (1990) Relative Positioning of Parts in Assemblies using Mathematical Programming. *Computer-Aided Design*, 22(7), 394-400.

Wang, M. Y. (2000) An Optimum Design for 3-D Fixture Synthesis in a Point Set Domain. *IEEE Transactions in Robotics Automation*, 16(6), 839-846.

Wang, H., and Ceglarek, D. (2008) Generation of Assembly Sequences with k-ary Operations. submitted to *Journal of Mechanical Design*.

Whitney DE, Gilbert OL, Jastrzebski M (1994) Representation of Geometric Variations using Matrix Transforms for Statistical Tolerance Analysis in Assemblies. *Research in Engineering Design*, 6:191-210.

Whitney, D. E. (2004) *Mechanical Assemblies*, Oxford University Press.

Zhou, S., Huang, Q., and Shi, J. (2003) State-space Modeling for Dimensional Monitoring of Multistage Machining Process using Differential Motion Vector. *IEEE Transactions in Robotics and Automation*, 19, 296-308.

Zhu, X., Hu, S. J., Koren, Y., and Marin, S. P. (2008) Modeling of Manufacturing Complexity in Mixed-Model Assembly Line. *Transactions of ASME, Journal of Manufacturing and Science Engineering*, 130 No. 051013-1, October 2008.

Zou, Z., and Morse, E. P. (2003) Applications of the GapSpace Model for Multidimensional Mechanical Assemblies. *Transactions of ASME, Journal of Computing and Information Science in Engineering*, 3, 22-30.

Zou, Z., and Morse, E. P. (2004) A Gap-based Approach to Capture Fitting Conditions for Mechanical Assembly. *Computer-Aided Design*, 36(8), 691-700.