MODELING AND ANALYSIS OF SECURE COLLABORATIVE DESIGN VIA FUNCTION-PARAMETER MATRIX

A Thesis

In

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Concordia University

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of applied Science (Quality Systems Engineering)

at Concordia University

Montreal, Quebec, Canada

December 2012

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Mehrnaz Mirhosseini

Entitled: Modeling and Analysis of Secure Collaborative Design Via Function-Parameter Matrix

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr Chun Wang _____ Chair

Dr Yong Zeng _____ Examiner

Dr Xiao Huang _____ Examiner

Dr Simon Li_____ Supervisor

Approved by _____
Chair of Department or Graduate Program Director

_____
Dean of Faculty

Date _____

ABSTRACT

Modeling and Analysis of Secure Collaborative Design via Function-Parameter Matrix

Mehrnaz Mirhosseini

In order to keep the competitive advantages in today's global business market, it is critical for the companies to establish an effective engineering collaboration and protect the intellectual properties of the original manufacturer companies. The purpose of this thesis is to develop the modularization method introduced by Li (2007), in order to protect intellectual property from being shared by suppliers or other manufacturers that attempt to do reverse engineering in collaborative design. This thesis uses this introduced Function-Parameter matrix which represents the dependency relationships between shared and protected (IP) design parameters, in order to group and isolate all the intellectual property information. This modularization includes three phases of clustering that matches hierarchical clustering rules. Due to this clustering, tracking the protected parameters via inferences and selecting one set of low risk parameters for sharing with suppliers will be more effective while this method mitigates the risk of information leakage for intellectual property. Based on a matrix-based modular structure, two formulations are proposed to estimate the leakage risk of protected parameters due to the disclosure of shared parameters to suppliers and potential inferences. At the end, the DC motor and the relief valve system are used to examine the proposed modularization and measurement methods.

Keywords: Collaborative design, intellectual property, cluster analysis, modularization, information leakage, dependency matrix, hierarchical clustering, parametric design

Acknowledgements

I would like to express my gratitude to all who helped me to complete my thesis. I would like to acknowledge my supervisor Dr. Simon Li for his support, guidance, and perseverance towards successful completion of this thesis. His encouragements helped me in all the time of research.

I would like to thank Concordia University and CIISE for giving me the possibility to pursue my Master degree in Quality Systems Engineering and providing financial support to accomplish the research pursued in this thesis.

I would like to thank my family for all their helps and encouragements in my whole life.
I would like to give my especial thanks to my husband whose love enabled me to complete this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Background

With globalization, companies need to focus on their competitive advantages in order to remain in the competitive markets. One competitive advantage is to shorten time-to-market via collaborative design with suppliers (Ma et al. 2009). In order to achieve this objective, companies need to share some of their design information during the collaboration. Wang et al. (2006) define the collaborative design as the process where all the stakeholders are one part of design decision-making process and all the product information of stakeholders can be shared across the boundaries of companies through internet. In collaborative design, the suppliers and the other manufacturers design the components with their own knowledge and independent from the original knowledge of the original manufacturer (OM) (Mun et al. 2009). However, it is important for OM to keep some vital information protected, which are considered as the intellectual properties (IP) of the company. In this thesis, we call intellectual properties as protected parameters or functions which companies try to protect them. Moreover, the design data which companies share with the other parties in collaborative design are called shared parameters and shared functions. In our assumptions, an inference or interaction happens when some protected or non-protected parameters inferred from some shared parameters due to some inherent relationships between design functions and parameters. It is assumed that the original manufacturers design and sell the final product for profits, and the suppliers receive the orders from the original manufacturers for producing component(s) of the final product.

In the phase of product design, the OM and the other suppliers and manufacturers need to share some information about design components for the other parties to be able to design their own design components (Mun et al. 2009). Due to the nature of collaborative design, many researches focused on information protection methods in order to minimize the risk of information leakage in collaborative design such as access control, generalization, simplification (Wang et al. 2006, Mun et al. 2009, Zhang et al. 2004).

These methods were used to control the information sharing of design components between different parties. Cera et al. (2006) defined information protection as creating "need to know" protections on sensitive and critical information. In product design, information protection methods cannot completely prevent leaking of design information for the other parties but these methods can just reduce the risk of leaking of sensitive information. For example, in designing a cell phone, one may need to know the exact shape of the device in order to design GPS for the cell phone but not the information about operating system or application applied in the cell phone being created by another parties. In this case, OM companies try to outsource the information which each designer needs for doing his own part and keep the other information protected. Sometimes, due to the inherent relationships between design functions of one product, some common design elements will be needed for different suppliers. Yet, Zhang et al. (2011,a) represented that the information leakage can take place due to some inferences that are not intellectual properties of companies but it can lead to disclosing these information. Then, the decision about which pieces of information should be shared for collaboration is not obvious. This thesis is intended to address this issue based on the inference notion by Zhang et al. (2011a).

In order to mitigate the risk of information leakage and select one secure set of parameters for sharing, we can group or isolate the protected and non-protected parameters. Clustering and grouping lead to the decomposition and allocation of product design parameters that have an important affect on controlling the risk of inferences and mitigating the risk of information leakage (Zhang et al. 2011a). One of the applications of clustering is in DSM (design structure matrix) and RM (rectangular matrix). Design Structure Matrix was proposed by Steward in 1981 and is a matrix which shows the relations and information flows between elements of one type. But RM represents the dependency among two types of elements. After using DSM/RM, different clustering method can be used to identify the interdependent groups of tasks or system elements. There is no universal and best algorithm for choosing the best clustering method and it totally depends on research objectives (Li 2007).

1.2. Thesis objectives and organization

In collaborative design, companies need to share some design information with the other participating companies in order to make the product design more effective and productive. However, the original manufacturer needs to protect some design information in order to prevent reverse engineering. In this thesis we develop a clustering method introduced by Li (2007) that can isolate and group the IP-sensitive parameters in few modules. By this modularization, we can separate all confidential parameters from non-confidential information. However there are some inferences that connect the modules which contain IP-sensitive parameters to the modules with non-IP sensitive information. In this case, we can control these kinds of interactions in order to mitigate the IP-sensitive information leakage. Furthermore, the quantitative approach is provided to measure and

estimate the risk of information sharing of each module in disclosing the IP-sensitive parameters which caused by potential inferences. Then we can analyze the results of this measurement to decide which set of information can be shared with the other suppliers and manufacturers with low risk of information leakage. The contribution of this thesis is in two-fold. First, we use the modularization method in order to protect the critical information and intellectual properties of original manufacturer using function-parameter (FP) matrix and apply the target coefficient (it has been discussed in Chapter 4) to address the need of clustering IP-sensitive parameters. Secondly, due to the use of the modular matrix, the risk estimation of information leakage which caused by potential inferences (to be discussed in Chapter 5) is also newly derived for the application of secure collaborative design. More justifications can be found in subsequent sections

The rest of the paper is organized as follows. Chapter 2 will review the related works. Chapter 3 will introduce a matrix-based modeling for secure parametric design, with the characterization of security-specific information. Chapter 4 will describe one clustering method for modularization and then develop this method for clustering IP-sensitive parameters. Based on the modular structure, Chapter 5 will propose two different measures to estimate the risk of information leakage. Chapter 6 will present two case studies by applying our method to relief valve and DC motor systems. In conclusions and future works chapter (i.e. Chapter 7) we summarize the contributions of this thesis and we propose several future research directions.

Chapter 2: Literature Review

2.1. Dependency modeling and information leakage

Sharing design information can have both positive and negative effects for original manufacturers (OM) in supply chain. Obviously, sharing the informtion with suppliers and other manufacturers lead to reduced costs and more efficient products (Zhang et al. 2011a). In contrast, sharing some confidentional infromation which are intellectual properties for companies in collaborative design, may lead in loosing the competitive edge for companies and information leakage. In order to summarize all the risks of information leakage in supply chain, Dye and Sridhar (2003), represented the negative effects of using outside consulting for the new defined project on information leakage that lead to reduced value of the project.

Referring to Mun et al. (2009), intellectual property is a concept that can be defined considering the positions of participating companies in collaborative design, i.e., OM and suppliers. Intellectual properties can specify the competitive advantages of each company. One issue of Computers in Industry (2012, vol.63) discusses secure collaboration in design and supply chain management (Zeng et al. 2012a). Therefore, in order to avoid losing the competitive advantages, the original manufacturer needs to be careful about sharing its intellectual property with the other collaborating companies during the product design process. On the other hand, the nature of collaborative design needs this information sharing and collaboration. Recently, Zeng et al. (2012b) have done a comprehensive literature review for secure collaboration in the global design and supply chain environment. In this paper they use EBD methodology in order to collect, organize, and analyze the literature by viewing literature review as a design problem.

Zhang et al. (2011a) discussed, information leakage in supply chain can occur in two different ways: 1- Direct leakage of information. 2- Information leakage that are placed via inferences. In this thesis we focus on both ways of information leakage and represent the critical role of inferences in information leakage by doing modularization, which will be discussed in Chapter 4.

2.1.1. Secure collaboration in collaborative design

As computer-aided design (CAD) becomes popular in engineering design, it is often used to store and transmit the design information in design collaboration. Mun et al. (2009) discussed in his research that in detail design of one product, all participating companies in collaborative design, need to access CAD data of the other parties due the inherent relationships between the design components. However, disclosing CAD data to another party in collaborative design may lead in disclosing the design components that include intellectual properties of a collaborating company. Nowadays, sharing information and collaborating with the other companies are critical for companies to survive in the competitive market. Due to this fact, the secured and effective sharing of design information is a vital concept in collaborative design.

In order to protect the information, some papers focused on the factors and conditions that lead to information leakage and proposed the methods to reduce them. Some researchers proposed to conceal the sensitive data or their sources and un-hide the summary measures like the mean and the variance (Zhang and Li 2006.) Li and Atallah (2006) used linear programming methods to solve supply chain problems by considering security. Then, Deitos et al. (2009) proposed a probability-based technique to improve overall practical performance of linear programming methods which was used to model

the collaborative problem. In order to mitigate the risk of information leakage in supply chain, Zhang et al. (2010) focused on the optimal supplier selection methods. In this thesis, we apply modularization in FP matrix which will be discussed in Chapter 4 in order to group the protected parameters and minimize the risk of information leakage.

Since IP protection is the main goal of information leakage protection methods, we review some important approaches in the literature for securing the IP information. Mun et al. (2009) divided the current approaches for securing the intellectual property of a company in product design collaboration into three categories: access control methods, model simplification and watermarking.

2.1.2. Access control methods for information protection

In the literature, access control is the most common method proposed to control the sharing of information especially CAD information and IP protection. In access control method, everyone has specific authorization to have access to the product information. Access control method has been used to protect sharing of information in different fields (Sun and Wang 2011). Carminati et al. (2011) used access control method for emergency management and Sun and Wang (2011) applied this method for providing information security in e-healthcare services. However, one of the most common usages of access control method is in protecting CAD information in collaborative design.

Wang et al. (2006) defined three major access control methods as: 1- discretionary access control; 2- mandatory access control; 3- role-based access control. In the first method, the access is defined based on the person who wants to have access to data but for the second method (i.e. mandatory access control) each data has specific security level and each user has defined security clearances, and users can only have access to data for which they are

clear. However, role-based access control method has received most attention in the domain of collaborative design. Basically, a role is a concept that connects job functions and users, and it can be used to derive the policy for information security (Sandhu et al. 1996).

Cera et al. (2004) applied role-based access control to information protection in design collaboration. This system was extended to a Scheduled Role-Based Distributed Data Access Control by Wang et al. (2006) that allowed for fine-grained data access control for the security of both sides of customers and servers. He believed that the current access control cannot be directly applied in CAD models due to their pure concentration on the appearance of 3D models. Based on decision making concept, Zheng et al (2012) also proposed a trust-based privacy authorization model which made the fine-grained authorization decision. Cera et al. (2006) developed role-based viewing through integration of multi-resolution geometry and security models by adding the design roles. In this research they focused on information assurance as a critical technique to protect intellectual property in collaborative design.

2.1.3. The other methods for information protection in collaborative design

Although many researches in the collaborative design domain have been devoted to the access control methods, some researchers believed that these methods are not able to address the formalization of a process in order to specify design parameters in a way which is more applicable (Mun et al. 2009). According to this problem in access control methods, simplification model was considered as a method which provided design model considering the accessibility of data for users with different levels of detail (LOD). However, due to the inherent simplification of this method, some critical design

information was deformed during the simplification process which would make problems in product design collaboration. In fact, this method had problems in considering the inferences between different parts of design components (Mun et al. 2009).

Unlike the simplification model, watermarking is a method which keeps the original form of design data (Benedens 1999). As Mun et al. (2009) represented, due to the frequent changing of data in design process, watermarking method cannot be applied in product design collaboration.

Suppression is the other method of controlling the security of information in supply chain (Zhang et al. 2011a). In this method, all confidential and sensitive data will be removed from data sets and documents and the other information can be released. In the domain of collaborative design, this method can be applied for CAD data which contain intellectual property when they exchange between different parties.

Mun et al. (2009) believed none of the above introduced methods could work effectively in securing CAD data in product design collaboration. These methods cannot specify the relationships between design components and their interfaces. He proposed a method to share a skeleton model among collaborating companies. By using this method, the participating companies receive the required information for doing detailed design while the security of the intellectual properties of the original manufacturers will be considered, simultaneously.

Some other proposed dependency models were used in the literature. Zanetti et al. (2008) proposed a framework in order to explain the correlations between the sensitive information and serial–level data. Then Rojas-Arciniegas and Kim (2012) proposed the matrix-based approach that considers information security in determining optimal

product architecture and component sharing decisions. He used DSM clustering using genetic algorithms in order to model the relationships between components. Kim et al. (2012) suggested a data model for the design template in order to specify product design data transmitted between a manufacturer and a design checking service provider by extending the skeleton model. In this thesis we use modularized FP matrix as our dependency model in order to consider the dependency between design parameters and their interfaces.

2.1.4. Information leakage and inferences

Although all the mentioned methods help to mitigate the risk of information leakage, but these methods are not able to control the effect of inferences correctly and they cannot evaluate the risk of information leakage caused by potential inferences and selected shared parameters.

Regarding this problem, some researchers have developed some methods respect to inferences and the following risk that will be concluded from the inferences besides using dependency models. Zhang et al. (2011a) represented the effect of inferences in information leakage by introducing a conceptual model (logical dependency graph) as the dependency model between design components and proposed quantitative method in order to evaluate the risk of information leakage respect to some shared information. In this research, they tried to answer to two questions in order to mitigate the risk of information leakage via inferences: “What inferences are possible, and what is the risk of information leakage caused by such inferences?”. They have also claimed that no satisfactory answers had been provided in other researches. Then, Deng et al. (2012) used DSM model in order to generate suitable product decomposition regarding IP protection

and manufacturing cost reduction issues considering different types of interactions and used the decomposition for an optimal supplier selection to mitigate the leakage risk of confidential information caused by inferences and minimize the manufacturing cost in the process.

The dependency model using in our thesis is function-parameter matrix based on three steps of hierarchical clustering. We also introduce modularization considering inferences which will be discussed in Chapter 4, in order to cover the questions discussed by Zhang et al. (2011a).

2.2. Security measures

In order to improve the process of collaborative design, different criteria and measures are defined in different papers. Yin et al. (2008) proposed 3-dimentional performance measurement model based on design dynamics, time and role-based performance measurement. This model helps the project managers to minimize the collaboration risks. Then, Dain et al. (2010) focused on the measures to evaluate the performance of each supplier during different stages of collaborative design.

During the process of designing a product in collaborative design, we should protect the intellectual property of original manufacturer from the other parties by security measures. However, these security measures should not be considered as a barrier for doing collaboration (Mun et al. 2009). Security measures are different based on the dependency models using in collaborative design. But, the goal of all security measures is to minimize the risk of information leakage in the whole product design process. In the literature, there are some different measures in order to evaluate the role of proposed models for secure collaboration.

Some researchers just compare the result of their experiences when they apply security in their model in order to evaluate the effectiveness of their proposed method in protecting the critical information. For instance, Cera et al. (2006) proposed Hierarchical Role-Based Viewing, for multilevel information security in collaborative 3D assembly design. By applying this method for computer mouse assembly, the costs and risks of collaboration reduced. Rojas-Arciniegas and Kim (2012) proposed a model with security consideration in selecting the set of shared parameters. The results of applying the security model in three printers, was considerable.

On the other hand, some papers focused on quantitative measures in order to evaluate the security and risk of information leakage after applying their proposed model. Zhang et al. (2011a) modeled the knowledge of inferrer and inferences for protected parameters. Based on this model, the probability distribution of protected parameters was defined and the risk of information leakage caused by inferences was calculated by quantitative method. One of the other risks of collaborative design is reverse engineering. Harston and Mattson (2010) used some measures and parameters in order to calculate the barrier and time to reverse engineering. By numerical calculation of time and barrier, the original manufacture can measure the security in collaborative design. As time and barriers are maximized, the security will be higher in design process.

In this thesis, we evaluate the risk of information leakage considering the FP-matrix by quantitative formulations based on two criteria: 1- size of modules in modular FP-matrix which is defined by the number of rows and columns involved in the modules 2- the number of entities (shaded boxes) exist in and between modules. These two formulations will be discussed in Chapter 5.

2.3. Modularization and information leakage

In our life, there is a large amount of data. To manage these data and in order to deal with them, we need to classify or group these data as cluster or module.

There are many definitions for clustering in different papers. Alex et al. (2009) defined clustering as a task whose goal is to determine a finite set of categories (clusters) to describe a dataset according to similarities among its objects. Most papers describe clusters as blocks, which patterns in the same blocks are most similar to each other while the patterns from different blocks are most different. Clustering can be used in many different areas, ranging from engineering, computer sciences, life and medical sciences, to earth sciences, social sciences and economics (Xu 2005).

Defining similarity (or dissimilarity) measure is the first step in cluster analysis in order to depict the closeness of any two objects. In fact similarity (or dissimilarity) measure determines whether two objects are close enough to be put in the same cluster or not. Two typical similarity (or dissimilarity) measures are Euclidean distance and Jaccard coefficient Li (2007).

Clustering techniques can be classified into three main types (Alex et al. 2009): 1- hirerechical clustering, 2- partitional clustering and 3- overlapping clustering.

Hierarchical clustering methods divide or merge existing groups to build hierarchical agglomerative or divisive structure respectively.

One of the most common applications of clustering is in grouping the elements of design structure matrix. A DSM is a matrix represents the dependency and information flow (in-to) between two elements or variables in a complex system from one type. As much as the dependency value is lager, the dependency between two elements is higher. There are

various methods in literature for calculating pair wise dependencies and constructing DSM (Nikanjam et al. 2010). In fact each row of the DSM represents the need of information supported by each column element for that element and each column represents that this elements give information to which elements of each row.

Compared to DSM, rectangular matrix (RM) represents the dependency of two different types of elements. For example according to Li (2007), RM can be a matrix which its columns are labeled by parts and its rows are labeled by machines. So the RM matrix indicates how these machines and parts related to each other. Since DSM and RM are methods that can represent the design problems, we focus on the history of these methods and review the clustering methods that lead in modularization in DSM or RM format with the concern of information protection.

According to Nikanjam et al. (2010), the main objective of DSM clustering is to find clusters that are basically the subsets of DSM elements or tasks such the elements or tasks inside a cluster have maximum relation and interacting and elements or tasks of different clusters have minimum relation and interacting. DSM clustering methods are applicable in architectural improvement in organizations and product design and development (Yu et al. 2009)

Different DSM clustering methods exist in literature to extract clusters from design structure matrix. Some researchers used the Analytic Hierarchy Process (AHP) to represent interdependency between elements or tasks for clustering. Chen et al. (2007) proposed an approach based on numerical coupling strengths and used similarity coefficient methods which are more suitable for numerical coupling based on their

opinion. After constructing the DSM, the authors used a robust approach, average linkage method, to cluster the elements with the lowest distance measure first.

Li (2007) proposed a DSM clustering method that is suitable for all DSM structures and RM. In his approach, he used hierarchical clustering algorithm (single linkage) as clustering method.

Another method was introduced by Fernandez (1998). He tried to find a good DSM clustering arrangements using simulated annealing search technique. According to Yu et al. (2007), "a clustering arrangement is considered to be "good" if only few (or none) interactions are left out and clusters are dense."At first step he considered each element as an individual cluster and evaluates the bids from other clusters, if there are any better bids from other clusters. But Yu et al. (2009) claim that this algorithm has simple objective function which is trade-off between the cost of being inside a module and the overall system benefit and is not suitable for more complex system or products. They also claim that this algorithm with many other suggested techniques such as GAs to form product modules based on Fernandez's (1998) approach, suffers from problems like oversimplified objective function and restricting parameters like maximum number of clusters. So they introduced another DSM clustering method called DSMGA to solve these problems. In DSMGA technique, the authors more focus on clustering metrics. Because they believed many problems are originally because of insufficient metrics for arranging good clustering. For this reason, Yu et al. (2009) proposed clustering metric based on the minimum description length principle and converted the DSM clustering problem to an optimization problem. The authors then used GAs to solve this optimization problem. This method acts well especially for overlapping modules. Some

other methods were proposed for removing the overlapping parts of modules to yield the independent blocks such as an extended CI method.

Some researchers used clustering and modularity concept in order to control the security of information. Rojas-Arciniegas and Kim (2012) proposed the matrix-based approach that considers information security in determining optimal product architecture and component sharing decisions. Then they used modularization in order to separate the sensitive information. Recently, Deng et al. (2012) applied clustering method to decompose the components of the product and allocate them to different suppliers regarding IP protection issues in order to mitigate the risk of information leakage.

While their work and this thesis both use the matrix models and the modularity concept, the work of this thesis is different. Firstly, the notion of inferences is not explicitly discussed in Rojas-Arciniegas and Kim (2012). Their security information is defined by identifying the critical functions (as the user input) and mapping them directly to a set of restricted components (i.e., resulting in the security matrix). Secondly, Rojas-Arciniegas and Kim (2012) and Deng et al. (2012), both perform DSM clustering, and this thesis applies hierarchical clustering for RM clustering. Finally, Deng et al. (2012) applies a matrix based method to decompose the product considering different types of interactions in order to allocate the components to the selected suppliers while mitigating the risk of information leakage. In this paper they focus on optimal supplier selection in order to mitigate the risk of information leakage. However, in our thesis, we focus on the selection of secured parameters for sharing and collecting all protected parameters in separated modules. We consider that these works propose different scopes and aspects in addressing the emerging security issues in product design.

The clustering method of this thesis is based on the framework by Li (2007), and the new element is the use of the target coefficient (to be discussed in Chapter 4), which promotes the modular structures with the concern of information leakage.

Chapter 3: Matrix-based Modeling for Secure Parametric Design

3.1. Function-parameter matrix

Mun et al. (2009) divided the design process to three phases: functional design, conceptual design, and detail design. According to the example in Chapter 6, in the functional design phase of one DC motor, the factors and requirements that satisfy the customers' needs are defined. In the conceptual design phase, a set of components of DC motor with their behavioral properties are selected which are well-known as design parameters in parametric design and their relationships between these parameters are specified by design functions. In the detail design phase, all design and geometric constraints with detailed geometry of parameters, according to the engineering design context of DC motor, are considered.

Consider one design problem with n design parameter and m design function. According to Li (2007), rectangular matrix (RM) represents the dependency of two different types of elements. As design functions and design parameters are two different types of entities, we use RM matrix in order to illustrate the dependency of these two types of entities. In this thesis, we call this RM matrix as FP matrix witch its rows represents the design functions (symbolized f_i) and its columns represent design parameters (symbolized p_i). Each element of this FP matrix can be symbolized as m_{ij} which means the element from i^{th} row and j^{th} column. The value of m_{ij} can be either 0 or 1. m_{ij} equals zero means there is no direct dependency between i^{th} design function and j^{th} design parameter. However, there is possibility of being indirect dependency via inferences which will be discussed in following sections. In contrast, m_{ij} equals one means there is direct dependency between

i^{th} design function and j^{th} design parameter. We represent this FP dependency matrix by M, which

$M = [m_{ij}]$, (i = 1, 2, …, m; j = 1, 2, …, n)

In order to clarify the application of this FP matrix in this step, we choose one small part of FP matrix related to design of DC motor from Chapter 6 as below:

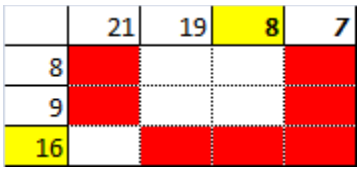


Figure 3-1: A sample FP matrix of DC motor

As shown in Figure 3-1, the FP matrix, M, has three design functions (i.e. f_8, f_9, f_{16}). There are also four design parameters that are labeled with numbers. In this matrix, p_{21}, p_{19}, p_8, and p_7 correspond to design voltage, number of slots or teeth on rotor, depth of slots, and rotor diameter in DC motor, respectively. The complete list of these parameters with their description will be found in Appendix B. The shaded cells of this matrix represent the dependency between related functions and parameters (i.e. m_{ij}=1) and the non-shaded cells remark that there is no dependency between the related functions and parameters (i.e. m_{ij}=0). These dependencies have been concluded from the constraints and physical equations of DC motor design.

3.2. Characterization for secure collaboration

In this thesis, we deal with two different types of entities. The entities that contain information which are important for the original manufacturer to be protected and the entities which their information are vital to be shared with the other parties (manufacturers and suppliers) in order to be able to design their own part correctly and

effectively. With reference to Zhang et al (2011a), we categorize these different types of entities for secure collaboration as follows:

1- Protected parameters: In collaborative design, they include the entities (parameters) which their information must be kept protected from other parties in order to protect the competitive advantages points of OM. In fact, these parameters are IP-sensitive parameters of the original manufacturer.
2- Protected functions: In collaborative design, they include the entities (functions) which contain the protected parameters and the critical information can be derived via these functions.
3- Shared parameters: These types of entities include parameters that need to be shared with the other parties (manufacturers and suppliers) in order to make the collaborative design process more effective.

In this thesis, IP-sensitive parameters, which need to be kept more secured, are identified by the original manufacturer. As the protected parameters are specified, the functions which contain the protected parameters are labeled as protected functions. The IP-sensitive parameters can be disclosed via these functions.

In order to specify the shared parameters, we use some methods in order to minimize the risk of disclosing the protected parameters that it can happen by controlling the inferences. These methods, as the main contribution of this thesis, are introduced in Chapter 4 with more details.

In order to discriminate different kinds of parameters and functions and for better visuals, for all FP matrices, we specify the protected parameters and functions with highlighted cells and we specify shared parameters with bolded and italic cells. Consider FP matrix in

Figure 3-1, p_8 is considered as a protected parameter. As f_{16} contains p_8, we label f_{16} as a protected function. As we can see in this example, p_7 is a shared parameter. Since f_{16} contains p_8 and p_7 as well, the possibility of disclosure of p_8 via f_{16} is maximized. Due to existence of these types of inferences, we need to control the selection of shared parameters in order to minimize the risk of disclosure of IP-sensitive parameters. In order to achieve this goal, we use modularization method which is the new element as compared to the work of Zhang et al. (2011a).

3.3. Utility of modular matrix

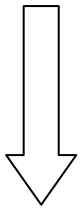
By increasing the number of design parameters in collaborative design, controlling the distribution of protected parameters and inferences in FP matrix will be more complicated. In this case, in order to overcome the complex dependency between parameters, we apply modularization method introduced by Li (2007). In this method, the parameters which are related to the specific structure of the whole design try to be put close to each other in the same module. For example, the parameters of Rotor in design of one DC motor try to be gathered in the same module. It can happen by re-arranging the columns and the rows of the original FP matrix. In the next section, we describe the steps that this modularization can take place.

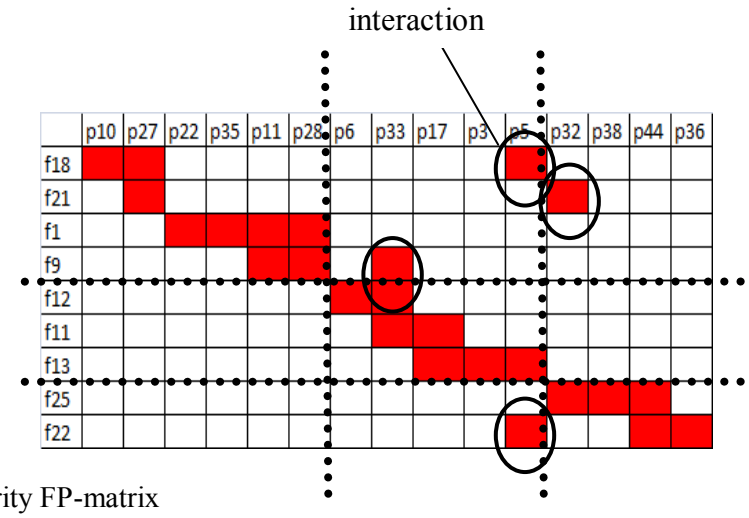
For better understanding, consider FP matrix in Figure 3-2 b. In the FP matrix, all modules are aligned along the diagonal and each module contains the subset of functions and parameters which are highly related to each other. For instance, Module 2 contains $p_6, p_{33}, p_{17}, p_3, p_5$ and f_{12}, f_{11}, f_{13}, which have high dependency. Besides these modules, we can see some non zero entities which represent the dependency between modules. In this thesis, we call these entities as interactions or inferences. The combination of all modules

and interactions covers the whole functions and parameters. By modularization, we can isolate the parameters which are as IP-sensitive of the company in one or few modules which we call them protected module(s). By this method, we can keep the information of these module(s) more secure. However, there are always some interactions between the protected and non-protected module(s) which can lead to disclosure of protected parameters. For instance, if we set p_{22} as protected parameter, Module 1 is labeled as protected module. Now, if we share some parameters from Module 2, such as p_{17}, the parameters of Module 1 can be disclosed via f_9 and f_{18} which are the interaction entities. So, it can show how much the tracking and controlling the interactions and selecting the appropriate set of parameters for sharing is important in secure collaborative design. In this thesis, the formation of a modular matrix is achieved by using the clustering method, which will be discussed in the next chapter.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		p3	p5	p6	p10	p11	p17	p22	p27	p28	p32	p33	p35	p36	p38	p44
1	f1					■		■		■			■			
2	f9					■				■		■				
3	f11						■					■				
4	f12			■								■				
5	f13	■	■				■									
6	f15			■												
7	f18		■		■				■							
8	f21								■		■					
9	f22		■											■		■
10	f25										■				■	■

a) A sample FP-matrix for DC motor





b) Modularity FP-matrix

Figure 3-2: Matrix-based modularization process

Chapter 4: Matrix-based Clustering Method for Modularization

Alex et al. (2009) defines clustering as a task whose goal is to determine a finite set of categories (clusters) to describe a dataset according to similarities among its objects. Clustering can be used in many different areas, ranging from engineering, computer sciences, life and medical sciences, to earth sciences, social sciences and economics (Xu 2005). In this thesis, we apply clustering method in order to generate modules which include design parameters and functions and in order to analyze the dependency of parameters within and between modules. For this reason, we use the clustering method which its steps are conformed to the steps of hierarchical clustering, particularly, the agglomerative (i.e., bottom-up) procedure.

4.1. Fundamental of hierarchical clustering

In an agglomerative hierarchical clustering method, there is no single step for partitioning of data to particular number of clusters and groups, instead in this method, some sequential partitioning take place which starts form one cluster containing all individual entities.

The general algorithm for hierarchical clustering is as below (Li 2007):

Step 1: Measure the similarity/dependency value between any two entities.

Step 2: Select the entities that illustrate the highest dependency value. Group them in one cluster and consider them a grouped entity.

Step 3: Update the dependency values considering the newly joint entity.

Step 4: Check if all the entities are clustered stop the procedure. Otherwise, go to Step 2.

The output of this clustering will be a tree which its branches are constructed from the merged entities in each phase. This tree represents how entities should be clusters progressively. For better visuals, we set a small example with six entities (a,b,c,d,e,f). The similarity/dependency matrix of these entities is shown in Table 4-1.

According to first step of hierarchical clustering method, we pick the entities with the highest similarity value. In this figure, ‘d’ and ‘e’ have the highest dependency equals to 0.93. We pick these two entities and group them as one entity (de). So, ‘de’ entity is the first branch of the tree structure. Then, we go to step 2 and update all the similarity values. The similarity values for ‘de ’are calculated by averaging the values related to entities‘d’ and ‘e’. The same procedure can be applied to the updated similarity matrix until it cannot be further reduced. The final tree with its all branches is shown in Figure 4-1.

	a	b	c	d	e	f
a	1	0.76	0.25	0.14	0.32	0.09
b	0.76	1	0.24	0.87	0.43	0.11
c	0.25	0.24	1	0.55	0.27	0.81
d	0.14	0.87	0.55	1	**0.93**	0.12
e	0.32	0.43	0.27	**0.93**	1	0.62
f	0.09	0.11	0.81	0.12	0.62	1

Table 4-1: Original similarity/dependency matrix

	a	b	c	f	de
a	1	0.76	0.25	0.09	**0.23**
b	0.76	1	0.24	0.11	**0.65**
c	0.25	0.24	1	0.81	**0.41**
f	0.09	0.11	0.81	1	**0.37**
de	**0.23**	**0.65**	**0.41**	**0.37**	**1**

Table 4-2: Updated similarity matrix

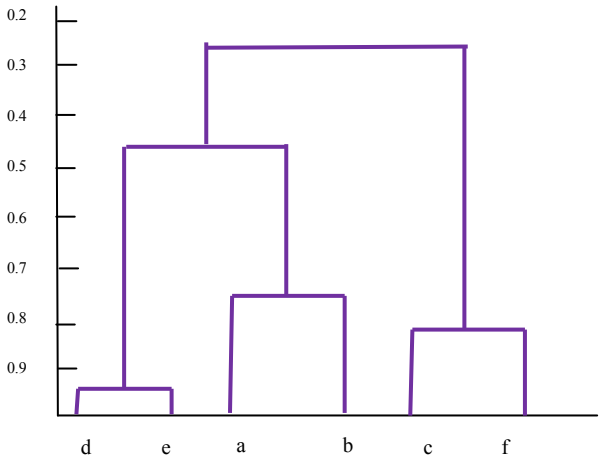


Figure 4-1: The tree structure of hierarchical clustering

By constructing the tree structure of our example, we can drive different clusters based on the desired number of clusters. For instance, if the desired number of clusters is three, the output clusters will be {de},{cf} and {ab}. But if the desired number of cluster is just two, the output clusters will be {abde} and {cf}.

This tree structure helps the grouping of entities in order to make the final clusters, but to finalize theses clusters, further information will be provided in the following chapters.

4.2. Three-phase matrix clustering framework

In this section, we introduce and develop the clustering phases defined by Li (2007) which correspond to the phases of hierarchical clustering method. These three phases are described briefly as follows:

1- Coupling analysis: represents the dependency information between any two design entities considering the similarity/dissimilarity index. Using this information, we can decide about putting the two entities in one module or separate one.
2- Sorting analysis: bring the entities with high coupling index, close to each other and sequence all design entities based on their coupling values. The diagonal/sorted matrix is as the result of this phase.
3- Partitioning: partition the sorted matrix and group the entities and identify the interactions between the groups based on various criteria such as intended size of modules, number of modules, and maximum number of interactions.

In the next section, we will describe each phase considering their application for clustering the FP matrix in collaborative design.

4.2.1. Coupling analysis

Secure sharing of information in collaborative design has the most priority for all participating parties. For this reason, the likelihood of information leakage of one design entity via the other design entities due to the existence of dependency between them is a

big deal for companies. Considering the FP matrix introduced in Chapter 3, this information leakage can be defined as the disclosure of some parameters information via the other parameters through the functional dependency.

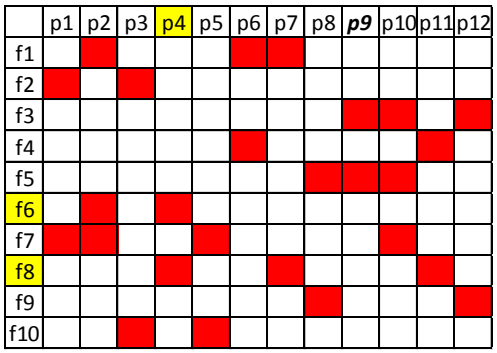


Figure 4-2: A sample FP matrix for relief valve

For better understanding about the coupling concept in FP matrix, consider the FP matrix in Figure 4-2 which represents 10 functions and 12 parameters. Without using any specific formulation, we can say that p_1 and p_2 are coupled and the information of p_1 can be leaked via f_7 (vice versa). However, the coupling degree is different between various entities. In order to normalize and formulate different coupling degree between entities, we count the number of functions that affect each two entities. In this example, we count the number of functions that contain p_1 and p_2. Using the notation of FP matrix formulated in (1), the coupling between two parameters p_i and p_j, can be calculated using the min/max coefficient formulated as follows:

$$R_{\min/\max}(p_i, p_j) = \frac{\sum_{k=1}^{m} \min(m_{ki}, m_{kj})}{\sum_{k=1}^{m} \max(m_{ki}, m_{kj})} \qquad i, j \in [1, n] \tag{2}$$

In the above formulation, the min operator counts the number of functions that contain both considered parameters and max operator counts the number of functions that contain each of the considered parameters. We can also use the same formulation for calculating the coupling value between each two functions. This formulation is as follows:

$$R_{\min/\max}(f_i, f_j) = \frac{\sum_{k=1}^{n} \min(m_{ik}, m_{jk})}{\sum_{k=1}^{n} \max(m_{ik}, m_{jk})} \qquad i, j \in [1, m] \tag{3}$$

In this thesis, we also need to calculate the coupling value between parameters and functions which are from two different types. In this context, we use the two-mode coupling coefficient introduced by Li (2011). This formulation is as follows:

$$R_{rc}(f_i, p_j) = \frac{2m_{ij}}{\sum_{k=1}^{m} m_{ik} + \sum_{k=1}^{n} m_{kj}} \qquad i \in [1, m], j \in [1, n] \tag{4}$$

When f_i and p_j are only related to each other not the other entities, the value of above formulation will be 1. In this case, these two entities will be definitely grouped in the same cluster. In contrast, if f_i and p_j are also related to the other entities, the chance of being in the same cluster will be lower. This situation is captured in the denominator of Formulation (4).

The point in these formulation is that the coupling values between the same type of entities and different type of entities are bounded between zero (i.e., no coupling) and one (i.e., two entities must be grouped in the same block).

Accordingly, three matrices denoted as CM_r, CM_c, and CM_{rc} are the outputs of each type of formulation respectively (i.e. p_i and p_j, f_i and f_j, p_i and f_j). In order to make one unit symmetric matrix denoted as CM, we combine three coupling matrices CM_r(n×n), CM_c(m×m), and CM_{rc}(m×n) as follows.

$$CM = \begin{bmatrix} w_r \cdot CM_r & w_{rc} \cdot CM_{rc} \\ w_{rc} \cdot CM_{rc}^T & w_c \cdot CM_c \end{bmatrix} \tag{5}$$

where w_c, w_r and w_{rc} are the weights for three coupling matrices. This combined CM is a square and symmetric matrix that records the coupling values between any two design entities, and a design entity can be either a design function or a design parameter. Since the coupling values in (5) are determined via two different coefficients (i.e., the min/max and two-mode coefficients), the weights in (5) are intended to minimize any bias from these coefficients. Thus, it is required that the average of the coupling values determined by the min/max coefficient be roughly equal to the average of the coupling values determined by the two-mode coefficient. Such criterion is expressed in the following formulation.

$$w_{rc}\left(\frac{\sum CM_{rc}}{m \cdot n}\right) \approx \frac{w_r}{2}\left(\frac{\sum CM_r}{m^2 - m}\right) + \frac{w_c}{2}\left(\frac{\sum CM_c}{n^2 - n}\right) \tag{6}$$

In order to increase the chance of protected parameters and functions being grouped in the same modules and keeping the related parameters and functions close to each other, we introduce one concept as target coefficient. In modularization, target coefficient makes the protected parameters and their related parameters and functions more highlight by affecting the coupling values, hence, the protected parameters and all their

dependencies can be grouped in the same modules in order to mitigate the risk of information leakage. The target coefficients are considered in three different situations between entities from the same and different types. These situations are classified as below:

- Two entities are protected. In this situation, we apply weights for the coupling values in order to increase the chance of being grouped together.
- One entity is protected, and another one is non-protected. In this situation, we still apply weights for the coupling values in order to increase the chance of being grouped together. Yet, the scale of increase should not be greater than that of the previous situation.
- Two entities are non-protected. In this situation, the coupling value remains unchanged.

Denote $R(e_i, e_j)$ the coupling value between the ith and jth entities, and the entity can be either a design function or a design parameter. Then, the target coefficient (denoted as $R_{target}(e_i, e_j)$) can be formulated as follows.

$$R_{target}(e_i, e_j) = \begin{cases} R(e_i, e_j)^{ttw} & \text{if both } e_i, e_j \text{ are protected entities} \\ R(e_i, e_j)^{tnw} & \text{if only one of } e_i, e_j \text{ is a protected entity} \\ R(e_i, e_j) & \text{if both } e_i, e_j \text{ are non-protected entities} \end{cases} \quad (7)$$

where *ttw* and *tnw* are the weights to increase the coupling values for the above first and second situations, respectively. The values of *ttw* and *tnw* are between 0 and 1. If the values of *ttw* and *tnw* are smaller, the weighting effects are stronger (i.e., higher increase on coupling values). For better understanding, we apply the target coefficient for the

sample FP matrix in Figure 4-2. In this example, we consider *ttw* = 0.1 and *tnw* = 0.3. As we can see in the following matrices, the coupling values for the target entities (protected parameters and functions) are relatively higher.

Applying target coefficient as a tool for helping to group the protected entities in the same modules is one of the contribution of this thesis.

	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
f1	-	0	0	0.25	0	0.66	0.17	0.62	0	0
f2	0	-	0	0	0	0	0.2	0	0	0.33
f3	0	0	-	0	0.5	0	0.17	0	0.25	0
f4	0.25	0	0	-	0	0	0	0.66	0	0
f5	0	0	0.5	0	-	0	0.17	0	0.25	0
f6	0.66	0	0	0	0	-	0.62	0.87	0	0
f7	0.17	0.2	0.17	0	0.17	0.62	-	0	0	0.2
f8	0.62	0	0	0.66	0	0.87	0	-	0	0
f9	0	0	0.25	0	0.25	0	0	0	-	0
f10	0	0.33	0	0	0	0	0.2	0	0	-

a) Row coupling matrix (CM_r)

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
p1	-	0.25	0.33	0	0.33	0	0	0	0	0.25	0	0
p2	0.25	-	0	0.66	0.25	0.25	0.25	0	0	0.2	0	0
p3	0.33	0	-	0	0.33	0	0	0	0	0	0	0
p4	0	0.66	0	-	0	0	0.72	0	0	0	0.72	0
p5	0.33	0.25	0.33	0	-	0	0	0	0	0.25	0	0
p6	0	0.25	0	0	0	-	0.33	0	0	0	0.33	0
p7	0	0.25	0	0.72	0	0.33	-	0	0	0	0.33	0
p8	0	0	0	0	0	0	0	-	0.33	0.25	0	0.33
p9	0	0	0	0	0	0	0	0.33	-	0.67	0	0.33
p10	0.25	0.2	0	0	0.25	0	0	0.25	0.67	-	0	0.25
p11	0	0	0	0.72	0	0.33	0.33	0	0	0	-	0
p12	0	0	0	0	0	0	0	0.33	0.33	0.25	0	-

b) Column coupling matrix (CM_c)

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
f1	0	0.33	0	0	0	0.4	0.4	0	0	0	0	0
f2	0.5	0	0.5	0	0	0	0	0	0	0	0	0
f3	0	0	0	0	0	0	0	0	0.4	0.33	0	0.4
f4	0	0	0	0	0	0.5	0	0	0	0	0.5	0
f5	0	0	0	0	0	0	0	0.4	0.4	0.33	0	0
f6	0	0.76	0	0.99	0	0	0	0	0	0	0	0
f7	0.33	0.29	0	0	0.33	0	0	0	0	0.29	0	0
f8	0	0	0	0.99	0	0	0.76	0	0	0	0.76	0
f9	0	0	0	0	0	0	0	0.5	0	0	0	0.5
f10	0	0	0.5	0	0.5	0	0	0	0	0	0	0

c) Row-column coupling matrix (CM_{rc})

Figure 4-3: Coupling matrices of the sample matrix after applying the target coefficient

4.2.2. Sorting analysis

The goal of this step is to rearrange all rows and columns of coupling matrix based on their coupling strength resulted from previous step. We do this rearranging or sorting by bringing all the functions and parameters, which are highly coupled, close to each other. The sequence of entities in rows/columns is such that the chance of high coupled entities is more to be closed together. As mentioned above, we apply hierarchical clustering method in order to do sorting analysis. The detailed description of sorting analysis steps with its application for sample FP matrix will be found in Appendix A.

4.2.3. Partitioning analysis

Considering the diagonal matrix in the previous section, partitioning analysis identifies the points that the clusters and modules can be formed. These partition points are specified using the tree from previous section and based on some criteria (such as size of module and number of modules). In case of having a large size matrix, this tree can facilitate finding good partition points. More algorithmic details of the tree-based approach can be found in Li (2011) and Appendix A.

To mitigate the risk of information leakage in collaborative design, we use this three phase clustering method in order to separate the protected parameters and functions in one or few modules. These modules which contain protected parameters are labeled as protected modules and their information will be kept secure. However, the information of the other modules will be shared with the other parties. Then, we use this modular matrix and propose the risk formulation in the next chapter in order to estimate the risk of sharing the information and mitigate it as much as possible.

Chapter 5: Estimation of Information Leakage

In order to formulate the estimation of information leakage risk, we first examine the distribution of shared and protected entities, respectively, in the modular FP matrix. In this dissertation, we propose two different types of information leakage risk formulation considering two factors: 1- size of modules in modular FP-matrix which is defined by the number of rows and columns involved in the modules, 2- the number of entities (shaded boxes) exist in and between modules. Then we briefly compare these two types of formulation. In fact, in this chapter, we define size of modules in two aspects.

In view of the distribution of shared parameters, any module that contains shared parameters is labeled with "S" (i.e., symbolize the meaning of "shared"). To illustrate, suppose that the sample modular FP matrix in Figure 5-1 has $p_{10}, f_{22}, f_{25,} f_1$ as protected entities and p_{26} as a shared parameter. Then, the corresponding module (i.e., Module 3) is termed "S-labeled". In this context, the suppliers are able to know the information of parameters pertaining to the S-labeled module (i.e., p_{12}, $p_{17,}$ p_{26} and p_{20} in Module 3) due to common functions. Then, we want to examine how the S-labeled modules interact with other modules. By definition, any module that is directly related to S-labeled module(s) via interactions in the modular FP matrix is described as S-related. In Figure 5-1, since p_{12} in the S-labeled module is related to f_{22} and $f_{25}$of Module 2 and f_2 of Module 1, Module 2 and Module 1 are identified as S-related. In sum, Module 3 is S-labeled, and Module 2 and Module 1 are S-related.

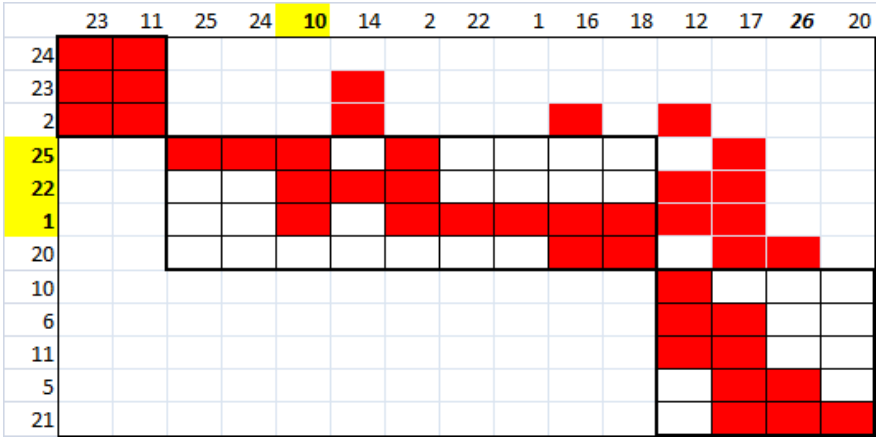


Figure 5-1: Modular FP matrix

In view of the distribution of protected entities, any module that contains protected parameters is labeled with "P" (i.e., symbolize the meaning of "protected"). To illustrate, suppose that P_{10} is a protected parameter in Figure 5-1. Then, the corresponding module (i.e., Module 2) is P-labeled. In this context, the original manufacturers intend to protect the information of the parameters in the P-labeled module. Then, we want to examine how the P-labeled modules are related to other modules. By definition, any module that is directly related to P-labeled module(s) via interactions in the modular FP matrix is described as P-related. In Figure 5-1, since P_{14} in the P-labeled module is related to f_{22} of Module 3 and f_2, f_{23} of Module 1, Module 1 and Module 3 are identified as P-related. In sum, Module 2 is P-labeled, and Module 1 and Module 3 are P-related in Figure 5-1.

5.1. Formulation type 1

In order to estimate the risk of information leakage, this type of formulation focuses on size of modules in modular FP-matrix. The size of a module is defined as the product of the number of its functions (rows) and the number of its parameters (columns). For

example, the size of Module 1 in Figure 5-1 is 3 (functions) multiplied with 2 (parameters), which is equal to 6.

In this type of formulation, we consider the modules which contain elements related to shared and protected parameters simultaneously. So we need to check the modules that are overlapped with "P" and "S" elements to identify the potential of information leakage. For instance, Module 2 in Figure 5-1 is both P-related and S-related, and information leakage may take place via this module. Particularly, we denote the sizes of four types of overlapping modules as follows:

- A_{LL} = the total size of modules that are P-labeled and S-labeled
- A_{LR} = the total size of modules that are P-labeled and S-related
- A_{RL} = the total size of modules that are P-related and S-labeled
- A_{RR} = the total size of modules that are P-related and S-related

In these four types of overlapping modules, A_{LL} represents the highest risk of information leakage since both protected and shared entities are contained in the same modules. Comparatively, A_{RR} represents the least risk due to indirect information leakage via interactions. Then, A_{LR} and A_{RL} represent the leakage risk between A_{RR} and A_{LL}. Accordingly, different weighting factors are applied to these terms to capture this reasoning. To normalize the measure of information leakage, we further define A_{p_label} as the total size of P-labeled modules and A_{p_relate} as the total size of P-related modules. Accordingly, the measure of information leakage can be formulated as follows.

$$L = \frac{A_{LL} + w_1 A_{LR} + w_2 A_{RL} + w_3 A_{RR}}{A_{p\_label} + w_4 \cdot A_{p\_relate}} \tag{9}$$

where w_1, w_2 , w_3, and w_4 are the weighting factors (between 0 and 1) to capture the effects of information leakage from P-related and S-related modules. In this formulation, the value of w_3 should be smaller than the values of w_1 and w_2 due to the absence of the direct effects from P-labeled or S-labeled modules (i.e., $w_3 \leq w_1$ and $w_3 \leq w_2$). In addition, since w_4 represents the effect of P-related modules, its value should be between the values of w_2 and w_3 (i.e., $w_3 \leq w_4 \leq w_2$). In this thesis, we set $w_1 = w_2 = w_4 = 0.5$ and $w_3 = 0.3$.

5.1.1. Demonstration

By considering P_{10} as protected parameter in Figure 5-2, we assume three different shared parameters in each module to represent three different cases of information leakage. Since P_{10} appears in Module 2, we label this module as P-labeled Module, and as mentioned before, due to the interaction entities between Module 2 and Modules 1&3, we label Module 1 and Module 3 as P-related modules.

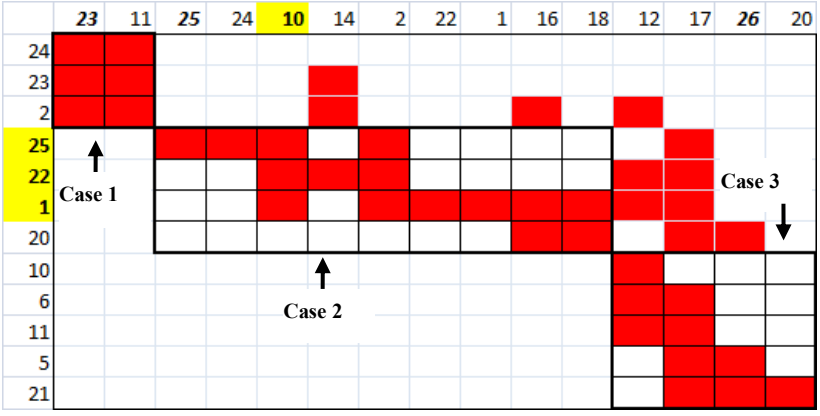


Figure 5-2: A sample FP matrix with three cases

In Case 1, we set P_{23} as a shared parameter which makes Module 1 as S-labeled and Modules 2&3 as S-related. In Case 2, we set P_{25} as a shared parameter which makes Module 2 as S-labeled and Modules 1&3 as S-related. In Case 3, we set P_{26} as a shared parameter which makes Module 3 as S-labeled and Modules 1&2 as S-related. Tables 5-1 and 5-2 have summarized the details of the case setup.

In the qualitative comparison, Case 2 should present the highest risk of information leakage because both protected and shared parameters appear in the same module. To estimate the risk of information leakage, at first step we calculate the size of modules by multiplying the number of parameters and functions in each module as follows:

Size of Module 1: 3*2= 6

Size of Module 2: 4*9= 36

Size of Module 3: 5*4 =20

We explain the estimation of information leakage risk for case 1 to clarify the suggested formulation. As Module 2 is P-labeled and Modules 1&3 are P-related in all cases, A_{p_label} = Size of Module 2= 36 and A_{p_relate} = Size of Module 1 plus Size of Module 3= 6+20 = 26.

In case 1, since P-labeled and S-labeled Modules are different, A_{LL} =0. The overlapping module (i.e. Module 2), makes A_{LR} = 36. On the other hand, Module 1 is another overlapping module leading to A_{RL} = 6. Finally, A_{RR} = 20, because the only overlapping module, is Module 3 which is S-related and P-related. By taking the weighting factors, the risk estimation in Case 1 is equal to:

Case 1: (0+(0.5*36)+(0.5*6)+(0.3*20))/(36+26) = 0.43

Case 2: (36+(0.5*0)+(0.5*0)+(0.3*(20+6)))/(36+26) = 0.70

Case 3: (0+(0.5*36)+(0.5*20)+(0.3*6))/(36+26) = 0.48

Table 5-1 has shown the resulting estimates of the information leakage for three cases. We can see that Case 2 yields the highest risk of information leakage and Cases 1&3 the medium risk which proves the above claim.

	Protected Parameter	P-labeled	P-related	Shared Parameter	S-labeled	S-related	Leakage Estimate
Case 1	P_{10}	Module 2	Modules 1&3	P_{23}	Module 1	Modules 2&3	0.43
Case 2	P_{10}	Module 2	Modules 1&3	P_{25}	Module 2	Modules 1&3	0.70
Case 3	P_{10}	Module 2	Modules 1&3	P_{26}	Module 3	Modules 1&2	0.48

Table 5-1: Case setup for demonstrating the risk estimation of information leakage (Formulation type 1)

5.2. Formulation type 2

Since formulation type1 is based on the size of modules while the number of interactions between modules, especially between protected module and shared modules, has no effect on the result of this formulation, we introduce the other type of formulation. In fact Formulation type 2 represents the effect of intensity of interactions between modules which are one of the most important reasons in information leakage. In order to make the Formulation type 2, we consider the number of dependency entities (shaded boxes) as the basis for information leakage risk estimation. Since the size of modules and interactions between modules affect the information leakage, we examine the size of modules that are

overlapped with shared and protected entities and the number of interaction parameters between the non-overlapped modules to estimate the risk of information leakage.

The quantification of the risk of information leakage is based on the proportion of P-labeled and P-related modules associated with shared parameters (in terms of S-labeled and S-related modules) and the interactions between modules. First of all, the size of a module is defined as follow:

$$M_k = \sum_{i_k=1}^{n_k} \sum_{j_k=1}^{m_k} e_{i_k j_k} \qquad e_{i_k j_k} = \{0 \text{ or } 1\} \tag{10}$$

Where M_k is the total size of kth module, $i_k = 1$ is the first row of the module K and n_k is the last row included in module K, $j_k = 1$ is the first column of Module K and m_k is the last column included in module k, $e_{i_k j_k}$ is the ijth entity of module k, which can be zero or one. In fact the size of each module is the number of entities that represent the dependency in the module (shaded entities in Figure 5-2). For example, the size of Module 1 in Figure 5-2 is 6, which means 6 non-zero dependencies exist in Module 1.

Secondly, we need to check the modules that are overlapped with "P" and "S" elements to identify the potential risk of information leakage. For instance, Module 2 in Figure 5-2 is both P-related and S-related, and information leakage may take place via this module. Particularly, we denote the sizes of four types of *overlapping* modules as follows:

- M_{LL} = the total size of modules that are P-labeled and S-labeled
- M_{LR} = the total size of modules that are P-labeled and S-related
- M_{RL} = the total size of modules that are P-related and S-labeled
- M_{RR} = the total size of modules that are P-related and S-related

Finally, we need to define different types of non-overlapping modules in order to consider the interactions between them (For example in Figure 5-2, the number of interaction entities between Module 2 and Module 3 is 7) as follows:

- P_LS_L = the total number of interaction entities between P-labeled and S-labeled Modules
- P_LS_R = the total number of interaction entities between P-labeled and S-related Modules
- P_RS_L = the total number of interaction entities between P-related and S-labeled Modules
- P_RS_R = the total number of interaction entities between P-related and S-related Modules

In order to formulate the risk of information leakage, we need to consider different risks, with different weights. Firstly, the risk of information leakage between P-labeled and S-labeled modules represents the highest risk, because information leakage of this type, directly lead to the information leakage of protected parameters. Comparatively, the risk of information leakage between P-related and S-related Modules represents the least risk due to indirect information leakage via interactions. Then, the risks of information leakage between P-labeled and S-related Modules and P-related and S-labeled Modules represent the leakage risk between two first types of risk. Accordingly, different weighting factors are applied to these terms to capture this reasoning.

Considering these classifications, the number of interaction entities between the overlapped Modules with "S" and "P" is equal to size of the overlapped Modules. For example, in Figure 5-2, if we consider p_{23} as shared parameter, the number of interaction

entities between P-labeled Modules and S-related Modules is equal to the size of Module 2 (Module 2 is the overlapped Module with P-labeled and S-related parameters) plus the interactions between Module 2 and Module 3. In fact, since the P-labeled and S-related parameters exist in the same module, all the dependency entities in Module 2 (i.e. size of Module 2) may lead to information leakage between P-labeled and S-related parameters. On the other hand, the other S-related module, i.e. Module 3 has some interaction with Module 2. So, the interaction between these two Modules can cause information leakage. Accordingly, the case which both protected and shared entities are contained in the same modules represents the highest risk of information leakage. In this case, there are no interaction entities between P-labeled and S-labeled, but all the entities in this module can directly lead to information leakage.

To normalize the measure of information leakage, we further define M_{p_label} as the total size of P-labeled modules plus the number of interaction entities related to P-labeled modules and M_{p_relate} as the total size of P-related modules plus the number of interaction entities related to P-related modules. Accordingly, the measure of information leakage considering the size of overlapped modules and the number of interaction entities between non-overlapped modules can be formulated as follows:

$$L = \frac{P_L S_L + M_{LL} + w_1(P_L S_R + M_{LR}) + w_2(P_R S_L + M_{RL}) + w_3(P_R S_R + M_{RR})}{M_{p\_label} + w_4 \cdot M_{p\_relate}} \qquad (11)$$

where w_1, w_2 , w_3, and w_4 are the weighting factors (between 0 and 1) to capture the effects of information leakage from P-related and S-related modules. In this formulation, the value of w_3 should be smaller than the values of w_1 and w_2 due to the absence of the direct effects from P-labeled or S-labeled modules (i.e., $w_3 \leq w_1$ and $w_3 \leq w_2$). In addition, since w_4 represents the effect of P-related modules, its value should be between the values of w_2 and w_3 (i.e., $w_3 \leq w_4 \leq w_2$). In this dissertation, we set $w_1 = w_2 = w_4 = 0.7$ and $w_3 = 0.5$.

5.2.1. Demonstration

The distribution of shared and protected parameters and the conditions of three cases are the same as the one we explained for Formulation type1. But this time, in order to represent the risk of information leakage, we use the formulation type 2 for these three cases.

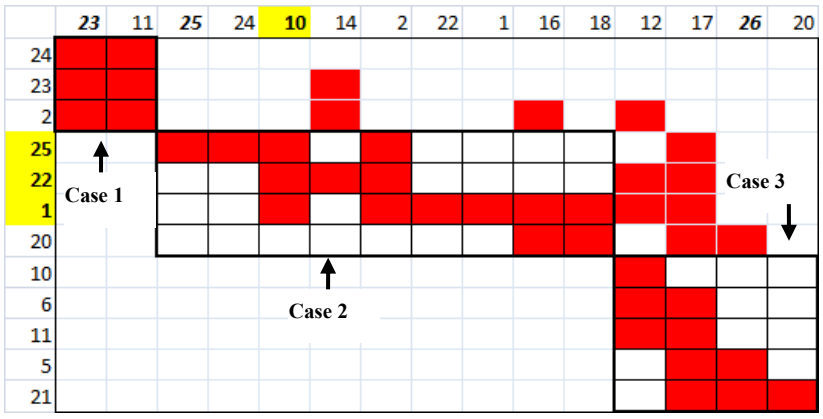


Figure 5-3: A sample FP matrix with three cases

Considering Figure 5-3, Module 2 is P-labeled and Modules 1&3 are P-related for all cases. The size of module 1, regarding the Formulation 10 is as follows:

$$M_1 = \sum_{i_1=1}^{n_1} \sum_{j_1=1}^{m_1} e_{i_1 j_1} \qquad e_{i_k j_k} = \{0 \text{ or } 1\}$$

Since in Module 1, there are 6 shaded boxes, M_1= 6. According to this explanation, M_2= 15 and M_3= 10. As Module 2 is P-labeled and Modules 1&3 are P-related in all cases, M_{p_label} = Size of Module 2 plus the number of interaction related to Module 2 = 36+10= 46 and M_{p_relate} = Size of Module 1 plus Size of Module 3 plus the number of interaction related to Modules 1&3= 6+20+11 = 37.

We explain the way of estimating the risk of information leakage for Case 1, step by step. Firstly, we need to calculate the size of overlapping Modules in Case 1. Considering the different S-labeled and P-labeled in Case 1, M_{LL}=0. Since Module 2 is P-labeled and S-related, M_{LR}=15 which is equal to size of Module 2. On the other hand, the overlapping Module 1which is S-labeled and P-related at the same time, makes M_{RL}=6. Consequently, Module 3 as P-related and S-related module, makes M_{RR}=10 which is equal to the size of Module 3.

Secondly, we need to calculate the number of interaction between non-overlapping modules. Since in Case 1, Module 2 is P-labeled and Module 1 is S-labeled, P_LS_L =3 (the number of shaded boxes between Module 1 and Module 2). As mentioned before, Modules 1&3 are P-related which make P_RS_L= 1, which means the number of interaction between Module 1and Module 3(non-overlapping modules). On the other hand, Modules 2&3 as S-related modules, leading to P_LS_R= 7 (the number of shaded boxes between Module 2 and Module 3). Since Module 2&3 are S-related and Modules 1&3 are P-related, we need to consider different situations in order to calculate P_RS_R: 1- the number of interactions between Modules 1&2, 2- the number of interaction between Modules 2&3, 3- the number of interactions between Modules 1&3. According to these different situations, the total number of P_RS_R= 11.

By applying the weighting factors, the risk estimation in Case 1 is equal to:

$$\text{Case 1: } L = \frac{3+0+0.7(7+15)+0.7(1+6)+0.5(11+10)}{46+0.7(37)} = 0.47$$

$$\text{Case 2: } L = \frac{0+15+0.7(10+0)+0.7(10+0)+0.5(1+16)}{46+0.7(37)} = 0.52$$

$$\text{Case 3: } L = \frac{7+0+0.7(3+15)+0.7(1+10)+0.5(11+6)}{46+0.7(37)} = 0.49$$

Table 5-2 has shown the resulting estimates of the information leakage for three cases. We can see that Case 2 yields the highest risk of information leakage. But as we will discuss in the next section, the difference between Case 2 and two other cases is not considerable.

	Protected Parameter	P-labeled	P-related	Shared Parameter	S-labeled	S-related	Leakage Estimate
Case 1	P_{10}	Module 2	Modules 1&3	P_{23}	Module 1	Modules 2&3	0.47
Case 2	P_{10}	Module 2	Modules 1&3	P_{25}	Module 2	Modules 1&3	0.52
Case 3	P_{10}	Module 2	Modules 1&3	P_{26}	Module 3	Modules 1&2	0.49

Table 5-2: Case setup for demonstrating the risk estimation of information leakage (Formulation type 2)

5.3. Comparison between two types of Formulation:

Considering Formulation type 1, the estimation of information leakage risk is just based on the selected size of modules and the number of columns and rows which are included in these modules. That is, if we select different partition points and consequently,

different size of modules, the results of the estimation of information leakage risk will be different, regardless the number of interactions exists between these modules. In contrast, Formulation Type 2 focuses on two factors, the number of entities which are included in the modules and the number of interaction between the selected modules.

As we can see in the results of estimation risk for Formulation type 2, Case 2 has the highest risk but comparatively, Case 3 has high risk which is too close to the risk of Case 1 due to the large number of interaction between Module 3(S-labeled) and Module 2 (P-labeled). It means, despite of having different P-labeled and S-labeled Modules, the large number of interaction between these two modules lead to the high risk of information leakage. But, Formulation Type 1 cannot support the effect of interactions on risk of information leakage.

Chapter 6: Application

In this chapter, we will justify our claim about the effects of modularization considering inferences in secure collaborative design. The inquiries in this context are: 1- Can "the grouping of target entities" minimize the chance of interferences that lead to information leakage? 2- Which parameters should be shared with the suppliers in order to minimize the risk of leaking the information of protected parameters? 3- Can the matrix-based metric appropriately estimate the leakage situations?

In order to answer to the above inquiries, we consider two different case studies: 1- DC motor 2-relief valve. Firstly, we yield the FP-matrix for each case to show how information leakage takes place and then we investigate the effect of grouping parameters and functions in information leakage by using the clustering method introduced in Chapter 4. Consequently, we set different cases for each case study by considering different set of shared parameters. In order to represent the effect of modularization considering inferences in secure collaborative design, we apply two types of formulations to estimate the risk of information leakage for each set of shared parameters.

Finally we consider two different situations: 1- by assigning target coefficient as one of our contributions in the clustering method, 2- removing target coefficient concept and doing modularization without it. Then we compare the leakage situations by considering protected parameters as our target entities and estimate the leakage information for all different situations in order to show how considering target coefficient can affect the secure collaboration.

6.1. DC motor

A DC motor is one kind of electric motor which produces torque in order to turn the motor by using electricity and magnetic field. In fact, the torque is provided by different forces of magnets with two opposite polarity (i.e. repellent and attractive electromagnetic forces of the magnets) (Kim 2011).

Since all the magnets are polarized with a positive and a negative side, the attraction between these two different poles and the repulsion of similar poles, convert the electricity to the motion. These properties cause motors turns. Besides the poles, one DC motor requires at least one electromagnet that is often located in the center of motor. This electromagnet is responsible to keep the motor running by changing the polarity. A schematic view of one simple DC motor is shown in Figure 6-1.

In this dissertation, the DC motor design is considered based on the motor weight. The optimal design for DC motor is the one with lower weight. Considering the motor design factors which affect the motor weight, the manufacturer needs to share some information with suppliers and at the same time, protect some important information to prevent reverse engineering in producing low weight DC motor which is the competitive advantage point for companies. In order to design a low weight DC motor, we consider P_8 (Depth of slots) and P_{10} (Height of field windings) as protected parameters which directly affect the weight of motor and investigate different information leakage situations by considering different shared parameters.

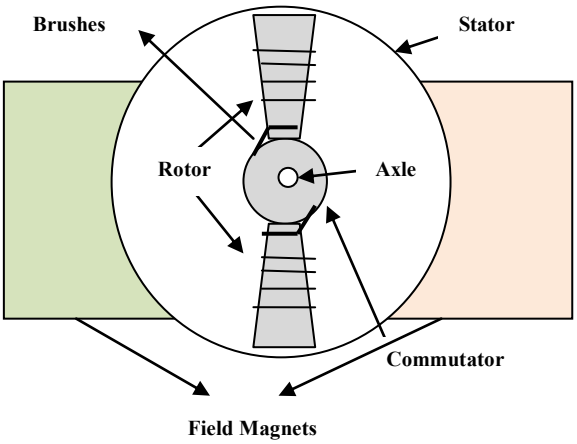


Figure 6-1: Schematic view of one simple DC motor (Seale 2003)[1]

In order to design this DC motor, 25 design functions are considered which affect the DC motor weight, and they involve 26 parameters. In this dissertation the optimal design for DC motor is the one with the lowest weight.

The definitions of the DC motor parameters are given in the Appendix B. Figure 6-2 represents the FP matrix that captures the dependency between these functions (represented in rows) and parameters (represented in columns). In this study, we want to protect the parameters which are directly related to motor weight, i.e. parameters P_8 and P_{10}. Thus, these two parameters are identified as protected parameters. These protected parameters are directly related to $f_1, f_7, f_{12}, f_{13}, f_{16}, f_{17}, f_{22}$ and f_{25} that are then identified as protected functions. For better visuals, the letters f and p are omitted in the labels of the FP matrix in Figure 6-2 and the labels of protected parameters and functions are shaded. In collaborative design, we need to disclose some parametric information to the suppliers.

[1]The idea of this figure is taken from: http://www.solarbotics.net, Seale,2003

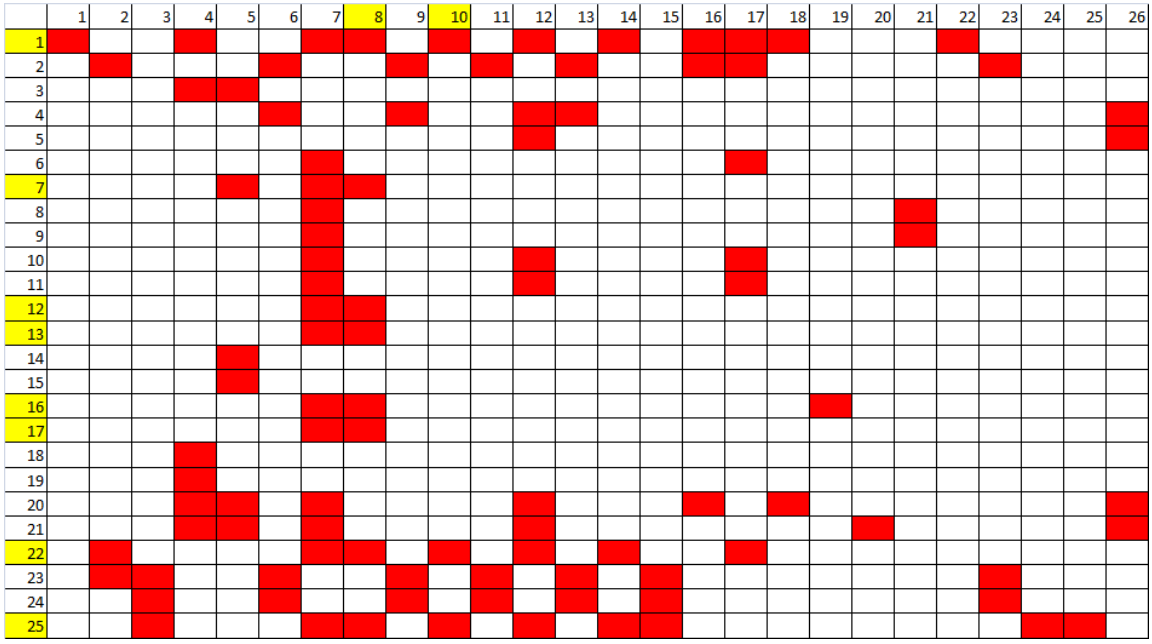


Figure 6-2: FP matrix of the DC motor highlighted with protected entities

6.1.1. How information leakage takes place?

Considering the dependency matrix in coupling analysis section, each non-zero dependency occurs in the matrix, may lead to information leakage. In order to protect the information of one parameter, we need to control the information of that parameter and all functions and parameters which have dependency with the mentioned parameter. For example, in FP-matrix of Figure 6-2, in order to protect the information of parameter P_{10}, we need to control f_{17}, f_{22} and f_{25} and all parameters that exist in these functions. Because, the information of P_{10} can leak via these functions and parameters. This kind of dependency between design parameters and functions (i.e. inferences) is the most important reason of information leakage.

6.1.2. Leakage situation without grouping

As mentioned before, p_8 and p_{10} are protected parameters. By considering p_3, p_4, p_7 and p_5 as the shared parameters, we can use the dependency path to track the information leakage situations for each protected parameters. For example, by considering p_7 as the

starting point of this dependency path, we can easily reach the information of protected parameters p_8 and p_{10} via f_{22}. The notation $p_7 \rightarrow f_{22} \rightarrow P_{8,}\ P_{10,}$ is to represent such a dependency path for each of the protected parameters. The dependency paths for all cases are presented below:

$P_7 \rightarrow f_{22} \rightarrow P_{8,}\ P_{10}$

$P_3 \rightarrow f_{25} \rightarrow P_{8,}\ P_{10}$

$P_4 \rightarrow f_1 \rightarrow P_{8,}\ P_{10}$

$P_5 \rightarrow f_{20} \rightarrow P_{12} \rightarrow f_1 \rightarrow P_{10,}\ \ P_5 \rightarrow f_7 \rightarrow P_8$

Selecting the appropriate set of shared parameters from FP-matrix has an important role in controlling of information leakage. By considering different shared parameters, different risks of information leakage are yielded. For each selected shared parameters, we represented the dependency path as above. The lengths of the above dependency paths roughly correspond to probability of information leakage. The shorter path is the better option to be selected as shared parameter. But these dependency paths are not exact criteria to select the appropriate set of parameters which can be shared without increasing the risk of information leakage for protected parameters. On the other hand, defining the set of parameters that should be protected in order to decrease the information leakage risk of the main protected parameters is not possible for two reasons. Firstly, by increasing the number of parameters, considering different dependency paths for all parameters in order to specify the most secured parameters for sharing, will be impossible. Secondly, when we share some parameters, there are no criteria to compare the risks of information leakage of the paths with the same length. For example, by

considering P_5 as a shared parameter, the information of P_8 can leak via f_7 but the information of p_{10} cannot be determined easily. So the decision for selecting the shared parameters cannot be made appropriately. In this case, the risk of information leakage for all protected parameters may not be considered simultaneously. To solve this problem, we need to share one group of parameters that cause less information leakage of protected parameters compared to the other group of parameters. This grouping helps us to avoid testing the risk of information leakage one by one, (the risk of information leakage for each protected parameters that occur by each share parameters). We can consider the total affect of shared parameters on the protected parameters by grouping them. As shown above, one shared parameter may cause leaking information of one protected parameter easily; on the other hand, it can act as a good shared parameter which has no effect on the information leakage of the other protected parameters (P_5).

So selecting one set of shared parameters that are lower risk parameters for protected parameters, seems difficult. In order to solve this problem, in the context of parametric design, this thesis proposed a clustering method defined in Chapter 4 that can group and isolate the IP-sensitive parameters in few modules. As the modules with IP-sensitive parameters indicate the core information for protection, the information inference can be deliberately managed by controlling the interactions among various modules. Furthermore, the modular structure is used to analyze the risk of information leakage if different modules of parameters are assigned to the suppliers for design and development activities.

6.1.3. Leakage situation with grouping

Clustering and grouping lead to the decomposition and allocation of product design parameters that have an important effect on controlling the risk of inferences and mitigating the risk of information leakage (Zhang, et al. 2011a). Grouping gives us the chance of selecting one set of shared parameters with minimum risk of information leakage of protected parameters by controlling the inferences (interactions). In the modularization process, the parameters that share more common functions tend to be grouped in the same module. In such a way, the parameters of the same module can be easily derived from each other due to the common functions. Thus, the preferable situation in secure collaboration is that the protected entities are grouped in one or few modules (denoted as protected modules). Then, the protected modules can be isolated from other modules via the control of the interactions. That is, if the shared parameters are only found in the non-protected modules, the parameters in the protected modules can be protected by controlling the relevant interactions. The three specific phases of grouping we used in this thesis are described briefly as follows.

- Coupling analysis: the couplings between any two parameters, between any two functions and between a parameter and a function are computed using Formulations (2), (3) and (4), respectively. Then, the target coefficients are applied using Formulation (7). The resulting coupling values are recorded in a concatenated coupling matrix as the analysis output.
- Sorting analysis: the concatenated coupling matrix is processed via the sorting algorithm discussed in Chapter 4. After executing the algorithm, the rows and columns of the original matrix are re-ordered to bring the high-coupled entities close

to each other. The diagonal matrix is obtained accordingly, and it is shown in Figure 6-3.

- Partitioning analysis: it is intended to construct three modules for the diagonal matrix. Based on the tree-based approach discussed in Chapter 4, two partition points are identified, and they are located in Figure 6-3. With these partition points, the corresponding modular structure is shown in Figure 6-3 as the result of the three-phase clustering method.

6.1.4. Leakage situation with target grouping

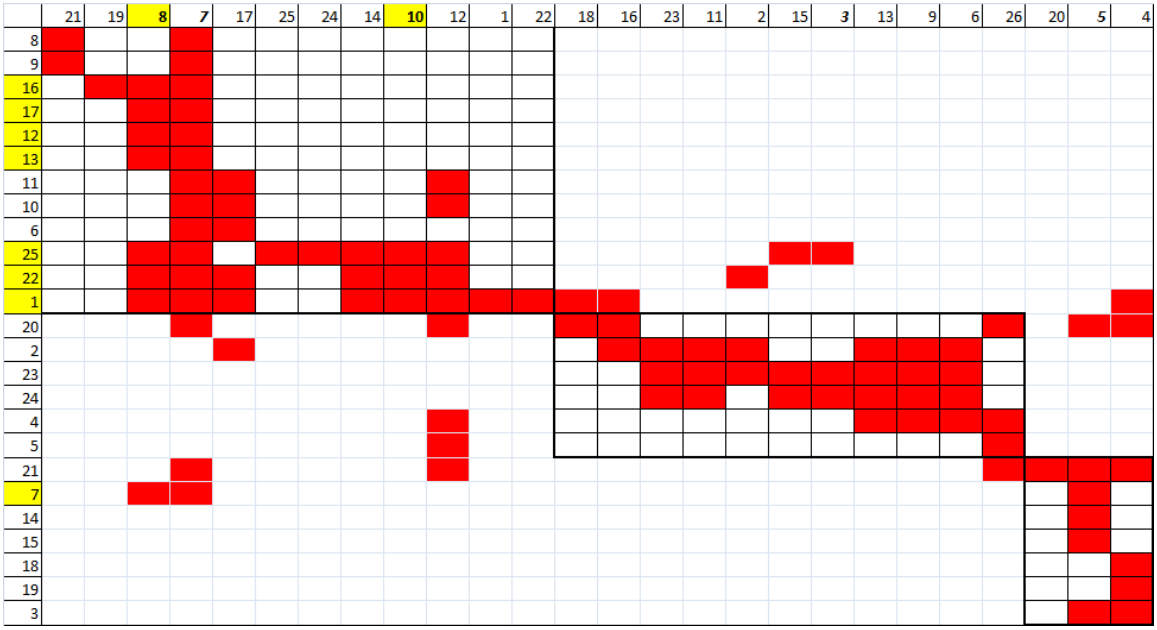


Figure 6-3: Modular structure of the DC motor FP matrix

By applying the target weighting factors, the modular structure in Figure 6-3 represents that the protected entities are grouped relatively close to each other. Particularly, Module 1 almost contains most of the protected entities except f_7 which is contained in Module 3. By looking at this matrix and without using any formulation, we can suggest that the

parameters contained in Module 2 can be shared with suppliers if we control the interaction entities between modules. Because, as we can see, there are no protected entities in Module 2. To further examine the information leakage issue, and in order to verify the above suggestion, three cases are set up based on the choice of shared parameters in different modules, and they are discussed as follows.

- Case 1: share p_7 (Rotor diameter) that is contained in Module 1.
- Case 2: share p_3 (Cross sectional area of field wire) that is contained in Module 2.
- Case 3: share p_5 (Maximum flux density) that is contained in Module 3.

In order to estimate the risk of information leakage related to these three cases, we use Formulation Types 1&2 as defined in chapter 5. Tables 6-1 and 6-2 have summarized the resulting estimates of the information leakage for three cases regarding both Formulations, respectively.

	Protected Parameter	P-labeled	P-related	Shared Parameter	S-labeled	S-related	Leakage Estimate
Case 1	P_8, P_{10}	Module 1	Modules 2&3	P_7	Module 1	Modules 2&3	0.91
Case 2	P_8, P_{10}	Module 1	Modules 2&3	p_3	Module 2	Modules 1&3	0.59
Case 3	P_8, P_{10}	Module 1	Modules 2&3	p_5	Module 3	Modules 1&2	0.55

Table 6-1: Case setup for demonstrating the risk estimation of information leakage in DC motors (Formulation type 1)

	Protected Parameter	P-labeled	P-related	Shared Parameter	S-labeled	S-related	Leakage Estimate
Case 1	P_8, P_{10}	Module 1	Modules 2&3	P_7	Module 1	Modules 2&3	0.86
Case 2	P_8, P_{10}	Module 1	Modules 2&3	p_3	Module 2	Modules 1&3	0.76
Case 3	P_8, P_{10}	Module 1	Modules 2&3	p_5	Module 3	Modules 1&2	0.76

Table 6-2: Case setup for demonstrating the risk estimation of information leakage in DC motors (Formulation type 2)

$M_1 = 42$; $M_2 = 30$; $M_3 = 10$

Case 1:

$$L = \frac{0+42+0.7(15+0)+0.7(15+0)+0.5(3+40)}{42+15+0.7(40+18)} = 0.86$$

Case 2:

$$L = \frac{10+0+0.7(5+42)+0.7(3+30)+0.5(18+10)}{42+15+0.7(40+18)} = 0.76$$

Case 3:

$$L = \frac{0+5+0.7(10+42)+0.7(3+10)+0.5(18+30)}{42+15+0.7(40+18)} = 0.76$$

Tables 6-1 and 6-2 have summarized the case setup with P/S-labeled and P/S-related modules, as well as the values of leakage estimates according to Formulations (9) & (11). As we can see, despite Module 2 contains no protected entities, it relatively represents medium risk of being shared with suppliers, due to the large amount of interactions between this Module and P-labeled Module (Module 1). Since most of the protected entities are contained in Module 1, the risk of information leakage risk should be lower

with the shared parameters located farther from Module 1. On the other hand, Module 3, which is the furthest Module from the P-labeled module, contains one protected entity. These two different situations of Module 3, lead to the range of low to medium risk of information leakage when this Module contains the shared parameters. This is basically observed in Tables 6-1 and 6-2. Particularly, as Case 1 shares the parameter that is directly related to the principal parameters of weight of DC motor (i.e. P_7) in Module 1, this case leads to the highest risk estimate. In contrast, Case 2 leads to medium risk of information leakage. Because, Module 2 contains the shared parameters close to the Module 1. On the other hand, Module 2 contains some parameters which affect the DC motor weight (as P_3) which cause large number of interactions between this module and Module 1. Case 3 which contains the protected entity, has its shared parameters in the module far from Module 1. So it demonstrates the risk in the range of low to medium. This Case also shares the parameters that are quite unrelated to the principal parameters of weight of DC motor. As a result, we reject the above claim about sharing the parameters of Module 2. Then we suggest sharing the parameters of Module 3, with lower risk compared to the other cases.

In order to verify the estimation of risk values in the example, we use the dependency path which was introduced earlier in this chapter. The dependency paths for all cases are shown below:

- Case 1: $P_7 \rightarrow f_{22} \rightarrow P_{8,} P_{10}$
- Case 2: $P_3 \rightarrow f_{25} \rightarrow P_{8,} P_{10}$
- Case 3: $P_5 \rightarrow f_{20} \rightarrow P_{12} \rightarrow f_1 \rightarrow P_{10,}$ $P_5 \rightarrow f_7 \rightarrow P_8$

The lengths of the above dependency paths roughly correspond to the ranking of the risk estimates in Tables 6-1and 6-2. As we discussed before, Case 1 has the highest value of leakage estimate and Case 2 relatively represents high value of leakage estimate. Accordingly, the dependency paths for these two cases are shorter. For Case 3, the above tables represent the risk value in the range of low to medium. If we consider P_{10} As the protected parameter, the dependency path for this case is longer than the other cases and for $P_8$the dependency path is shorter. The dependency paths for Case 3, corresponds to the risk of estimate in the range of low to medium. The proposed module-based leakage measures do not capture this level of fidelity, and further research is required for a more detailed measure. Nevertheless, the module-based approach allows us to approximately identify which subsets of parameters can be potentially shared with suppliers, and it is the original contribution of this paper.

6.1.5. Leakage situation with non-target grouping

By applying three phases of clustering method in Chapter 4 and without applying target coefficient, we have the new modular structure matrix as Figure 6-4. Considering modular structure of Figure 6-4, the protected parameters exist in two different modules (Module 1 and Module 3) that are far from each other and the protected functions are distributed among all the modules. By keep tracking the parameters which are related to the weight of DC motor, we can see these parameters are included in all modules. For instance, Module 1 contains P_3 (Cross sectional area of field wire), Module 2 contains P_{12} (Rotor axial length) and Module 3 contains P_7 (Rotor diameter).

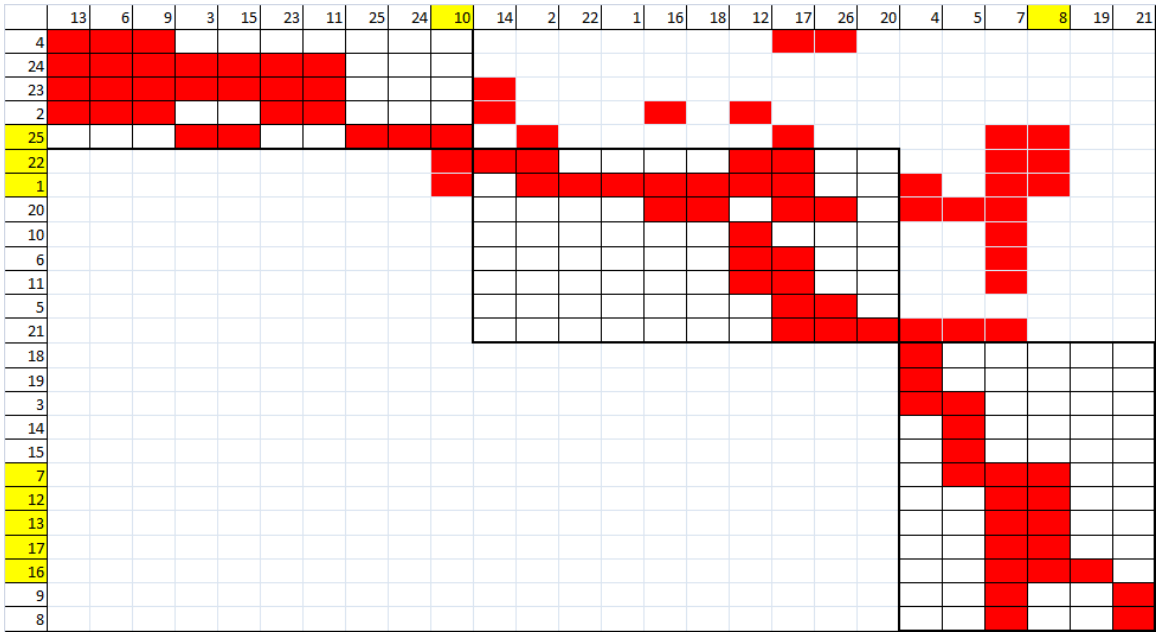


Figure 6-4: Modular structure of the FP matrix without using target coefficient

6.1.6. The effect of target grouping

Comparing modular structure matrix in Figures 6-3 and 6-4, the tracking of information leakage in Figure 6-4 is not quite possible. Firstly, the protected entities are distributed among all modules. So, all modules are P-labeled or P-related and as mentioned above, all these modules contain parameters that have direct affect on weight of DC motor. Secondly, when there is no possibility to consider all protected parameters in the same module, selecting one set of shared parameter that impose the lower risk of information leakage for all protected parameters is quite difficult. In this situation, tracking the interaction of various modules and controlling the risk of information leakage seems more difficult. Comparatively, since the modular structure in Figure 6-3 can roughly group the protected entities, it is easier to identify and control the interactions around the protected entities to prevent Information leakage.

Conventionally, defining different protected parameters, lead to different forms of modular structure matrix. Regarding the idea of using target coefficient, each of these different modular structure matrix attempts to keep the protected parameters close to each other and make them group. However the design functions and parameters and the modular structure of a design is often considered stable according to its functional and physical decomposition and considering different protected parameters have no affect on them. The feature of forming different modular structures based on emerging protected parameters is relatively new in this field of research (Rojas-Arciniegas and Kim 2012).

6.2. Relief valve

The relief valve system is adapted from Kannapan and Marshek (1992), which schematic is shown in Figure 6-5. To design this valve system, 29 functions concerning geometry and fluid properties are identified, and they involve 49 parameters. The definitions of the relief valve parameters are given in the Appendix C. Figure 6-6 shows the FP matrix that captures the dependency between these functions (represented in rows) and parameters (represented in columns). In this study, we want to protect helical spring specifications that correspond to the parameters p_{23}, p_{32} and p_{37}. Thus, these three parameters are identified as protected parameters. These protected parameters are directly related to $f_{15,}$ $f_{16,}f_{21,}f_{24,}f_{25}$ and f_{28} that are then identified as protected functions. For better visuals, the letters f and p are omitted in the labels of the FP matrix in Figure 6-6, and the labels of protected parameters and functions are shaded. In collaborative design, we need to disclose some parametric information to the suppliers. The inquiry in this context is which parameters should be shared with the suppliers in order to minimize the risk of leaking the information of protected parameters.

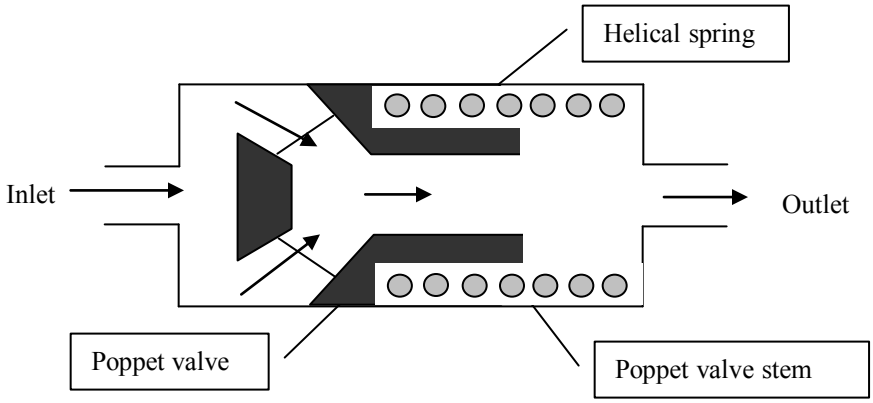


Figure 6-5: schematic of a relief valve system

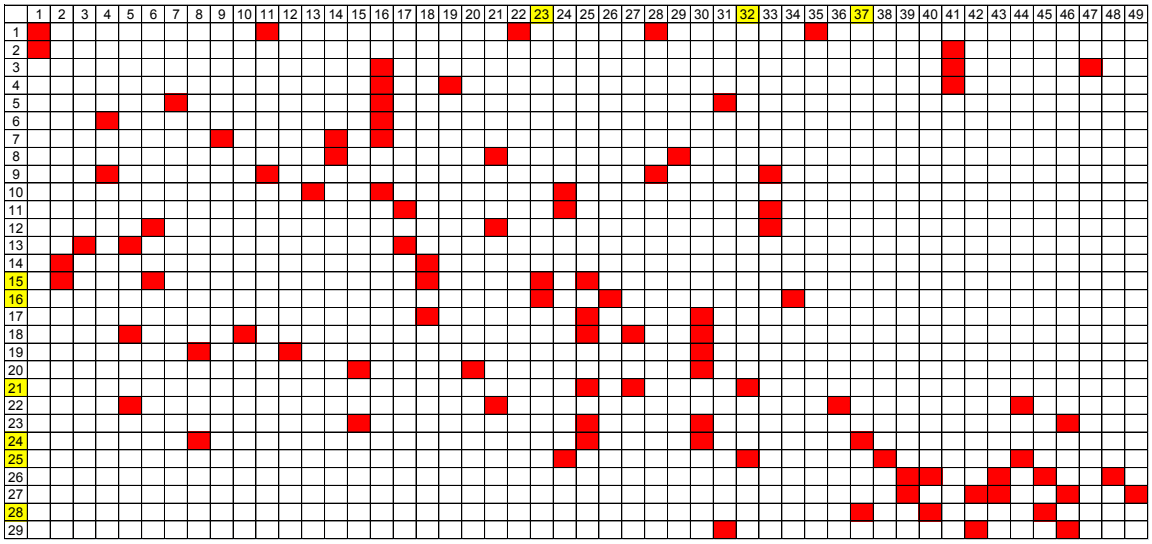


Figure 6-6: FP matrix of the relief valve system highlighted with protected entities

6.2.1. Demonstration

The modular structure in Figure 6-7 shows that the protected entities are grouped relatively close to each other. Particularly, Module 3 contains most of the protected entities except f_{25}, which is contained in Module 2. By properly controlling the interactions between modules, we can roughly suggest that the parameters contained in

Modules 1 & 4 (about valve external geometry and poppet valve specifications) can be shared with suppliers. To further examine the information leakage issue, four cases are set up based on the choice of shared parameters in different modules, and they are discussed as follows.

- Case 1: share p_9 (seal thickness) that is contained in Module 1.
- Case 2: share p_{38} (clash allowance ratio) that is contained in Module 2.
- Case 3: share p_{25} (helical spring wire diameter) that is contained in Module 3.
- Case 4: share p_{42} (valve cylinder thickness) that is contained in Module 4.

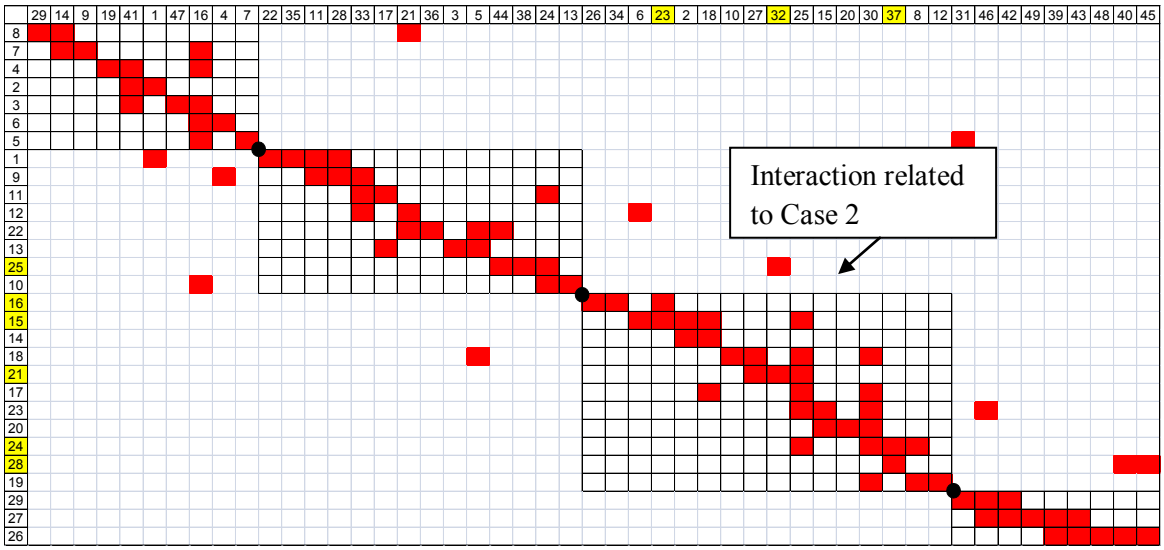


Figure 6-7: Modular structure of the relief valve FP matrix

In order to estimate the risk of information leakage related to these four cases, we use Formulation Types 1&2 as defined in Chapter 5. Tables 6-3 and 6-4 have summarized the resulting estimates of the information leakage for four cases regarding both Formulations.

	Protected Parameter	P-labeled	P-related	Shared Parameter	S-labeled	S-related	Leakage Estimate
Case 1	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_9	Module 1	Modules 2 & 4	0.17
Case 2	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_{38}	Module 2	Modules 1 & 3	0.59
Case 3	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_{25}	Module 3	Modules 2 & 4	0.89
Case 4	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_{42}	Module 4	Modules 1 & 3	0.41

Table 6-3: Case setup for leakage risk analysis demonstrating the risk estimation of information leakage in relief valve (Formulation type 1)

	Protected Parameter	P-labeled	P-related	Shared Parameter	S-labeled	S-related	Leakage Estimate
Case 1	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_9	Module 1	Modules 2 & 4	0.35
Case 2	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_{38}	Module 2	Modules 1 & 3	0.66
Case 3	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_{25}	Module 3	Modules 2 & 4	0.82
Case 4	p_{23}, p_{35}, p_{37}	Module 3	Modules 2 & 4	p_{42}	Module 4	Modules 1 & 3	0.61

Table 6-4: Case setup for demonstrating the risk estimation of information leakage in relief valve (Formulation type 2)

$M_1 = 17$; $M_2 = 24$; $M_3 = 34$; $M_4 = 13$

Case 1:

$$L = \frac{0+0+0.7(6+0)+0.7(5+0)+0.5(0+24+13)}{34+6+0.7(24+13+11)} = 0.35$$

Case 2:

$$L = \frac{3+0+0.7(0+34)+0.7(0+24)+0.5(0+11)}{34+6+0.7(24+13+11)} = 0.66$$

Case 3:

$$L = \frac{0+34+0.7(6+0)+0.7(6+0)+0.5(0+24+13)}{34+6+0.7(24+13+11)} = 0.82$$

Case 4:

$$L = \frac{3+0+0.7(6+0)+0.7(6+0)+0.5(0+24+13)}{34+6+0.7(24+13+11)} = 0.61$$

Tables 6-3 and 6-4 have summarized the case setup with P/S-labeled and P/S-related modules, as well as the values of leakage estimates according to Formulations (9) & (11). As most of the protected entities are contained in Module 3, the leakage risk should be lower with the shared parameters located *farther* from Module 3. This is basically observed in Table 6-4. Particularly, as Case 3 shares the parameter that is directed to the helical spring (i.e., p_{25}) in Module 3, this case leads to the highest risk estimate. In contrast, Cases 2 and 4 have their shared parameters in the modules right next to Module 3, leading to the medium risk estimates. Case 1 has demonstrated the lower risk as it shares the parameter that is quite unrelated to the helical spring.

Notably, the proposed leakage risk estimate in Formulation (9) is based on the modular solution. To verify the risk estimate values in the example, we use the dependency paths introduced earlier in this chapter to trace the functional dependency from a shared parameter to one of protected parameters without considering the modularity information. The dependency paths of all cases are presented below.

- Case 1: $p_9 \rightarrow f_7 \rightarrow p_{14} \rightarrow f_8 \rightarrow p_{21} \rightarrow f_{12} \rightarrow p_6 \rightarrow f_{15} \rightarrow p_{23}$
- Case 2: $p_{38} \rightarrow f_{25} \rightarrow p_{32}$
- Case 3: $p_{25} \rightarrow f_{21} \rightarrow p_{32}$
- Case 4: $p_{42} \rightarrow f_{29} \rightarrow p_{46} \rightarrow f_{23} \rightarrow p_{25} \rightarrow f_{21} \rightarrow p_{32}$

The lengths of the above dependency paths roughly correspond to the ranking of the risk estimates in Table 6-4. For instance, Cases 2 and 3 have the higher values of leakage estimate, and their dependency paths are shorter than those of Cases 1 and 4. Yet, the proposed module-based leakage measure is only an approximation, and some detailed distinction may not be captured. For instance, sharing the parameter p_{38} in Case 2 directly leads to the information of one protected parameter (i.e., p_{32}) via the interaction at f_{25} and p_{32} (pointed in Figure 6-7). This specific interaction causes the same length of dependency paths between Cases 2 and 3. The proposed module-based leakage measures do not capture this level of fidelity, and further research is required for a more detailed measure. Nevertheless, the module-based approach allows us to approximately identify which subsets of parameters can be potentially shared with suppliers, and it is the original contribution of this paper.

6.2.2. Effects of target coefficient

To examine the effect of the target coefficient, we re-run the three-phase clustering method without applying the target coefficient, and the corresponding modular structure is shown in Figure 6-8. Compared with Figure 6-7, this modular structure has the protected entities scattered in different modules (i.e., Modules 1, 2 and 4). In the context of sharing the parameters with suppliers, this modular structure implies not to share

parameters in Modules 1, 2 and 4. While finding shared parameters in Module 3, the interactions with this module need to be carefully handled to prevent information leakage.

Considerably, the modular structure in Figure 6-8 is not quite useful for handling the information leakage issue as compared to the modular structure in Figure 6-7 for two reasons. Firstly, we tend to select the parameters from the modules that are not P-labeled and P-related. However, the modular structure in Figure 6-8 does not have such a module, increasing the difficulty of selecting shared parameters. Secondly, since the protected entities are scattered in Figure 6-8, it becomes more difficult to keep track of the interactions of various modules and control the risk of information leakage. Comparatively, since the modular structure in Figure 6-7 can roughly group the protected entities, it is easier to identify and control the interactions around the protected entities to prevent information leakage.

Conventionally, the modular structure of a design is often considered stable according to its functional and physical decomposition. The idea of using the target coefficient essentially implies that the modular structures can be varied according to the identification of protected parameters. That is, if different protected parameters are defined, the resulting modular structure should be specifically formed in attempt to group and isolate these protected parameters.

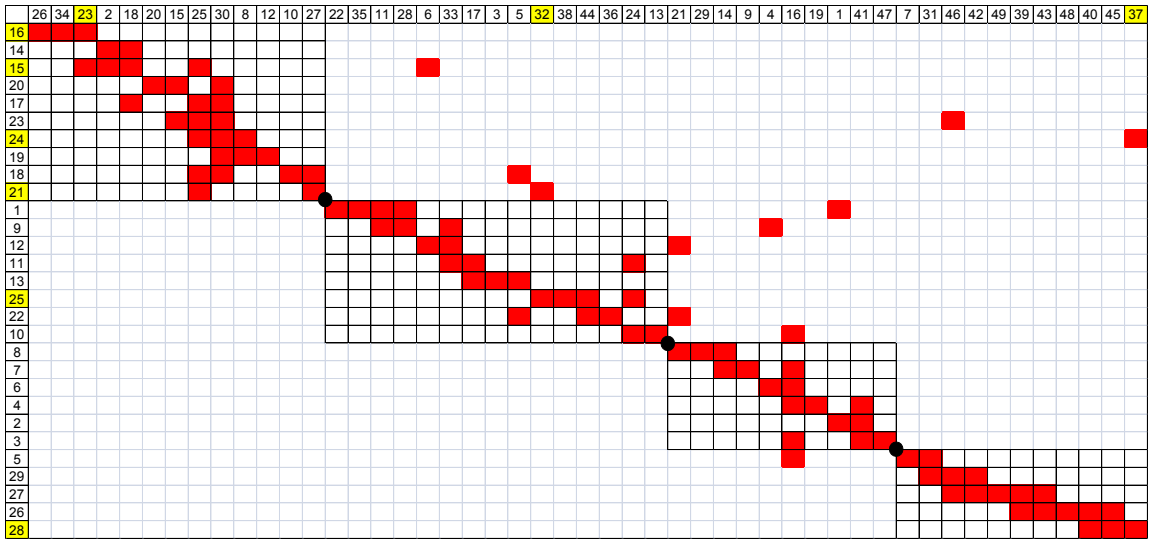


Figure 6-8: Modular structure of the FP matrix without using target coefficient

6.3. General discussions

By applying the modularization method in obtained FP matrices from DC motor and relief valve system and based on two formulations regarding these FP matrices, we can summarize the results as follows. It is necessary to mention that, the results of two different case studies (i.e. Dc motor and relief valve system) are consistent with each other.

- The effect of target coefficient: After using target coefficient in grouping the entities, all the protected parameters have been grouped in one module and close to each other in both cases. However, without using target coefficient concept, the protected entities can possibly be distributed among all modules. So, all modules are P-labeled or P-related. Consequently, when there is no possibility to consider all protected parameters in the same module, selecting one set of shared parameter

that impose the lower risk of information leakage for all protected parameters is quite difficult

- Risk estimation based on two types of formulations: By considering the results of risk estimation for different cases and based on two formulations for both case studies, we can conclude that selecting the set of shared parameters from P-labeled modules (i.e. modules which include protected parameters) yield the highest risk of information leakage for protected parameters. We can also claim that the risk of information leakage will be high when the selected shared parameters are from modules that have so many interactions with the p-labeled modules. Unlike two previous situations, selecting one set of shared parameters from modules which are far from the p-labeled modules and have the least interactions with the p-labeled modules, yield the minimum risk of information leakage.

Chapter 7: Conclusions

7.1. Summary

Collaborative design is one common practice for companies to reduce the cost and time of designing a product. However, this collaboration may result in leaking the IP-sensitive information of original manufacturer to the other parties in collaborative design. In order to mitigate the risk of information leakage due to the nature of collaborative design, some methods and approaches have been suggested.

In this thesis, we focused on modularity concept and clustering in order to group and isolate the IP-sensitive information for their protection. So, we developed a matrix-based modularization method. However, there is always the risk of leaking the information of IP-sensitive parameters via the interaction parameters. This modularization helps the original manufactures to track leaking of protected elements via interactions. MATLAB codes have been utilized in this thesis order to implement the algorithm of modularization.

In order to evaluate the risk of information leakage for protected information, caused by sharing different set of parameters, we introduced two formulations based on the size of modules and the intensity of interactions between and within the modules. By using these formulations, we can facilitate the selection of one set of secure parameters for sharing with the other parties in collaborative design. In order to justify the proposed model and formulations, we applied them for relief valve system and DC motor. As a result, this modularization led in isolating the protected parameters in one module. Moreover, it can be concluded that selecting the set of parameters for sharing from the modules which are far from the protected module and have the least interaction with the protected module

imposes the lowest risk for original manufacture. All the results verify our research works in this thesis.

7.2. Contributions

While some researchers have focused on the information protection methods in collaborative design, only few of them have considered the modularity concept as a method for information security. Compared to previous works, the contribution of this thesis is in developing the modularization method for information protection considering the inferences, based on hierarchical clustering steps. In this process, using target coefficient concept as a tool for keeping the protected information close to each other in FP matrix was original.

In order to evaluate the risk of information leakage, two new formulations based on two different criteria were introduced. The first formulation was based on the size of modules and the second one was produced in order to clarify the role of inferences between modules in information leakage.

Since the selection of parameters for sharing in collaborative design can affect the risk of information leakage, so, another contribution of this thesis is in identifying the subsets of parameters based on the results of the risk of information leakage which can be potentially shared with suppliers in order to minimize the information leakage in collaborative design.

7.3. Future works

In future, the matrix-based modularization approach can be applied for optimizing supplier selection, different modules can be assigned to different suppliers, and also the optimal number of suppliers can be specified.

We can also further elaborate the risk estimation formulation by considering different types of interactions between different parameters.

Moreover, in order to improve risk estimation formulation, in future, we can estimate the risk of sharing one set of parameter in leaking all protected parameters simultaneously. In other words, we can control the information leakage for all protected parameters, even from different modules, at the same time.

References

Abu, AO., 2008, "Comparision between Data Clustering Algorithms", The International Arab Journal of Information Technology, Vol. 5, No.3.

Alex, A., Freitas, Andr´e, C. and Ponce Leon, F., 2009, "A Survey of Evolutionary Algorithms for Clustering", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 39, No. 2.

Anand, K. and Goyal, M., 2009, "Strategic Information Management under Leakage in a Supply Chain," Management Science, Vol. 55, pp. 438-452.

Arai, K. and Barakbah, A.R., 2007, "Hierarchical K-means: An Algorithm for Centroids Initialization for K-Means", Reports of the Faculty of Science and Engineering, Saga University, Vol. 36, No.1

Beauvais, K., 2003, "The Simple DC Motor", Research Experience for Teachers Center for Materials Science and Engineering Massachusetts Institute of Technology

Benedens, O., 1999, "Geometry-Based Watermarking of 3D Models," IEEE Computer Graphics and Applications, Vol. 19, pp. 46-55.

Browning, T.R., 2001, "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," IEEE Transaction on Engineering Management, Vol. 48, pp. 292-306.

Carminati, B., Ferrari, E. and Guglielmi, M., 2011, "Secure Information Sharing on Support of Emergency Management", IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

Cera, C., Kim, T., Han, J. and Regli, W., 2003, "Role-Based Viewing Envelopes for Information Protection in Collaborative Modeling," Computer-Aided Design, Vol. 36, pp. 873-886.

Cera, C., Braude, I., Kim, T., Han, J. and Regli, W., 2006, "Hierarchical Role-Based Viewing for Multilevel Information Security in Collaborative CAD," ASME Journal of Computing and Information Science in Engineering, Vol. 6, pp. 2-10.

Chen, F., 2003, "Information Sharing and Supply Chain Coordination," S. Graves, T. de KoK, eds., Handbooks in Operations Research and Management Science: Supply Chain Management, Vol. 11, North Holland, Amsterdam, pp. 341-421.

Chen, G. and Huang, E., 2007, "A Systematic Approach for Supply Chain Improvement Using Design Structure Matrix", Springer Science + Business Media, LLC

Chen, J., 2005, "Comparison Of Clustering Algorithms and Its Application To Document Clustering", A Dissertation Presented To The Faculty Of Princeton University In Candidacy For The Degree Of Doctor Of Philosophy

Chen, T., Chen, Y., Chu, H. and Wang, C., 2007, "Development of an Access Control Model, System Architecture and Approaches for Resource Sharing in Virtual Enterprise," Computers in Industry, Vol. 58, pp. 57-73.

Chen, T., Chen Y. and Chu, H., 2008, "Developing a Trust Evaluation Method between Co-workers in Virtual Project Team for Enabling Resource Sharing and Collaboration," Computers in Industry, Vol. 59, pp. 565-579.

Chu, C., Wu, P. and Hsu, Y., 2009, "Multi-Agent Collaborative 3D Design with Geometric Model at Different Levels of Detail," Robotics and Computer-Integrated Manufacturing, Vol. 25, pp. 334-347.

Dain, M., Calvi, R. and cheriti, S., 2010, "Measuring Supplier Performance in Collaborative Design: Proposition of a Framework", R&D Management, Vol. 41, Issue 1, pp. 61-79.

Danilovic, M. and Browning, T.R., 2007, "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices", International Journal of Project Management, Vol. 25, pp. 300-314.

Deitos, R. and Kerschbaum, F., 2009, "Improving Practical Performance on Secure and Private Collaborative Linear Programming", 20th International Workshop on Database and Expert Systems Application, IEEE Computer Society

Deng, X., Huet, G., Tan, S. and Fortin, C., 2012, "Product decomposition using design structure matrix for intellectual property protection in supply chain outsourcing", Computers in industry, Vol.63, pp. 632–641

Dye, R. and Sridhar, S., 2003, "Investment Implications of Information Acquisition and Leakage", Management Science, Vol.49, No.6, pp.767-783

Everitt, B., Landau, S., Leese, M. and Stahl, D., 2011, Cluster Analysis, 5th Edition, Wiley, West Sussex.

Fernandez, C., 1998, "Integration Analysis of Product Architecture to Support Effective Team Co-location," Master of Science thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Harston, S.P. and Mattson, C.A., 2010, "Metrics for Evaluating the Barrier and Time to Reverse Engineer a Product," ASME Journal of Mechanical Design, Vol. 132, DOI: 041009.

Hoecht, A. and Trott, P., 2006, "Outsourcing, Information Leakage and the Risk of Losing Technology-based Competencies," European Business Review, Vol. 18, pp. 395-412.

Kannapan, S.M. and Marshek, K.M., 1992, "An Approach to Parametric Machine Design and Negotiation in Concurrent Engineering," in Concurrent Engineering: Automation, Tools and Techniques, A. Kusiak (Ed.), pp. 509-533, John-Wiley, New York.

Kim, H, 2011, "Real Time Control of Servo Motor" ,A project report submitted to SIM University in partial fulfillment of the requirements for the degree of Bachelor (Hons) of Engineering in Electronics

Kim, I., Lee, J., Mun, D., Jun, H., Hwang, J., Kim, JT. and Han, S., 2012, "Securing Design Checking Service for the Regulation-Based Product Design", Computers in Industry, Vol.63, pp. 586–596

Kim, T., Cera, C., Regli, W., Choo, H. and Han, J., 2006, "Multi-Level Modeling and Access Control for Data Sharing in Collaborative Design," Advanced Engineering Informatics, Vol. 20, pp. 47-57.

Lee, H.L., Padmanabhan, V. and Whang, S., 1997, "Information Distortion in a Supply Chain: The Bullwhip Effect," Management Science, Vol. 43, pp. 546-558.

Lee, H.L., So, K.C. and Tang, C.S., 2000, "The Value of Information Sharing in a Two-Level Supply Chain," Management Science, Vol. 46, pp. 626-643.

Li, J. and Atallah, MJ., 2006, "Secure and Private Collaborative Linear Programming," Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Work sharing

Li, L., 2002, "Information Sharing in a Supply Chain with Horizontal Competition," Management Science, Vol. 48, pp. 1196-1212.

Li, L. and Zhang, H., 2008, "Confidentiality and Information Sharing in Supply Chain Coordination," Management Science, Vol. 54, pp. 1467-1481.

Li, S., 2007, "Matrix-based Decomposition Algorithms for Engineering Applications: Survey and Generic Framework", International Journal of Product Development, Vol. 9, pp. 78-110

Li, S., 2009, "A Revised Two-Phase Method for Decomposition of Design Problems", International Conference on Engineering Design

Li, S., 2011, "A Matrix-based Clustering Approach for the Decomposition of Design Problems," Research in Engineering Design, Vol. 22, pp. 263-278.

Lindemann, U., Maurer, M. and Braun, T., 2009, Structural Complexity Management: An Approach for the Field of Product Design, Springer, Berlin.

Ma, Y., Tang, S., Au, C.K. and Chen, J., 2009, "Collaborative Feature-Based Design Via Operations With a Fine-Grain Product Database" Computers in Industry, Vol. 60, pp. 381-391.

Mao, J. and Liu, Sh., 2010, "Application of LOD in Collaborative Design and Information Security", International Conference on Networking and Digital Society

Michelena, N.F. and Papalambros, P.Y., 1995, "Optimal Model-based Decomposition of Powertrain System Design," ASME Journal of Mechanical Design, Vol. 117, pp. 499-505.

Mun, D., Hwang, J. and Han, S., 2009, "Protection of Intellectual Property based on a Skeleton Model in Product Design Collaboration," Computer-Aided Design, Vol. 41, pp. 641-648.

Niakanjam, A., Sharifi, H., Helmi, H. and Rahmani, A., 2010, "A New DSM Clustering Algorithm for Linkage Groups Identification", Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation

Niakanjam, A., Sharifi, H., Helmi, H. and Rahmani, A., 2010, "Enhancing the Efficiency of Genetic Algorithm by Identifying Linkage Groups Using DSM Clustering", IEEE Congress on Evolutionary Computation (CEC)

Otto, K, 1995, "Measurement Methods for Product Evaluation," Research in Engineering Design, Vol. 7, pp. 86-101.

Pimmler, T.U. and Eppinger, S.D., 1994, "Integration Analysis of Product Decompositions," Proceedings of ASME 6th International Conference on Design Theory and Methodology, Minneapolis, Minnesota.

Qiu, Z., Kok, K., Wong, Y. and Fuh, J., 2007, "Role-Based 3D Visualization for Asynchronous PLM Collaboration," Computers in Industry, Vol. 58, pp. 747-755.

Reyer, J., 2000, "Combined Embodiment Design and Control Optimization: Effects of Cross-Disciplinary Coupling", A Dissertation Submitted in Partial Fulfillment of the Requirement for the Degree of Doctor of Philosophy (Mechanical Engineering) in the University of Michigan

Rojas-Arciniegas, A.J. and Kim, H.M., 2012, "Incorporating Security Considerations into Optimal Product Architecture and Component Sharing Decision in Product Family Design," Engineering Optimization, Vol. 44, pp. 55-74.

Romesburg, H.C., 2004, "Cluster Analysis for Researchers", Lulu Press, North Carolina.

Sandhu, R.S. and Samarati, P., 1994, "Access Control: Principles and Practice," IEEE Communications Magazine, Vol. 32, pp. 40-48.

Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E., 1996, "Role-Based Access Control Models," IEEE Computer, Vol. 29, pp. 38-47.

Steward, D. V., 1981, "The Design Structure System: A Method for Managing the Design of Complex Systems," IEEE Trans. Eng. Manage., 28, pp. 71–74.

Suh, N.P., 1990, "The Principle of Design", Oxford University Press, New York.

Sun, L. and Wang, H., 2011, "A Purpose Based Usage Access Control Model for E-Healthcare Services", International Conference on Data and Knowledge Engineering (ICDKE), pp.41-46

Sun, X., Zeng, Y. and Liu, W., 2011, "Formalization of Design Chain Management using Environment-based Design (EBD) Theory," Journal of Intelligent Manufacturing, Online First™, 08 December 2011.

Thomas, B., Raju, G. and Wangmo, S., , 2009, "A Modified Fuzzy C-Means Algorithm for Natural Data Exploration", World Academy of Science, Engineering and Technology,

Wang, Y., Ajoku, P., Brustoloni, J. and Nnaji, B., 2006, "Intellectual Property Protection in Collaborative Design through Lean Information Modeling and Sharing," ASME Journal of Computing and Information Science in Engineering, Vol. 6, pp. 149-159.

Xu, R., 2005, "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, VOL. 16, NO. 3

Yao, D.Q., Yue, X. and Liu, J., 2008, "Vertical Cost Information Sharing in a Supply Chain with Value-Adding Retailers," Omega, Vol. 36, pp. 838-851.

Yin, Y., Qin, Sh. and Holland, R., 2008, "Development of a Project Level Performance Measurement Model for Improving Collaborative Design Team Work", 12th International Conference on Computer Supported Cooperative Work in Design, pp.135-140

Yu, T., Goldberg, D., Sastry, K., Lima, C. and Pelikan, M., 2009, "Dependency Structure Matrix, Genetic Algorithms, and Effective Recombination," Evolutionary Computation, 17(4), pp. 595-626.

Zanetti, D. and Capkun, S., 2008, "Protecting Sensitive Business Information while Sharing Serial–Level Data", 12th Enterprise Distributed Object Computing Conference Workshops, pp.183-191

Zeng, Y. and Lingyu, 2012a, "Secure Collaboration in Design and Supply Chain Management", Editorial, Computers in Industry, Vol. 63, pp. 543-544.

Zeng, Y., Lingyu, W., Deng, X., Cao, X. and Khundker, N., 2012b, "Secure Collaboration in Global Design and Supply Chain Environment: Problem Analysis and Literature Review", Computers in Industry, Vol. 63, pp. 545-556.

Zhang, C. and Li, S., 2006, "Secure Information Sharing in Internet-Based Supply Chain Management Systems", Journal of Computer Information Systems, Vol. 46, pp. 18-24.

Zhang, D., Zeng, Y., Wang, L., Li, H. and Geng, Y., 2011a, "Modeling and Evaluating Information Leakage Caused by Inferences in Supply Chains", Computers in Industry, Vol. 62, pp. 351-363.

Zhang, D., Cao, X., Wang, L. and Zeng, Y., 2011b, "Mitigating the Risk of Information Leakage in a Two-Level Supply Chain through Optimal Supplier Selection," Journal of Intelligent Manufacturing, Online First™, 30 March 2011.

Zhang, S., Shen, W. and Ghenniwa, H., 2004, "A Review of Internet-based Product Information Sharing and Visualization," Computers in Industry, Vol. 54, pp. 1-15.

Zhang, Y., Bai, S. and Guo, Y., 2010, "The Application of Design Structure Matrix in Project Schedule Management", International Conference on E-Business and E-Government

Zheng, J., Huang, Zh., Hu, J., Wei, O. and Liu, L., 2012, "Trust-Based Privacy Authorization Model for Web Service Composition", Software Engineering and Knowledge Engineering: Vol. 2, AISC 115, pp. 307–313

Appendix A

Sorting analysis steps and partitioning analysis

- Step 1: Constructing the tree branches: Considering obtained coupling matrix, select two entities which yield highest value. Then label the leaves of the tree according to the selected entities. Form the branch of the tree by combining the leaves (or branches). The vertical axis of the tree is labeled with the coupling values. The leaves (or branches) are merged to form a new branch at their coupling value.
- Step 2: Updating the coupling matrix: Update the coupling matrix by considering the two selected entities as one unit branch and recalculate the new coupling values through the average distance formulation as follows.

$$r_{(ij)k} = \frac{r_{ik} + r_{jk}}{2} \qquad 1 \le k \le n \quad k \ne i \quad k \ne j$$

(8)

where r_{ik} is the coupling value between the ith and kth entities, and the subscript ij refers to the newly combined branch.

- Step 3: Iteration check. Repeat Step 1 and Step 2 until the coupling matrix cannot be further reduced.

The output of sorting analysis step is the tree (or dendrogram). In our thesis, we use sorting analysis in order to rearrange the sequence of rows and columns in FP matrix. Figure 7-1 has shown the sorted/diagonal matrix of the sample matrix, and Figure 7-2 shows the corresponding tree.

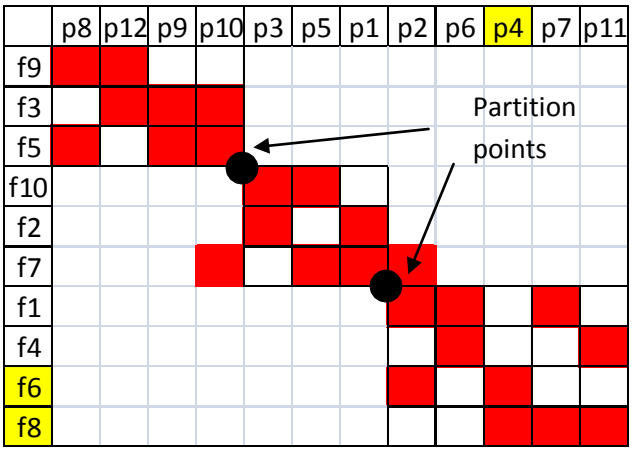


Figure 7-1: Diagonal matrix with partition points

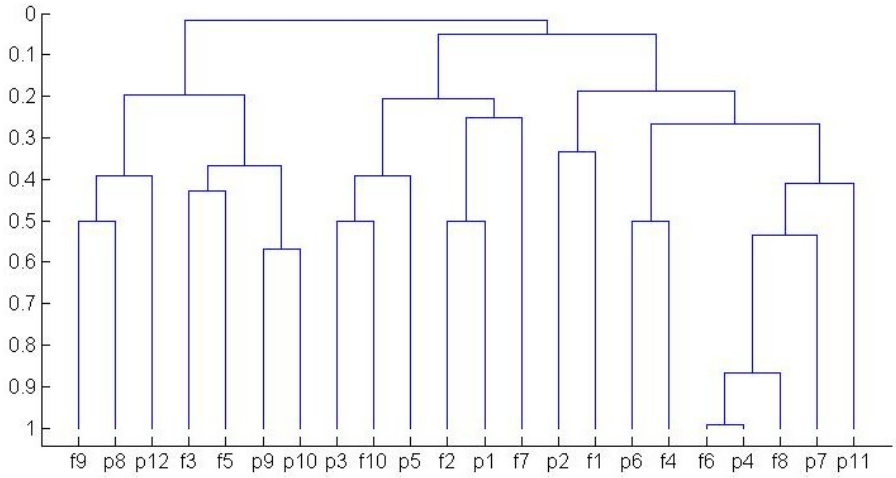


Figure 7-2: The resulting tree of sample FP matrix

In order to illustrate the partition points, consider the diagonal matrix and the tree in Figures 7-1 and 7-2. By looking at the branches of tree in Figure 7-2, we can see the first partition points can be reached by breaking the first branch between p_{10} and p_3. By keeping breaking the other level of branches we reach the second partition points between f_7 and p_2 .Based on the required number of modules and the other criteria, we stop or keep breaking the other branches. Similarly, the second top branch suggests another partition point.

Appendix B

Description of DC motor parameters

Parameters		Functions	Description
p_1	a	f_1	Number of parallel paths
p_2	$A_{wa,}$	f_{22}, f_{25}	Cross sectional area of armature wires
p_3	A_{wf}	$F_2, f_{23}, f_{24}, f_{25}$	Cross sectional area of field wires
p_4	ac	$f_1, f_3, f_{18}, f_{19}, f_{20}, f_{21}$	Specific electrical loading
p_5	B_g	$f_3, f_7, f_{14}, f_{125}, f_{20}, f_{21}$	Maximum flux density
p_6	b_{fc}	f_2, f_4, f_{23}, f_{24}	Depth of field coil
p_7	D	$f_1, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{16}, f_{17}, f_{20}, f_{21}, f_{22}, f_{25}$	Rotor diameter
p_8	d_s	$f_1, f_7, f_{12}, f_{13}, f_{16}, f_{17}, f_{22}, f_{25}$	Depth of slots
p_9	f_{cf}	f_2	Copper space factor
p_{10}	h_f	f_2, f_5, f_{23}, f_{24}	Height of field windings
p_{11}	l_f	f_2, f_{23}, f_{24}	Field current
p_{12}	L	$f_1, f_4, f_5, f_0, f_{11}, f_{20}, f_{21}, f_{22}, f_{25}$	Rotor axial length
p_{13}	L_{mtf}	f_2, f_4, f_{23}, f_{24}	Mean turn length of field coil
p_{14}	L_{wa}	f_1, f_{22}, f_{25}	Length of armature wire
p_{15}	L_{wf}	f_{23}, f_{24}, f_{25}	Length of filed wire
p_{16}	n_d	f_1, f_2, f_{20}	Derivative controller gain
p_{17}	p	$f_1, f_2, f_6, f_{10}, f_{11}, f_{22}$	Number of poles
p_{18}	P_{mind}	f_1, f_{20}	Minimum required power
p_{19}	S	f_{16}	Number of slots or teeth on rotor
p_{20}	$T_{min\ d}$	f_{21}	Minimum required torque
p_{21}	V_d	f_8, f_9	Design voltage
p_{22}	η	f_1	Motor efficiency
p_{23}	ρ	f_2, f_{23}, f_{24}	Resistivity
p_{24}	$\rho_{cu,}$	f_{25}	Densities of copper
p_{25}	ρ_{fe}	f_{25}	Densities of iron
p_{26}	Ψ	f_4, f_5, f_{20}, f_{21}	Pole arc to pole pitch ratio

Appendix C

List of parameters of the relief valve system

Parameter	Description	Parameter	Description
p_1	orifice diameter	p_{26}	allowable helical spring material stress
p_2	Wahl spring rate	p_{27}	number of helical spring coils
p_3	spring rate reduction ratio (for stability)	p_{28}	fluid specific gravity
p_4	pipeline cross-sectional area	p_{29}	cracking pressure
p_5	actual spring rate	p_{30}	mean helical spring diameter
p_6	total force on helical spring	p_{31}	valve hole diameter
p_7	valve hole size ratio	p_{32}	helical spring solid length
p_8	radial clearance between helical spring and enclosure	p_{33}	dynamic fluid force
p_9	seal thickness	p_{34}	helical spring factor of safety
p_{10}	shear modulus of spring material	p_{35}	pressure drop from valve inlet to outlet
p_{11}	fluid flow rate	p_{36}	helical spring installed length
p_{12}	helical spring outer diametral clearance ratio	p_{37}	outer diameter of spring enclosure
p_{13}	flow area variation ratio	p_{38}	clash allowance ratio
p_{14}	seal outer diameter	p_{39}	maximum fluid pressure in pipeline
p_{15}	radial clearance between helical spring and poppet stem	p_{40}	valve outer diameter
p_{16}	flow line diameter	p_{41}	equivalent orifice diameter
p_{17}	computed spring rate	p_{42}	valve cylinder thickness
p_{18}	helical spring index	p_{43}	allowable stress for pipe material
p_{19}	flow coefficient (head loss factor)	p_{44}	helical spring free length
p_{20}	helical spring inner diametral clearance ratio	p_{45}	pipe thickness
p_{21}	cracking force on helical spring	p_{46}	inner diameter of spring enclosure
p_{22}	orifice coefficient	p_{47}	valve configuration factor
p_{23}	helical spring maximum stress	p_{48}	corrosion resistance allowance for valve enclosure
p_{24}	maximum deflection of valve and spring due to fluid force	p_{49}	corrosion resistance allowance for poppet valve stem
p_{25}	helical spring wire diameter		