The Pennsylvania State University

The Graduate School

Industrial and Manufacturing Engineering Department

A NEW METHOD FOR EVALUATING DESIGN DEPENDENCIES IN PRODUCT ARCHITECTURES

A Dissertation in

Industrial Engineering

by

Oyku Asikoglu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2012

The dissertation of Oyku Asikoglu was reviewed and approved* by the following:

Timothy W. Simpson
Professor of Mechanical and Industrial Engineering
Dissertation Adviser
Chair of Committee

Soundar R. T. Kumara
Distinguished Professor of Industrial Engineering

Gül E. Okudan Kremer
Associate Professor of Industrial Engineering and Engineering Design

Sven G. Bilén
Associate Professor of Engineering Design,
Electrical Engineering, and
Aerospace Engineering

Paul Griffin
Professor of Industrial Engineering
Head of the Harold and Inge Marcus Department of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Many different approaches to the problem of defining and measuring modularity and complexity in product architectures have been developed. Redesign complexity and change propagation are two of the main foci for such work, but there have been relatively few attempts to quantify design dependencies in product architectures. Most of these approaches rely on "engineering intuition" of the design teams for determining the redesign effort for a product; however, this introduces uncertainty and bias to metrics from two important aspects:

- Subjective opinions of engineers can lead to company- and product-related results that do not provide good metrics for different types of product architectures; and
- Relying solely on engineering intuition may overlook some important indirect connections in the product architecture.

As the complexity of systems increase, the difficulty of accounting for the discrete indirect interactions in these systems also increases. Considering the disadvantages of using subjective engineering expertise, the need for an unbiased metric is apparent. For that reason this research develops a simple metric that can be calculated directly using input data retrieved from Design Structure Matrices (DSMs). DSMs facilitate visualization of both inter- and intra-modular connections that are encompassed in the product, which are critical for understanding the architecture of the product. Connection weights between modules or components are captured by referring to the number of interfaces and different interface types. After determining connection weights within the product architecture, a new algorithm to calculate design dependencies is employed. This algorithm incorporates the effects of indirect connections in the system to the redesign effort by using a circuit analogy from electrical engineering.

Lastly, this work presents a step-by-step method and analyzes a set of twenty-one small electro-mechanical products. Important statistical data about the interface occurrence frequencies are collected and weight for individual interface types are assigned. The results are validated by several tests including sensitivity analysis and design change analysis within different product architectures. Comparison with existing methods is also provided.

The research fills an important gap in the literature by providing a simple, unbiased metric that can be used to compare different product architectures based on their design dependencies as well as evaluating the “change sensitivity” of individual components in a product architecture. This provides a very important decision-making tool for design engineers to reduce the duration of the redesign process by pointing out the less flexible design architectures or highly change sensitive components.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I thank Dr. Timothy W. Simpson for advising me in this research, providing much needed guidance and for all his patience and efforts. He was always here to answer any of my questions and provided me invaluable lessons and insights that were essential to the workings of academic research in general. I also thank my committee: Dr. Gül E. Okudan Kremer, Dr. Soundar R. T. Kumara, and Dr. Sven G. Bilén for their interest, instruction, and valuable feedback.

I would like to thank Martin Samayoa, for his unconditional love, support and encouragement even under the most difficult circumstances and for being by my side from the beginning of my Ph.D. journey to the end. I also would like to thank to my brother Övünç Aşıkoğlu and to my dearest friends, Didem İkis, Sema Erten, Orhan Kışlal, Ayşe Pamir Cırdı, Günsü Bağcı and Emre Artun for their companionship and moral support.

Finally and mostly, I thank my parents, Mustafa Aşıkoğlu and Oya Aşıkoğlu, who were always caring and provided me with everything I always needed to receive the best education, for their patience and understanding during the preparation and completion of this work.

To My Parents

Mustafa Aşıkoğlu and Oya Aşıkoğlu

CHAPTER 1

INTRODUCTION

1.1 Background

Different strategies for providing a wide variety of goods at low cost have been practiced to achieve “mass customization” of goods. Companies should quickly move into creating niche markets and to be the first offering new features to increase their market share. To achieve this, the design focus should shift from mastering primary functions towards offering additional features with decreased costs. However, product design is a costly and lengthy process comprising many risks and uncertainties, and volatile demand changes in the market require quick responses from companies to stay ahead of their competitors. Hence, it is not typically feasible (or cost-effective) for companies to design new products for each niche market.

In this case, new strategies such as reconfiguration of pre-existing products or utilization of highly compatible modules need to be employed [1]. The complexity of the overall product architecture increases substantially as features are added to respond to customer needs. Adding features can cause major design changes within the base product. Each new feature in a product necessitates configuration alterations within the system such as component and functional changes. This also brings further uncertainty and component reliability issues that elongate the design and testing process.

Modularity is often employed to facilitate product architecture design [1]. It is motivated by speed-to-market pressures since its implementation can reduce product development time and increase market share while providing competitive advantages [1]. It also entails promoting design robustness to reduce the marginal effort (and cost) of redesign [2]. Decomposition of the system into modular "chunks" also allows simplifying the architecture in order to minimize and localize any possible malfunction or module incompatibilities, thus reducing uncertainties [1].

Modular design representations often utilize Design Structure Matrices (DSMs) to map all of the connections between components in the product and define modules by clustering components into groups that are intensely connected to each other [3]. DSMs facilitate visualization of both inter- and intra-modular connections that are encompassed in the product, which are critical for understanding the structure of the product [3]. For modular designs, DSMs are used to define the interactions between modules and are used to group the components that are highly dependent on each other due to both structural and functional reasons. In modular designs, the number of connected modules (either functionally or physically) represents the design dependency of that module within the product; therefore, fewer connections imply fewer dependencies, and fewer design dependencies increases the flexibility of the product for future design changes [4].

Even though clustering and grouping of parts into modules helps define subsystems and isolate highly interdependent components, intra-modular connections also become highly linked as the product gets more complex. Even if a module has connections to only one module, it will be indirectly connected to other modules in the system through the modules to which it is connected. This creates a "network" of modules in which every module is connected to each other with direct or indirect links [5]. It might be hard to comprehend the indirect networks in the system, but a network graph can be constructed by using the connection information from a DSM to visualize the linkages between modules. As an example, consider the components in the optical computer mouse shown in Figure 1.1.

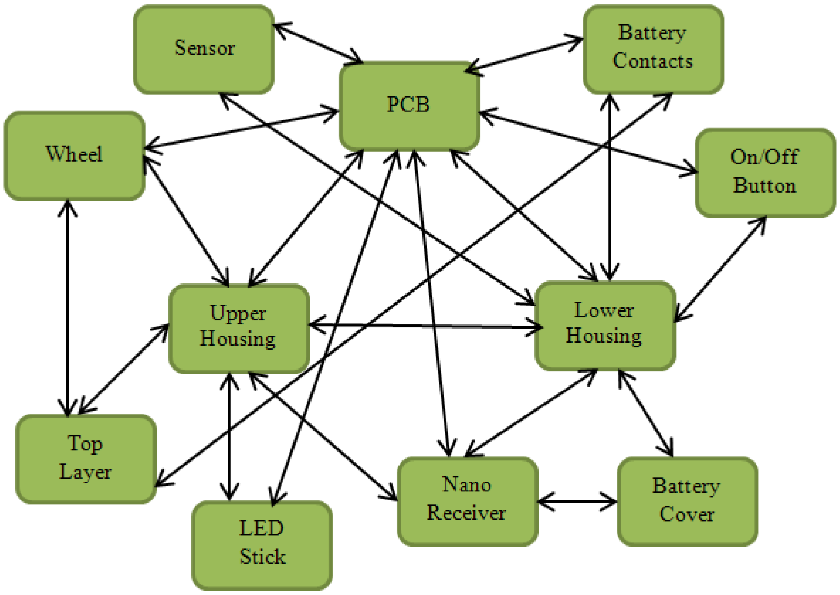


Figure 1.1. Component Network Representation of an Optical Computer Mouse

As shown in Figure 1.1, in such a network, modules can be indirectly connected through other modules even though they do not share interfaces; hence, changes in one module can propagate through the whole system via a combination of direct and indirect connections [6]. In order to estimate the “ripple effect” of change propagation in the whole system that is triggered by one module, it is very important to understand the nature of the interactions that each module has with each other. Intense intra-modular dependencies often reduce desired robustness and increase the complexity of a system

[6]. Consequently, even small-scale modifications may amplify and affect numerous subsystems, leading to substantial design alterations. In order to avoid prolonged design processes and increased costs, the flexibility of a design should be determined in both structural and functional models. Evaluation is necessary both for pre-existing architectures for platforming purposes as well as for new product design processes.

In addition to the functional properties of the module (e.g., range of power, speed or force that it can supply), the characteristics of interfaces also affect the change propagation and redesign flexibility within the system [7, 8]. Even though a module might provide several different functions (force, speed, power), it might not share all those functions with all the other modules to which it connects. Therefore, for a realistic evaluation of redesign flexibility and modular design dependencies within the system, the type of individual connections should be evaluated. Interaction types have been determined depending on the information being carried [3, 8, 9]. Lai and Gershenson [4] define design dependencies based on the connectivity of components. Representation of these interfaces is one of the most fundamental aspects of modular design strategies. Increasing interest in modularity studies created more attention for research in inter-connections and flows between modules. Since then, there have been numerous studies on the topic. Various approaches using functional and structural elements and mixtures of both have been adopted, and several types of description systems have been proposed. For example some studies have classified the types of interactions into generalized categories such as: electrical, mechanical, thermal, controls and vibrations [3, 9]. The type of the information being carried also might indicate the difficulty (or complexity) of the interface [7, 8].

As shown in Figure 1.1, a system includes numerous direct and indirect connections and can be effectively visualized as a network of modules [5]. Considering that the redesign flexibility of a system depends largely on the module connections, a comprehensive calculation of the complexity embedded in the design dependencies should include the direct *and indirect* effects of all the interactions in the system. However, such a method remains absent in the literature as discussed in Chapter 2, and this research addresses that gap.

In complex products, the number of parts and the relationship between them has been referred to as the main source of developing design problems [6]. Related factors that contribute to such problems include: wrong design decisions, insufficient clarification of the task and inadequate design or redesign processes, as well as insufficient communication between design engineers and incorrect, inconsistent architecture representations [6]. Therefore, to be able to avoid faulty designs and to minimize redesign efforts, proper interface representation is fundamental.

1.2 Statement of the Problem

Design dependencies affect many of the important architectural decisions of a product. Even in modular structures, managing interfaces becomes much more difficult as the product becomes more complex as new features are added. Due to the amount and different types of connections that they may contain, some architectures might be more susceptible to amplification of design changes within the system than others [6]. Such architectures are most likely to increase the cost and duration of the redesign process;

hence, it is very important to be able to determine the impact of design dependencies in advance.

Despite several metrics proposed for assessing the complexity and flexibility of product architectures, there are only a few studies on quantifying design dependencies, and they rely mostly on “engineering intuition” of the design teams for determining the redesign effort of a product [5, 7]. In these studies, the input data are collected by surveying engineering teams about the design dependencies of modules in particular products. Using subjective information also hinders the comparability of the metric even within the same engineering design team. Rankings by numbering or ratings are based on subjective opinions, and they can vary based on the experience or the engineers’ understanding of the product architecture [6, 7].

Recognizing sources of uncertainty is also very important in engineering design. Overlooking the uncertainty in a design process can lead to under-designed products [10]. The most important uncertainty in design is introduced during the customer assessment stage when product designers need to assess the customer’s willingness to make tradeoffs between attributes. In this process, customers rate their preferences for product features and rank them based on the importance of the feature. Previous studies show that poor product evaluations introduce uncertainty into the process since customers tend to misconduct their preferences in the surveys, and an inadequate amount of data might misrepresent overall customer preferences [6]. This type of uncertainty is also introduced into data collection processes based on engineering team interviews. Engineering opinion of experienced design teams (or members) compared to novice design teams might be different for the same product [7]. Collective decision-making also introduces bias to the situation; so, decision-makers may be influenced by the decisions of other group members. Bias in group decision-making can be reduced by employing several techniques, but it is hard to eliminate completely [11].

Therefore, data collected for the proposed metric can lead to company- and product-related results that cannot be generalized for use with different types of products. As more companies strive to apply strategies to increase their speed to market, the importance of flexible designs increases [12]. While there are several studies focusing on the flexible designs [12-14], only a few of them understand the importance of interfaces in the product architecture, and most of them remain application- and case-specific [5, 7, 8]. Resorting to engineering intuition might also cause the importance of some indirect connections to be overlooked. In such methods, only the most apparent connections that are recognized by the design engineers have been included in the calculations whereas most of them are completely ignored [7, 12]. The importance of change propagation within the system has also been limited to direct (and the most obvious indirect) connections [12]. This underestimates the effects of a design change transferred by indirect connections within the architecture. As the complexity of the overall system increases, the difficulty of accounting for the discrete indirect interactions in the system also increases. For that reason this research focuses on developing a simple metric that can be calculated directly by using input data retrieved from a DSM that incorporates both direct and indirect connections in the whole system for calculating intra-modular dependencies.

Considering the disadvantages of using subjective engineering expertise and the lack of focus on the indirect connections within the system, the need for an unbiased metric is apparent based on the previous discussion.

A system to represent interactions in a product architecture should encompass characteristics of:

- *Consistency:* so that reliable analysis throughout different products or different brands can be provided;
- *Simplicity:* to include only needed features reducing calculation complexity;
- *Practicality:* to be learned, implemented and modified quickly, and
- *Objectivity:* so that different teams of engineers can reach the same conclusions during the analysis of the same product.

The metric should be flexible even to be used in the conceptual (and later) design stages as well as during benchmarking (i.e., data that can be captured through product dissection). Benchmarking helps to assess alternative designs and compare the cost and variety that can be provided by a given product architecture [15]. Developing such a metric would help pinpoint inflexible designs in advance and avoid related costs.

This research proposes a new method to evaluate design dependencies in a product architecture by using interaction data from product DSMs. This work also introduces a hypothesis that product architectures can be modeled using an electrical circuit analogy and that relationships between design change and connections within the architecture can be explained by analogy to Ohm’s Law. This helps evaluate all direct and indirect relationships in the physical structure. The following sections introduce motivation and the research objectives for this study with an example case study and outline the rest of this dissertation.

1.3 Research Objectives

This research aims to develop a novel objective method to assess the design dependencies in a product's architecture and provide a guide on how it is used with applications on existing products. In accomplishing this overall objective there are three main objectives that are accomplished:

1) The first objective is to compare all previous efforts on defining the flows and connections in product architectures and trace the evolution of these approaches over time.

2) The second objective is to reconcile the inconsistencies that arise due to the sequestration of design knowledge in interfaces. To accomplish this, a simplified and practical system to be used with analysis of existing architectures is introduced.

3) The third objective is to provide an objective method to quantify the design dependencies within products architectures. An electrical circuit model is introduced to measure flexibility in product architectures, and guidelines to apply the proposed method are also provided.

1.4 Outline of Dissertation

There have been several related studies in the past [16-18]; however, most fall short on delivering an unbiased metric accounting for all direct and indirect connections within the architecture. In addition to delivering such a method to fill the gap in this area, this study completes three main objectives. The proposed method is applied to and validated by case studies. The scope of the case study has been limited to small electro-mechanical products to keep the results comparable. Twenty-one small household products have been inspected by systematic dissection and physical relations within the product have been mapped.

Details of the introduced method and outcomes are explained in the following chapters. Chapter 2 reviews related work on the subject. Studies on modularity, interface representation, complexity, and change propagation lay the foundation for the proposed work. Chapter 3 encompasses a detailed assessment of existing taxonomies of interface classification systems and proposes a new interface classification system. Quantifying design dependencies and modeling complex product architectures are elaborated in Chapter 4. The proposed method for evaluating design dependencies is explained step-by-step in Chapter 5, and validation of the method is provided in Chapter 6 by analyzing twenty-one small electro-mechanic products. Finally, Chapter 7 highlights the research contributions and gives closing remarks for the dissertation.

CHAPTER 2

LITERATURE REVIEW

To understand how change propagation and design dependencies are managed to reduce complexity in product architectures, key studies based on modularity, complexity and change propagation in engineering design have been reviewed to lay the foundation for the proposed research.

2.1 Modularity in Product Design

Determining problematic modules or design architectures is beneficial in terms of reducing costs of the product throughout its life cycle. The cost benefits of modularity and commonality have been classified under categories of product design, manufacturing, inventory, and use costs [19]. Developing modular architectures has become the dominant strategy to offer new products with shorter lead times, as well as reducing complexity [20].

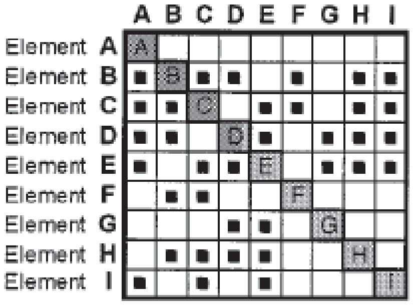
Modularity has been investigated from different aspects including product, process, organization, or innovation and has several definitions [19]. In one of several studies, modularity is defined as: a one-to-one mapping between functional elements and physical components including "decoupled component interfaces" [21]. Baldwin and Clark [22] advocate modularity as a solution to the growing design complexity problem and define modularization as a process of creating complex products or processes from components that can be designed independently yet are compatible together. Otto and de Weck [20] define a module as “a chunk that is highly coupled within but only loosely coupled to the rest of the system.”

Several studies also focus on developing methods for defining modules. There are three main approaches to defining the modules in a product architecture: (1) heuristic models based on functional modularization [23], (2) Design Structure Matrix (DSM) clustering analysis [24], and (3) modular functional deployment methods [8, 18]. One of the first heuristic algorithms for modularization was proposed by Stone et al. [23] to cluster modules based on a functional model; their approach includes three heuristics that target dominant flow, branching flow and conversion-transmission functions. In another study, Meng et al. [25] merge four important principles of module identification: (1) isolation of individualization, (2) localization of change, (3) functional independency and (4) structural independency. They introduce quantitative models for each of these principles and employ a genetic algorithm based approach to combine and solve all four models as an optimization problem [26]. Another approach of modularization is developed by Ericsson and Erixson [18] to classify components based on their shared functions. In their work the authors utilize DSMs to represent design dependencies between components [18]. In addition to other functional heuristics, their study also considers customer preferences in the process [8].

In one of most important studies on modularity, Sosa et al. [5] investigate various disciplines that are used to define and measure modularity in product architecture. The study mostly focuses on modularity at the component level, and they introduce a measure called distance modularity that captures the indirect design dependencies in the system based on the distance of the components [5]. Even though the measure might be practical for small products, as the analyzed product gets bigger and more integrated, the computational complexity for this metric would increase substantially. In a recent study, Gupta and Okudan [27] combine modularity, DFA, and DFV approaches to provide better insights on the conceptual design selection.

A common tool used to identify and define modules in a product architecture is the Design Structure Matrix (DSM) [3, 8, 20, 28]. DSM facilitates one-to-one mapping of all components or subsystems within a structure in a visual manner (see Figure 2.1). DSMs have also been used to modularize the design process to combine similar activities to determine inter-task dependence in projects in order to reduce costs and increase quality and efficiency [2]. DSMs are actually classified based on their application areas: component-based or architecture DSMs, team-based or organization DSMs, activity-based or schedule-based DSMs and parameter-based or low-schedule DSMs [3]. Component-based and organizational-based DSMs are static DSMs that can be analyzed with clustering algorithms; whereas activity and parameter-based DSMs are considered as time-based DSMs and analyzed with sequencing algorithms [3]. Component-based DSM analysis assists representation of the system architecture by decomposing the system into sub-systems, decoding the relations between these subsystems and analyzing possible reintegration of those subsystems [3]. The focus in this work is on component-based DSMs.

A component-based DSM is a square matrix consisting of the same elements for both row and column labels. Such DSMs usually map the relationships between components with a binary system. The existence of a relationship is typically represented with a dot or a number “1”, whereas cells for non-existing relationships are left blank. The diagonal of a DSM is also left blank since a component does not connect to itself.



	A	B	C	D	E	F	G	H	I
Element A	A								
Element B	■	B	■	■		■		■	■
Element C	■	■	C		■	■		■	■
Element D	■	■		D	■		■	■	■
Element E	■		■	■	E		■	■	■
Element F		■	■			F			
Element G				■	■		G		
Element H		■	■	■	■			H	
Element I	■		■		■				I

Figure 2.1. Example DSM [3]—Non-Blank Cells Indicate Connected Elements

In modular architectures, well-defined and well-designed interfaces are as important as defining the modules since compatibility and interchangeability depends on it [20]. Pertinent literature on interface classification includes several studies on the representation of modular interfaces [3, 8, 9]. Lai and Gershenson use [29] component-to-component DSMs to calculate the dependency and similarity relations between components in order to develop modules that facilitate the assembly process. They use different factors for dependency weights assigned based on shapes, joining methods, or handling techniques of parts [29].

Browning’s [3] work on different applications of DSM addresses different types of interactions, such as spatial, energy, information and material and offers to weigh the interaction relative to each other. However, his study assigns only one type of interaction per interface, which overlooks the importance of other types of connections available in the same interface. In his work, interactions are weighted by design engineers with a range from +2 to −2 regarding the “necessity of physical adjacency of components” [3].

Similarly, Holtta and Otto [7] and Sosa et al. [5] acknowledge the importance of determining different types of interfaces and try to capture their relative weights by interviewing design teams. However, using subjective ratings based on engineering experience introduces bias to the assessment process. The proposed research overcomes this limitation and provides an objective and practical measure.

2.2 Complexity in Design

Modularity strategies are used to manage the complexity in product design, product design processes or manufacturing practices. A proper measure to quantify system complexity is one of the most important aspects when it comes to managing it, and there have been several studies that propose different methods. In one of the earlier studies, Braha and Maimon [30] focus on design process complexity and claim that functional information content is fundamental to develop successful design. The information content in terms of physical artifacts are loosely defined as relations and modules [30]. Another study that adopts a mathematical approach to define complexity is by El-Haik and Yang [31], in which they calculate the complexity of a product by using mathematical representations of the independence and the information axioms introduced by Suh [32]. They also incorporate the Boltzmann entropy theory as a measure of complexity since it enables calculation with continuous random variables, which they associate with complexity in their work [31]. In his axiomatic design studies, Suh [32] defines two main axioms to maintain the independence of the functional requirements (FRs) and minimizing the information content of the design. Finally, product design complexity has been studied by Holtta and Otto [7] using a “black box” method to define function structures. They acknowledge the importance of the interface types within the architecture but go into too much detail in calculations where the metric loses practicality and generality.

Other investigations of design complexity utilize different disciplinary tools such as social network analysis [5, 33, 34]. For instance, Sosa et al. [5] focus on modularity at the component level and define modules based on the “level of independence or disconnectivity” from other components. They define three different types of modularities (degree, distance, and bridge modularity), which depend on a different centrality measure [5]. Their work falls short of generalizing the results beyond the engine example and introduces the uncertainty of design expert opinion in building up the initial component network.

In a recent study reviewing complexity metrics, Summers and Shah [33] classify studies into three groups: design product, design problem, and design process. As a result of analysis of close to seventy studies on complexity, their study proposes three different measures of complexity based on size, coupling, and solvability [33]. Although, they approach the topic form a wide perspective, they do not offer a unified measure and leave the question of advantages and disadvantages of these metrics open. Crespo et al. [35] compile exiting complexity metrics and provide a detailed comparison of them by using hip replacement devices. Finally, the authors propose a new complexity metric incorporating the intended use for the product along with functions, number of interactions, and number of components [35].

2.3 Change Propagation in Product Architectures

Interest in the effects of design change in product architectures started increasing as an extension to the research on product platform architectures and design for variety [12]. Martin and Ishii [12] focused on variety within the existing product line as well as future generations and introduced the Generational Variety Index (GVI) to be able to determine the components that are most likely to change. GVI is used to determine the connectivity of components and track change propagation in the system [12]. They also introduce a Coupling Index (CI) to measure the relative strength of connection between sub-systems by using a rating method ranging between 0 and 9 (uncoupled to strongly coupled) [12]. Coupling strengths are judged by engineering design teams. The index uses a one-to-one matrix to map the interactions between components, which are also classified by their type as heat output, pressure and voltage supply [12].

One of the most prominent works on the characterization of the engineering design propagation is done by Eckert et al. [6], in which they identify the problems of grasping the relationships of a complete system as “a major source of emergent problems”. Their study defines a complex product as a product with closely linked parts and systems where a system change most likely transfers to other systems [6]. They examine a complex system of a helicopter and their interviews with the engineering team determines that only 70% of the whole system was completely understood [6]. The lack of understanding the complete system comprehensively also brings major downfalls in determining how design changes propagate through a system. They elaborate on the sources of change, system problems, change propagation and types of components behavior under change propagation, change prediction and consequences of change has been described under the example of helicopter design and some shortcomings of relying on engineering intuition for design change are also pointed out [6].

In a follow-up study, Clarkson et al. [14] introduce a method to predict change propagation using DSMs. They describe change relationships as a combination of likelihood and impact [14]. Likelihood is defined as the probability of a design change requiring a change in the product architecture and impact is defined as the average proportion of the redesign if the change propagates in the system [14]. The data needed for the construction of the likelihood and impact matrices are collected from the historic information on previous design changes and engineering expertise [14], which incorporates personal bias into the analysis. Even though they acknowledge the importance and try to capture the effects of indirect connections in the system by using "propagation trees", their method fails to provide a practical solution to be used in products with lower-level system granularity in which there are increased numbers of components. Later on, the method of using "propagation trees" is formalized into the Change Prediction Method (CPM) to visualize direct and indirect change propagation in computer software setting [36].

In an application-based study on design flexibility, product architecture is defined by its elements: number of parts and interfaces, the types of interfaces, functions, and the number modules in the architecture [13]. Influenced by Failure Modes and Effects Analysis (FMEA) methodology, they introduce the Change Modes and Effects Analysis (CMEA) method to determine product flexibility [13]. Similar to the method proposed in this work, the CMEA method also starts with product decomposition for data collection

[13]. This enables the engineers to control the granularity level of the analysis. As a significant difference from the proposed method in this work, CMEA uses a functional approach and decomposes the products based on their functions [13]. In addition to the data collected from the product itself, the information needed to calculate the flexibility of the product also requires information on probability of change occurrence or possible effects of change or change readiness [13]. Unfortunately, their study gathers the vital information on change from engineering expertise, which introduces bias to the analysis.

Suh et al. [37] also focus on managing change propagation in systems providing a change propagation index that enables classifying components based on their change transmittance characteristics such as constants, multipliers, carriers and absorbers. The work provides a detailed look into the mechanism of change propagation and extends the work to address of cost determination and uncertainty analysis [37]. However, the construction of the change propagation matrix heavily depends on observations that increase the risk of incomplete or incorrect data.

Building on the studies of Suh et al. [37] and Eckert et al. [6], Giffin et al. [34] introduce a network-based analysis technique that can be applicable to large data sets. The study defines change propagation characteristics of the system and uses a DSM to visualize the connections in terms of physical connections with information and energy flows [34]. Even though the study is comprehensive, it focuses on the complex network system composed of software, hardware and documentation areas.

2.4. Chapter Summary

This chapter reviewed key studies related to the definition and quantification of flexible product designs and change propagation and modularity. There have been numerous studies on product architecture representation, and numerous different interface type classification methods have been introduced. Since correct product architecture representation is one of the most important aspects of assessing of design dependencies in product architectures, it is necessary to review those studies in detail to understand their limitations. While methods have been developed to assess change propagation and complexity in product architectures, a method that provides an unbiased, practical approach to evaluate the design dependencies within product architectures is absent. The next chapter focuses on reviewing of existing taxonomies on product architecture representation.

CHAPTER 3

PRODUCT ARCHITECTURE REPRESENTATION

3.1 Introduction

In engineering design, analysis of the product architecture provides important insights on solving in any design-related problems. Decomposition methods are often used to understand the product architecture and its interconnections. Awareness for the importance of product architectures and studies of product structures began in the early 1980s when Steward [38] proposed using the design structure technique to represent inner connections of product architectures. As the number of studies tackling the problem of improved design (from different perspectives: DFA, DFM, DFV) increased, the need for representing product architectures became prominent. Several researchers used different approaches to find the best way for representing a product's architecture. Using a standard language to represent product architectures provides reliable and repeatable interpretations of product architectures that can be understood and shared by different design engineers or design teams. Numerous attempts to develop such a language have been made, and this chapter reviews these attempts in a chronological order to shed light on their evolution within the research community.

3.2 Assessment of Existing Taxonomy

Based on a thorough literature search and review of 18 pertinent studies on interface classifications systems, three main approaches have been observed: (1) functional, (2) structural, and (3) hybrid approach. Between functional and structural approaches, functional approaches focus on the primary function that is intended for each component, whereas structural approaches focus on the physical connections between components. Most of the time, these two different approaches are used for two different purposes. Functional approaches are mostly preferred in new product development when the physical connections within the product architecture are yet to be determined. On the other hand, the structural approaches are used to describe existing product architectures. There also are some studies that adopt a combination of these approaches, namely, hybrid approaches that try to capture both aspects in a design structure. The advantages and disadvantages of each approach is reviewed based on its purpose and the basis of its proposed classification. Classification definitions as well as any available case studies and/or outcomes of the applications are examined.

3.2.1 Functional Approaches

In one of the earliest studies on mechanical systems, Hubka [39] introduces an extensive study in classification of various categories based on functions, working principles, complexity, functional dependencies, quantification potential and more. He identifies the relations between the elements in a system as “couplings” and indicates that the couplings among the system elements can be of various kinds such as mechanical, electrical, chemical, magnetic, time and spatial [39]. However, he only focuses on geometrical and locational relationships in his book and does not articulate on the details of other structural connections.

Another earlier study visualizes the physical dynamic systems by using energy exchanges in the system which are known as bond graphs [40]. Developed by Paynter [40] in 1959, bond graphs describe the systems under different classifications such as energy domain, effort, and flow. Energy domain includes two different definitions of mechanical energy namely: rotational and transitional.

Leveraging Hubka's [39] descriptions, Pahl and Beitz [41] also treat technical artifacts as systems that are connected to the environment with inputs and outputs that are defined by a system boundary. In their work, each system can be defined as a summation of subsystems and includes processes involving conversions of matter that can be found in forms of material, energy or information. The conversion of these different types of matter is called *flow*. A conversion of the matter from one form to another is facilitated by input/output relationships that necessitate different types of *functions*.

While Pahl and Beitz [41] list the types of functions as convert, connect, change, vary, and store, they also distinguish between *main functions,* which affect the overall function of the artifact directly, from the *auxiliary functions,* which have indirect input to the system. However, they acknowledge difficulty of definitively differentiating these two types and point out that using deterministic function definitions to describe the system can jeopardize finding solutions in higher granularity levels. Being one of the most fundamental and most cited publications in engineering design, the authors prefer breadth to depth and visit a wide range of engineering design-related topics limiting the amount of detailed information for each chapter. Consequently, their study remains short on providing enough detail for different levels of hierarchy or abstraction for the purpose of this research.

Suh [32], being another pioneer on engineering design issues, points out that most of the problems such as long development processes and faulty designs arise from poor translation of functional requirements of a design into structural models [32]. Functional requirements are defined as “the minimum set of independent requirements that completely characterize the functional needs of the product design in the functional domain” and design parameters (DPs) are the physical incarnation of the required functions [31]. As the solution for the design problems arises in the system, Suh [31] proposes a set of axioms to govern the translation of functional requirements into design parameters. These two main axioms of independence and information refer to “maintaining the independence of the FRs” and “minimizing the information content of DPs”, respectively [31]. By using the independence axiom, one makes sure that a given change in a particular DP only affects its corresponding FR while the information axiom minimizes the amount of connections (or coupling) amongst DPs and FRs to reduce complexity in the system. Both of these axioms aim to minimize change propagation and design complexity of the architecture. Suh [31] provides some of the most fundamental information on engineering design in his book and discusses the details his axioms with its corollaries but does not articulate properties or interconnections of physical structures of product architectures.

As the number of studies on systematic design increased, the need for defining the connections and the functions of the sub-systems of a product became prominent. Being one of the first studies on this topic, Hundal [42] focuses on developing a systematic design method to be used with a computer program to help design engineers during the conceptual design process. Influenced by the studies of Pahl and Beitz [41], he describes the systematic design method in four steps: (1) clarification of the task, (2) conceptual design, (3) preliminary or embodiment of design, and, lastly, (4) detailed or final design [42]. The second step of the method deals with the functional requirements of the product [42]. He suggests that at this stage the designer uses the decomposition techniques to obtain task-specific functions from the overall function of the product by using engineering expertise [42]. These functions claim to represent material, energy or information (signals) [42]. Defining the main flows (in terms of energy, information or material) in the system is the first step in developing an initial product structure. After determining the main flow, the second step includes naming the most important functions. His study classifies primary categories of these basic functions as: channel, store/supply, connect, branch, change magnitude, and convert [42]. He proposes that most of the physical functions can be represented as either one or combination of these function categories [42]. His database includes detailed description of these task-specific functions [42]. The complete list of basic functions and their sub-categories can be viewed in Figure 3.1. Hundal's [42] work serves as a useful method to easily match function structures for necessary features and describing the changes caused by the physical effects of these selected functions and using computerized databases helps to consider all possible combinations for the design process; however, addressing physical effects of the functions such as solid expansion, friction, and amplification falls short on describing the physical elements of product architecture. For this reason, Hundal's [42] method does not provide an interface classification system to describe the nature of the connections between elements in an architecture.

Written as a chapter in PDMA-Handbook of New Product Development [43], Cutherel's study on product architectures serves as a simple guide to how to develop product architectures as well as define the basic concepts related to the topic. Defining product structures from both functional and structural aspects, his starting point is the work of Ulrich and Eppinger [44] on product design and development. Cutherel's chapter includes some insights on the requirements of developing a good architecture as well as classification of product architectures as modular and integral [43]. Later in the study, he introduces guidelines for developing product architectures [43]. During the description of the steps on product architecture development he mentions the importance of defining interactions within the architectures [43]. The relations between chunks of components are defined as *interfaces,* and flows across these interfaces are called *interactions* in his study. His functional approach defines four types of interactions, namely, spatial, energy, information and material interactions [43]. Even though the study includes brief definitions of these interactions, he does not provide any basis or any previous work on which he builds his classifications. In addition, he also touches on the topics on product families and the effects of product architecture on various issues ranging from product change to impacts of product service; however, the importance of the physical connections are not mentioned in this section of this work either [43].

Advancements in modular product development made researchers pay attention to inner connections of modules within the architecture. In such a study, Kusiak and Huang [45] propose a methodology for developing modular products considering cost and performance. Building on the work of Suh [32], they define design as “creating solutions in form of products processes or systems to fulfill perceived needs by mapping functional requirements in functional domain” and “choosing appropriate design parameters from physical domain” [45]. Following the definition from a previous study of Ulrich and Seering [46], which describes design as “compilation of functional elements and their interconnections,” they claim that functional elements in a design correspond to mechanisms while the interactions between mechanisms correspond to functional flows [45]. They define six different types of “functional similarities” to represent modular components that exist in their study: geometric, temporal, force, electrical, thermal, and photometric. These types are visualized in interaction graphs during conceptual design [45]. They also use a weight density metric obtained from the number of functional interactions among components to define the quality of the cluster. This metric measures the number of the occurrence of specific function interaction between two components in different products. The information is then used for redefining modules [45].

They also define different types of similarities and rules that interaction functions should be similar or compatible for components to be grouped in the same module [45]. However, explanation on which interaction similarities (or edges) are compatible enough to be in the same module is not included in their study. The types of interaction similarities also seems to be generic and remains within the lines of their case study; therefore, their study falls short in providing an interaction classification system to be generalized for use in different products.

The attempts of developing a universal language to define all the relationships and flows within the product architecture have been significant and produced several publications in the literature in the past two decades [41, 42]. Due to employing a functional approach, most of these studies come close in terms of using similar classifications where terms can overlap or contradict each other. This situation creates confusion amongst researchers and engineers who adopt these languages for their studies or product architecture analysis. In addition, in some cases these studies do not cover different types of architectures and to stay specific to case study at hand. With an attempt of reconciling the differences between these languages and solving locality issues, Hirtz et al. [47] provide a comprehensive study in 2002 covering over 40 of the most significant articles up to that date. The authors review three different types of pertinent studies: functional modeling researches, NIST research efforts, and functional basis efforts [47].

Since both NIST taxonomy and functional basis were built extensively on previous work, Hirtz et al. [47] focus on reconciling these two languages [47]. According to their comparison, the NIST research efforts and functional basis classification studies are highly similar [47]; therefore, creating a system to reconcile these languages methodically is necessary. Constructing a hierarchical relationship for several levels of abstraction (from higher granularity to more detailed levels) is completed along with a set of integration rules [47]. Based on this set of rules, a new term is added when it is necessary for coverage. If new the term is a subset of an existing term, it would be placed in a lower level. If it is superset of an existing term, then it replaces existing term, and the existing term is moved to a lower level [47]. Using these rules, they review the functional basis and the NIST taxonomy to make sure that core function and flow terms do not overlap. This process is followed for all levels of abstraction, and different terms representing the same concept or same concepts represented by different terms are reconciled. As a result of their study, they establish a comprehensive functional classification system including detailed definitions, primary and secondary classes for functions, and secondary and tertiary classes of flow definitions. A detailed list of the classes and primary function for their study can be seen in Figure 3.1.

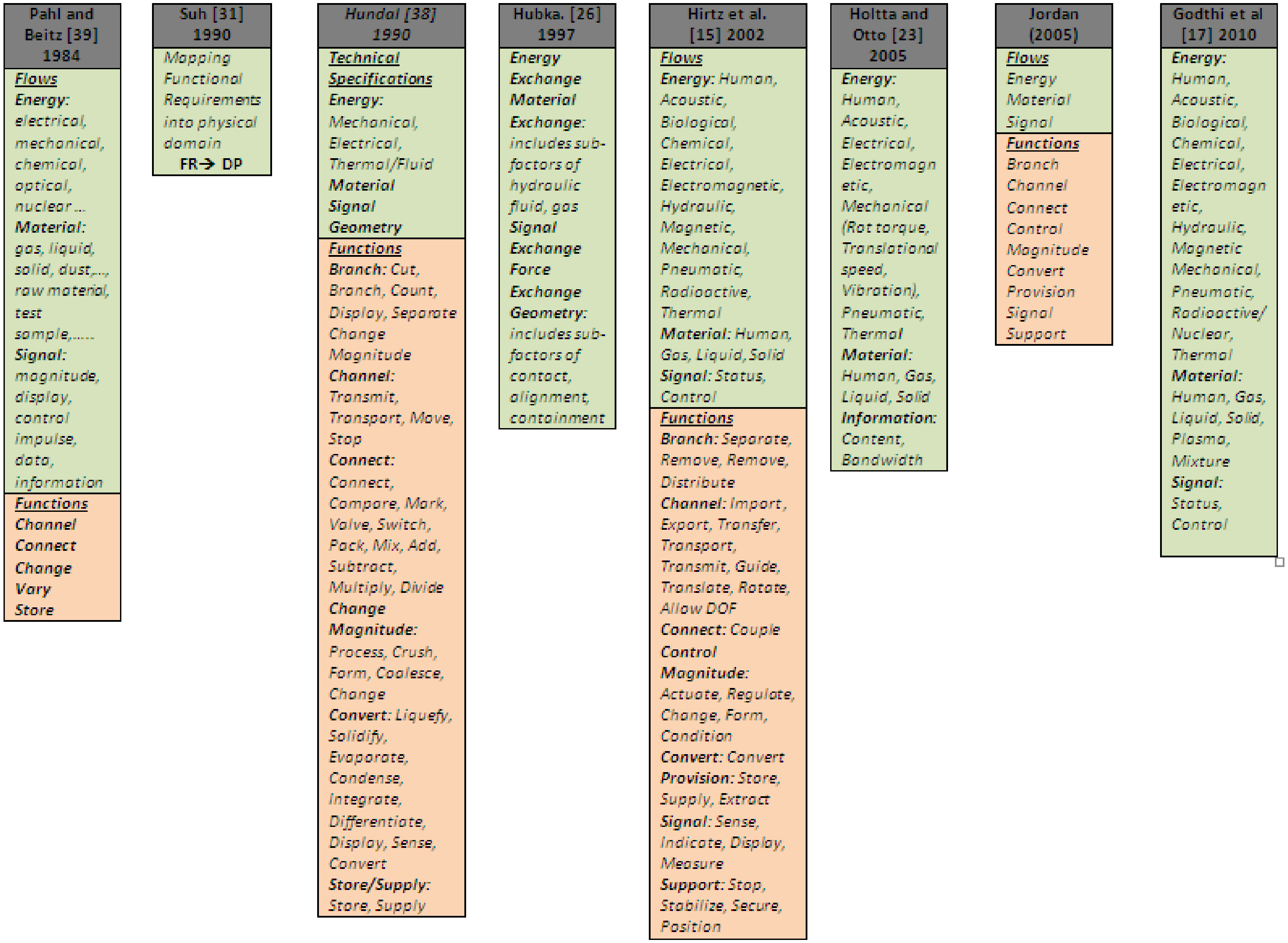
One of the most important studies on redesign complexity that focuses on interfaces has been published by Holtta and Otto [7]. In their study, they develop a redesign effort complexity metric to define the module boundaries by using information from the connections between modules [7]. According to their work, defining module boundaries helps minimize the change propagation in the product architecture [7]. By converging previous studies of Otto and Wood [48] and Thebeau [49], they introduce a six-step method from identifying the customer needs to defining module boundaries with the objective of minimizing the redesign effort in the system [7]. A modified version of the functional basis by Hirtz et al. [47] is used as the interface classification system to understand the redesign effort complexity of the product architecture [7]. To create a metric for early stages of the design process (before any project planning is done), they use a functional approach to define the interface characteristics [7]. Their classification system consists of the three main categories of material, energy and information and includes numerous sub-categories including electrical, mechanical, pneumatic, information, and material energy flows [7]. The complete list of flows categories and sub-categories can be seen in Figure 3.1.

Holtta and Otto's [7] procedure starts by identifying the customer needs and building function structures of the product. After using the DSM clustering algorithm developed by Thebeau [49], they determine the critical interfaces by decomposing the product and building function structures by using material, energy and information flows [7]. Function structures are then analyzed based on their inclination to increased complexity by assigning design effort complexity values for each flow and calculating the summation to reach the total design complexity number for each interface [7]. These values are assigned by consulting a design engineering group while considering only the primary intention of the flow. The last step uses a DSM representation of the product architecture to visualize all critical interfaces in the system to make modularization decisions [7]. They suggest modularizing the components as a way to keep complex interfaces within the module in order minimize their effects on the entire architecture [7].

In spite of providing structured guidelines and detailed information on the function structures of the product architecture, the use of an engineering team's opinion on the strength of connections and the change propagation in the system creates bias in the analysis and affects the repeatability of the results. Furthermore, the use of a functional approach in the early stages of product planning limits the level of abstraction and does not fit the purposes of this research.

One of the most recent studies introducing a detailed interface classification system is published by Godthi et al. [50], who develop a digital application to develop modular structures. Their goal is to develop an interface classification representation that can be used to create a machine-readable language to define the modules with an automated modular product design software [50]. This cyber-infrastructure is intended to support global manufacturing and be updated frequently in real-time. Developing such a language necessitates a vast amount of detail especially if the software is going to assist in developing the actual modular structures. Since their proposed system aims to design concepts based on the given requirements, they adopt a functional interface definition approach [50]. As mentioned previously, when it comes to developing new product architectures, functional approaches are always preferred since the structural connections of the system have not been established yet.

Building on the previous work of Hirtz et al. [47] and Stone et al. [23], Godthi et al. [50] incorporate several other factors in a design repository to create a complete database that includes all needed information for product development [50]. The repository includes seven different classifications of information including the function classification with primary, secondary and tertiary flow terms as well as performance types to specify the flow [50]. The details of these flows can be seen in Figure 3.1 in comparison with other proposed classifications. Even though the classification is built meticulously to provide information about the necessary functions, the system does not provide sufficient information to map the product architecture to track propagating design changes.



Pahl and Beitz [39] 1984	Suh [31] 1990	*Hundal [38] 1990*	Hubka. [26] 1997	Hirtz et al. [15] 2002	Holtta and Otto [23] 2005	Jordan (2005)	Godthi et al [17] 2010
Flows ***Energy:*** *electrical, mechanical, chemical, optical, nuclear ...* ***Material:*** *gas, liquid, solid, dust,..., raw material, test sample,.....* ***Signal:*** *magnitude, display, control impulse, data, information*	*Mapping Functional Requirements into physical domain* **FR→ DP**	***Technical Specifications*** ***Energy:*** *Mechanical, Electrical, Thermal/Fluid* ***Material*** ***Signal*** ***Geometry***	***Energy Exchange*** ***Material Exchange:*** *includes sub-factors of hydraulic fluid, gas* ***Signal Exchange*** ***Force Exchange*** ***Geometry:*** *includes sub-factors of contact, alignment, containment*	***Flows*** ***Energy:*** *Human, Acoustic, Biological, Chemical, Electrical, Electromagnetic, Hydraulic, Magnetic, Mechanical, Pneumatic, Radioactive, Thermal* ***Material:*** *Human, Gas, Liquid, Solid* ***Signal:*** *Status, Control*	***Energy:*** *Human, Acoustic, Electrical, Electromagnetic, Mechanical (Rot torque, Translational speed, Vibration), Pneumatic, Thermal* ***Material:*** *Human, Gas, Liquid, Solid* ***Information:*** *Content, Bandwidth*	***Flows*** *Energy* *Material* *Signal*	***Energy:*** *Human, Acoustic, Biological, Chemical, Electrical, Electromagnetic, Hydraulic, Magnetic, Mechanical, Pneumatic, Radioactive/ Nuclear, Thermal* ***Material:*** *Human, Gas, Liquid, Solid, Plasma, Mixture* ***Signal:*** *Status, Control*
Functions ***Channel*** ***Connect*** ***Change*** ***Vary*** ***Store***		***Functions*** ***Branch:*** *Cut, Branch, Count, Display, Separate* *Change Magnitude* ***Channel:*** *Transmit, Transport, Move, Stop* ***Connect:*** *Connect, Compare, Mark, Valve, Switch, Pack, Mix, Add, Subtract, Multiply, Divide* ***Change Magnitude:*** *Process, Crush, Form, Coalesce, Change* ***Convert:*** *Liquefy, Solidify, Evaporate, Condense, Integrate, Differentiate, Display, Sense, Convert* ***Store/Supply:*** *Store, Supply*		***Functions*** ***Branch:*** *Separate, Remove, Remove, Distribute* ***Channel:*** *Import, Export, Transfer, Transport, Transmit, Guide, Translate, Rotate, Allow DOF* ***Connect:*** *Couple* ***Control Magnitude:*** *Actuate, Regulate, Change, Form, Condition* ***Convert:*** *Convert* ***Provision:*** *Store, Supply, Extract* ***Signal:*** *Sense, Indicate, Display, Measure* ***Support:*** *Stop, Stabilize, Secure, Position*		***Functions*** *Branch* *Channel* *Connect* *Control* *Magnitude* *Convert* *Provision* *Signal* *Support*	

Figure 3.1. Studies with Functional Approach

3.2.2 Structural Approaches

The first explicit study done from the structural approach on interface classification systems was by Sanchez [51]. In his work, Sanchez [51] analyzes the product development processes of successful companies and, drawing from their development strategies, proposes a scientific approach of product design characterized by objectives, techniques and logics. Based on the analysis of product development strategies of these companies, he introduces two important design principles, namely, *system design* and *component modularity* to increase the flexibility of design in order to introduce a large variety of low-cost products with shorter lead times [51]. In his system design principle, a product is defined as a system of components connecting to each other by a pre-specified array of interfaces, which allows enough flexibility to connect different variations of the same component without any major design change [51]. The second principle, component modularity, suggests that there exists a wide range of input and output variations in between components within a system and modular components are designed to function within a subset of these variations stated by the system design [51]. Sanchez [51] identifies five different types of input-output ranges, in other words, interface types: attachment interfaces, transfer interfaces, control and communication interfaces, spatial interfaces, and environmental interfaces. Descriptions of interface types and relevant examples are provided in his study; they are summarized as follows.

- *Attachment interfaces* are described as the structural connections between two components, which may or may not consist of an actual connector such as a bolt or a screw. Snap-fit connections are also considered as an attachment type of interface [51].

- *Transfer interfaces* represent the flow of materials or power between components. Transfer of electrical energy in a different range of current or voltages and transfer of motions such as torque in a different range of rpm's are examples of this type of interface [51].
- *Control and communications interfaces* describe the relation of signal or information flow between two components. Explicitly, this interface type represents a controlling relationship between two components where signals from one component can alter the state of the other component [51].
- *Spatial interfaces* define the geometrical and locational constraints of a component in the overall arrangement of other components [51].
- *Environmental interfaces* represent the interaction between two components where one component affects the performance of the other component or the whole system due to characteristics of its weight, sound, smell, light, taste, tactile surface, vapor, corrosions, heat, electromagnetic, and radiation [51].

In addition to defining these interface types, Sanchez also classifies components into strategic roles such as variety-driven, technology-driven and cost-driven components [51]. Further in the study, he points out the potential benefits of the proposed principles and provides examples of strategic product design from industry [51]. As one of the first studies in the topic, he mostly relies on his own experience in addition to analysis of the strategies of the aforementioned companies; however, he neither justifies nor limits the scope of his interface definitions to a certain type of products (for example small electro-mechanical household items). He also does not elucidate the list of products or the method of analysis used in his study to be able to define the design strategies. The case studies do not make use of any proposed interface type definitions in actual product architecture despite detailed component and interface type definitions in the work.

Another study investigating the importance of the connections between components has been published by Pimmler and Eppinger [16] around the same time as Sanchez [51]. In their study, they focus on the analysis of product design decomposition and vital information that can be obtained about the product structure [16]. In addition to using the information for organizing product development teams, their method is used to find the optimal product structure that could be carried out for several product generations, leading to a decrease in development costs and easy troubleshooting at any stage of the product life-cycle [16]. Building on the previous studies of Pahl and Beitz [41], Suh [32], and Ulrich [21] on mapping functional structures on product architectures, Pimmler and Eppinger [16] propose a three-step method for product decomposition analysis, starting with decomposing the system into elements, then continuing with documenting the interactions between those elements. Their method concludes with the last step of clustering elements into chunks [16].

Pimmler and Eppinger's [16] study is significant in terms of how they document the interactions between components. Similar to the Sanchez study [51], the interactions between components are classified into categories of spatial, energy, information and material. Even though some interaction definitions such as spatial and information interfaces correspond one-to-one to the definitions of *spatial* and *control and* *communications* interfaces, other descriptions show slight differences from Sanchez's study [16, 51]. Unlike Sanchez [51], Pimmler and Eppinger [16] do not cover structural connections or environmental effects between connections explicitly in their study. In addition, Sanchez [51] describes energy or material exchange between components under the same category (transfer interfaces), while Pimmler and Eppinger [16] distinguish the two by separating them into different interface categories called energy and material interactions.

Taking a step further from Sanchez [51], Pimmler and Eppinger [16] propose a system to quantify the interaction importance in the architecture. In their study, they acknowledge that some connections can be relatively important compared to others and some connections are desired while some other might be detrimental [16]. They employ a rating system to assign weights to connections on a five-point scale ranging from +2 to −2 [16]. Their system is proposed to quantify the comparative necessity of each interaction [16]. As described in their three-step method, after decomposing the product into its elements (whether in structural or functional terms), all information on connection properties and their relative importance are presented in an interaction matrix, which is clustered into chunks based on a strategy determined by the engineering design team [16]. They put their method into practice with a case study of an air conditioning system. Acquiring interaction information from design engineers of the company, they construct an interaction matrix that is used to cluster the components into chunks. The result of their analysis is in accordance with the two existent chunks utilized by the company, revision of a third chunk is made, and a fourth chunk of controls/connections has been added to the system architecture.

Their method simplifies product decomposition analysis and provides additional insights about the overall architecture. However, use of the information obtained from the design engineers as the starting point introduces bias when determining the relative importance of the connections. Furthermore the interface types and their explanations are limited to the product chosen for analysis. The limitations of their study make these definitions case-specific and insufficient for universal application.

The next study that considers defining structural connections within the product architecture is Ericsson and Erixon's [18] work on modular product platforms. Their study discusses how modular design can help reduce the lead times in product development and introduces a method of Modular Function Deployment (MFD) to find optimal modular design [18]. Their method includes five steps starting from defining customer requirements, then selecting technical solutions, generating module concepts, evaluating these concepts, and, lastly, optimizing the modules [18]. In the fourth step of their method, they evaluate the module concepts and point out the essential importance of connections between modules [18]. For this step, they find it beneficial to define different types of interfaces such as fixed, moving and media-transmitting interfaces [18]. Generic definitions are used to explain these interface types. Fixed interfaces are used to physically connect modules and transfer forces, whereas moving interfaces are used to transfer rotating or alternative energy. Lastly, media interfaces are used to define the transmission of fluids or electricity between modules [18]. Even though there are several different types of connections within these categories, only two of the defined interface types are used throughout the study: Geometry (G) and Energy (E) [18]. In their case study of a vacuum cleaner analysis, they point out these two specific connection types and base their decision of “chassis based assembly” on the high concentration of G type interfaces in the chassis module [18].

Even though there have been several notable studies on modularity and interface characterization, Ericsson and Erixon [18] do not refer to any of these previous studies. Similar to the previous studies, their study falls short on providing a standardized interface type classification system that can be applied to a variety of different product types. The definitions of interface types and their classification do not quantify the connections between components; they prefer to emphasize the optimization of modularity by focusing on each module individually rather than the connections between these modules, which relies on the individual expertise of the design engineers.

Another study that considers the structural connections between components is done by Sosa et al. [52], which compares the coupling between product architectures and development organizations from integral and modular development perspectives. The study focuses on complex systems for encompassing both modular and integrative systems [52]. Decomposing both product architectures and organization structures, they compare the design interfaces with team interactions [52]. During product architecture decomposition and interface determination, they define five different types of interfaces [52]. The definitions build on Pimmler and Eppinger’s [16] work adding a *structural interface* type, which defines the requirement of transferring loads or containment between components [52]. Later in their study they categorize the interface types into two major groups called spatial-type and transfer-type interfaces, where spatial-type interfaces only include spatial relations between components while transfer type interfaces include all the rest of the interface types except information interfaces [52].

Even though they explain different types of interfaces in the product architecture, their study does not use these definitions effectively. The information on the interface characteristics is only used statistically to show that the development teams have stronger preference to use spatial type interfaces [52]. Although preference of spatial interactions over types is an important finding, it provides a limited amount of information considering the other types of interfaces defined at the beginning of their study. More detailed analysis of the preferences among the other interface types (and maybe a preference ranking) and the possible reasons for these preferences among the design engineers might have shed more light into product architecture design and widened the scope of the study.

Jarrat et al. [17] take interest in the topic of design change and investigate the complexity and change propagation in systems in their study. Describing the engineering change process and its impacts on products, they find out that these changes can remain local as well as propagate through the system. They point out that when one of the connections between components needs to be changed, then the complexity of the architecture increases substantially [17]. They call this phenomenon "interface-overlapping change" [17]. Reviewing some previous studies (Pimmler and Eppinger [16], Pahl and Beitz [41], and Suh [41]) in addition to the interviews with engineers and designers, they introduce a list of "linkage" definitions that represent structural connections in the system [17]. A team of design engineers (consisting of four members) are chosen to breakdown and analyze a diesel engine to determine the linkages in the system [17]. After some elaboration, decomposition of the engine is reduced to 26 major assemblies for simplification purposes. After initial suggestions of mechanical, spatial, thermal and electrical linkages, the design team creates their own comprehensive list of linkages [17]. Their list includes eight different connections: Mechanical Steady State (Ms), Mechanical Dynamic (Md), Spatial (S), Thermal Steady State (Ts), Thermal Dynamic (Td), Electrical Signal (Es), Electrical Earth (Ee), and Electrical Dynamic (Ed) [17]. Some of the existing linkages, such as fluid flow linkage, are excluded from the investigation due to perceived lack of importance [17]. Based on the review of a lead designer, the scope of the product teardown is reduced even more for simplification purposes and, to minimize the duration of the exercise, the number of analyzed connections in the system is reduced even more [17]. The team rates the two aspects of every linkage: likelihood of change propagation and impact of change propagation based on a 0 to10 point scale [17]. The opinions of design engineers about the most and least design changes in the system are validated by analyzing the past change occurrences of the product of interest [17]. No other measure is introduced to evaluate the data collected from the exercise. Only 11 out of 31 system changes were captured by consulting the engineering team [17]. The high abstraction level of the analysis is shown as the reason for the low success rate in the study [17].

The results of their investigation remain focused on the applicability and the benefits of using a DSM and investigating the effects of change propagation by product teardown. As there is a lot to critique about the method for the study, from the interface classification representation perspective, it can be said that the study is case specific. Use of the engineering opinion of a team of four engineers for determining the definition of the linkages limits information about the product architecture to the limits of expertise and experience of the selected engineers.

In addition to application purposes, some studies focus solely on developing a standard language to be used to describe product architectures [53]. One of the first studies to develop a standard language to document the product architecture by focusing on the interface types from a structural point of view has been published by Bettig and Gershenson [9] in 2006. Aiming to develop a design database that can be used and maintained in CAD/PDM environment, they question how interfaces should be represented and what information is important to represent [9]. They refer to some of the previous studies like Sosa et al. [52] and Otto and Wood [48]; however, when determining the interface types, they prefer to start with a current implementation of their industrial partner [9]. Their recommended interface classification includes: attachment, transfer, control and communications, power (electrical), spatial, field, and environmental interfaces [9]. Influenced by the interface classification system in Design Theory [38], which introduces spatial, energy, material, and information types, they decide to narrow down their list to the four categories of attachment, control and communications, transfer, and field interfaces. Even though, they are not explicit about how they decided on these specific interface types, they provide a comparison of the three different categorizations and how each type corresponds to each other. A partial hierarchy chart is represented in the study through description of the attributes for each interface type. Lastly, they instantiate their system with those interface types and CAD models in their design database.

In a more recent study, Dobberfuhl and Lange [8] focus on defining the ideal number of modules for a product and focus on intra-modular connections. The authors refer to the previous studies of Bettig and Gershenson [9] and Gershenson et al. [53].

They extend Bettig and Gershenson's [9] work to identify seven different types of interfaces where they eliminate a separate classification for electrical connections and add in a new type for user interfaces (U), but their explanation falls short. In their study, they use the defined interface classifications effectively to represent the connection properties on their "interface matrix" [8]. The interface matrix is used to visualize all different types of interfaces between two components in one setting [8]. Their study also acknowledges the complexity of the connection increases as the variety of connections between two components increase; however, they assume that all different kinds of connections have the same importance (i.e., attachment vs. communications and controls) [8]. Dobberfuhl and Lange [8] use one interface DSM to represent all seven of their structural interface types to represent the existing relationship between two components without including the number of structural connections [8]. Indirect connections within the system are also not mentioned in their study but not evaluated.

The latest study on using structural approach in interface classification is published by Ariyo et al. [54] in 2010. They mainly focus on using a modularized approach to develop connectivity models of products for effort reduction [54]. The authors aim to minimize the connectivity model building efforts by assigning different modules to different groups to work in parallel. Connectivity modules include information on the architecture, interface and standards, which is represented in partitioned DSMs [54]. The interface information provided in their study lists four types of interfaces, namely: mechanical steady state, spatial, electrical signal and air flow [54]. They base their interface types on the previous studies of Locklegde and Salustri [55] and Jarratt et al. [17]; however, the limited description of their interface types points to a case-specific definition to be able to simplify the parallel connectivity modeling exercise done by the subjects during the study. Consequently their study fails to provide a standardized interface classification system for use in other products.

Sanchez [48] 1994	Pimmler and Eppinger [16] 1994	Ericsson and Erixon [18] 1999	Sosa et. al [49] 2000	Jarrat et al. [17] 2004	Bettig and Gershenson [9] 2006	Dobberfuhl and Lange [8] 2010	Ariyo et al. [51] 2010
Attachment Interfaces	***Spatial Interactions:***	***Fixed Interfaces:*** *only connect modules and transmit forces (such as geometry denoted as G)*	***Structural:*** *indicates a requirement related to transferring loads, or containment*	***Movement:*** *Translational (LRM) and Rotational (RRM)*	***Attachment Interfaces:*** *mechanical connection to restrict relative motion and transmit force*	***Attachment Interface (A):*** *physically connections*	***Mechanical Steady State (M):*** *relations of physical contact*
Spatial Interfaces	***Material Interactions:*** *material exchange*	***Moving Interfaces:*** *transmits energy in forms of rotating or alternating forces (such as energy transmitting denoted as E)*	***Spatial:*** *indicates a requirement related to physical adjacency for alignment, orientation, serviceability, assembly or weight*	***Spatial:*** *Proximity (P) and Alignment (A)*	***Transfer Interface:*** *intends to convey energy, material, or signal as through mechanical means*	***Spatial Interfaces (S):*** *module boundaries*	***Spatial (S):*** *important interactions of adjacency and orientation*
Transfer Interfaces: *defines the allowable flows of materials or power from one component to another component*	***Energy Interactions:*** *energy transfer between two elements*	***Media Interface:*** *fluids or electricity*	***Material:*** *indicates a requirement related to transferring airflow, oil, fuel or water*	***Material:*** *Human (HM), Gas (GM), Liquid (LM), Solid (SM), Plasma (PM), Mixture (MM)*	***Control and Power Interfaces:*** *transmits electrical power or*	***Transfer interfaces (T):*** *power or media transfer*	***Airflow(A):*** *material (air) exchange between components*
Communications and Controls	***Information Interactions:*** *information or signal exchange*		***Energy:*** *indicates a requirement related to transferring heat energy, vibration energy, electrical energy or noise.*	***Energy:*** *Human (HE), Acoustic (AE), Biological (BE), Chemical (CE), Electrical (EE), Electromagnetic (EME), Hydraulic (HYE), Mechanical (ME), Magnetic (MAG), Pneumatic (PE), Radioactive (NE), Thermal (TE), Strain Energy (SE),*	***Field Interface:*** *transmits energy, material, or signal as an unintended side effect of a module*	***Command And Control Interfaces (C):*** *the state of communication and/or controls between modules*	***Electrical Signal (E):*** *relations of signal transfer*
Environmental Interfaces				***Information:*** *Status (SI) and Control (CI)*		***Field Interfaces (F):*** *the functioning of one component can generate heat, magnetic fields, vibrations or other*	
						User Interfaces (U): *human interaction with features within a module system.*	
						Environmental Interfaces (E): *effect of ambient conditions*	

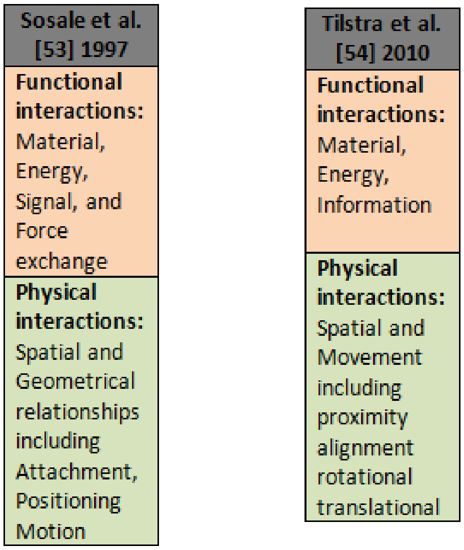
Figure 3.2. Studies with Structural Approach

3.2.3 Hybrid Approaches

The third kind of interface classification system includes hybrid systems that combine functional and structural domains together. Sosale et al. [56] focus on modularity in product architectures considering recycling and reuse issues in products. They approach the problem from both original design and adaptive (redesign) design perspectives; therefore, both functional and structural approaches are considered as relevant for the module determination [56]. Identification of modular boundaries first starts with the decomposition of the product, then identification of the characteristic of the design problem and then interaction analysis [56]. For interaction analysis, they introduce different types of relevant factors that range from objective factors (e.g., life-cycle expectancy, material worth, component worth, recyclability, homogeneity) to functional and structural factors [56]. The list of functional and structural factors can be seen at Figure 3.3. After identifying the specific relevant factors, interaction values and factor weight are calculated [56]. These interaction values are determined by the relative relationship strength of the objective in the system and range from 0 to 10 based on whether the relationship is weak or strong in the system [56]. These decisions are made by the design engineers, and the weighted average interaction values are calculated by a formula introduced in the study, which introduces bias to the analysis [56]. They do not base their classification system on any previous study or provide a classification system where they categorize and name the different types of connections as either structural or functional [56]. The provided case study does not have a detailed analysis of the connections or the list of important factors, either. Even though both functional and structural aspects of a product architecture are considered, the interface classification (either functional or structural) proposed in the study does not provide enough information to represent the product architecture effectively to track change propagation.

Tilstra et al. [57] also follow this approach realizing that the functional basis is not sufficient to describe the physical products; they introduce a two-tier classification system and introduce five general structural interaction types where each includes several secondary class functional interaction types. They base their five general structural interaction types of interfaces on the work of Pimmler and Eppinger [16] and add movement interaction [57]. They add in the fifth “movement” type of interface to better understand the interactions between components. As mentioned before, they also propose a secondary function-based classification to their system. The secondary class of flow set is based on the studies of Hirtz et al. [47]. The subtypes related to other interactions are listed in Figure 3.3.

Tilstra et al. [57] also propose interface DSMs to represent the product architectures. They use their High Definition-DSM model to compare the flexibility between different products where they use different layers for each of their interaction types to represent the product architecture [57]. Even though the method provides detailed information about product interactions, it is far from being easy. They explain the method to be applied in two different stages of whole system and subsystems [57]. Within these stages, one should apply ten different steps to construct the HD-DSM of the analyzed product [57]. Considering the necessity of creating the “black box” models for each element and HD-DSMs for each different interaction in each subsystem, the method becomes a daunting task even for a small electro-mechanical product such as the electrical screwdriver (only 32 parts) provided as the case study [57]. They analyze the screwdriver based on flexibility guidelines provided in their study. These guidelines include 24 rules to increase flexibility adopted from previous work [57]. During the analysis, five selected flexibility rules are applied to the given product. The basis for the selection of these five specific rules for analysis is not justified. Even though different types of connections are analyzed separately, neither their relative importance nor the effects of change in the system are mentioned in their study.



Sosale et al. [53] 1997	Tilstra et al. [54] 2010
Functional interactions: Material, Energy, Signal, and Force exchange	**Functional interactions:** Material, Energy, Information
Physical interactions: Spatial and Geometrical relationships including Attachment, Positioning Motion	**Physical interactions:** Spatial and Movement including proximity alignment rotational translational

Figure 3.3. Studies with Hybrid Approach

3.3 A New Structural Approach to Map Design Dependencies

Accurate definitions of interface types are fundamental for a correct understanding of a product’s architecture. As discussed in the previous sections, relatively few metrics for design complexity based on design dependencies exist, and most of them are case-specific or impractical for frequent use. This work proposes a simple classification to evaluate the design complexity of a product that can be calculated directly from DSMs and be generalized for use with different product lines. As explained in previous sections, DSMs represent interactions between modules (or parts) in a product. While some DSMs only refer to the existence or absence of a connection, more detailed DSMs also include information about the nature of the interactions. For instance, the intensity of these interactions can be captured by referring to the number and types of interfaces.

Depending on the purpose of the study, the researcher might prefer to use functional, structural or hybrid approaches. After careful review of the previous studies and seeing the strengths and weaknesses of each approach, structural approach appears to be the best fit for purposes of this work since it minimizes any possibility of bias on deciding the interface type. Building on the previous studies that utilize structural approach, a new classification system is introduced. The definitions of the different types of interfaces are described as follows, and examples for each type of interface can be seen in Table 3-1.

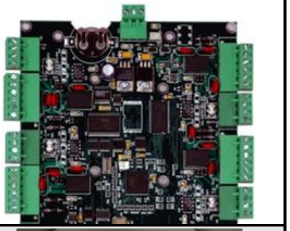
- *Attachment interfaces* are described as the structural connections between two components that require a type of connector. Such as bolts, screws and rivets.
- *Spatial interfaces* define the geometrical and locational constraints of a component with respect to other components. Snap-fit connections as well as any other kind on connection that does not require a connector (e.g., weldment) are considered as spatial type of interface.

- *Transfer interfaces* represent the flow of materials or power between components. Transfers of motion such as torque in different range of rpm's as well as any kind of material flow (e.g., water flow in a coffee maker or in an iron) are examples of this type of interface. Transfer interfaces can be identified by the existence of additional material to facilitate the desired action, e.g., lubricants, bearings, water-tight connections are identifiers of this type of connection.
- *Control and communications interfaces* describe the relationship of signal or information flow between two components in which the state of one component is communicated or controlled by another component. Most electronic components such as circuit boards enable this kind of interface.
- *Field interfaces* represent the interaction between two components in which one component can generate heat, vibration, or magnetic field and this affects the performance of the other component or the whole system due to the field's characteristics. Hard to identify, this interface type mostly relies on material properties of components.
- *Power interfaces* represent the electrical connection between two components unlike communications and controls interfaces. In power interfaces, different currents and voltages are not represented; only simple power on/off relationship is represented.

The first main difference of this classification from previous work [8, 51] is at the physical connections. Sanchez [51] identifies physical and geometrical relations under one interface type, whereas Dobberfuhl and Lange's [8] identify spatial interfaces as geometrical relations and attachments as the physical connections. The differences between two different types of connections are not clear in either study. The proposed classification system clearly differentiates between the two based on the need for an additional connector (e.g., bolt, screw, or rivet). Spatial interfaces represent two different relations between components: the first one is dimension relation (the size and fit of the component within the architecture), the second facilitates a physical connection without the need of an additional connector.

Another main difference in the proposed classification system is electrical connections. While previous studies [8, 51, 53] include electrical connections in transfer interfaces, the proposed work assigns electrical connections to a separate grouping. The use of the electrical connections is more frequent compared to other transfer interfaces and they also differ in terms of required assembly procedures. Lastly, this work omits the definition or use of user interface (U) type, which is defined in Dobberfuhl and Lange's [8] work, since the design dependencies within the product architecture defines the main focus of this study.

Table 3-1. Definitions and Examples of Interface Types

Notation	Definition	Picture
A	Attachment interfaces that physically connect components	
S	Spatial interfaces that determine the boundaries of components in relation to other components	
T	Transfer interfaces represent channeling of power or media from component to component	
C	Communication interfaces are used to define the control relations	
F	Field interfaces that represent any kind of field interaction such as radiation, heat and vibration	
P	Power interfaces represent the transfer of electrical energy	

Intra-modular connections can include more than one type of interface, and a component-based DSM should map all of the different types of interactions. There are numerous combinations of interactions between components based on the notations listed in Table 3-1. All possible combinations of interactions are summarized in Table 3-2. For instance, a commonly found "AS" type of interface defines a connection that entails a snap-fit connection reinforced with the addition of a bolt or a screw. In reality, not all of the combinations may co-exist. This is explained in more detail in Chapter 6.

Table 3-2. Possible Interface Types Investigated

Possible Combinations of Interaction Types					
A	AC	TF	ATC	STP	ASTF
S	AF	TP	ATF	SCF	ASTP
T	AP	CF	ATP	SCP	ATCF
C	ST	CP	ACF	SFP	ATCP
F	SC	AST	ACP	TCF	ACFP
P	SF	ASC	AFP	TCP	ASTCF
AS	SP	ASF	STC	TFP	ASTCP
AT	TC	ASP	STF	ASTC	ASTCFP

3.4 Chapter Summary

In engineering design, product architecture representation is a fundamental issue in order to visualize the product structure and provide fast and reliable solutions to any design problem at hand. Using a standard language to describe the product supports improved communication and correct interpretation of problems amongst engineers. There have been several attempts to provide a unifying language from different perspectives, namely functional, structural and hybrid. This chapter reviews these studies and explores the motives and basis behind each proposed classification. It is seen that, depending on the purpose of the study, whether new product development or building a database to create a repository, the approach might shift from functional to structural or be a hybrid combination of the two. This section also introduces a structural classification system building on the previous structural approaches and provides combinations of possible interaction types acknowledging that there might be more than one type of interface between components, which might increase the dependency between the components of interest. The next chapter focuses on assessing the degree of design dependencies quantitatively by assigning weights to the interface types and calculating a dependency value for each connection.

CHAPTER 4

QUANTIFYING SYSTEM DEPENDENCIES

4.1 Introduction

This chapter summarizes the calculation of system dependencies within a product architecture. A product architecture is a network of components where the effect of a change in any component in the system might create related changes in the whole architecture. *As an addition to this concept, our approach offers a new perspective where the design dependency between components can be seen as the nature of the connection rather than component itself.* As the number of components increases in the system, the number of direct and indirect connections also increases in the product architecture. Increased number of connections leads to an increased number of components that are affected by a single component change and the change propagates to more components in the system due to the indirect connections. The complexity of the system increases with the number of components; hence, the number of connections increases due to the increased change propagation. In order to simplify the calculations for the whole architecture, the product architecture is modeled using an electrical engineering analogy.

4.2 Determining Relative Importance of Connections

Considering all these different possibilities of interactions between components, one can easily realize that not all connections have the same intensity; therefore, not all the connections have the same dependency. This work introduces a novel approach to calculate the strength of design dependencies by using connection information contained in a DSM. The relative strength of direct connections between modules is first calculated using the Module Complexity Score (MCS), see Eq. 4.1, as proposed by Dobberfuhl and Lange [8]. Equation 4.1 takes into account the fact that each module/component might have a different level of complexity while connecting to other modules and that each can carry more than one type of interface at the same time [8]. Therefore, a connection that carries two different types of interfaces (such as AT, an attachment and transfer interface) would have two levels of complexity, hence, twice as much complexity as a connection with one interface type [8]. Respectively, as the number of interface types increases, the complexity of that connection increases.

$$\text{MCS} = (\text{\# of 1's} \times 1) + (\text{\# of 2's} \times 2) + (\text{\# of 3's} \times 3) + (\text{\# of 4's} \times 4) + \\ (\text{\# of 5's} \times 5) + (\text{\# of 6's} \times 6) + (\text{\# of 7's} \times 7) \qquad (4.1)$$

As an enhancement to this equation, we allow interface types to have different weights. As an example, a communication interface, C, might create more dependencies in a product architecture than a simple attachment, A type, interface; therefore, any change in a C type interface may propagate to more components. It is very important to account for the different effects of interfaces in a system for accurate representations of real changes. Input data for the method is derived by incorporating different weights and complexity levels. This data is later used to denote the connection intensity weight between modules and can be represented in a product network graph on the corresponding links. Weighted MCS formulation can be seen in Eq. 4.2.

$$\text{w-MCS} = [w_i^*(\#\text{ of 1's} \times 1)] + [\sum w_i\,(\#\text{ of 2's} \times 2)] + [\sum w_i\,(\#\text{ of 3's} \times 3)] + [\sum w_i\,(\#\text{ of 4's} \times 4)] + [\sum w_i\,(\#\text{ of 5's} \times 5)] + [\sum w_i\,(\#\text{ of 6's} \times 6)]$$

where: w_i=weight of interface type and I = A, S, T, P, C or F (4.2)

The relative weights of different types of interactions are determined based on the frequency of occurrence of the different types of interfaces by using a weighting function. By using the non-linear weighting function given in Eq. 4.3, weights that range between 1 to10 are assigned (to determine an upper and lower level to weights) to different interface types based on their frequency of occurrence. A non-linear weight function is preferred for calculations since it captures the nature of the input data (frequency of occurrences of interface types). W(α) represents a weighting function defined over the interval [0,1], which normalizes the statistical information, by the frequency of occurrence α:

$$W(\alpha) = C_1 e^{-C2\alpha} + C_1 C_2 e^{-C2\alpha} \tag{4.3}$$

where constants C_1 and C_2 are estimated from the desired sensitivity that characterizes the interface type-occurrence relationship. The interface type-occurrence relationship suggests that commonly employed interfaces are preferable over the less used ones due to economies of scale and additional compatibility benefits throughout the architecture.

4.3 Modeling the Product Architecture

Calculating the cumulative effect of direct and indirect connections in the product architectures brings computational difficulties. To overcome these difficulties this study uses an electrical engineering algorithm and proposes modeling the product architecture as an electrical circuit where the connections between components are represented by resistors. This section elaborates on the electrical elements and the modeling approach.

4.3.1 Electrical Engineering Concepts

Electrical components work in accordance with basic circuit theory. These elements define the relationships between fundamental electrical parameters such as current and voltage. Other disciplines such as bioengineering and mechanical engineering also use analogies of circuit theory [58], in which "circuit elements" define relationships between other fundamental parameters. This study aims to extend these analogies to product design and to show that they are not only applicable but also very beneficial in terms of calculating design dependencies in product architectures.

DeCarlo and Lin [58] define a circuit as an energy or signal/information processor. Similar to a product architecture, an electrical circuit includes interconnections of circuit elements or devices such as battery, resistor, capacitor, inductor, and operational amplifier. Resistance is the opposition to the flow of current through an electrical element and determines the amount of current flowing through the circuit across a given voltage; therefore, the higher the resistance, the lower the current that flows. An ideal resistor satisfies Ohm's Law, $V = I \times R$, which states that the potential difference between two points in an electrical circuit is proportional to the current flowing between those two points and the proportionality ratio given by the resistance. Let us consider the simple electrical circuit given in Figure 4.1. This simple circuit includes a voltage source and a resistor, across which electrons to flow as current (by convention, current flows in the direction opposite to electron flow).

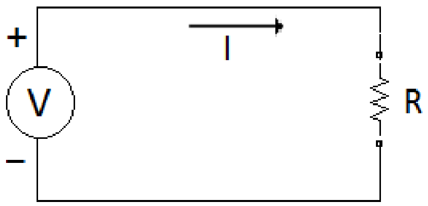


Figure 4.1. Simple Electrical Circuit

Capacitors are another type of simple electrical device that store electrical charge. Inductors store energy in the form of magnetic flux. The circuit analogy being presented in this work could be extended to include various additional types of electrical circuit elements; however, that remains beyond of the scope of this research and is left as future work. This work focuses solely on the resistor element.

4.3.2 Electrical Circuit Model

As mentioned in Chapter 1 and shown in Figure 1.1, it is possible to consider a product architecture as a network of components [52]. This research employs a similar approach to the problem by proposing a novel electrical circuit analogy. Similar to an electrical circuit, the components within a product also form a network wherein all subsystems connect to each other directly or indirectly. This work considers components having relationships with other components of different strengths. Unlike most DSMs where interactions are represented in a binary nature, this study works with an interface-based DSM where connections have calculable values. This shift in perspective is reflected by the analogy of an electrical circuit where weighted-MCS values represent the concept of electrical conductance. This approach proposes that, as an interface between two components becomes more complex (as the weighted-MCS value increases), its ability to "conduct" or "transmit a design change" also increases.

The Electrical circuit analogy illustrates this "change propagation" throughout the product architecture. As mentioned above, in an electrical circuit, the flow of the current is dependent on the resistance within the circuit. Similarly, change propagation in a product architecture is related to the design dependencies between components. Figure 4.2 represents a component change and resulting dependency change between components. In the figure the bars represent the changing connection characteristics between components.

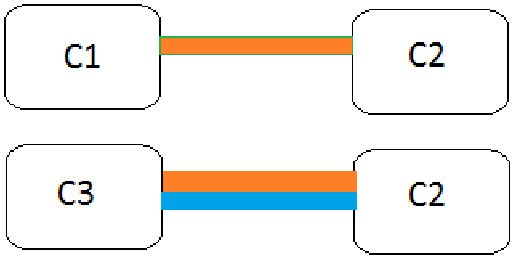


Figure 4.2. Change and Dependency between Components

This work defines design dependencies as characteristics of the interfaces between components. Therefore, a design change in a product architecture is considered as "current" flowing through the electrical circuit. Similar to an electrical circuit in which the resistance of a resistor defines the amount of current passing through it, the characteristics of the connections between components (interfaces) determine the amount of change transmitting throughout the product architecture. If we refer back to Ohm's Law, V = I × R, then the relationship can be described in similar structure in which R = design dependencies between components, I = design change, and V is redesign effort.

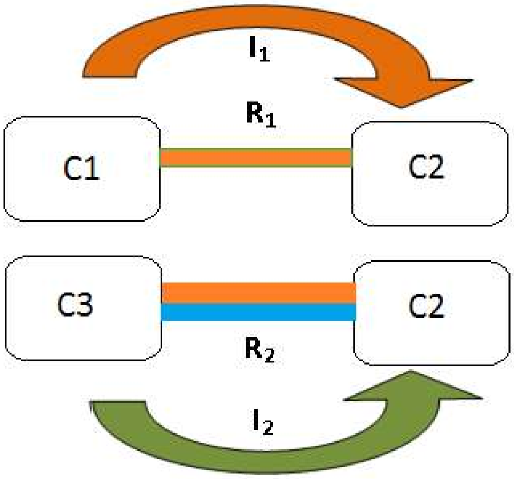


Figure 4.3. Ohm's Law Analogy

Following this analogy, calculating design dependencies based on the given interfaces becomes practical. First, product architectures are modeled as electrical circuits representing all direct connections in the system and then the reciprocal of the weighted-MCS values (named as: change resistance weights) are used as resistance values in these models to represent the complexity of each individual connection. By simulating the product architectures as electrical circuits and running the models with given change resistance scores, the impedance results for any chosen connection are calculated. These impedance calculations represent the overall design dependency for a given interface in the system by incorporating all of its direct and indirect connections.

Calculating design dependency between two components can be explained with a simple example. Figure 4.4 represents a simple module network consisting of four modules. The design dependency between modules (or nodes) M1 and M2 (denoted as W1 in the Figure 4.4) can be calculated by accounting for all connections within the system, translating them into an electrical circuit representation. The electrical circuit representation begins with the network shown in Figure 4.4 and proceeds step-by-step to create the equivalent electrical circuit shown in Figure 4.5.

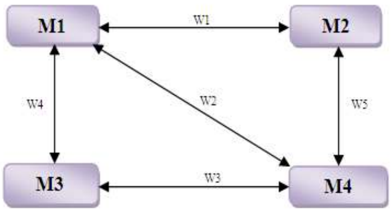


Figure 4.4. Simple Representation of a Four Module Network

As taught in basic electrical circuits, resistors can be connected in series or in parallel. The total impedance can be calculated into single impedance by adding the individual impedances when resistors are connected in series or if the resistors are connected in parallel adding the inverses of the individual impedances. Impedance calculations represented in Figure 4.1 are formulated as follows:

$$\mathrm{WT} = \frac{1}{\mathrm{W1}} + \frac{1}{\mathrm{WX} + \mathrm{W5}}, \qquad \text{where :} \tag{4.4}$$

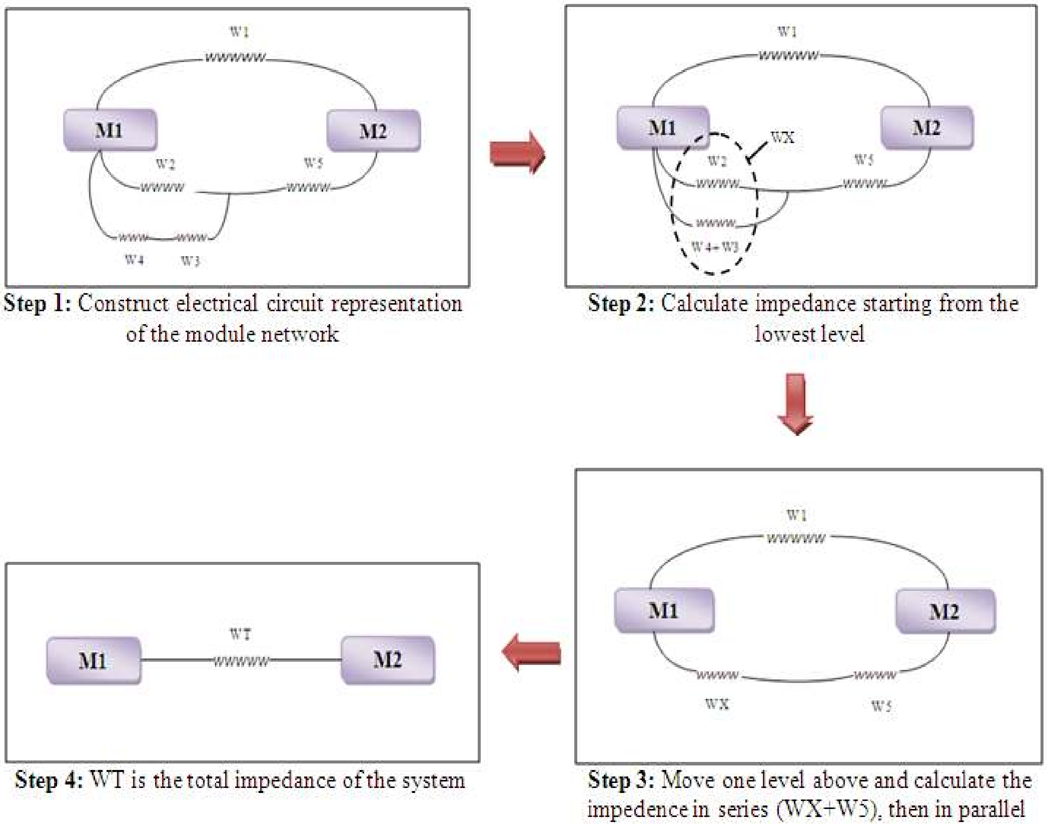
$$\mathrm{WX} = \frac{1}{\frac{1}{\mathrm{W2}} + \frac{1}{\mathrm{W3} + \mathrm{W4}}} \tag{4.5}$$

Substituting Eq. 4.4 into Eq. 4.3 yields:

$$\mathrm{WT} = \frac{1}{\mathrm{W1}} + \frac{1}{\frac{1}{\frac{1}{\mathrm{W2}} + \frac{1}{\mathrm{W3+W4}}} + \mathrm{W5}} \tag{4.6}$$

As seen in Eqs. 4.4 to 4.6, as the number of nodes (i.e., components) in the system increases, the calculations get more complicated even though the individual calculations are very simple.

In the proposed approach, connections between different components or modules are considered as resistors in the electrical circuit. Input data for the system is obtained from the reciprocal values of weighted-MCS in the DSM. Therefore, the result does not include any subjective opinions from engineering design teams, and it provides a reliable measure that can be used to compare different products designed by different teams. In addition, by using only empirical data, the metric reduces the time spent by design engineers when analyzing design dependencies.



Step 1: Construct electrical circuit representation of the module network

Step 2: Calculate impedance starting from the lowest level

Step 3: Move one level above and calculate the impedence in series (WX+W5), then in parallel

Step 4: WT is the total impedance of the system

Figure 4.5. Step-by-Step Impedance Calculation in a Four Module System

As discussed in Chapter 5, the circuit models are constructed by using MATLAB Simulink application package with Simscape extension [59]. The Simscape extension to the program provides a graphical interface to develop dynamic systems and includes a library of a wide range of elements from including electrical, hydraulic, magnetic, mechanical, physical, pneumatic, and thermal to use constructing the models [59].

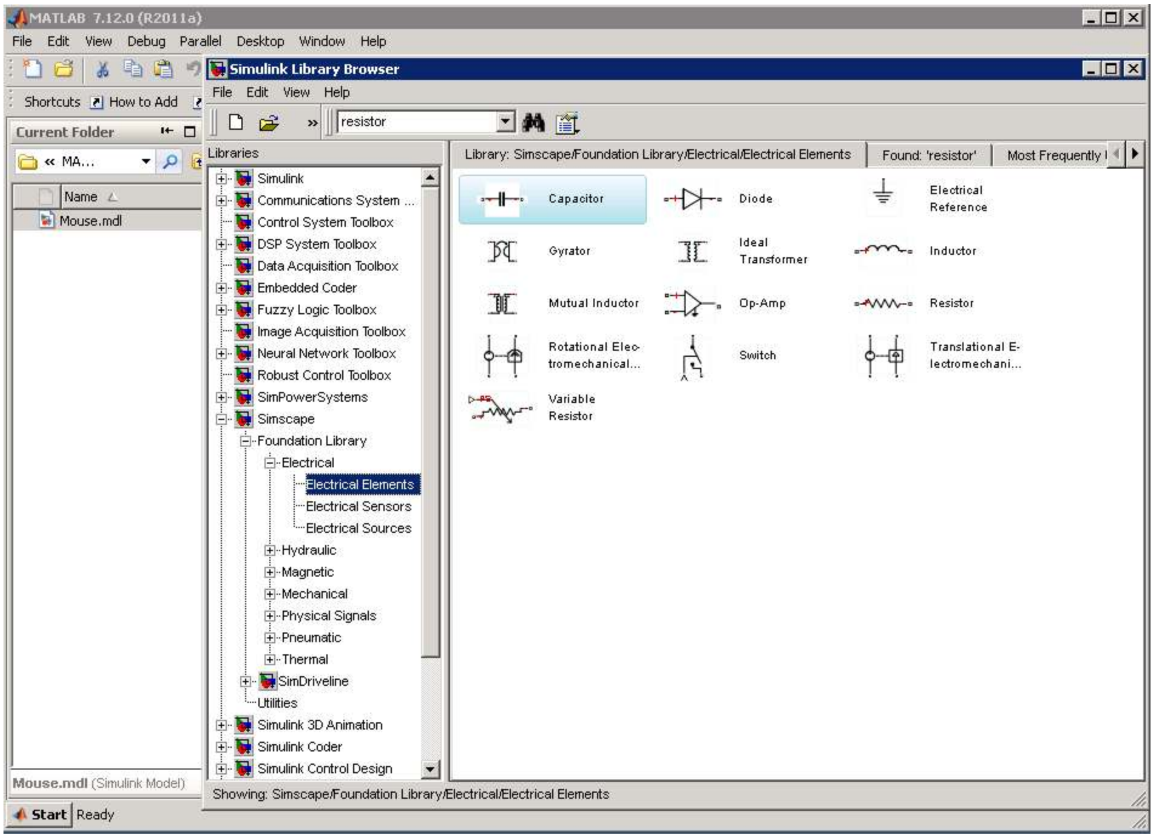


Figure 4.6. MATLAB Simscape Electrical Elements Library [59]

The model is constructed by adding resistors to the Simulink interface. The resistors are connected to each other using the linkages at the ends of the resistors. The linkages are controlled by clicking the mouse and dragging to the desired spot. Unconnected linkages remain red while the completed connections appear as a black line. The example can be seen from Figure 4.7.

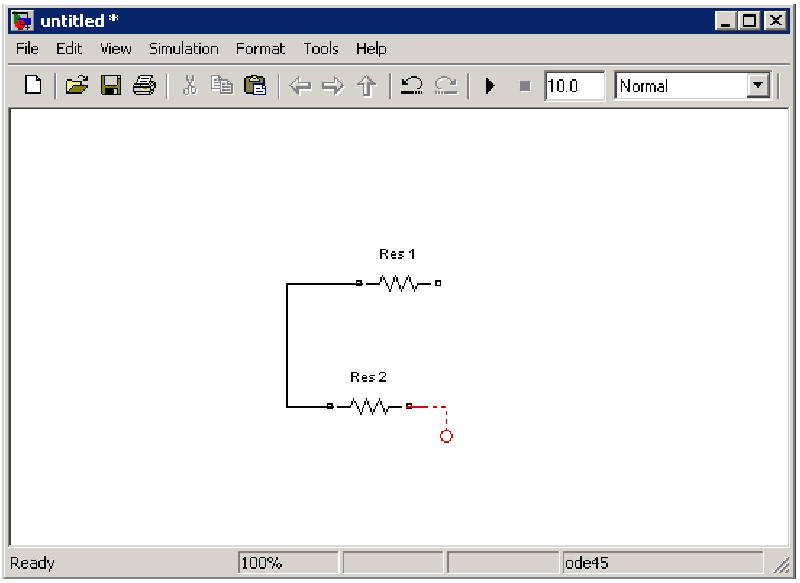


Figure 4.7. Model Construction Steps in MATLAB Simulink [59]

Model construction is an important step of the method, and one should remember that the resistors represent the connections between two components; therefore, correct abbreviations or naming of the model is crucial to avoid mistakes during model construction. In this study resistors (a.k.a. connections between two components) are named by the components that they are connecting. For example, the resistor that represents the connection between Components 1 and Component 3 is named as C1-C3. Using the smaller component number first is recommended since Simulink does not allow name repetitions for simulation elements. This way, in larger products, where the number of connections is really high, connection repetitions can be avoided.

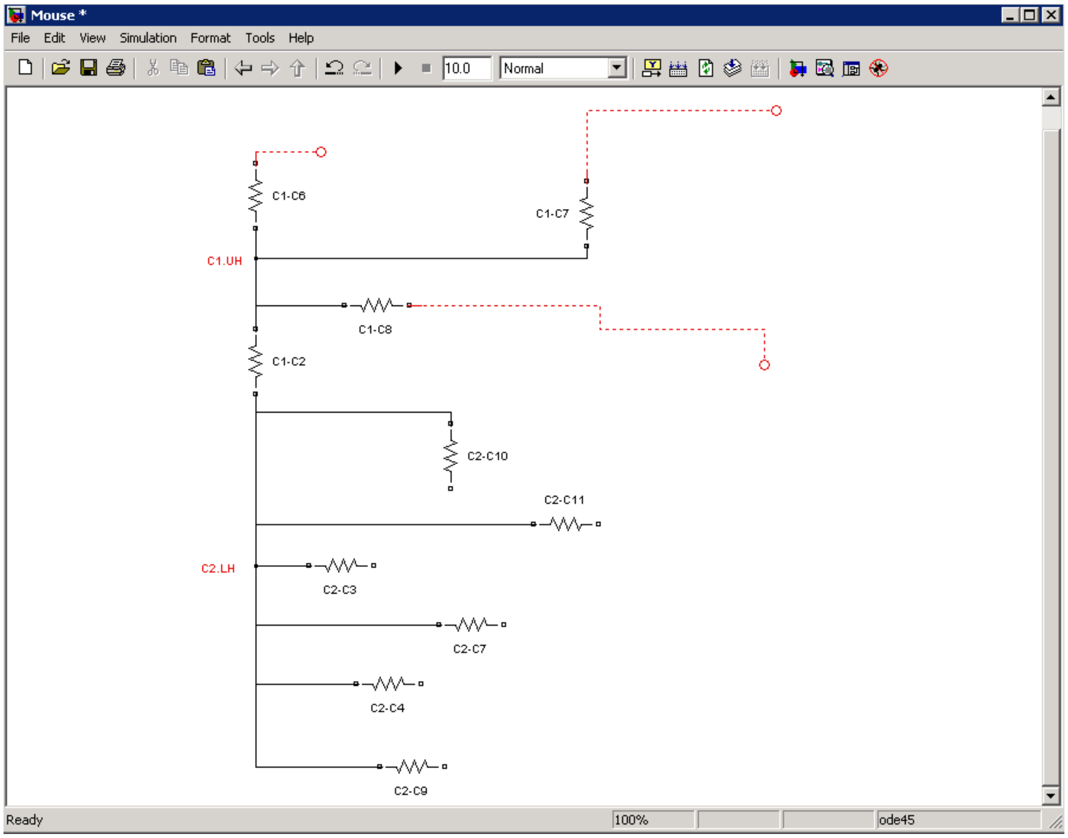


Figure 4.8. Model Construction Example [59]

Another important recommendation for building the model is to keep developing nodes as the model gets more interconnected by accumulating the relevant connections to the same spot. This can be easily understood by following the mouse example illustrated in Figure 4.9. In this figure the two connections of two components for the mouse has been constructed in the circuit model. The resistors representing the connections of Component 1 with other components in the system have been developed and connected to each other. At this point it can be seen that the engineer has preferred connecting one end of all these resistors to one spot, creating a node labeled as C1.UH in red. This label is the abbreviation for Component 1: Upper Housing for the mouse product. By collecting one end of all resistors at the same spot and giving a specific label, the engineer ensures to represent every component in the system. The label and the connections for Component 2—lower housing can also be seen in Figure 4.5. This labeling approach is used to avoid any confusion on the connection points when the model gets more complicated. In addition to completing the connections the network should also include a power source, a resistance measurement element, and a display to the simulation for data collection. These elements can be found in the Simulink library [59]. The equivalent resistance measurements are obtained by connecting the nodes of the resistance measurement element to the components of interest.

4.4 Chapter Summary

Determination of system dependencies in a product architecture provides essential information for design engineers about the nature of the product structure. Highly dependent systems (integral architectures) may amplify the any design changes in the system and create a domino effect of change propagation to other components in the system. It has been discussed that estimating the change propagation in a product architecture is very difficult and even most experienced engineers cannot define them 100% of the time [6]. This chapter introduces a new method to quantify different levels of the design dependencies in product architectures based on the nature of the connections between components. In addition to determining the individual connections, the method also considers the indirect connections in the system. The method employs an electrical engineering analogy to model and calculate the overall effect of individual changes in whole architecture. The next chapter includes step-by-step guidelines to apply the proposed method to analyze the product architecture by using the mouse product as an example. Chapter 6 includes the analysis done on twenty-one different products, and Chapter 7 presents conclusions, limitations, and future work.

CHAPTER 5

A NEW METHOD FOR DETERMINING CHANGE SENSITIVITY IN PRODUCT ARCHITECTURES

5.1 Introduction

This chapter introduces a method to determine the change sensitivity in product architectures. The proposed method includes three main phases. Phase 1 is the “Data Collection” phase and includes two steps: (1) dissecting the products to a pre-determined level and (2) collecting structural data. Phase 2 requires calculation of the connection values, and in Phase 3 the product structure is modeled. To maintain consistency and comparability amongst results in any study, it is fundamental to follow standardized procedures. The phases are explained in detail in this chapter. Even though the phases are sequential in execution order, the two steps of Phase 1 should be handled in parallel to avoid any information gaps. A Gantt chart explaining the proposed sequencing of phases is illustrated in Table 5-1.

Table 5-1 Gantt Chart Representing the Recommended Sequence of Phases

Phase	Step	Description							
1	1	*Dissect the Product*	🟦	🟦					
	2	*Prepare Interface DSM*		🟦	🟦				
2	1	*Calculate Weighted-MCS*				🟪			
	2	*Compute Reciprocals*					🟪		
3	1	*Build Electrical Circuit Model*						▨	
	2	*Analyze the Output*							▨

5.2 Phase 1: Data Collection

The method begins with data collection. If done correctly, the data collection ensures the consistency of the results gathered for the analysis. In the proposed method, the data are collected from an actual product by disassembly. Even though technical drawings and a bill of materials might be available, they may not reveal enough detail about the nature of the connections and might not be consistent with the needed granularity of disassembly. Collecting data from actual products has been recommended in previous studies and called product archeology by Ulrich and Pearson [60]. This approach is preferred since it does not require any authorization or confidential information from the manufacturer of the product of interest.

The data collection phase of the method has two main steps: (1) product dissection and (2) product architecture representation. These two steps together facilitate transferring the physical product structure into a digital representation. Once the product architecture is digitized accurately, any modification on the architecture can be analyzed without making any actual changes to the physical product. This saves time and money in terms of redesign efforts and prototype development.

5.2.1 Step 1: Dissect the Product

The product dissection step entails disassembling the actual product into its sub-systems and components. During this step, data are collected by reverse engineering the actual product and removing its parts one by one. Based on the purposes of the analysis, the level of dissection should be determined first and held constant throughout the process to ensure consistency of the analysis. In some cases, the disassembly of a part might become extremely difficult. Electronic parts such as printed circuit boards are such an example and can be considered as a single component to simplify the process. Not listing connecting elements (e.g., nuts, bolts, screws, cables) as components is also strongly suggested to keep the component matrix at a manageable size and to avoid connection duplications in the data.

Data collection should start from the outer most layer of the product and slowly move to inner sections as each layer is removed. Taking pictures at the beginning and throughout the process is beneficial. An example dissection sequence of an optical computer mouse is illustrated in Figures 5.1. In almost all cases, dissection would start by examining any covers and/or snap-fit parts. The process continues with detaching the outer shell of the product. As shown in Figure 5.1(c) and Figure 5.2, some products might have battery covers and small transmitters that might be housed on the product. These should be noted, and part names and types of connections should be recorded in the component-to-component interface DSM.

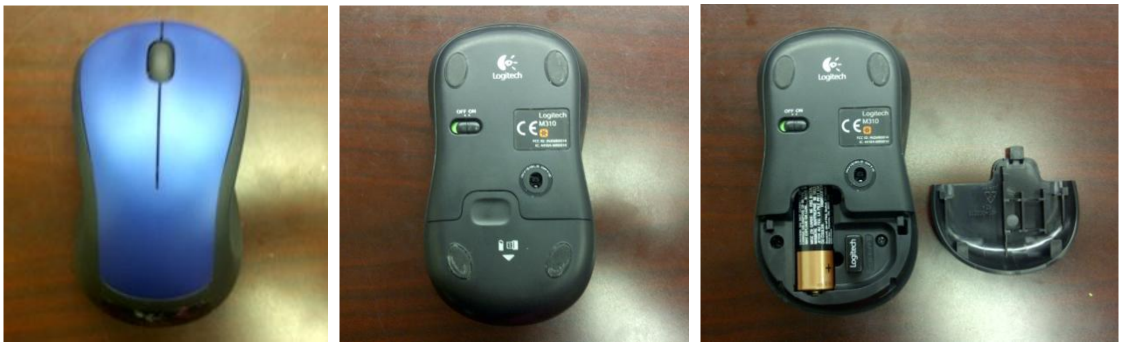


Figure 5.1. (a), (b) and (c): Example Product Dissection Steps

Figure 5.2. Bluetooth Transmitter for the Mouse

The next step of the dissection process involves detaching the outer housings of the product. There might be several different ways that the housing might be attached, e.g., screws, snap-fits. Even though snap-fit connections do not require any additional connector (nuts, bolts, and/or screws), they heavily rely on the geometrical properties of the connecting parts. Snap-fit connections are mostly parts made out of plastic material and are produced by injection molding [61]. This is a relatively cheap manufacturing process and reduces assembly time compared to other connection types [61].

The outer housing of a product may also depend on screws for attachment. In most cases, if the product includes any kind of battery or outer cover then these screws may be hidden for aesthetic purposes. Similarly in the example in Figure 5.1(c), two screws are used to attach the upper and lower housing of the mouse including snap-fit connections. This information is translated into the component-to-component interface DSM as shown in Table 5-2, where "A(2)" represents the two screw attachments and "S" represents the spatial interface from the snap fit.

Table 5-2. Sample Interface Representation

Mouse Interface DSM	Upper Housing	Lower Housing
Upper Housing		A(2)S
Lower Housing	A(2)S	

It should be noted that the DSM is generated by dissecting the product into its components and analyzing the direct connections between every component or subsystem. Therefore, no matter how many different kinds of interface types a component has, the nature of the connection between two components is considered to be the maximum number of different interface types connecting the components. This makes the DSM symmetric across the diagonal of the matrix.

As the dissection process continues, the inner parts of a product are revealed. At this stage, the designer might observe multiple parts with multiple connections and geometrical relations. Geometrical relations might seem subtle and easy to overlook. However, they are an important part of a product structure and might be affected by a design change in any of the related parts. A better explanation can be seen in the example in Figure 5.3.

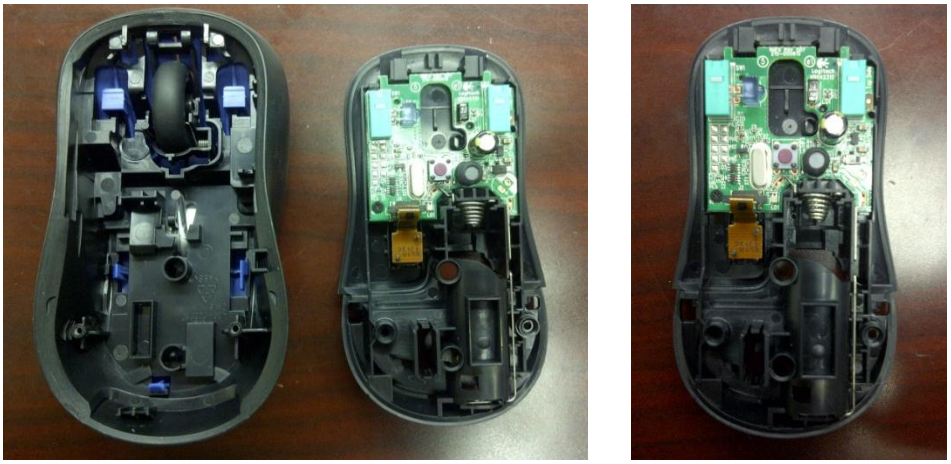


Figure 5.3. (a) and (b): Inner Parts of the Mouse

As can be seen in Figures 5.3(a) and (b), when the outer housings are detached, a printed circuit board (PCB) is revealed. Even though the PCB seems to be mainly connected to the lower housing (with a screw in addition to a snap-fit connection), the upper housing has been molded in a way to accommodate the size and volume of the PCB. An untrained eye also might miss the small detail that the PCB has an opening in the middle to accommodate the wheel since they are attached to different sides of the housing (see Figures 5.3 and 5.4). Therefore changes to the PCB might affect the upper housing as well as the lower housing in addition to the wheel. Paying close attention to such details during the data collection phase increases the reliability of the analysis.

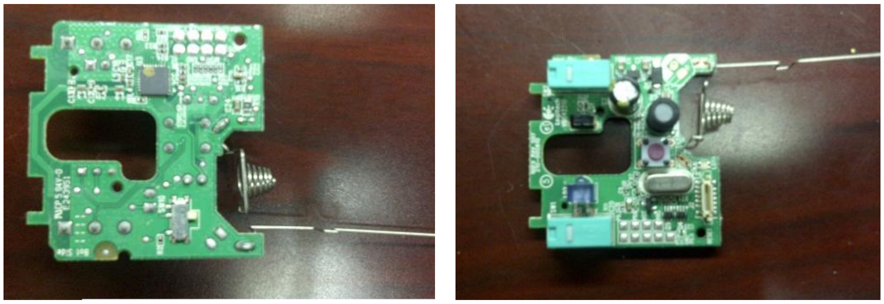


Figure 5.4. (a) and (b): PCB

As the remaining parts of the mouse are dissected, an LED light extension rod between the blue cover and the upper housing is revealed. These parts can be observed in Figures 5.5(a) and (b).

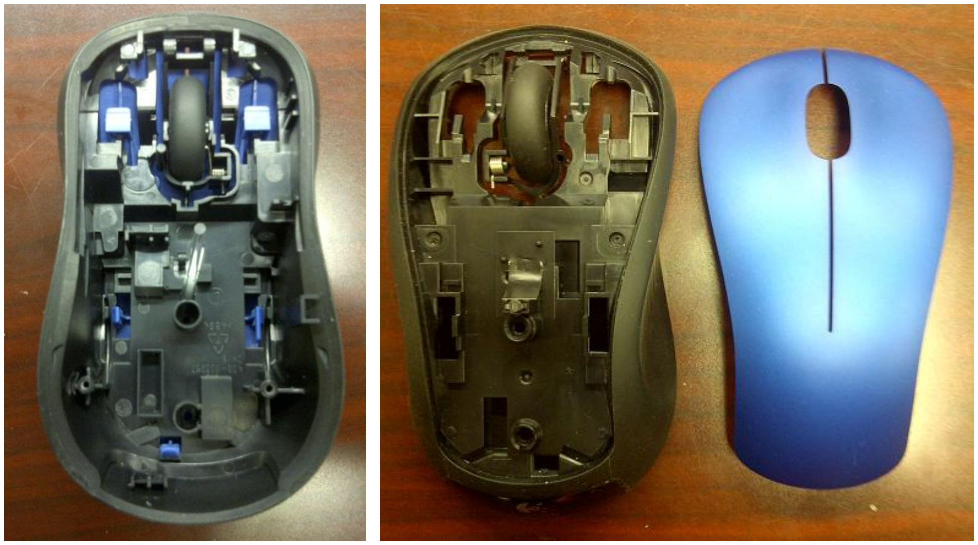


Figure 5.5. (a) and (b) Dissection of the Remaining Parts

5.2.2 Step 2: Prepare Interface DSM

The second step of the data collection phase defines and records the types of connections between components. Since it occurs concurrently with the dissection process, the DSM will also be complete at the end of the dissection process. The DSM is constructed by naming all parts and listing them in the matrix rows and columns. Diagonal cells of the matrix are irrelevant since they correspond to the same part. Each connection and the nature of the connection needs to be recorded to the matrix.

As explained in Section 3.3, there are six types of interfaces (A, S, T, C, P, F), and components might have:

- A single type of interface that only occurs once;
- More than one type of the same interface such as two attachments: A(2);
- Combination of interfaces: AT or APC; or
- Both A(2),APC

between them. If there are no connections between two parts, then the corresponding box in the matrix is left blank. Following this procedure, the DSM for the mouse is given in Table 5-3 as dissection is completed. The next phase includes determination of the connection values based on the types of interfaces identified during Phase 1.

Table 5-3. Complete DSM for the Optical Mouse

MOUSE		1	2	3	4	5	6	7	8	9	10	11
Upper Housing	1		A(2),S			S	S	S	S			
Lower Housing	2	A(2),S		AS	S			S		S	A(4),S	S
PCB	3		AS				S	F	S	P	PC	S
Battery cover	4		S					S				
Blue Top Layer	5	S					S		S			
Wheel	6	S		S		S						
Nano Receiver	7	S	S	F	S							
LED Stick	8	S		S		S						
Battery Contacts	9		S	P								
Sensor	10		A(4),S	PC								
On/Off Button	11		S	S								

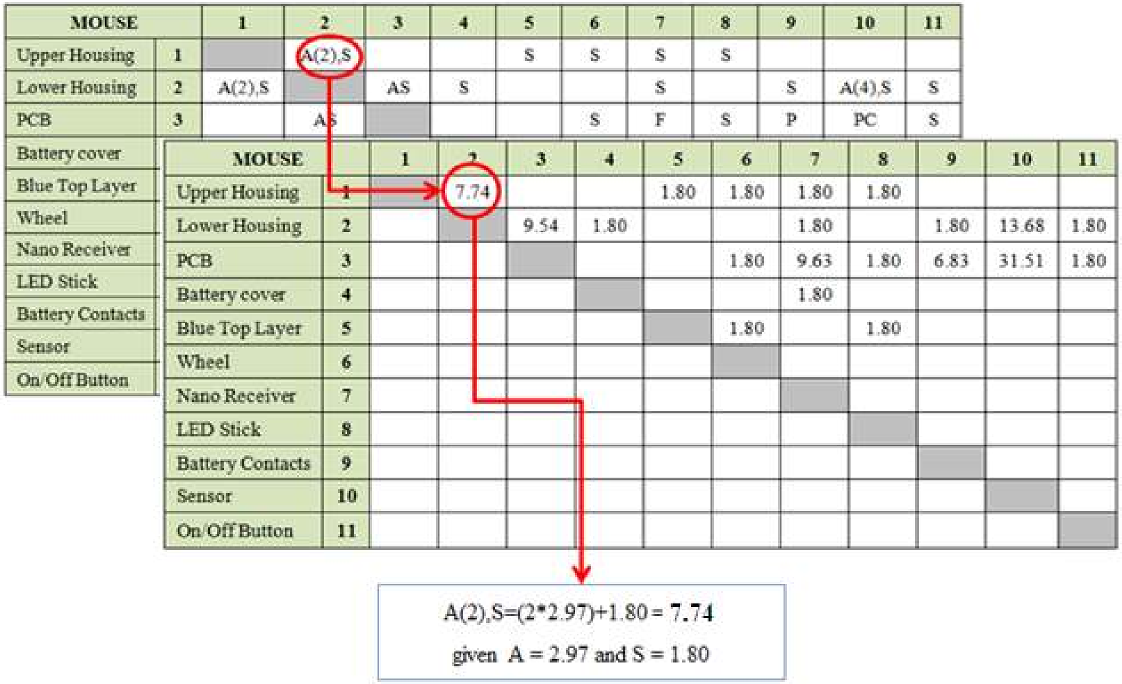
5.3 Phase 2: Determining Connection Values

As can be seen in the DSM in Table 5-3, the connection types between components vary and are used as input values in the circuit models for the product architecture. These input values are calculated in two steps.

5.3.1 Step 1: Calculate Weighted-MCS

First, connection values are determined by using the weighted Module Complexity Score equation (w-MCS) (see Eq. 4.2). The weights for individual interface types are calculated based on the frequency of occurrence of that interface type in the data set. To calculate the relative weights for the interface types, all the products in the analysis set should be dissected and necessary statistical data should be processed. Then these values are used in a non-linear weight function (see Eq. 4.3). Two constants C_1 and C_2 are estimated to characterize the interface type-occurrence relationship. Details on the calculations of the interface type weights are discussed extensively in Section 4.2. An example for the w-MCS calculation can be observed in Table 5-4 with complete MCS values for the optical mouse.

Table 5-4. Weighted-MCS Calculations of the Mouse



MOUSE		1	2	3	4	5	6	7	8	9	10	11
Upper Housing	1		A(2),S			S	S	S	S			
Lower Housing	2	A(2),S		AS	S			S		S	A(4),S	S
PCB	3		AS				S	F	S	P	PC	S
Battery cover												
Blue Top Layer												
Wheel												
Nano Receiver												
LED Stick												
Battery Contacts												
Sensor												
On/Off Button												

MOUSE		1	2	3	4	5	6	7	8	9	10	11
Upper Housing	1		7.74			1.80	1.80	1.80	1.80			
Lower Housing	2			9.54	1.80			1.80		1.80	13.68	1.80
PCB	3						1.80	9.63	1.80	6.83	31.51	1.80
Battery cover	4							1.80				
Blue Top Layer	5						1.80		1.80			
Wheel	6											
Nano Receiver	7											
LED Stick	8											
Battery Contacts	9											
Sensor	10											
On/Off Button	11											

A(2),S=(2*2.97)+1.80 = 7.74

given A = 2.97 and S = 1.80

5.3.2 Step 2: Compute Reciprocals

After calculating the weighted-MCS values of connections, it is necessary to obtain reciprocals of these values, which represent the "change resistance" of each corresponding connection as discussed in Section 4.3.2. Calculated reciprocal values are listed in Table 5-5.

Table 5-5 Resistance Values for the Mouse

MOUSE		1	2	3	4	5	6	7	8	9	10	11
Upper Housing	1		0.129			0.557	0.557	0.557	0.557			
Lower Housing	2			0.105	0.557			0.557		0.557	0.073	0.557
PCB	3						0.557	0.104	0.557	0.146	0.032	0.557
Battery cover	4							0.557				
Blue Top Layer	5						0.557		0.557			
Wheel	6											
Nano Receiver	7											
LED Stick	8											
Battery Contacts	9											
Sensor	10											
On/Off Button	11											

To obtain accurate results on the change resistance of a given connection, both direct and indirect connections that may exist in the product architecture must be included in the analysis. This is calculated using the electrical circuit model that simulates the propagating effects of design change in the system (see Section 4.3). The last phase of the analysis requires building such a model to determine the change sensitivity of components in the product architecture.

5.4 Phase 3: Modeling the Product Architecture

This section elaborates on modeling the product architecture. As explained in Chapter 1, product architectures can be considered as complex networks that include several direct and indirect connections.

5.4.1 Step 1: Build Electrical Circuit Model

As discussed in Chapter 4, an electrical circuit analogy is used to represent a product architecture given the complications of networks with many components. The models are created by using the MATLAB/Simulink software package to construct an electrical circuit from electrical elements (resistors, capacitors, power sources, etc.) in its user interface database. The connections between components are represented as resistors, while the actual components act like junction points for connections. Figure 5.6 shows the representation of the electrical circuit model created for the optical mouse. The tags labeling the small points in the model represent physical components in the product, which are connected to other components by with resistors.

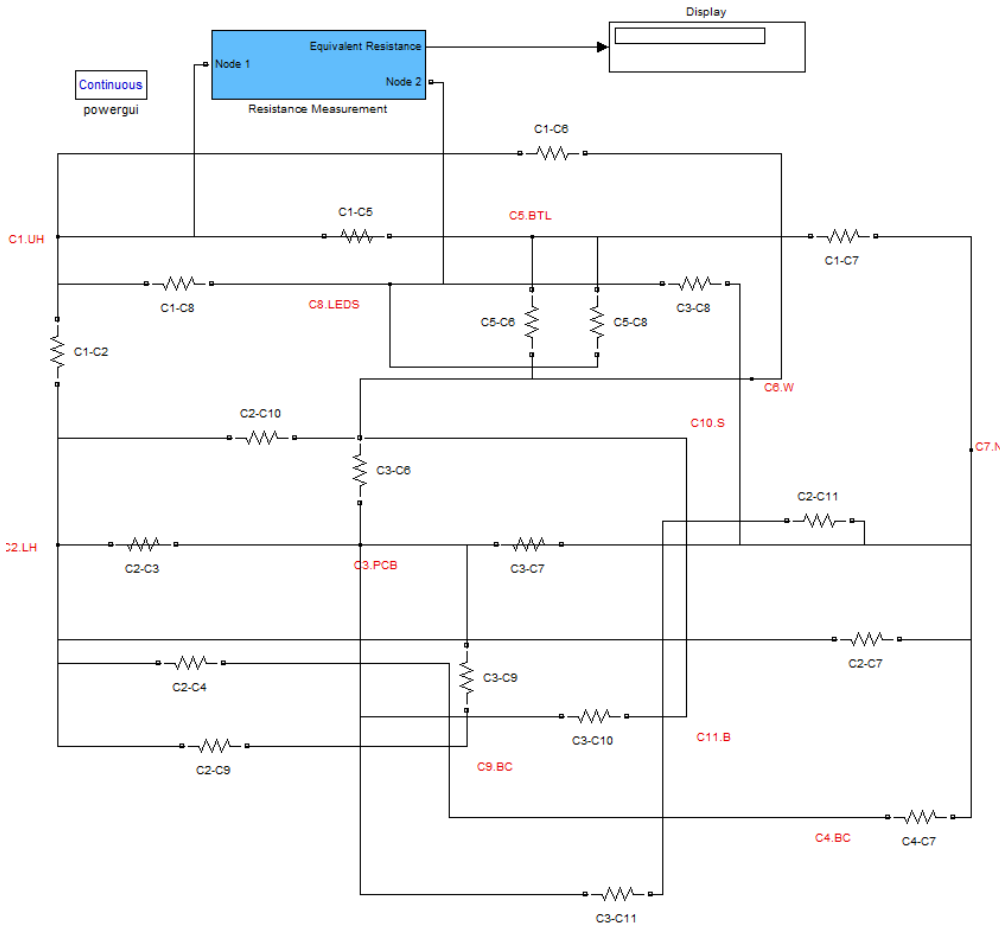


Figure 5.6. Electrical Circuit Model of the Mouse

As discussed in Section 4.3.2, the proposed method prioritizes the nature of connections rather than the nature of the components. The calculated reciprocal values (see Table 5-5) are used as resistance values in the model. To be able to calculate the equivalent resistance between two components, the nodes in the model need to be connected to points of interest, and the model automatically displays the equivalent resistance between those two points in the display box in the model. All electrical circuit models of the analysis set can be found in Appendix B.

5.4.2 Step 2: Analyze the Output Data

Analysis of the output data obtained from the electrical circuit model is the last step for the method during which the engineer reviews results and makes comparisons with other products. During this step, the engineer needs to look out for any inconsistent results that might arise since it might be an indication of an error within the model. Simple errors such as typos or incorrect connections within the electrical circuit model can be observed by paying attention to abnormalities in the output. Output comparisons of very similar products are shown to be beneficial to spot any errors. The raw output data collected for the optical mouse can be seen in Table 5-6. The values in the output data represent the equivalent resistance (minimum resistance value) for that connection, which represents that connection's "resistance to transmit change". Therefore, the minimum value in the matrix represents the connection and corresponding component that is "most likely to transmit change" to other components in the architecture.

Table 5-6. Output Data for Optical Mouse

MOUSE		1	2	3	4	5	6	7	8	9	10	11		
Upper Housing	1		0.091			0.248	0.232	0.145	0.232				0.091	5
Lower Housing	2	0.091		0.039	0.302			0.093		0.140	0.041	0.041	0.039	7
PCB	3		0.039				0.247	0.075	0.247	0.117	0.026	0.026	0.026	7
Battery cover	4		0.302					0.302					0.302	2
Blue Top Layer	5	0.248					0.294		0.294				0.248	3
Wheel	6	0.232		0.247		0.294							0.232	3
Nano Receiver	7	0.145	0.093	0.075	0.302								0.075	4
LED Stick	8	0.232		0.247		0.294							0.232	3
Battery Contacts	9		0.140	0.117									0.117	2
Sensor	10		0.041	0.026									0.0258	2
On/Off Button	11		0.041	0.026									0.0258	2
													0.128	

5.5 Chapter Summary

This chapter presents a novel method to determine the change resistance in product architectures. The method includes three main phases. Phase 1 is data collection and includes two steps: (1) dissection and (2) product architecture representation. These two steps are critical for the consistency of the results; therefore, they should be handled with utmost attention. The first phase captures the structural information of the product architecture in the form of a DSM. After the development of the DSM that maps connections within the dissected product, Phase 2 is used to quantify the relative importance of these connections within the architecture. Using a weighted-MCS formulation and calculating the reciprocals of these values, the resistance value for each connection in the system is obtained. However, the effect of all direct and indirect connections in the product architecture should be included for a complete analysis. For that purpose, an electrical circuit analogy approach is adopted. The third and last phase of the method requires modeling the product architecture as an electrical circuit in which the resistance values calculated in Phase 2 are used as inputs to the model. Electrical circuit models facilitate measurement of the equivalent resistance between any chosen components including the effect of the whole system. Determining connections with high change sensitivity provides information to designers about the components that are likely to transfer design changes the most. This method is applied to several electro-mechanical products in the next chapter and compared with previous methods to validate the proposed approach.

CHAPTER 6

CASE STUDY & ANALYSIS

In this chapter, the method proposed in Chapter 5 is validated by using data from twenty-one different electro-mechanical household products. This chapter includes the statistical information collected based on the interface characteristics of the selected products as well as the individual results from the application of the proposed method with comparison to other products.

6.1 Product Selection

One of the very important decisions for model validation is selecting the correct set of products. Selected products should include the characteristics of the group that they represent. This group of products should be also consistent with each other to provide comparable results. The consistency of results amongst these products is a big indicator of valid modeling. In this work, validation of the proposed model was performed by analyzing small electro-mechanical household products. Low numbers of components and ease of dissection were important factors when choosing this set of products. These characteristics provided consistency during data collection phase.

As mentioned earlier, each interface type includes several different types of connections within it. The products selected for the analysis of the proposed method cover all of the different types of interfaces defined in the study. The complexity level for these products has been kept at the same level, i.e., maintaining one level DSMs. All of the selected products have integral structures with an average of eighteen components.

A set of cordless Durabuilt® power tools has been chosen for analysis (seen in Figure 6.1). Durabuilt tools are known to offer high functionality with low cost. This power tool family has been designed for at-home do-it-yourself projects; hence, they compromise the durability and long life-span of heavy duty power tools for low cost. The functionality for the products in this line has been kept basic for non-professional use. The set of tools include: flashlight, sander, circular saw, and jigsaw. The set also included a drill that was lacking parts due to a previous dissection study; therefore, another drill has been selected for analysis. The varieties of functions the tools provide also cover the interface types defined in this study; hence, the products provide good examples for small mechanical or electro-mechanical household items group.

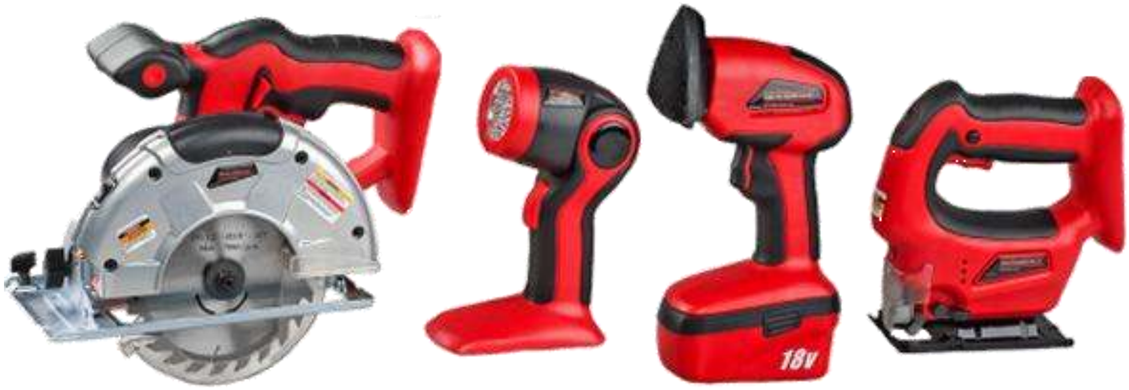


Figure 6.1. Durabuilt Cordless Power Tool Set [62]

Due to the unsuitable condition of the drill included in the Durabuilt toolset, a Black and Decker® DR260B 3/8″ 5.2-amp corded drill/drive, which can be seen in Figure 6.2, has been chosen for dissection. This product provides variable speeds for drilling or driving with a keyless chuck, which can be used to attach either drilling parts or screw heads to provide two different functions. It also includes several rotating parts that require high precision fitting between gears to transfer motion.

Figure 6.2. Black and Decker DR260B 3/8″ 5.2 Amp Corded Drill/Driver [63]

This study also includes analysis of two different types of rice cookers. The smaller rice cooker, a Rival® CKRVRCMO63, has 3-cup capacity, one-touch operation and auto keep-warm features. On the other hand, the Aroma® ARC-150SB has an enlarged capacity of 10 cups with digital controller that offers features such as different cooking functions for different kinds of rice and a keep-warm function in addition to delay timer. Model differences can be seen in Figure 6.3. These two different types of rice cookers were selected for the analysis due to their different principles of operation. Specifically, the Rival rice cooker uses ferromagnetic properties to switch from cooking mode to keep warm mode, whereas the Aroma rice cooker is completely digitized and controls its features by a PCB.

Figure 6.3. (a) Rival CKRVRCMO63[64], (b) Aroma ARC-150SB Rice Cookers [65]

Another cooking appliance analyzed in this study is the GE countertop oven with rotisserie model 168947 (see Figure 6.4). This product was selected due to its simple architecture and the low part count and includes heat regulation, timer and rotation features.

Figure 6.4. GE Countertop Oven with Rotisserie Model 168947 [66]

Several standard AA or AAA battery-operated products have also been analyzed for this study. A radio clock, See'n Say® toy, Kodak single-use camera, Revlon® electric face brush and Logitech® optical mouse are examples. The radio clock is a standard alarm clock with and LED digital screen that also includes a radio feature (see Figure 6.5). The product has standard snooze and sleep functions with the preference of buzzer or radio alarm to wake and includes built-in speakers. It has single type of buzzer alarm and has 0.6″ red LED display. The radio has both AM/FM bands that can be adjusted manually.

Figure 6.5. GPX Brand Radio Clock [67]

The Revlon RVSP3505B1 is an electric face brush that is operated by battery power. The product consists of a rotating head to be used with any cleansing product for facial cleansing. This particular product also includes interchangeable heads offering different features such as: exfoliation, blemish extraction, rolling ball massage and cleansing sponge applicators with low or high speed options. Different from all other applicators, the blemish extraction head has suction property instead of a rotating feature, which is enabled by a small propeller beneath the tube on the attachment head. The product and its different applicator heads can be seen from Figure 6.6.

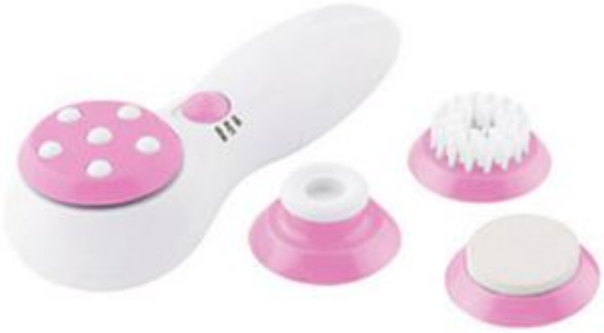


Figure 6.6. Revlon Electric Brush with Different Applicators [68]

Another product that operates with battery power is Mattel’s See 'n Say The Farmer Says infant toy (seen in Figure 6.7). The barn-shaped product includes songs and tips to teach kids about 16 different animals. It is operated by pulling the lever on the side and pointing the arrow. The toy also includes a quiz mode and plays two different songs. The product includes a built-in speakers and PCB that controls the gaming process. Even though the architecture is not complex, the product includes many small parts that are all secured by screws to the housing to minimize choking hazards.

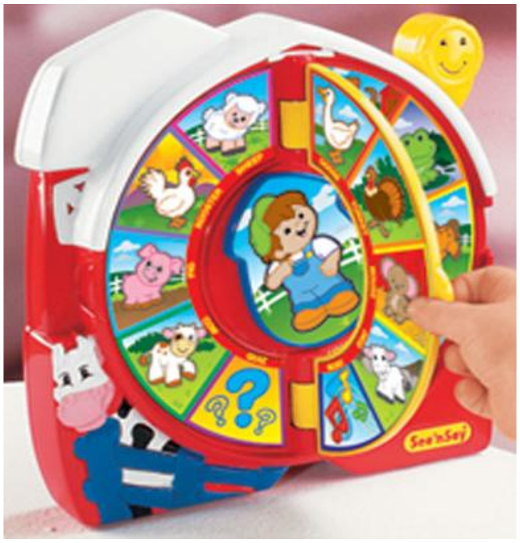


Figure 6.7. Mattel See 'N Say The Farmer Says [69]

Another product that includes many interconnected small parts is the Kodak Fun Saver single-use camera. This purely mechanical product only uses the battery to power its flash. It includes twenty-seven exposures and an exposure counter. The product, which is illustrated in Figure 6.8, uses mostly spatial interfaces and almost all components are plastic, excluding the flash connections to the battery and the shutter.

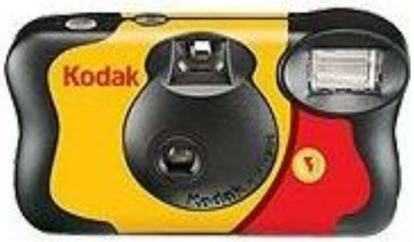


Figure 6.8. Kodak Fun Saver Single-Use Camera [70]

The last alkaline battery-powered product in the set is an optical mouse. The Logitech M310 computer mouse is also discussed in detail in Chapter 5 when demonstrating the proposed method step-by-step. The product includes two PCBs for control and uses Bluetooth technology for connecting to the computer. It carries the Bluetooth connector under the lower housing, which can be seen in Figure 6.9 with the mouse.

Figure 6.9. The Logitech M310 Wireless Computer Mouse [71]

The study also includes analysis of a product with purely mechanical architecture. The PowerShot 5700M staple/nailgun (see Figure 6.10) uses springs to push the staple or nail into the desired surface. The product is assembled in such a way that the mechanism inside is under stress. Once the housings are disassembled, the inner components of the product spring out of their places to release the tension.

Figure 6.10. PowerShot 5700M Staple/Nailgun [72]

Similar to the Revlon electric brush and the Black and Decker power drill, another product that provides rotation as the main function is the 5-Speed 100-watt Toast Master® Hand mixer (see Figure 6.11). Unlike the more complicated inner workings of the power drill, this product includes a couple of gears that transfer the motion from the motor to the beaters. The upper housing includes a button for speed control and another to release the beaters.

Figure 6.11. 5-Speed 100-Watt Toast Master Hand Mixer [73]

Coffee makers are one of the most used products in product architecture analysis due to their low cost and wide range of different types of connections they include. They are easy to dissect and re-assemble. This study also includes a coffee maker in the case study set. The Mr. Coffee® TF13 12-cup coffee maker has been chosen for analysis (see Figure 6.12). The product works with a very simple one-button operation; however, it includes various connections of electrical, spatial and field interactions due to its relations between heating coil and connecting components.

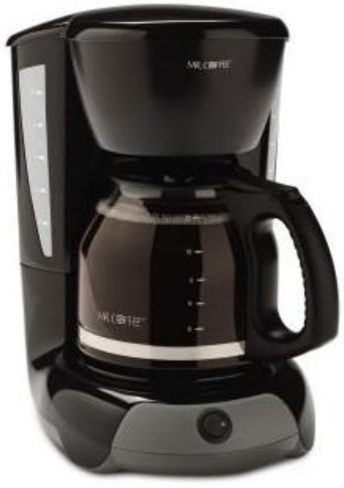


Figure 6.12. Mr. Coffee TF13 12-cup Coffee Maker [74]

Another household item with a heat function analyzed in this study is a simple clothes iron. The Rival lightweight iron (see Figure 6.13) is one of the most basic models on the market. It includes only the most fundamental features: adjustable temperature control, steam/dry option, pump spray, heel rest, and transparent water tank. The heating elements are placed just under the base and isolated from the plastic water tank. The handle of the product includes the steam control and water pump with an opening for reservoir refill.

Figure 6.13. Rival Lightweight Iron [75]

Conair® Ionic Ceramic Cord-keeper 209 GWP is another product with heat function. The product, which is illustrated in Figure 6.14, has features of ceramic technology to prevent heat damage, and ionic technology for shinier and healthier hair. The hairdryer has three levels of heat and two levels of speed. The cord retractor mechanism tucks in the cord automatically with a push of a button. It includes a lot of additional functions compared to a base model hairdryer.

Figure 6.14. Conair Ionic Ceramic Cord-Keeper 209 GWP [76]

The last product with a heating function is the Honeywell® Surround Heat model HZ-220 room heater. The product has the capability of radiating heat 360°, and the heat can be directed in one direction if desired. The heater outputs 1500 watts on its high setting and also includes safety features such as a child-resistant power knob and a tip-over switch that disables the unit. Flame resistant plastic, an adjustable thermostat that maintains the same temperature, and overheat protection are the other features of this product. The product can be viewed in Figure 6.15.

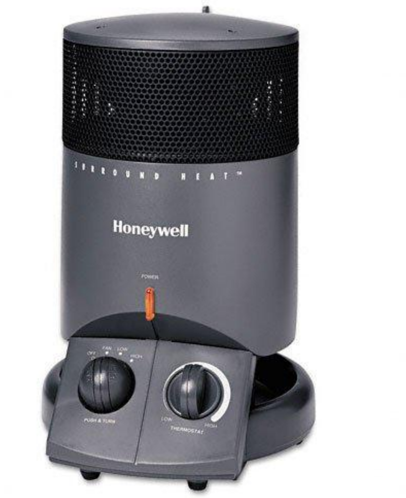


Figure 6.15. Honeywell Surround Heat Model HZ-220 Room Heater [77]

The Conair corded phone model PR5007w is a simple corded dial phone that is analyzed in this study (see Figure 6.16). This product carries the simplest features of its kind. The compact slim phone can be placed on a desk or also can be mounted on the wall. The only additional feature for the phone is a redial feature. The ringer volume can be adjusted to high or low.

Figure 6.16. Conair Corded Phone Model PR5007w [78]

The last product in the analysis set is a handheld 14.4-volt Black & Decker model CHV1408 Dustbuster®, which is seen in Figure 6.17. This bag-less handheld mini vacuum cleaner works with rechargeable batteries just like the products in the Durabuilt power tools set. The product description claims cleaner exhaust due to three-stage filtration. The outer dial provides instant filter cleaning to restore suction. The product has different stages of fans and cone tubings to facilitate suction function. Most components are connected by spatial interactions and rotate within the housings.

Figure 6.17. Black & Decker Model CHV1408 Dustbuster 14.4-Volt [79]

As mentioned, the set of products for analysis has been chosen in such a manner as to enable comparison and cover all different types of interactions defined in this study. The individual types of interfaces found in each product are summarized in Table 6-1.

Table 6-1. Types of Interfaces that Exist in Analyzed Products

Product Name	**A**	**S**	**P**	**C**	**T**	**F**
Aroma Rice Cooker	✓	✓	✓	✓	-	✓
Circular Saw	✓	✓	✓	✓	✓	-
Coffee Maker	✓	✓	✓	✓	✓	✓
Drill	✓	✓	✓	-	✓	✓
Dustbuster	✓	✓	✓	✓	✓	-
Electric Brush	✓	✓	✓	✓	✓	-
Flashlight	✓	✓	✓	✓	-	-
Hair Dryer	✓	✓	✓	✓	-	✓
Handmixer	✓	✓	✓	-	✓	-
Heater	✓	✓	✓	✓	-	✓
Iron	✓	✓	✓	-	✓	✓
Jigsaw	✓	✓	✓	✓	✓	-
Mouse	✓	✓	✓	✓	-	✓
Oven	✓	✓	✓	✓	✓	-
Phone	✓	✓	✓	✓	-	-
Radio Clock	✓	✓	✓	✓	-	✓
Rival Rice Cooker	✓	✓	✓	✓	-	✓
Sander	✓	✓	✓	✓	✓	-
See’n Say	✓	✓	✓	✓	-	-
Single Use Camera	✓	✓	✓	-	-	-
Stapler	✓	✓	-	-	-	-

6.2 Analysis

Application of Phase 1 of the method (which includes product dissection and architecture representation phases) is carried out individually for each product to avoid any confusion during dissection. Each product in the set of twenty-one was dissected separately, and the required information on the nature of the interfaces has been recorded in individual interface DSMs as described in Chapter 5. These matrices can be viewed in Appendix A for all selected products. After completion of Phase 1 for all twenty-one products in the set, statistical analysis for determination of the frequency of occurrence for individual interfaces is conducted. Based on the analysis, descriptive product architecture data was also collected (see Table 6-2). This sets the stage for Phase 2: Determination of the Connection Values of the proposed method.

Table 6-2. Descriptive Product Architecture Data

Product Name	# of Connections	# of Parts	Average Connection/part	Number of Different Interfaces/Interface Combinations
Aroma Rice Cooker	44	15	2.93	6
Circular Saw	66	19	3.48	7
Coffee Maker	54	16	3.38	8
Drill	86	16	5.38	6
Dustbuster	80	19	4.21	7
Electric Brush	44	13	3.38	5
Flashlight	58	15	3.87	5
Hairdryer	96	28	3.43	6
Handmixer	38	10	3.8	5
Heater	104	25	4.16	6
Iron	62	13	4.77	8
Jigsaw	88	22	4	8
Mouse	40	11	3.64	6
Oven	114	28	4.07	7
Phone	54	16	3.38	3
Radio Clock	64	15	4.26	6
Rival Rice Cooker	56	25	4.16	8
Sander	46	15	3.07	8
See’n Say	66	17	3.82	6
Single Use Camera	118	20	5.9	5
Stapler	68	16	4.25	2

The relative weights of different types of interactions are calculated based their frequency of occurrence by using the weighting function as detailed in Eq. 4.1. Frequency of occurrence statistics are collected during the product dissection phase where all the products of interest are disassembled to the individual component level, and the connections between them are recorded in a component-to-component DSM.

Based on the collected data, spatial interfaces are the most prominent type of interface in product architectures with an occurrence rate of 51%. The second most prominent type of interface is attachment interface: 34% of all interfaces in the analysis set are comprised of attachment interfaces. Electrical interfaces make up 10% of all interfaces, communication and controls interfaces represent 3% and transfer and field interfaces remain at 1%. The respective occurrence ratios are plotted in the pie chart in Figure 6.18.

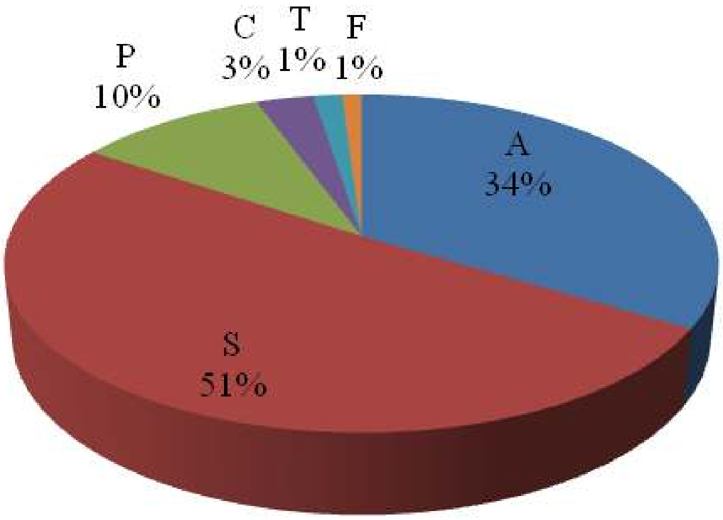


Figure 6.18. Interface Type Distribution of Case Study Set

The relative weights of different types of interfaces can be now calculated based on the given data in Figure 6.18 by using the weighting function (in Eq. 4.1 in detail). Figure 6.19(a) and (b) demonstrate the proposed weighting function and its corresponding sensitivity, respectively. The constants C_1 and C_2 were chosen such that the sensitivity approaches zero as the likeliness of occurrence increases. For instance, interfaces that occur frequently such as attachment and spatial will have similar weights compared to interfaces that occur less frequently (e.g., communication or transfer).

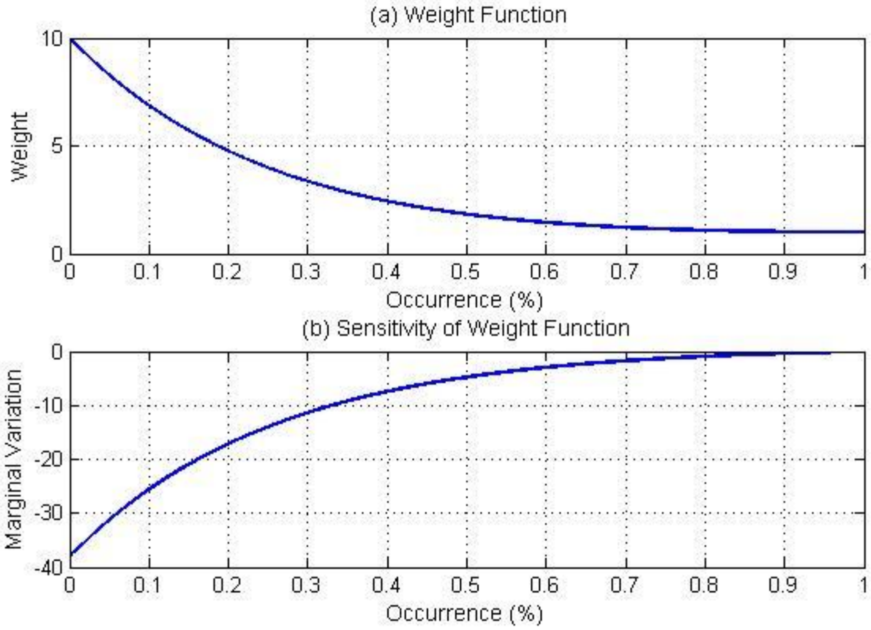


Figure 6.19. (a) Weight Function and (b) Sensitivity of the Weight Function

Given the frequency of occurrence and increased sensitivity towards the less frequent interface types, individual weights for interface types have been determined. The final weights can be viewed in Table 6-3.

Table 6-3. Final Assigned Weights for Different Interface Types

	A	S	P	C	T	F
Occurrence %	33.634	50.719	10.062	3.039	1.520	1.027
Weight	2.974	1.795	6.832	8.923	9.445	9.627

6.2.1 Results

Validation of the assigned scores is very important for getting rational results that gives design engineers insight about the behavior of the component within the product architecture. In order to define the accuracy of the assigned weights, a sensitivity analysis is conducted to examine the variation in weights when calculated with different numbers of products in the set. The analysis has been made by using 5, 10, 15, and 21 products.

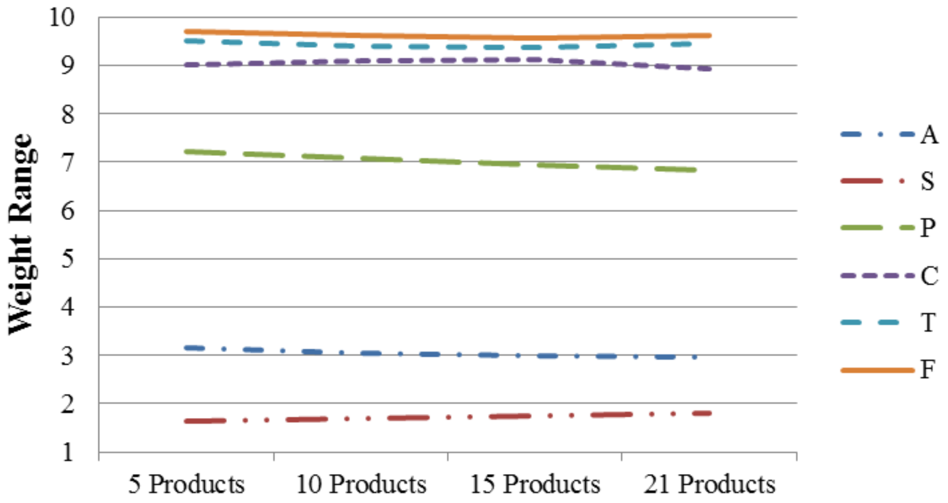


Figure 6.20. Sensitivity Analysis of Weights

The weights are calculated using the approach based on the occurrence of interfaces in each subset of products. As seen from Figure 6.20, the obtained weights remained fairly consistent for each interface type and biggest change in the actual score values did not exceed 10 percent. Individual change rates can be viewed in Table 6-4.

Table 6-4. Sensitivity of Weights for Different Interface Types

Weights	A	S	P	C	T	F
5 Products	3.145	1.626	7.225	9.025	9.517	9.700
10 Products	3.043	1.698	7.091	9.093	9.409	9.627
15 Products	2.993	1.754	6.934	9.128	9.374	9.554
21 Products	2.974	1.795	6.832	8.923	9.445	9.627
% change	**5.453**	**9.398**	**5.449**	**1.878**	**1.509**	**1.510**

As seen in Table 6-4, the largest change rate of interface weights remained less than 10%. One can observe that as the frequency of occurrence of individual interface types decreases, the percentage of change also decreases. Since the accuracy of the assigned weights are validated within this product set, the connection calculations for different combinations of interface types is completed by using weighted-MCS by Eq. 4.2. A complete list of calculated values of different types of interactions can be seen in Table 6-5.

Table 6-5. Weight Calculations of Different Types of Interactions Used in Analysis

A	2.974	TF	38.143	STP	54.215
S	1.795	PF	32.916	SCF	61.032
P	6.832	CF	37.098	SCP	52.647
C	8.923	PC	31.508	SFP	54.759
T	9.445	AST	42.642	PCF	76.142
F	9.627	ASC	41.074	TCP	75.598
AS	9.538	ASF	43.186	TFP	77.710
AT	24.838	ASP	34.801	ASTC	92.546
AC	23.793	ATC	64.025	ASTF	95.362
AF	25.201	ATF	66.137	ASTP	84.182
AP	19.611	ATP	57.752	ATCF	123.872
ST	22.480	ACF	64.568	ATCP	112.692
SC	21.435	ACP	56.184	ACFP	113.418
SF	22.843	AFP	58.296	ASTCF	163.815
SP	17.253	STC	60.488	ASTCP	149.841
TC	36.735	STF	62.600	ASTCFP	237.568

After calculating the connection values for the different types of interfaces, the corresponding values have been placed in interaction DSMs for all products. Electrical circuit models are built for individual products, and the reciprocals of the calculated connection values are used as the input data for the electrical circuit models for each product. These models enable the determination of the strength of the design dependencies of the connections between any modules by taking into account all direct and indirect connections in the system. Calculating the reciprocal values of w-MCS values for each connection in the product architecture concludes Phase 2 of the proposed method.

As explained in Chapter 4, each product architecture is modeled as an electrical circuit in which connections between components are represented by resistors. Each individual resistor is given its specific resistance, which makes a total of 720 individual connections for all twenty-one products in the analysis set. Circuit models are completed once every resistance value has been entered for its corresponding resistor (connection), and all connections are completed in the system. Calculation of the equivalent resistance of each connection has been made by connecting the two measurement nodes to the components that develop the connection of interest. Every connection in the whole analysis set has been individually measured, and the outcomes have been recorded on the final output matrices. Individual electrical circuit models and the output matrices for all products can be seen in Appendix B.

Table 6-6 lists the change resistance (C.R) values obtained from the electrical circuit models of each product. The first and second columns of the table represent the average minimum change resistance value in each product and its corresponding components. The change resistance value of a component represents the ease of change propagation through that component. As the change resistance gets smaller, the resistance to conduct change will be smaller, and consequently any design change in the corresponding component will be propagated to the connecting components. Therefore, the minimum C.R value in the system represents the most change-sensitive component in the system. Average C.R values have been linearly normalized between [0,1] to facilitate the ranking and grouping amongst the product set.

Table 6-6. Summary of Model Results

Rank	Product	Normalized Average C.R	Min. Average Change Resistance	Corresponding Component	Number of Connected Comp
1	Phone	0.000	0.013	Controls PCB	10
2	Iron	0.056	0.023	Controller	5
3	See'n Say	0.117	0.015	Speaker & PCB	3
4	Jigsaw	0.123	0.015	Motor	5
5	Sander	0.130	0.031	Motor	5
6	Drill	0.148	0.019	Motor	9
7	Circular Saw	0.160	0.029	Motor	4
8	Handmixer	0.167	0.023	Motor	8
9	Oven	0.198	0.024	T&F Control	6
10	Aroma Rice Cook.	0.259	0.030	Controls PCB	5
11	Heater	0.284	0.019	Motor& Contr.	7
12	Radio Clock	0.284	0.014	PCB	11
13	Rival Rice Cooker	0.309	0.014	Control PCB	4
14	Single Use Cam.	0.321	0.045	Flash PCB	7
15	Mouse	0.364	0.026	PCB	7
16	Flashlight	0.377	0.031	PCB	4
17	Coffee Maker	0.438	0.026	Heating coil	6
18	Dustbuster	0.469	0.015	Motor	4
19	Stapler/Nailgun	0.525	0.029	Housing	9
20	Electric Brush	0.593	0.027	Motor	7
21	Hairdryer	1.000	0.100	Heating coil	7

It can be seen from Table 6-6 that the proposed approach consistently assigns the same type of components as the most change sensitive ones in the system. The motors have been consistently identified as the most change sensitive component in the products in which they exist. Moreover, if the product does not have a motor but has electrical elements, then PCBs are designated as the most change sensitive component in the architecture. In the case of purely mechanical (non-electrical) products such as stapler/nail gun, results indicate that the housing component (which holds and connects all springs and tension trigger mechanisms together) proves to be the most change sensitive component in overall structure.

To compare different products in the analysis set, the average minimum C.R value of each product has been calculated, and products have been ranked in Table 6-6 from 1 to 21 where the most design dependent product is ranked first and the least design dependent product is ranked last. Figure 6.21 represents a scatter diagram for minimum average change resistance values and average number of connections for each product that was analyzed for this study. Based on this analysis, the phone is determined to be the most change sensitive product in the analysis set, i.e., a design change in this system creates more change in the overall architecture than any other products in the analysis set. As it can be seen from Table 6-6, the phone does not have either the highest number of components or the most complex connections. The average connection per part is 3.38, which slightly below average for the analysis set. The hairdryer, which is ranked the least change sensitive in the set, has a total number of connections way above the average and lower than average connection per component value (calculated as 3.43 connections/component). However, the results indicate that those aspects, in addition with the effect of indirect connections, contribute to determine the design dependency.

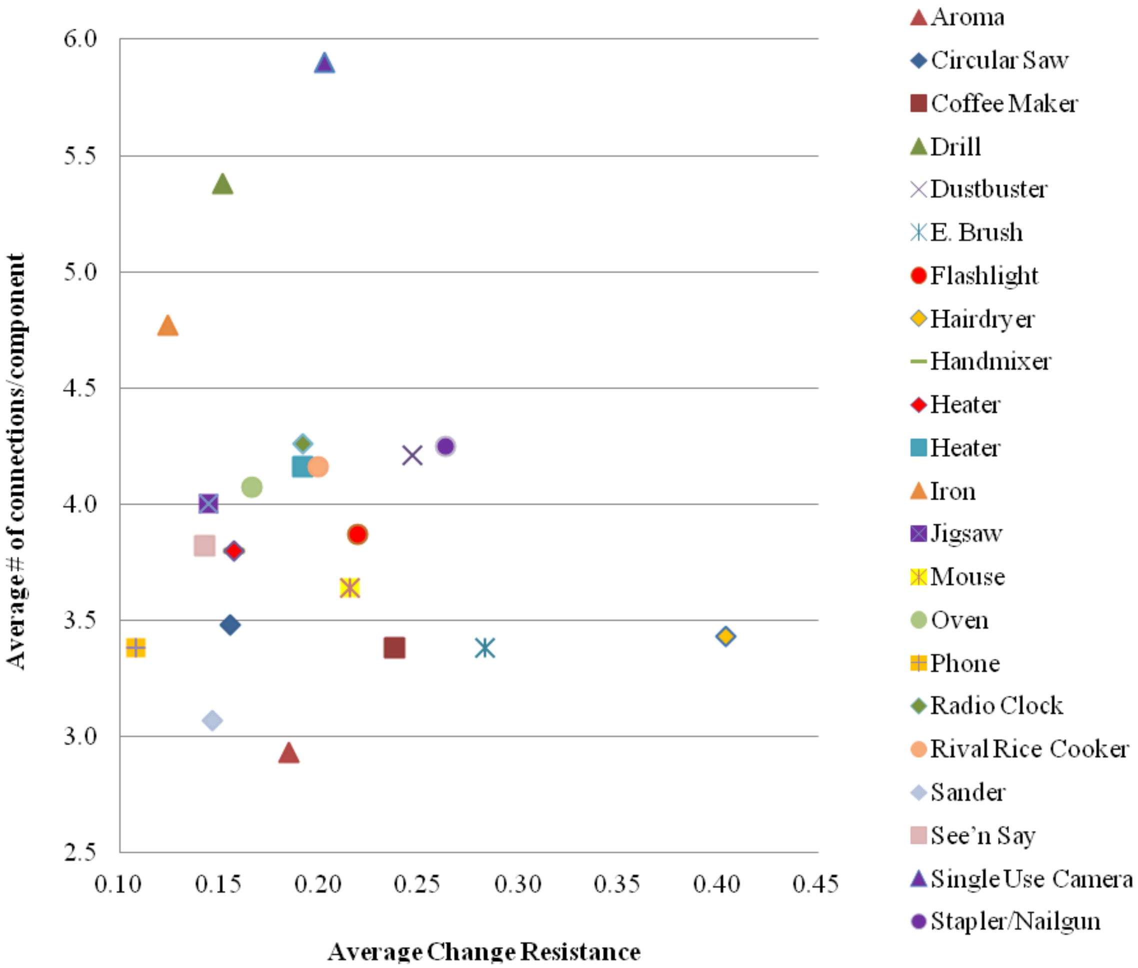


Figure 6.21. Average Connections/Component vs Average Min C.R

The proposed method also enables designers to analyze the products at the individual component level to determine the change sensitive components in the product architecture. Individual components are ranked and divided into groups that represent their change sensitivity (see Table 6-7). The components that obtain normalized min C.R between 0.0 and 0.3 are placed in "high change sensitivity" group indicating that that particular component is very likely to transmit a design change to other components that are connected to it. The products are placed in the medium level change sensitivity group when their normalized minimum C.R value is between 0.31 and 0.60 and low change sensitivity range is 0.61 to 1.00. These groupings of high, medium, or low change sensitivity and the defined ranges are presented with the purpose of simpler visualization of comparison between different products or the outcomes of different methods. Design engineers can pick different ranges based on their preferred tolerance of change for their components.

Table 6-7. Component Sensitivity Ranking for the Phone

Phone	#	N. CR	
Controls PCB	3	0.000	**High**
Redial PCB	5	0.000	
Front Housing	2	0.016	
Numbers PCB	4	0.032	
Numbers/Buttons	9	0.032	
Receiver	6	0.049	
Curly Cable	11	0.059	
Holster PCB	15	0.062	
Main wall cable	16	0.064	
Ringer	7	0.067	
Mic	8	0.069	
Holster Upper Housing	12	0.192	
Holster Lower Housing	13	0.192	
Weight	14	0.419	**Medium**
Volume Button	10	0.973	**Low**
Back Housing	1	1.000	

The product design dependency rankings presented in Table 6-6 also indicate that the hairdryer product has the least design dependency in the analysis set, which means that it is the most robust product and transfers the minimum amount of change compared to other products given a design change in the system. The hairdryer product includes 28 components. Change sensitivity levels of the individual components within the hairdryer can be seen in Table 6-8.

Table 6-8. Component Sensitivity Ranking for the Hairdryer

Hairdryer	#	N. CR	
Heating Coil	14	0.000	High
Heat Selector Switch	19	0.000	
Cool Down Switch	21	0.014	
GFI Plug/power	18	0.030	
Motor/Fan	12	0.035	
Speed Selector Switch (ON/OFF)	20	0.088	
Impeller	10	0.110	
Black Airflow Diverter	11	0.110	
Black Electrical Comp	15	0.110	
Retractor trigger	28	0.126	
Power Cord Retractor	25	0.128	
Back Housing	2	0.140	
Front Housing	4	0.140	
Fins	13	0.226	
Power Cord	17	0.370	Medium
Switch Retainer Back1	22	0.418	
Switch Retainer Back2	23	0.426	
Cool Down Button	7	0.537	
Retractor button	27	0.554	
Insulation Cone	8	0.559	
Heat Selector Switch Cover	5	0.600	
High/Low Switch Cover	6	0.602	Low
Removable Inlet Protective Screen	1	1.000	
Metal Inlet Protector	3	1.000	
Ceramic Ring	9	1.000	
Exit Protective Screen	16	1.000	
Cord Spacer	24	1.000	
Spacer	26	1.000	

6.2.2 Validation

To validate these results, a design change is imposed on a product architecture to observe the outcomes. This way, the change sensitivity of the products can be compared, and the model can be validated. Three products have been chosen based on the overall design dependency rankings presented in Table 6-6, to investigate the change propagation in their architectures. The most change resistant product (hairdryer), the least change resistant product (phone), and a third product (radio clock) that scored in the middle have been chosen for further analysis (see Table 6-6).

The hairdryer is chosen as the main product for inflicting a design change due to its resistance to transmit changes in the architecture (based on the results from the proposed method). To be able to make the most change in the overall system (since this is the most change resistant product in the set), the connection with the least change resistance within the product is chosen for modification. The change effects will be more prominent in the system to observe the propagation on directly and indirectly connected components. Figure 6.22 indicates the most change sensitive connection within the hairdryer product architecture.

Hairdryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet Pr	1		S																										
Back Housing	2	S		S	A(2),S													S								A(2),S			
Metal Inlet Protecto	3		S																										
Front Housing	4		A(2),S			S	S	S	S					S		S	S	S				S	S	S	S		S	S	S
Heat Selector Switch	5				S																		S						
High/Low Switch Co	6				S																			S					
Cool Down Button	7				S																	S							
Insulation Cone	8				S									S															
Ceramic Ring	9													S															
Impeller	10											A(3),S							S							A(2),S			AS
Black Airflow Divert	11										A(3),S		AS	S															
Motor/Fan	12											AS			PC														
Fins	13				S				S	S		S			S	A													
Heating Coil	14												PC	S		P			PC	P,PCF	P	PC							
Black Electrical Com	15				S									A	P						P								
Exit Protective Scree	16				S																								
Power Cord	17		S		S																					S			
GFI Plug/power	18										S				PC						P					S			
Heat Selector Switch	19														P,PCF							PC	S						
Speed Selector Swit	20														P	P			P					S					
Cool Down Switch	21				S			S							PC					PC									
Switch Retainer Back	22				S	S														S									
Switch Retainer Back	23				S		S														S								
Cord Spacer	24				S																								
Power Cord Retracto	25		A(2),S								A(2),S							S	S										S
Spacer	26				S																								
Retractor button	27				S																								S
Retractor trigger	28				S						AS															S		S	

Figure 6.22. Interface DSM for the Hairdryer

The connection between heating coil and the heat selector switch is assigned the most change sensitive connection in the architecture by the model. If we make a design change to increase the range of heat that the hairdryer provides, then we might need to add heating coils to increase the heat and add electrical connections to associate with that function as well as add an additional connection to the control mechanism to regulate the heat over a wider range. As a result of making these changes in that particular connection, assume that the interfaces for that connection double in strength. After the new dependency values are calculated for the modified connection, the change sensitivity of the new architecture is calculated following the proposed method. The connection values for the interface DSM and resistance input values for the circuit model are calculated. Since the interface types have pre-determined weights and the weighted MCS formula incorporates the number of interfaces, doubling the amount of interfaces in that connection doubles the value of the design dependency for that connection. The original w-MCS value and the value for the modified architecture can be seen the Figure 6.23.

Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet	1	■	1.8																										
Back Housing	2		■	1.8	7.74													1.8								7.74			
Metal Inlet Protec	3			■																									
Front Housing	4				■	1.8	1.8	1.8	1.8					1.8		1.8	1.8	1.8				1.8	1.8	1.8	1.8		1.8	1.8	1.8
Heat Selector Sw	5					■																	1.8						
High/Low Switch	6						■																	1.8					
Cool Down Butto	7							■														1.8							
Insulation Cone	8								■					1.8															
Ceramic Ring	9									■				1.8															
Impeller	10										■	10.71							1.8							7.74			9.54
Black Airflow Di	11											■	9.54	1.8															
Motor/Fan	12												■		31.51														
Fins	13													■	1.8	2.97													
Heating Coil	14														■	6.83			31.51	82.97	6.83	31.51							
Black Electrical C	15															■					6.83								
Exit Protective Sc	16																■												
Power Cord	17																	■								1.8			
GFI Plug/power	18																		■		6.83					1.8			
Heat Selector Sw	19																			■		31.51	1.8						
Speed Selector Sv	20																				■			1.8					
Cool Down Switc	21																					■							
Switch Retainer E	22																						■						
Switch Retainer E	23																							■					
Cord Spacer	24																								■				
Power Cord Retr	25																									■			1.8
Spacer	26																										■		
Retractor button	27																											■	1.8
Retractor trigger	28																												■

Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet	1	■	1.8																										
Back Housing	2		■	1.8	7.74													1.8								7.74			
Metal Inlet Protec	3			■																									
Front Housing	4				■	1.8	1.8	1.8	1.8					1.8		1.8	1.8	1.8				1.8	1.8	1.8	1.8		1.8	1.8	1.8
Heat Selector Sw	5					■																	1.8						
High/Low Switch	6						■																	1.8					
Cool Down Buttor	7							■														1.8							
Insulation Cone	8								■					1.8															
Ceramic Ring	9									■				1.8															
Impeller	10										■	10.71							1.8							7.74			9.54
Black Airflow Di	11											■	9.54	1.8															
Motor/Fan	12												■		31.51														
Fins	13													■	1.8	2.97													
Heating Coil	14														■	6.83			31.51	165.94	6.83	31.51							
Black Electrical C	15															■					6.83								
Exit Protective Sc	16																■												
Power Cord	17																	■								1.8			
GFI Plug/power	18																		■		6.83					1.8			
Heat Selector Sw	19																			■		31.51	1.8						
Speed Selector Sv	20																				■			1.8					
Cool Down Switc	21																					■							
Switch Retainer E	22																						■						
Switch Retainer E	23																							■					
Cord Spacer	24																								■				
Power Cord Retra	25																									■			1.8
Spacer	26																										■		
Retractor button	27																											■	1.8
Retractor trigger	28																												■

Figure 6.23. w-MCS Values for Original and Modified Hairdryer Architectures

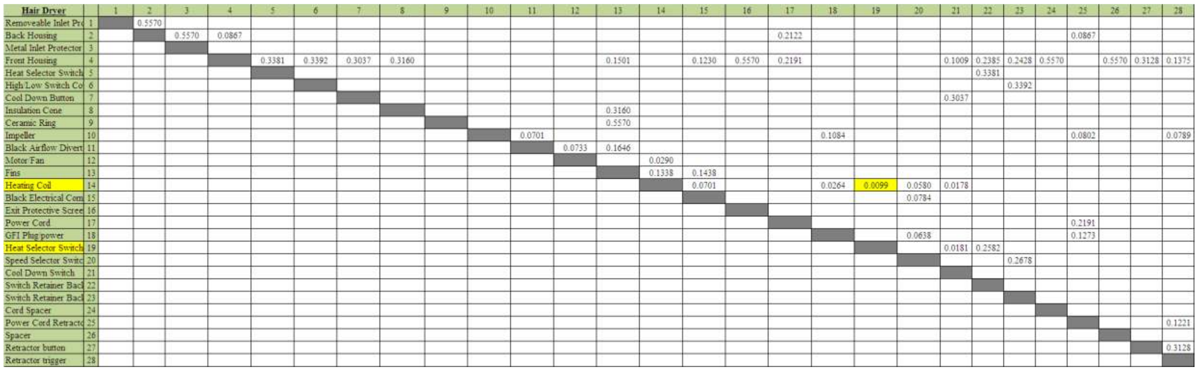
As it was explained in the previous chapters, due to the characteristics of the resistivity theory, the values that are obtained from weighted-MCS equation are not used directly in the circuit model. Since the higher values represent high resistance to change in the electrical circuit model, the reciprocals of the weighted-MCS values are used as input. Therefore, any change that doubles the weighted-MCS value of a connection would halve the input resistance value for that connection. The original input resistance is decreased from 0.012 to 0.006 as it can be seen in the Figure 6.24.

Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet	1	▨	0.557																										
Back Housing	2		▨	0.557	0.129													0.557								0.129			
Metal Inlet Protec	3			▨																									
Front Housing	4				▨	0.557	0.557	0.557	0.557					0.557		0.557	0.557	0.557				0.557	0.557	0.557	0.557		0.557	0.557	0.557
Heat Selector Sw	5					▨																	0.557						
High/Low Switch	6						▨																	0.557					
Cool Down Butto	7							▨														0.557							
Insulation Cone	8								▨					0.557															
Ceramic Ring	9									▨				0.557															
Impeller	10										▨	0.093							0.557							0.129			0.105
Black Airflow Div	11											▨	0.105	0.557															
Motor/Fan	12												▨		0.032														
Fins	13													▨	0.557	0.336													
Heating Coil	14														▨	0.146			0.032	0.012	0.146	0.032							
Black Electrical C	15															▨					0.146								
Exit Protective Sc	16																▨												
Power Cord	17																	▨								0.557			
GFI Plug/power	18																		▨		0.146					0.557			
Heat Selector Sw	19																			▨		0.032	0.557						
Speed Selector Sv	20																				▨			0.557					
Cool Down Switc	21																					▨							
Switch Retainer E	22																						▨						
Switch Retainer E	23																							▨					
Cord Spacer	24																								▨				
Power Cord Retra	25																									▨			0.557
Spacer	26																										▨		
Retractor button	27																											▨	0.557
Retractor trigger	28																												▨

Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet	1	▨	0.557																										
Back Housing	2		▨	0.557	0.129													0.557								0.129			
Metal Inlet Protec	3			▨																									
Front Housing	4				▨	0.557	0.557	0.557	0.557					0.557		0.557	0.557	0.557				0.557	0.557	0.557	0.557		0.557	0.557	0.557
Heat Selector Sw	5					▨																	0.557						
High/Low Switch	6						▨																	0.557					
Cool Down Butto	7							▨														0.557							
Insulation Cone	8								▨					0.557															
Ceramic Ring	9									▨				0.557															
Impeller	10										▨	0.093							0.557							0.129			0.105
Black Airflow Div	11											▨	0.105	0.557															
Motor/Fan	12												▨		0.032														
Fins	13													▨	0.557	0.336													
Heating Coil	14														▨	0.146			0.032	0.006	0.146	0.032							
Black Electrical C	15															▨					0.146								
Exit Protective Sc	16																▨												
Power Cord	17																	▨								0.557			
GFI Plug/power	18																		▨		0.146					0.557			
Heat Selector Sw	19																			▨		0.032	0.557						
Speed Selector Sv	20																				▨			0.557					
Cool Down Switc	21																					▨							
Switch Retainer E	22																						▨						
Switch Retainer E	23																							▨					
Cord Spacer	24																								▨				
Power Cord Retra	25																									▨			0.557
Spacer	26																										▨		
Retractor button	27																											▨	0.557
Retractor trigger	28																												▨

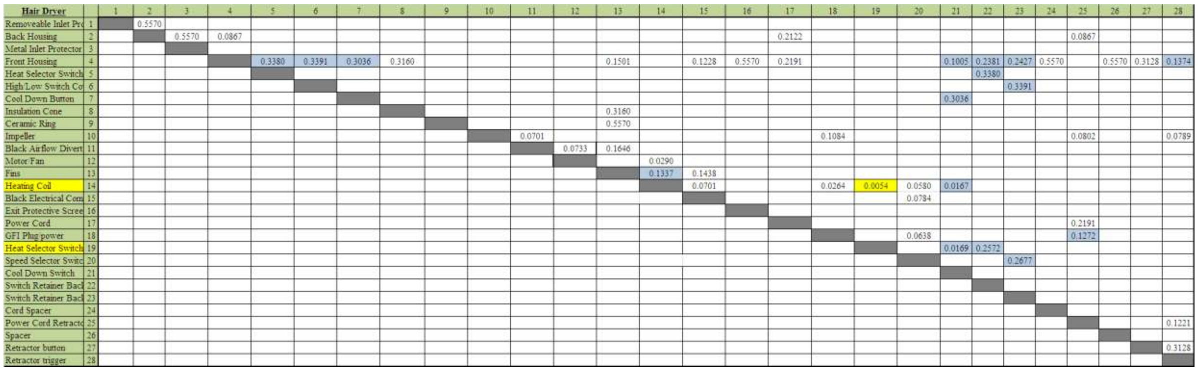
Figure 6.24. Resistance Input Values for Original and Modified Hairdryer Architectures

The analysis for both the original and the modified architecture are completed, and the results are shown in Figures 6.25 and 6.26, respectively. Based on comparison of the changes of the output values for both models, it can be observed that the change does not propagate to all connections in the architecture, but it does propagate.



Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet Pr	1		0.5570																										
Back Housing	2			0.5570	0.0867													0.2122								0.0867			
Metal Inlet Protector	3																												
Front Housing	4					0.3381	0.3392	0.3037	0.3160					0.1501		0.1230	0.5570	0.2191				0.1009	0.2385	0.2428	0.5570		0.5570	0.3128	0.1375
Heat Selector Switch	5																						0.3381						
High/Low Switch Co	6																							0.3392					
Cool Down Button	7																					0.3037							
Insulation Cone	8													0.3160															
Ceramic Ring	9													0.5570															
Impeller	10											0.0701							0.1084							0.0802			0.0789
Black Airflow Divert	11												0.0733	0.1646															
Motor/Fan	12														0.0290														
Fins	13														0.1338	0.1438													
Heating Coil	14															0.0701			0.0264	0.0099	0.0580	0.0178							
Black Electrical Con	15																				0.0784								
Exit Protective Scree	16																												
Power Cord	17																									0.2191			
GFI Plug/power	18																				0.0638					0.1273			
Heat Selector Switch	19																					0.0181	0.2582						
Speed Selector Switc	20																							0.2678					
Cool Down Switch	21																												
Switch Retainer Bac	22																												
Switch Retainer Bac	23																												
Cord Spacer	24																												
Power Cord Retracto	25																												0.1221
Spacer	26																												
Retractor button	27																												0.3128
Retractor trigger	28																												

Figure 6.25. Output Values for the Original Design



Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet Pr	1		0.5570																										
Back Housing	2			0.5570	0.0867													0.2122								0.0867			
Metal Inlet Protector	3																												
Front Housing	4					0.3380	0.3391	0.3036	0.3160					0.1501		0.1228	0.5570	0.2191				0.1005	0.2381	0.2427	0.5570		0.5570	0.3128	0.1374
Heat Selector Switch	5																						0.3380						
High/Low Switch Co	6																							0.3391					
Cool Down Button	7																					0.3036							
Insulation Cone	8													0.3160															
Ceramic Ring	9													0.5570															
Impeller	10											0.0701							0.1084							0.0802			0.0789
Black Airflow Divert	11												0.0733	0.1646															
Motor/Fan	12														0.0290														
Fins	13														0.1337	0.1438													
Heating Coil	14															0.0701			0.0264	0.0054	0.0580	0.0167							
Black Electrical Con	15																				0.0784								
Exit Protective Scree	16																												
Power Cord	17																									0.2191			
GFI Plug/power	18																				0.0638					0.1272			
Heat Selector Switch	19																					0.0169	0.2572						
Speed Selector Switc	20																							0.2677					
Cool Down Switch	21																												
Switch Retainer Bac	22																												
Switch Retainer Bac	23																												
Cord Spacer	24																												
Power Cord Retracto	25																												0.1221
Spacer	26																												
Retractor button	27																												0.3128
Retractor trigger	28																												

Figure 6.26. Output Values for the Modified Design

The connections that are marked with blue in Figure 6.26 illustrate how design change propagates in the architecture. By using the difference in the output values and normalizing it, relative change is calculated. The percentage of relative change per connection in the hairdryer architecture is calculated as 1.25%. This value may not mean much by itself and might even be considered as negligible; however, one should remember that this is not an exact value of change but a comparative value of change. Therefore, it needs to be used in comparison to other products or to compare the effect of the different design changes.

This design change experiment validates the results on design dependency ranking presented in Table 6-6. The results of the analysis set suggests that the hairdryer is the least design dependent product in the set, whereas the phone is the most design dependent product and the radio clock falls in between them in ranking. To be able to validate this outcome, the design change investigation is applied to all three products. To keep the consistency of the approach and comparability of the results and the amount of change that is inflicted on the product architecture, the design changes are imposed on the most change-sensitive connection in each product architecture by doubling the design dependency of that connection. The results are presented in Table 6-9.

Table 6-9. Percentage Change Values Obtained for Three Products

	Rank	Product	% Change/Connection
Most Change Transmitting	1	Phone	3.26
	12	Radio Clock	2.04
Least Change Transmitting	21	Hairdryer	1.25

The results represent the percentage change between two models after design modifications for three products. As ranked in Table 6-6, the phone created the most change in the system given a design modification, and the hairdryer created the least change in the overall architecture compared with the other products. It can be seen that phone being the most change transmitting product in the group transmits more than two times the change compared to hairdryer. This is an important result not to be overlooked since it provides important information for design engineers to use while making decisions about the flexibility of the product architectures. Lastly, the value obtained by the radio clock adds another validation for the method since it remain between highest and the lowest percentage change as predicted and ranked by the circuit model.

In addition to validating the model at the product level, one additional test has been carried out to validate the results at the component level. A design change has been made on the "ceramic ring" component that is assigned to "low change sensitivity" group in hairdryer (see Table 6-8) and with one connection. The dependency value for this component in ceramic ring and fins connection has been doubled in value. The output for the modified architecture did not make any relative changes in the system.

Validating the model in the component level, this outcome also determines that the change is created by the nature of the connections in the system rather than the characteristics of the individual component. Therefore, with this result it is safe to state that, no matter how wide the capabilities or the functions of an individual component, the design dependency between two components will only be as high as receiving capacity of the simpler component.

6.2.2 Comparison with Other Methods

Even though there are similar studies that quantify design dependency between components objectively or considers the effect of indirect connections within product architectures, comparison of the method is needed to demonstrate the advantages over other metrics. Table 6-10 presents a comparison of different methods for the analysis set. The term “MCS” represents the calculations without weighted interface type or considering the indirect connections in the architecture. This approach proposed by Dobberfuhl and Lange [8] assumes that all of the different interface types (e.g., A and T) have the same weight and focuses instead on the importance of the degree of complexity and the number of connections made by one component. Compare this to the MCS method, which has no-weights and does not use an electrical circuit analogy to include indirect connections.

Table 6-10. Comparison of Proposed Method with MCS

Rank	Proposed Method	MCS	Average MCS	# of comp
1	Phone	Phone	11.13	16
2	Iron	Coffeemaker	9.00	16
3	See'n Say	Oven	8.82	28
4	Jigsaw	See'n Say	8.65	17
5	Sander	Drill	7.88	16
6	Drill	Iron	7.69	13
7	Circular Saw	Stapler	7.50	16
8	Handmixer	Circular Saw	7.37	19
9	Oven	Heater	7.12	25
10	Aroma 20 Cup	Handmixer	7.00	10
11	Heater	Radio clock	6.53	15
12	Radio Clock	Single use	6.40	20
13	Rival Ricer	Jigsaw	6.36	22
14	Single Use	Rival	6.13	16
15	Mouse	Dustbuster	5.90	20
16	Flashlight	E. Brush	5.69	13
17	Coffee Maker	Sander	5.53	15
18	Dustbuster	Aroma	5.47	15
19	Stapler/Nailgun	Mouse	5.09	11
20	Electric Brush	Flashlight	4.80	15
21	Hair dryer	Hairdryer	4.68	28

As can be seen from Table 6-10, the MCS method accurately defines the most and the least design-dependent products in the analysis set and provides a fairly close prediction on ranking of the radio clock; however, the similarities end there. For the analysis of the design dependencies at the component level, the MCS method provides different results. Components for all three products are grouped based on their dependencies with the values that are calculated by using the MCS method. Table 6-11 represents the differences between results of two approaches for the hairdryer. This analysis has been completed in all three products (representing most, middle, and the least dependency) and the rest of the comparison tables can be viewed in Appendix C. Normalized MCS values are used for the simplification of the grouping process.

Table 6-11. Comparison of Change Sensitivity at the Component Level

Hairdryer	N. CR	Level	Hairdryer	N. MCS	Level
Heating Coil	0.000	High	Front Housing	1.000	High
Heat Selector Switch	0.000		Heating Coil	0.706	
Cool Down Switch	0.014		Impeller	0.529	Medium
GFI Plug/power	0.030		Power Cord Retractor	0.471	
Motor/Fan	0.035		Heat Selector Switch	0.471	
Speed Sel. Switch	0.088		Back Housing	0.353	
Impeller	0.110		Black Airflow Div.	0.294	Low
Black Airflow Div.	0.110		Fins	0.294	
Black Elec.Comp	0.110		Cool Down Switch	0.294	
Retractor trigger	0.126		GFI Plug/power	0.235	
Power Cord Retr.	0.128		Retractor trigger	0.235	
Back Housing	0.140		Motor/Fan	0.176	
Front Housing	0.140		Black Electrical Comp	0.176	
Fins	0.226		Speed Sel. Switch	0.176	
Power Cord	0.370	Medium	Power Cord	0.118	
Switch Retainer B1	0.418		Switch Retainer Back1	0.118	
Switch Retainer B2	0.426		Switch Retainer Back2	0.118	
Cool Down Button	0.537		Heat Sel. Sw. Cover	0.059	
Retractor button	0.554		High/Low Switch Cover	0.059	
Insulation Cone	0.559		Cool Down Button	0.059	
Heat Sel. Sw. Cover	0.600		Insulation Cone	0.600	
High/Low Sw. Cover	0.602	Low	Retractor button	0.602	
Removable Inlet Protective Screen	1.000		Removable Inlet Protective Screen	1.000	
Metal Inlet Protector	1.000		Metal Inlet Protector	1.000	
Ceramic Ring	1.000		Ceramic Ring	1.000	
Exit Pro. Screen	1.000		Exit Protective Screen	1.000	
Cord Spacer	1.000		Cord Spacer	1.000	
Spacer	1.000		Spacer	1.000	

To validate the results of the MCS method at the product level and to see if the method can capture the design changes in product architectures accurately, an additional design change analysis is completed. To be able to keep the results comparable with the proposed method, the analysis has been applied on same products (phone, radio clock and hairdryer) by using the same approach (doubling the interface types in the same connections). The study compared the MCS value before and after the design change for each product. The results for average change per component are represented in Table 6-12.

Table 6-12. MCS Change Analysis for Selected Products

Rank	Product	% Change/Component
1	Phone	0.17
12	Radio Clock	0.39
21	Hairdryer	0.13

The results show that, even though the average design-dependency rankings were correctly predicted, the amount of the percentage change for a given product change in the system does not agree with the design dependency results. Based on the design dependency rankings obtained from the MCS method, the phone was expected to have the highest percentage change per component; however, it can be seen that the radio clock resulted with significantly higher values of change within the product architecture. These results do not trend as expected and do not create the projected relative amount of change.

In addition to the analysis on the product level, a component-level analysis is also completed to monitor the differences in assignments for the low, medium and high change sensitivity groups. Since there has been a single change in the architecture on the most change-sensitive connection, the sensitivity grouping of components were not affected as expected; however, a slight change in normalized values is observed. The comparison of MCS analysis on both original and modified architectures on the component level can be seen in Table 6-13.

Table 6-13. Component Level Change Analysis Comparison for MCS

Hairdryer	N Mod. MCS	Level	Hairdryer	N. MCS
Front Housing	1.00	High	Front Housing	1.00
Heating Coil	0.94		Heating Coil	0.71
Impeller	0.59	Medium	Impeller	0.53
Power Cord Retractor	0.53		Power Cord Retractor	0.47
Heat Selector Switch	0.47		Heat Selector Switch	0.47
Back Housing	0.47		Back Housing	0.35
Black Airflow Diverter	0.29	Low	Black Airflow Diverter	0.29
Fins	0.29		Fins	0.29
Cool Down Switch	0.29		Cool Down Switch	0.29
GFI Plug/power	0.24		GFI Plug/power	0.24
Retractor trigger	0.24		Retractor trigger	0.24
Motor/Fan	0.18		Motor/Fan	0.18
Black Electrical Comp	0.18		Black Electrical Comp	0.18
Speed Switch (ON/OFF)	0.18		Speed Switch (ON/OFF)	0.18
Power Cord	0.12		Power Cord	0.12
Switch Retainer Back1	0.12		Switch Retainer Back1	0.12
Switch Retainer Back2	0.12		Switch Retainer Back2	0.12
Heat Sel. Switch Cover	0.06		Heat Sel. Switch Cover	0.06
High/Low Switch Cover	0.06		High/Low Switch Cover	0.06
Cool Down Button	0.06		Cool Down Button	0.06
Insulation Cone	0.06		Insulation Cone	0.06
Retractor button	0.06		Retractor button	0.06
Removable Inlet Protective Screen	0.00		Removable Inlet Protective Screen	0.00
Metal Inlet Protector	0.00		Metal Inlet Protector	0.00
Ceramic Ring	0.00		Ceramic Ring	0.00
Exit Protective Screen	0.00		Exit Protective Screen	0.00
Cord Spacer	0.00		Cord Spacer	0.00
Spacer	0.00		Spacer	0.00

Unlike the proposed method, the MCS method neither assigns weights for different interface types nor incorporates the effect of indirect connections in the system for determining the module or part complexity. To demonstrate the importance of both of these aspects further analyses have been completed.

The first analysis investigates the importance of incorporating the effects of indirect connections given a design change in a product. To be able to show the difference in the results without indirect connections, the design change analysis has been carried out for the weighted-MCS values without use of the circuit model. This calculation is done by just applying the w-MCS formulation provided in Eq. 4.2 to calculate each connection and adding up the individual connection values for each component. The results are represented in Table 6-14.

Table 6-14. Analysis Ignoring Indirect Connections

Rank	Product	w-MCS of Original Des.	w-MCS of Modified Des.	% Change per connection
1	Phone	16.23	18.56	0.14
12	Radio Clock	8.17	9.16	0.12
21	Hair dryer	7.29	9.02	0.24

As it can be observed from the values in Table 6-14, the percentage change per connection without using the circuit model the accurate level of change per connection cannot be calculated. Due to the increased number of more complex connections within the product architecture as compared to other products, the change ratio for the hairdryer is higher relative to the other products. The phone, which is ranked as the least change resistant product in the set, shows less change per connection ratio in the calculations without the effects of indirect connections. Analysis on the individual component level is also completed and can be seen from Table 6-15. Similar to the previous case the relative percentage change values are not trending as anticipated by the model.

Table 6-15. Component Level Analysis Ignoring Indirect Connections

Hair Dryer	Non Cir	Level
Heating Coil	1.000	**High**
Heat Selector Switch	0.599	**Medium**
Cool Down Switch	0.339	
GFI Plug/power	0.210	**Low**
Motor/Fan	0.205	
Front Housing	0.172	
Impeller	0.146	
Speed Selector Switch	0.107	
Black Airflow Diverter	0.106	
Back Housing	0.100	
Power Cord Retractor	0.100	
Black Electrical Comp	0.087	
Retractor trigger	0.069	
Fins	0.053	
Power Cord	0.019	
Switch Retainer Back1	0.019	
Heat Selector Switch Cover	0.009	
High/Low Switch Cover	0.009	
Cool Down Button	0.009	
Insulation Cone	0.009	
Switch Retainer Back2	0.009	
Retractor button	0.009	
Removable Inlet Prot. Screen	0.000	
Metal Inlet Protector	0.000	
Ceramic Ring	0.000	
Exit Protective Screen	0.000	
Cord Spacer	0.000	
Spacer	0.000	

Table 6-15 represents the change sensitivity levels of components within the hairdryer calculated without considering the effects of indirect connections in the system. Compared with the results of the proposed method in Table 6-8, these results seem to under-evaluate the change sensitivity of most components. This situation might create design incompatibilities within the product, and if not realized earlier in the process, might increase the costs and duration of the redesign phase.

One last analysis is needed to validate another key aspect of the proposed method, which is the use of weights. As explained before, not all connections are created equal and they create different design dependencies in the product architecture. To be able to determine the effect of using weights in analysis, the use of the MCS method with circuit model is suggested. In this way the effect of weights in the analysis can be calculated by considering the effect of indirect connections in the system. Just like the previous analyses, hairdryer, radio clock and phone are analyzed with their original and modified architectures and the percentage of change per component in each product is compared. The results can be seen in the Table 6-16.

Table 6-16. Product Change Analysis Ignoring Weights of Interfaces

Rank	Product	MCS/w cir	Mod. MCS/w cir	% Change per connection
1	Phone	0.128	0.127	2.69
12	Radio Clock	0.159	0.156	2.08
21	Hair dryer	0.244	0.242	2.19

Table 6-16 represents the results for the case of non-weighted model, in which the effects of the indirect connections are incorporated in the study by use of circuit model.

In comparing the minimum average change resistance values for each product, the ranking of the products remains compatible with the proposed method suggesting that the phone is assigned as the most change transmitting product in the set. Even though this conclusion might seem in accordance with the other two products, the sensitivity of the “percent change per connection” in the analysis seems low, and change ratios for the radio clock and hairdryer products are not supporting the ranked results; therefore, it is hard to decide that the method without weights provides conclusive results. The validity of the method is also investigated at the component level, and the results are shown in Table 6-17.

Table 6-17. Component Level Analysis Ignoring Weights of Interfaces

Hairdryer	N.MCS Cir	Level
Fins	0.00	High
Cool Down Button	0.00	
Back Housing	0.02	
Power Cord Retractor	0.02	
Spacer	0.03	
Ceramic Ring	0.03	
Heat Selector Switch	0.04	
GFI Plug/power	0.04	
Switch Retainer Back2	0.07	
High/Low Switch Cover	0.12	
Retractor trigger	0.14	
Power Cord	0.18	
Black Electrical Comp	0.19	
Insulation Cone	0.22	
Heat Selector Switch Cover	0.27	
Retractor button	0.33	Medium
Removable Inlet Protective Screen	0.35	
Speed Selector Switch (ON/OFF)	0.48	
Impeller	0.49	
Cord Spacer	0.49	
Motor/Fan	0.53	
Front Housing	0.53	
Heating Coil	1.00	Low
Cool Down Switch	1.00	
Black Airflow Diverter	1.00	
Switch Retainer Back1	1.00	
Metal Inlet Protector	1.00	
Exit Protective Screen	1.00	

Component-level results for the hairdryer obtained from the analysis without the weights illustrate a significant difference from the results of the proposed method. The reader should take note that in this analysis different types of interfaces are not discriminated, and they are assumed to have equal weights in the product architecture; hence, they create equal levels of design dependencies within the product. In other words, the method automatically considers that the components with more connections would create more design dependencies within the system no matter how simply they are connected. Adding the effect of indirect connections to the equation amplifies the importance of interconnectivity and increases the change sensitivity of more interconnected components significantly, which results in exaggerated sensitivity levels of components because of their increased cumulative connections (direct + indirect connections) in the architecture. In the non-weighted method, even a very simple part such as a spacer (see Table 6-17) can be rated as highly change sensitive just because a component that it connects has with high a level of connectivity to other components. The chain reaction of connectivity can add up in such a way to misrepresent the change sensitivity level of individual components in the product architecture.

6.2.3 Comparison of Existing Architectures

As mentioned previously, one of the benefits of this method is enabling comparisons between similar product architectures in order to define the change sensitivity of the product. Three existing architectures for the same product have been compared. First, a comparison for the base model of two different brands has been completed for benchmarking purposes and secondly two different products of the same brand has been compared. This section presents the comparison of Mr. Coffee 12-cup base model (see Figure 6.12) with a base model of Krups® coffee makers. Additionally, the architecture of Mr. Coffee 12-cup base model is compared to another Mr. Coffee model that carries additional timer and programming features. Compared products can be seen in Figures 6.27 and 6.28.

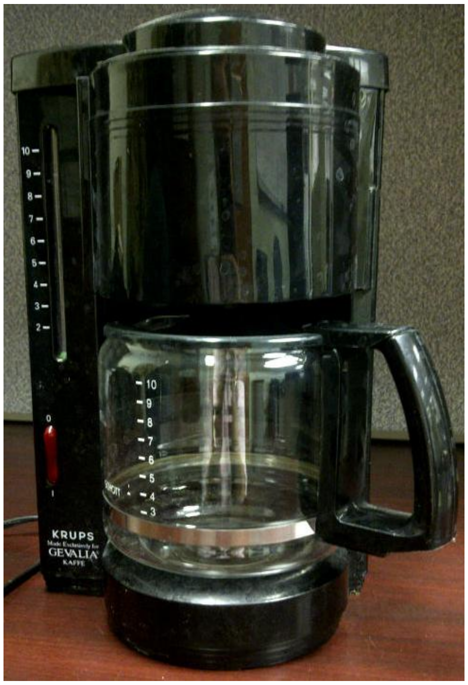


Figure 6.27. Krups Base Model Coffee Maker

Figure 6.28. Mr. Coffee with Additional Features [80]

The comparison of these three products has been made following the method step by step. As the result of the benchmarking between Mr. Coffee base model and Krups base model, it is observed that both coffee makers share very similar architectures, as expected. Architecture differences between two products have been observed in minor cosmetic parts such as the water level display. Both products have almost the same number of components. The model results also support similarity between the architectures. The most change-sensitive component is identified as the heating coil in Krups product as in the Mr. Coffee base model, and minimum change resistance values for these products are the same. These results can be seen in Table 6-18.

Table 6-18. Comparison of Base Models

Product	Min C.R	Component	Average Min C.R
Mr. Coffee	0.026	Heating Coil	0.14
Krups	0.026	Heating Coil	0.13

In comparing the Mr. Coffee base model to the Mr. Coffee model with display and programming features, the difference between calculated values is more pronounced. As can be seen in Table 6-19, in the advanced model PCBs are selected as the most change sensitive components of the architecture. Additionally, the change resistivity of Mr. Coffee with display feature is much lower than the base model, which indicates that it is less robust architecture compared to the base model.

Table 6-19. Comparison of Different Mr. Coffee Models

Product	Min C.R	Component	Average Min C.R
Mr. Coffee	0.026	Heating Coil	0.14
Mr. Coffee Display	0.025	PCBs	0.11

6.3 Discussion

This research provides very important information on product architectures for design engineers on change sensitivity of individual components as well as comparison of different product architectures. This kind of information is very beneficial, especially when there is a high probability of developing a family of products from a base architecture or future redesign or the product. The benefits of knowing the most change-sensitive component can be illustrated with an example. If a coffee maker is chosen as an example from the model results (see Appendix B), we know that the heating coil is the most change sensitive component in the product. Considering this finding closer it can be seen that the results are pretty accurate with respect to realistic industrial engineering issues. Imagine that a design team wants to change the power of heating coil and wants to install a more powerful one to keep the coffee warmer or to add a water boiling feature for the new product. Since the heating coil has a field interface with surrounding parts in addition to the physical attachments with power connections, changing the heating coil might create a lot of change propagation in the system. Increased heat in the coil might have an adverse effect on the surrounding materials: plastic materials might deform, leak water, or release hazardous chemicals in addition to the fire hazard created by miscellaneous surrounding cables or the glass carafe, which sits directly above the heating coil and might fail and crack. In other words, if the product is not designed with the specific design change in mind, material properties or surrounding components might not be compatible with the new heating coil. This creates an increased redesign period with major changes to the base architecture and maybe the need to change suppliers or manufacturing settings. Depending on the strategy or the reason for the upgrade there might be several recommendations to the design engineer. For this specific example, if the company is looking to develop a family of coffee makers, and the high-end segment needs the additional feature of water boiling, it might be more feasible to determine the material properties for all products in the family at the beginning. Even though at the individual level using a high quality material at a low quality low cost market segment might reduce profit margins, buying high quality components in bulk might provide savings in overall operations for that product family. Reduced supply chain costs (including transportation and cost/part) and minimal change in manufacturing set ups are some of the areas where cost savings can be achieved.

The results also show that ignoring indirect connections or the importance of different interface weights results in inaccurate and inconsistent results that might lead to unreliable designs or increased redesign efforts. This method provides accurate and comparable results by minimizing any bias that might be introduced by collecting data from engineering opinions. This can be used to the design engineers' advantage and provide fast and reliable results either during the design stage or as a benchmark against other products, especially in defining change sensitive components in complex products with complex interfaces with high integrality. The effects of adding new features or components to an existing architecture can be also predetermined just by using this approach of modeling product architectures without the need of building product prototypes. All the benefits of this approach would decrease the redesign time and effort for the engineering design team.

Another use of this method is as a metric to determine the more flexible architectures. This can be a very beneficial tool to be used during the new product development process when deciding between different product architectures of the same product. Depending on the strategies of the company and the market trends, design engineers might have to choose between two different product prototypes to manufacture and release to market. If the product is expected to evolve quickly with developing technologies, design engineers might prefer a more robust architecture to enable more easily modifications. The use of this method provides a simple objective manner to be able to discriminate between similar products based on their flexibility and enables better decisions during the design process.

CHAPTER 7

CONCLUSIONS & FUTURE WORK

This research presents a novel and objective method to assess design dependencies within product architectures and the effects of design change. The method was introduced, demonstrated and validated, and this chapter summarizes the principles of the method, its importance, and its limitations. This chapter also outlines the potential future work and possible enhancements for the method to widen its scope.

7.1 Dissertation Summary

Motivated by the efforts of reducing efforts in the redesign process, this study first focuses on the product architecture representation literature. In addition to other motivations discussed in Chapter 1, understanding the structure of a product architecture and relaying the necessary information correctly has been one of the most fundamental problems in engineering design. There have been several cases in the past where very important projects failed due to miscommunication of information such as the three-year delay of Boeing 787 Dreamliner [81]. First delays of this mega-project that were announced in September 2007 were explained as "ongoing challenges with out-of-sequence production work, including parts shortages, and remaining software and systems integration activities" [81]. As it can be seen, information sharing problems might derail progress even in the biggest and most expensive projects. Due to the importance of the correct interpretation and communication of product architecture information, numerous studies have been carried out on this topic, and this work begins by reviewing past studies in the area. Pertinent studies have been analyzed and grouped based on their approaches, and their different classification systems have been compared to see the corresponding definitions.

After the review of existing classifications and their shortcomings, this study consolidates these previous efforts in order to create a common language of interface representation for tracking change propagation within product architectures. As explained in Chapter 3, different approaches of architecture representations (functional approach, physical approach, or hybrid approach) serve different objectives. The functional approach has been proven to be more beneficial in terms of module determination, whereas the physical approach has been commonly used in design complexity calculations. As mentioned before, the nature of the connections between components defines the design dependency within the product architecture, [6] and the more complex the connections between the components, the more interdependent the product becomes [20]. Therefore, it is important to be able to determine the complexity of connections between components for any kind of design/redesign efforts. To minimize the engineering bias when determining design dependencies in a product architecture, this study uses the data that is extracted directly from the product itself. Doing so requires using the characteristics of the physical connections of the product architecture, which mandates that the proposed classification method employ a physical approach.

The interface classification system proposed in this study includes six different types of interfaces: attachment interfaces (A), spatial interfaces (S), communications and control interface (C), power interfaces (P), transfer interfaces (T), and field interfaces (F). However, one should note that using six basic interface types is not sufficient to define the connection characteristics within product architectures as combinations of these types also occur frequently. This study offers a new perspective on design dependencies where it is defined by the nature of the connection between components rather than the components themselves. Based on the complexity and the nature of the connections between two components in the product, the interface between them can be represented by one or more of these interface types. As the combination of interface types gets more complex (see the Table 3-2 to see the list of possible interface combinations) the dependency between components increase. Quantification of the relative strength of design dependencies is necessary to be able to determine the cumulative effect of each component in the system. The relative magnitude of these dependencies are formulated by using Modular Complexity Score (MCS) equation (see Eq. 4.1), which was first introduced by Dobberfuhl and Lange [8]. This study enhances the existing equation to acknowledge that different interface types might have different importance levels relative to each other and might carry different weights in terms of the dependencies that they create in the system. Therefore, modifications have been made to the formulation to create weighted-MCS (refer to Eq. 4.2).

In addition to the unbiased interpretation of product architecture representation, the weights for individual interface types should also be determined objectively. Just as has been done with architecture representation, the information that is extracted directly from the product itself has been used to determine the relative weights. The frequency of occurrence of different interface types is used as the input data to calculate their respective weights. Based on the assumptions in Section 4.2, as the number of occurrences of a specific interface type increases in the system, their respective weights compared to other interface types decrease. This assumption is based on the fact that the selection of interface types is a combination of several factors, such as the availability in the system, cost, ease of assembly, and so on.

After assigning weights to individual interface types, the value of each connection is calculated by using the weighted-MCS formulation; however, this calculation reflects only the dependency between two specific components within the architecture. This study considers a product as a network of components in which every part in the system is interconnected with each other either directly or indirectly. Therefore, any change in a connection has the potential to affect any component in the system. To be able to include the effect of indirect connections in the system, a novel electrical engineering analogy is proposed to model product architecture as an electrical circuit network. The connections between components are considered as resistors, and the components are considered as nodes where one or more resistors connect. Using the reciprocals of connection scores calculated from the weighted-MCS equation as resistance values in the circuit, this simple approach enables calculation of equivalent resistance between any desired components. The output retrieved from the model for each individual connection is referred as its “Change Resistance” (C.R) value, which represents the resistance to change transmission for that connection. This means that connections with lower C.R scores are more prone to transmit any changes to the next component given any changes in the system. The component which has the connection with minimum C.R value in the whole architecture is named as the most "change sensitive" component in the product, and engineers are advised to pay attention when making any design changes to that component. Based on the "change sensitivity" values defined at the component level, the sensitivity levels of high, medium, or low can be defined and design strategies can be developed to manage different levels of sensitivities.

In addition to its step-by-step demonstration, the proposed method has also been validated by using 21 example applications. Each product in the analysis set is chosen to include similar characteristics of small electro-mechanical household items. Different analyses have been made to confirm the validity of the results. Sensitivity analysis was used to validate the weight determination for the individual interface types. The analysis has been carried out using different subsets of products to calculate individual weights. The sensitivity analysis revealed consistent weight ranges when tested with five, ten, and fifteen random products from the set. Consistency of the results for 21 different products also proved the validity of the method. The results from individual models were also compatible with each other and the type of components selected as the most change sensitive in the architecture was consistent for all the products in within the analysis set. Final analysis carried on the existing coffee makers also follow the trends suggested by the proposed model by showing that a newer product with additional features might be more sensitive to design changes compared to a more simpler model. This analysis also aligns with the hypothesis introduced in the problem statement, which claims that product architectures can be modeled by using an electrical circuit analogy.

This work contributes to industrial engineering discipline in several topics. The next section elaborates more on the details of these contributions.

7.2 Research Contributions

The main objective in this research was to develop a novel and objective method to assess the design dependencies in a product's architecture and provide a guide on how it is used with applications to existing products. While accomplishing this objective there are main three objectives that are completed.

7.2.1 Comparison of Previous Approaches of Product Architecture Representation

As discussed in Chapter 3, correct product architecture representation is considered one of the most important aspects of product design. There have been numerous studies done in the topic starting in the 1980s, and while some of the studies overlap in terms of their classifications, others differentiate even from the point of approach. This research reviews all pertinent studies starting from Sanchez [51] in 1980 and classifies them into groups based on their approaches: functional, physical, and hybrid. In addition to grouping these studies, the research also correlates and maps the overlapping classifications to highlight unique definitions. It also provides a chart to represent the chronological evolution of the topic. This chart enables the reader to see how the past research builds on each other as time has progressed.

7.2.2 Introduction of a New Classification System for Tracking Change Propagation

Another objective completed in this research is the reconciliation of the inconsistencies that arise due to the different approaches in product architecture representation. To accomplish this goal, a simplified and practical system of existing architectures is introduced. A representation scheme that includes six main interface types has been defined (see Table 3-1). In addition, a method that enables the use of their combinations is introduced to define more complex connections.

7.2.3 Quantification of Design Dependencies within Product Architectures

By using data that can be directly extracted from the product itself, the research proposes an objective method to calculate the design dependencies within a product. Some studies use binary approaches to define the connections between components. Weighted-MCS formulation enables the quantification of relative importance of connections within the product architecture using an unbiased approach.

Overall, this research delivers a detailed analysis of the different interface types that exist in different product architectures and an unbiased step-by-step approach to evaluate the complexity of a product architecture based on its design dependencies. This approach provides an equation to determine relative connection strengths between modules and an electrical circuit analogy-based calculation procedure that enables one to estimate the effects of both direct and indirect connections within a product. The proposed method provides the first unbiased metric to quantify design dependencies in a product architecture by including effects of both direct and indirect connections of the system.

There are several benefits to having such an objective approach to assess the design dependencies in a product's architecture. Calculating the intensity of the connections gives a perspective on the "sensitive components" that might impact other components when redesigned. Evaluating the product architecture or any other design problem with an objective measure also enables comparisons between different product architectures during the conceptual design stage. This would also provide a benchmarking tool for evaluating competitors' products since product dissection is the only reliable method for collecting the necessary information on competitors' product architectures. With the proposed approach, problematic or inflexible designs can be easily determined, and costly changes can be avoided in advance.

7.3 Research Limitations & Potential Future Work

As this research reflects only applications and findings in the area of small electro-mechanical products, further research is necessary to expand the scope and use the full potential of the proposed method. Even though similar results are expected through the investigation of other industries with more complex product architectures such as automotive or aircraft/aerospace systems, investigation of a large variety of products from different size ranges might increase the accuracy of the information on interface type frequencies and relative weights. Additionally, existence of all different combinations of possible interface types can be investigated in more technologically advanced products. Moreover, quantification of economic benefits of determining change sensitive components and savings on redesign efforts could be another possible direction for a future study.

This study introduces modeling product architectures via analogy to electrical circuits with resistors representing the connections/interfaces between components. Investigation into the use of additional electrical circuit elements to visualize product architectures and represent other design elements is recommended. Additionally, in this work the frequency of occurrence of interface types is used to create a surrogate for the design dependencies in the product. Determination of the contributing factors for design dependencies such as supply chain issues, material cost, set-up cost, or assembly time could be used to determine the actual change resistance values rather than use of a comparative measure.

Finally, the electrical circuit models used in this work are static due to the deterministic nature of connections of components. However, investigating the effects of using a dynamic model might be beneficial and provide different insights for understanding the product architecture.

REFERENCES

[1] J. H. Mikkola, "Capturing the degree of modularity embedded in product architectures," *Journal of Product Innovation Management,* vol. 23, pp. 128-146, Mar 2006.

[2] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, "A Model-Based Method for Organizing Tasks in Product Development," *Research in Engineering Design-Theory Applications and Concurrent Engineering,* vol. 6, pp. 1-13, 1994.

[3] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: A review and new directions," *IEEE Transactions on Engineering Management,* vol. 48, pp. 292-306, Aug 2001.

[4] X. Lai and J. K. Gershenson, "DSM-based Product Representation for Design Process and Modularity," presented at the ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Brooklyn, NY, 2008.

[5] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "A network approach to define modularity of components in complex products," *Journal of Mechanical Design,* vol. 129, pp. 1118-1129, Nov 2007.

[6] C. Eckert, P. J. Clarkson, and W. Zanker, "Change and customisation in complex engineering domains," *Research in Engineering Design,* vol. 15, pp. 1-21, Mar 2004.

[7] K. M. M. Holtta and K. N. Otto, "Incorporating design effort complexity measures in product architectural design and assessment," *Design Studies,* vol. 26, pp. 463-485, Sep 2005.

[8] A. Dobberfuhl and M. W. Lange, "Interfaces per Module Is There an Ideal Number?," presented at the ASME International Design Engineering Technical Conferences-Computers and Information in Engineering Design Conference, San Diego, CA, 2009.

[9] B. Bettig and J. K. Gershenson, "Module Interface Representation," presented at the ASME Design Engineering Technical Conferences, Philadelphia, PA, 2006.

[10] J. M. Aughenbaugh, "The value of using imprecise probabilities in engineering design," *Journal of Mechanical Design,* vol. 128, pp. 969-979, Jul 2006.

[11] A. H. Van De Ven and A. L. Delbecq, "The Effectiveness of Nominal, Delphi, and Interacting Group Decision Making Processes," *The Academy of Management Journal,* vol. 17, pp. 605-621, 1974.

[12] M. V. Martin and K. Ishii, "Design for variety: developing standardized and modularized product platform architectures," *Research in Engineering Design-Theory Applications and Concurrent Engineering,* vol. 13, pp. 213-235, Nov 2002.

[13] P. K. Palani Rajan, M. Van Wie, M. I. Campbell, K. L. Wood, and K. N. Otto, "An empirical foundation for product flexibility," *Design Studies,* vol. 26, pp. 405-438, 2005.

[14] P. J. Clarkson, C. Simons, and C. Eckert, "Predicting Change Propagation in Complex Design," *Journal of Mechanical Design,* vol. 126, pp. 788-797, 2004.

[15] H. J. Thevenot and T. W. Simpson, "A Product Dissection-Based Methodology to Benchmark Product Family Design Alternatives," *Journal of Mechanical Design,* vol. 131, pp. 1-9, Apr 2009.

[16] T. U. Pimmler and S. D. Eppinger, "Integration analysis of product decompositions," in *ASME 6th International Conference on Design Theory and Methodology*, Minneapolis, MN, 1994.

[17] T. Jarrat, C. Eckert, and P. J. Clarkson, "Development of a Product Model to Support Engineering Change Management," in *TMCE*, Lausanne, Switzerland, 2004.

[18] A. Ericsson and G. Erixon, *Controlling Design Variants*. New York: ASME Press, 1999.

[19] S. K. Fixson, "Modularity and commonality research: past developments and future opportunities," *Concurrent Engineering-Research and Applications,* vol. 15, pp. 85-111, Jun 2007.

[20] K. Holtta-Otto and O. de Weck, "Degree of modularity in engineering systems and products with technical and business constraints," *Concurrent Engineering-Research and Applications,* vol. 15, pp. 113-126, Jun 2007.

[21] K. Ulrich, "The Role of Product Architecture in the Manufacturing Firm," *Research Policy,* vol. 24, pp. 419-440, May 1995.

[22] C. Y. Baldwin and K. B. Clark, "Managing in an age of modularity," *Harvard Business Review,* vol. 75, pp. 84-94, Sep-Oct 1997.

[23] R. B. Stone, K. L. Wood, and R. H. Crawford, "A Heuristic Model for Identifying Modules for Product Architectures," *Design Studies,* vol. 21, pp. 5-31, 2000.

[24] K. Otto and O. de Weck, "Metrics for Assessing Coupling Density and Modularity in Complex Products and Systems," presented at the ASME International Design Engineering Technical Conferences-Computers and Information in Engineering, Las Vegas, NV, 2007.

[25] X. H. Meng, Z. H. Jiang, and G. Q. Huang, "On the module identification for product family development," *International Journal of Advanced Manufacturing Technology,* vol. 35, pp. 26-40, Nov 2007.

[26] B. K. Bairstow, O. de Weck, J. Sobieszczanski-Sobieski, J. P. L. B. Bairstow, Pasadena, CA; O. de Weck, Massachusetts Institute of Technology, Cambridge, MA; and J. Sobieszczanski-Sobieski, NASA Langley Research Center, Hampton, VA, and AIAA-2007-1879, "System-level Optimization and Concurrent Engineering: A Comparative Experiment," in *3rd AIAA Multidisciplinary Design Optimization Specialist Conference*, Honolulu, HI, 2007.

[27] S. Gupta and G. E. Okudan, "Computer-Aided Generation of Modularised Conceptual Designs with Assembly and Variety Considerations," *Journal of Engineering Design,* vol. 19, pp. 553-551, Dec 2008.

[28] K. Holtta-Otto, V. Tang, and K. Otto, "Analyzing module commonality for platform design using dendrograms," *Research in Engineering Design,* vol. 19, pp. 127-141, Nov 2008.

[29] X. Lai and J. K. Gershenson, "Representation of similarity and dependency for assembly modularity," *International Journal of Advanced Manufacturing Technology,* vol. 37, pp. 803-827, Jun 2008.

[30] D. Braha and O. Maimon, "The measurement of a design structural and functional complexity," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 28, pp. 527-535, 1998.

[31] B. El-Haik and K. Yang, "The components of complexity in engineering design," *IIE Transactions,* vol. 31, pp. 925-934, 1999.

[32] N. P. Suh, *The principles of design*. New York: Oxford University Press, 1990.

[33] J. D. Summers and J. J. Shah, "Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability," *Journal of Mechanical Design,* vol. 132, pp. -, Feb 2010.

[34] M. Giffin, O. de Weck, G. Bounova, R. Keller, C. Eckert, and P. J. Clarkson, "Change Propagation Analysis in Complex Technical Systems," *Journal of Mechanical Design,* vol. 131, pp. -, Aug 2009.

[35] P. Crespo, O. K. Kremer, C. S. Tucker, and L. Medina, "An Analysis of Compexity Measures in Product Design and Development," presented at the ASME International Design Engineering Technical Conferences-Computers and Information in Engineering Conference, Chicago, Illinois, 2012.

[36] R. Keller, T. Eger, C. Eckert, and P. J. Clarkson, "Visualising change propagation," presented at the 15th International Conference on Engineering Design (ICED05), Melbourne, Australia, 2005.

[37] E. Suh, O. de Weck, and D. Chang, "Flexible product platforms: framework and case study," *Research in Engineering Design,* vol. 18, pp. 67-89, 2007.

[38] D. V. Steward, "The Design Structure-System - a Method for Managing the Design of Complex-Systems," *Ieee Transactions on Engineering Management,* vol. 28, pp. 71-74, 1981.

[39] V. Hubka, *Theorie der Maschinensysteme: Grundlagen einer wissenschaftlichen Konstruktionslehre*: Springer, 1973.

[40] H. M. Paynter, *Analysis and design of engineering systems*. Boston: The M.I.T. Press, 1961.

[41] G. Pahl and W. Beitz, *Engineering design : a systematic approach*. London ; New York: Springer, 1996.

[42] M. S. Hundal, "A Systematic Method for Developing Function Structures, Solutions and Concept Variants," *Mechanism and Machine Theory,* vol. 25, pp. 243-256, 1990.

[43] D. Cutherell, "Product Architecture," in *The PDMA Handbook of New Product Development*, M. D. Rosenau, Ed., ed New York: Wiley 1996, pp. 217-235

[44] K. T. Ulrich and S. D. Eppinger, *Product design and development*. New York: McGraw-Hill, 1995.

[45] A. Kusiak and C. C. Huang, "Development of modular products," *Ieee Transactions on Components Packaging and Manufacturing Technology Part A,* vol. 19, pp. 523-538, Dec 1996.

[46] K. T. Ulrich and W. P. Seering, "Synthesis of schematic descriptions in mechanical design," *Research in Engineering Design,* vol. 1, pp. 3-18, 1989.

[47] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood, "A functional basis for engineering design: Reconciling and evolving previous efforts," *Research in Engineering Design-Theory Applications and Concurrent Engineering,* vol. 13, pp. 65-82, Mar 2002.

[48] K. N. Otto and K. L. Wood, *Product design : techniques in reverse engineering and new product development*. Upper Saddle River, NJ: Prentice Hall, 2001.

[49] R. E. Thebeau, "Knowledge Management of System Interfaces and Interactions for Product Development Processes," Master of Science, System Design & Management Program, Massachusetts Institute of Technology, Boston, 2001.

[50] V. Godthi, J. Yoo, C. B. Arnold, T. W. Simpson, and S. Kumara, "Product Design Using Interface-based Module Description," presented at the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Montreal, Canada, 2010.

[51] R. Sanchez, "Towards a Science of Strategic Product Design: System Design, Component Modularity, and Product Leveraging Strategies," presented at the 2nd International Product Development Management Conference on New Approaches to Development and Engineering, Gothenburg, Sweden, 1994.

[52] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "Designing Modular and Integrative Systems," presented at the ASME 2000 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Baltimore, MD, 2000.

[53] J. K. Gershenson, G. J. Prasad, and Y. Zhang, "Product modularity: measures and design methods," *Journal of Engineering Design,* vol. 15, pp. 33-51, Feb 2004.

[54] O. Ariyo, C. Eckert, and P. J. Clarkson, "Towards a Decntralised Appraoch to Modelling Connectivity in Complex Products," presented at the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Montreal, Canada, 2010.

[55] J. C. Lockledge and F. A. Salustri, "Defining the Engine Design Process," *Journal of Engineering Design,* vol. 10, pp. 109-124, 1999/06/01 1999.

[56] S. Sosale, M. Hashemian, and P. Gu, "Product modularization for reuse and recycling," presented at the ASME International Mechanical Engineering Congress and Exposition, FAIRFIELD, NJ, 1997.

[57] A. H. Tilstra, C. C. Seepersad, and K. L. Wood, "Analysis of Product Flexibility for Future Evolution Based on Design Guidelines and a High-Definition Design Structure Matrix," *ASME Conference Proceedings,* vol. 2009, pp. 951-964, 2009.

[58] R. A. DeCarlo and P. M. Lin, *Linear Circuit Analysis: Time Domain, Phasor, and Laplace Transform Approaches*: Oxford University Press, 2001.

[59] "MATLAB," R2011a ed. Natick, Massachusetts: The MathWorks Inc., 2011.

[60] K. T. Ulrich and S. Pearson, "Assessing the Importance of Design Through Product Archaeology," *Management Science,* vol. 44, pp. 352-369, 1998.

[61] G. Boothroyd, P. Dewhurst, and W. A. Knight, *Product Design for Manufacture and Assembly*: Marcel Dekker, 2002.

[62] Available: http://www.toolselect.com/product/detail/Durabuilt-085-06-0102

[63] Available: http://blackanddeckermediakit.com/power-tools-accessories/images/38-5-2-amp-corded-drilldriver/

[64] Available: http://www.rivalproducts.com/product.aspx?pid=968

[65] Available: http://chefdruck.blogspot.fr/2010/06/aroma-rice-cooker-giveaway.html

[66] Available: http://www.bondfaro.com.br/forno-eletrico--pg2.html

[67] Available: http://salestores.com/gpxc208bamfm.html

[68] Available:http://www.epinions.com/prices/Revlon-Spa-Pore-Facial-Brush-Set/display_~latest_prices/pp_~1/sort_~avg_rating/sort_dir_~des/sec_~wtb?sb=1

[69] Available: http://www.amazon.com/Fisher-Price-Seen-Say-TFS/dp/B001D0C70C=1

[70] Available: http://store.kodak.com/store/ekconsus/en_US/pd/FUN_SAVER_Camera/productID.164407600

[71] Available:http://egagroupusa.net/logitech-m310-wireless-mouse-peacock-blue-910-001917.html

[72] Available:http://www.toolking.com/1/arrow_fastener_5700_staple_nail_gun_1051804563.php

[73] Available: http://www.sears.com/kalorik-lime-hand-mixer/p-00872290000P

[74] Available: http://www.amazon.com/dp/B001KBY9I2/ref=asc_df_B001KBY9I22092284?smid=ATVPDKIKX0DER&tag=nextagus0006062-20&linkCode=asn&creative=395093&creativeASIN=B001KBY9I2

[75] Available: http://www.rivalproducts.com/product.aspx?pid=2307

[76] Available: http://www.tristatecamera.com/product/Conair_Conair_Hair_Dryer_Ionic_Ceramic_Cord_Rewind_Conc_209GWP.html

[77] Available: http://newyorkrenovator.com/2009/10/looking-at-electric-heaters.html

[78] Available:http://www.ebay.com.sg/itm/Conairphone-PR5007-Big-button-assorted-colors-/180545284916

[79] Available: http://blackanddecker.com/power-tools/CHV1408.aspx

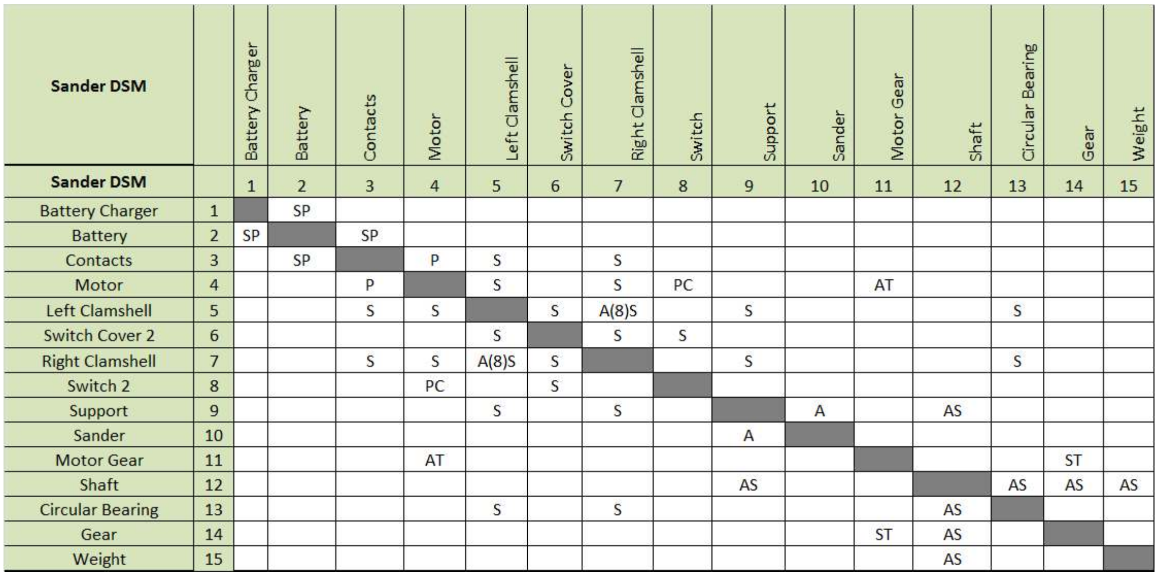
[80] Available:http://www.buy.com/prod/mr-coffee-12-cup-programmable-coffee-maker-black-tfx23/203497362.html

[81] D. Terdiman. (2011, January 13). *Boeing's 787 Dreamliner: A legacy of delays*. Available: http://news.cnet.com/8301-13772_3-20028854-52.html

APPENDIX A

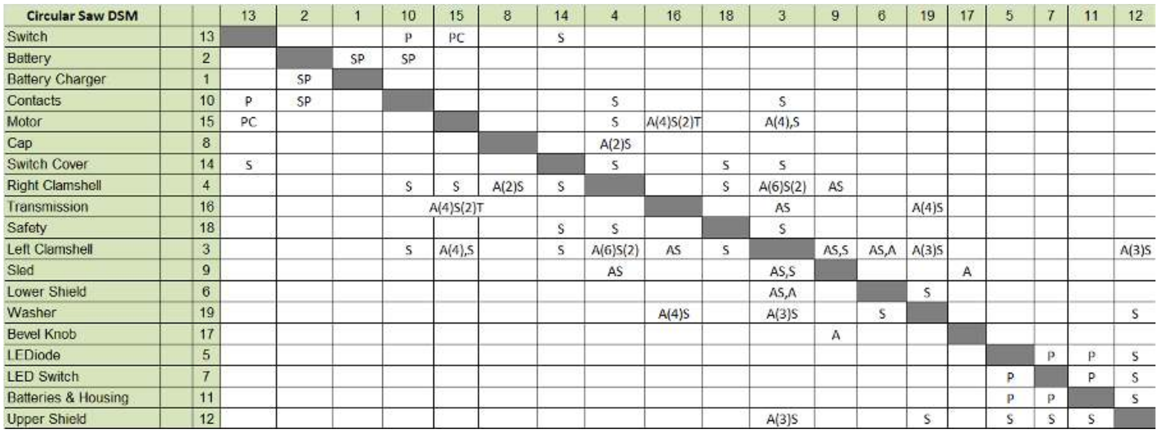
INTERFACE DSMS OF ANALYZED PRODUCTS

A.1. Sander Interface DSM



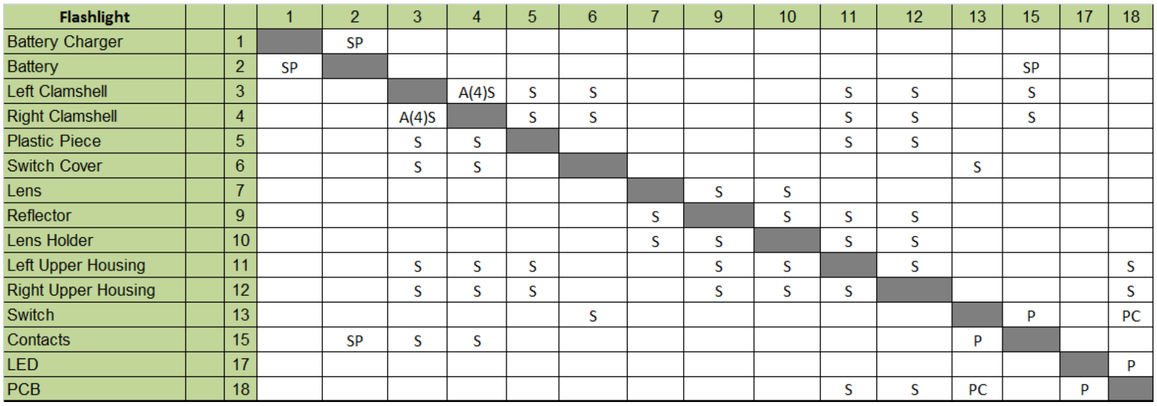
Sander DSM		Battery Charger	Battery	Contacts	Motor	Left Clamshell	Switch Cover	Right Clamshell	Switch	Support	Sander	Motor Gear	Shaft	Circular Bearing	Gear	Weight
Sander DSM		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Battery Charger	1		SP													
Battery	2	SP		SP												
Contacts	3		SP		P	S		S								
Motor	4			P		S		S	PC			AT				
Left Clamshell	5			S	S		S	A(8)S		S				S		
Switch Cover 2	6					S		S	S							
Right Clamshell	7			S	S	A(8)S	S			S				S		
Switch 2	8				PC		S									
Support	9					S		S			A		AS			
Sander	10									A						
Motor Gear	11				AT										ST	
Shaft	12									AS				AS	AS	AS
Circular Bearing	13					S		S					AS			
Gear	14											ST	AS			
Weight	15												AS			

A.2. Circular Saw Interface DSM



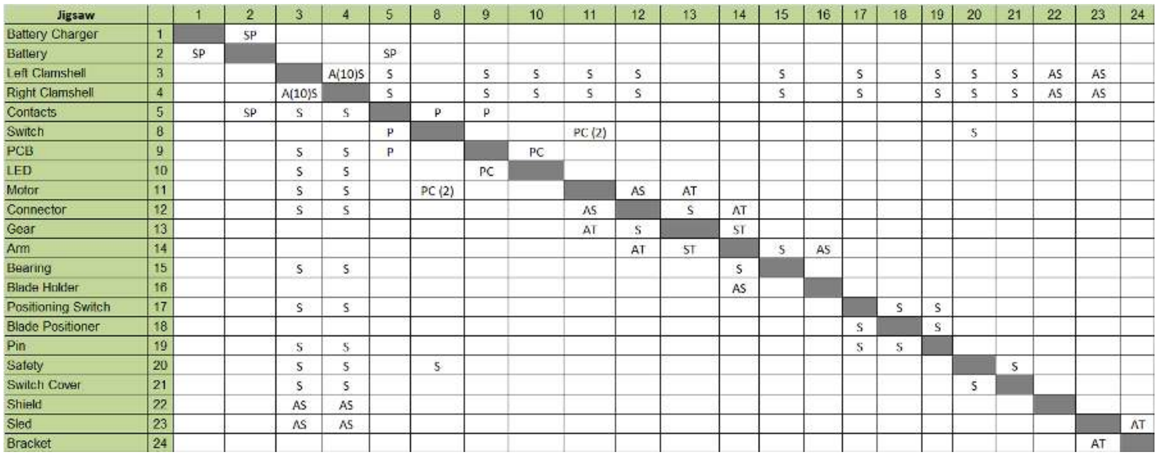
Circular Saw DSM			13	2	1	10	15	8	14	4	16	18	3	9	6	19	17	5	7	11	12
Switch		13				P	PC		S												
Battery		2			SP	SP															
Battery Charger		1		SP																	
Contacts		10	P	SP						S			S								
Motor		15	PC							S	A(4)S(2)T		A(4),S								
Cap		8								A(2)S											
Switch Cover		14	S							S		S	S								
Right Clamshell		4				S	S	A(2)S	S			S	A(6)S(2)	AS							
Transmission		16					A(4)S(2)T						AS			A(4)S					
Safety		18							S	S			S								
Left Clamshell		3				S	A(4),S		S	A(6)S(2)	AS	S		AS,S	AS,A	A(3)S					A(3)S
Sled		9								AS			AS,S				A				
Lower Shield		6											AS,A			S					
Washer		19									A(4)S		A(3)S		S						S
Bevel Knob		17												A							
LEDiode		5																	P	P	S
LED Switch		7																P		P	S
Batteries & Housing		11																P	P		S
Upper Shield		12											A(3)S			S		S	S	S	

A.3. Flashlight Interface DSM



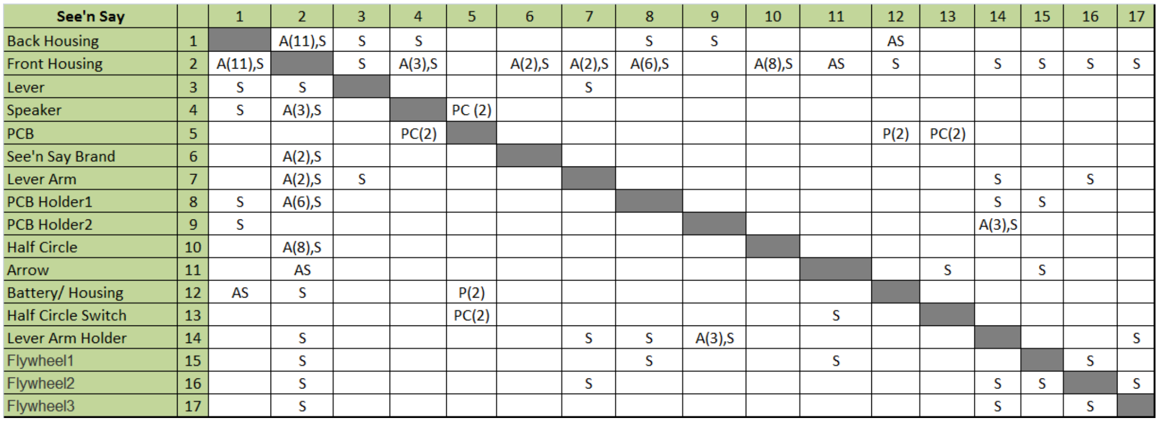
Flashlight			1	2	3	4	5	6	7	9	10	11	12	13	15	17	18
Battery Charger		1	▨	SP													
Battery		2	SP	▨											SP		
Left Clamshell		3			▨	A(4)S	S	S				S	S		S		
Right Clamshell		4			A(4)S	▨	S	S				S	S		S		
Plastic Piece		5			S	S	▨					S	S				
Switch Cover		6			S	S		▨						S			
Lens		7							▨	S	S						
Reflector		9							S	▨	S	S	S				
Lens Holder		10							S	S	▨	S	S				
Left Upper Housing		11			S	S	S			S	S	▨	S				S
Right Upper Housing		12			S	S	S			S	S	S	▨				S
Switch		13						S						▨	P		PC
Contacts		15		SP	S	S								P	▨		
LED		17														▨	P
PCB		18										S	S	PC		P	▨

A.4. Jigsaw Interface DSM



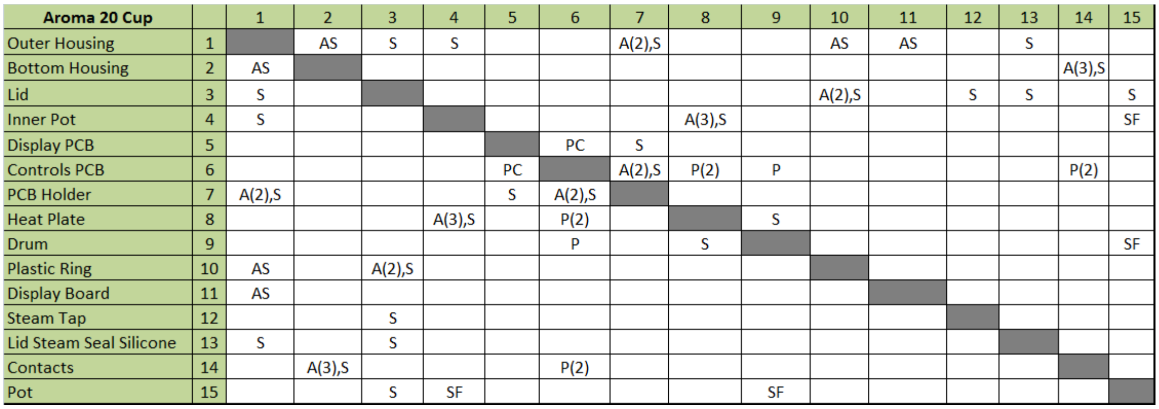
Jigsaw		1	2	3	4	5	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Battery Charger	1	▨	SP																				
Battery	2	SP	▨			SP																	
Left Clamshell	3			▨	A(10)S	S		S	S	S	S			S		S		S	S	S	AS	AS	
Right Clamshell	4			A(10)S	▨	S		S	S	S	S			S		S		S	S	S	AS	AS	
Contacts	5		SP	S	S	▨	P	P															
Switch	8					P	▨			PC (2)									S				
PCB	9			S	S	P		▨	PC														
LED	10			S	S			PC	▨														
Motor	11			S	S		PC (2)			▨	AS	AT											
Connector	12			S	S					AS	▨	S	AT										
Gear	13									AT	S	▨	ST										
Arm	14										AT	ST	▨	S	AS								
Bearing	15			S	S								S	▨									
Blade Holder	16												AS		▨								
Positioning Switch	17			S	S											▨	S	S					
Blade Positioner	18															S	▨	S					
Pin	19			S	S											S	S	▨					
Safety	20			S	S		S												▨	S			
Switch Cover	21			S	S														S	▨			
Shield	22			AS	AS																▨		
Sled	23			AS	AS																	▨	AT
Bracket	24																					AT	▨

A.5. See'n Say Interface DSM



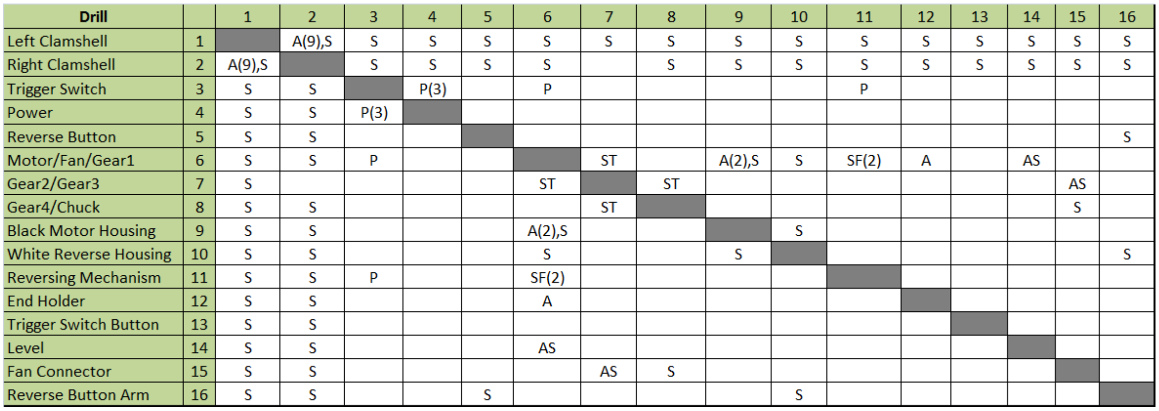
See'n Say		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Back Housing	1	■	A(11),S	S	S				S	S			AS					
Front Housing	2	A(11),S	■	S	A(3),S		A(2),S	A(2),S	A(6),S		A(8),S	AS	S		S	S	S	S
Lever	3	S	S	■				S										
Speaker	4	S	A(3),S		■	PC (2)												
PCB	5				PC(2)	■							P(2)	PC(2)				
See'n Say Brand	6		A(2),S				■											
Lever Arm	7		A(2),S	S				■							S		S	
PCB Holder1	8	S	A(6),S						■						S	S		
PCB Holder2	9	S								■					A(3),S			
Half Circle	10		A(8),S								■							
Arrow	11		AS									■		S		S		
Battery/ Housing	12	AS	S			P(2)							■					
Half Circle Switch	13					PC(2)						S		■				
Lever Arm Holder	14		S					S	S	A(3),S					■			S
Flywheel1	15		S						S			S				■	S	
Flywheel2	16		S					S							S	S	■	S
Flywheel3	17		S												S		S	■

A.6. Aroma Rice Cooker Interface DSM



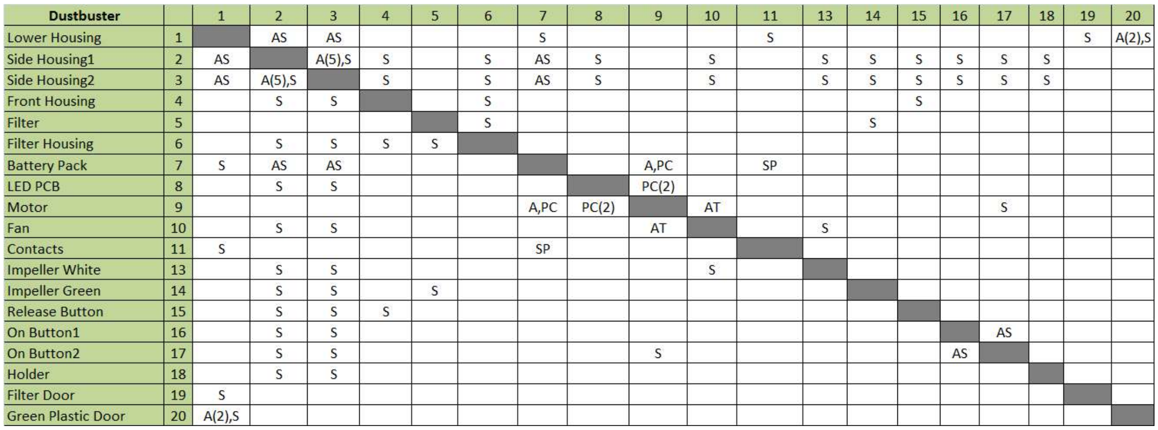
Aroma 20 Cup		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Outer Housing	1	■	AS	S	S			A(2),S			AS	AS		S		
Bottom Housing	2	AS	■												A(3),S	
Lid	3	S		■							A(2),S		S	S		S
Inner Pot	4	S			■				A(3),S							SF
Display PCB	5					■	PC	S								
Controls PCB	6					PC	■	A(2),S	P(2)	P					P(2)	
PCB Holder	7	A(2),S				S	A(2),S	■								
Heat Plate	8				A(3),S		P(2)		■	S						
Drum	9						P		S	■						SF
Plastic Ring	10	AS		A(2),S							■					
Display Board	11	AS										■				
Steam Tap	12			S									■			
Lid Steam Seal Silicone	13	S		S										■		
Contacts	14		A(3),S				P(2)								■	
Pot	15			S	SF					SF						■

A.7. Drill Interface DSM



Drill		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Left Clamshell	1		A(9),S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Right Clamshell	2	A(9),S		S	S	S	S		S	S	S	S	S	S	S	S	S
Trigger Switch	3	S	S		P(3)		P					P					
Power	4	S	S	P(3)													
Reverse Button	5	S	S														S
Motor/Fan/Gear1	6	S	S	P				ST		A(2),S	S	SF(2)	A		AS		
Gear2/Gear3	7	S					ST		ST							AS	
Gear4/Chuck	8	S	S					ST								S	
Black Motor Housing	9	S	S				A(2),S				S						
White Reverse Housing	10	S	S				S			S							S
Reversing Mechanism	11	S	S	P			SF(2)										
End Holder	12	S	S				A										
Trigger Switch Button	13	S	S														
Level	14	S	S				AS										
Fan Connector	15	S	S					AS	S								
Reverse Button Arm	16	S	S			S					S						

A.8. Dustbuster Interface DSM

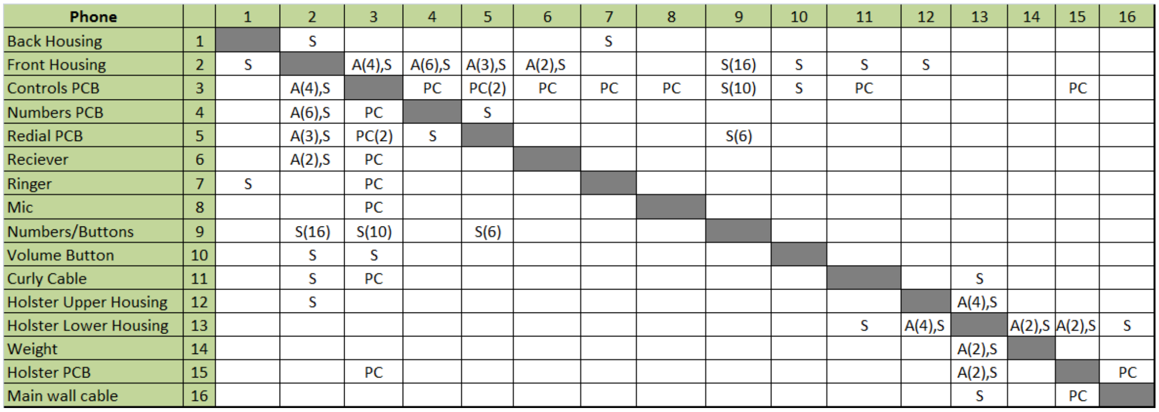


Dustbuster		1	2	3	4	5	6	7	8	9	10	11	13	14	15	16	17	18	19	20
Lower Housing	1		AS	AS				S				S							S	A(2),S
Side Housing1	2	AS		A(5),S	S		S	AS	S		S		S	S	S	S	S	S		
Side Housing2	3	AS	A(5),S		S		S	AS	S		S		S	S	S	S	S	S		
Front Housing	4		S	S			S								S					
Filter	5						S							S						
Filter Housing	6		S	S	S	S														
Battery Pack	7	S	AS	AS						A,PC		SP								
LED PCB	8		S	S						PC(2)										
Motor	9							A,PC	PC(2)		AT						S			
Fan	10		S	S						AT			S							
Contacts	11	S						SP												
Impeller White	13		S	S							S									
Impeller Green	14		S	S		S														
Release Button	15		S	S	S															
On Button1	16		S	S													AS			
On Button2	17		S	S						S						AS				
Holder	18		S	S																
Filter Door	19	S																		
Green Plastic Door	20	A(2),S																		

A.9. Hairdryer Interface DSM

Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Removeable Inlet Protective	1		S																										
Back Housing	2	S		S	A(2),S													S								A(2),S			
Metal Inlet Protector	3		S																										
Front Housing	4		A(2),S			S	S	S	S					S		S	S	S				S	S	S	S		S	S	S
Heat Selector Switch Cover	5				S																		S						
High/Low Switch Cover	6				S																			S					
Cool Down Button	7				S																	S							
Insulation Cone	8				S									S															
Ceramic Ring	9													S															
Impeller	10											A(3),S							S							A(2),S			AS
Black Airflow Diverter	11										A(3),S		AS	S															
Motor/Fan	12											AS			PC														
Fins	13				S				S	S		S			S	A													
Heating Coil	14												PC	S		P			PC	P,PCT	P	PC							
Black Electrical Comp	15				S									A	P						P								
Exit Protective Screen	16				S																								
Power Cord	17		S		S																					S			
GFI Plug/power	18										S				PC						P					S			
Heat Selector Switch	19														P,PCT							PC	S						
Speed Selector Switch (ON/O	20														P	P			P					S					
Cool Down Switch	21				S			S							PC					PC									
Switch Retainer Back1	22				S	S														S									
Switch Retainer Back2	23				S		S														S								
Cord Spacer	24				S																								
Power Cord Retractor	25		A(2),S								A(2),S							S	S										S
Spacer	26				S																								
Retractor button	27				S																								S
Retractor trigger	28				S						AS															S		S	

A.10. Phone Interface DSM



Phone		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Back Housing	1	■	S					S									
Front Housing	2	S	■	A(4),S	A(6),S	A(3),S	A(2),S			S(16)	S	S	S				
Controls PCB	3		A(4),S	■	PC	PC(2)	PC	PC	PC	S(10)	S	PC				PC	
Numbers PCB	4		A(6),S	PC	■	S											
Redial PCB	5		A(3),S	PC(2)	S	■				S(6)							
Reciever	6		A(2),S	PC			■										
Ringer	7	S		PC				■									
Mic	8			PC					■								
Numbers/Buttons	9		S(16)	S(10)		S(6)				■							
Volume Button	10		S	S							■						
Curly Cable	11		S	PC								■		S			
Holster Upper Housing	12		S										■	A(4),S			
Holster Lower Housing	13											S	A(4),S	■	A(2),S	A(2),S	S
Weight	14													A(2),S	■		
Holster PCB	15			PC										A(2),S		■	PC
Main wall cable	16													S		PC	■

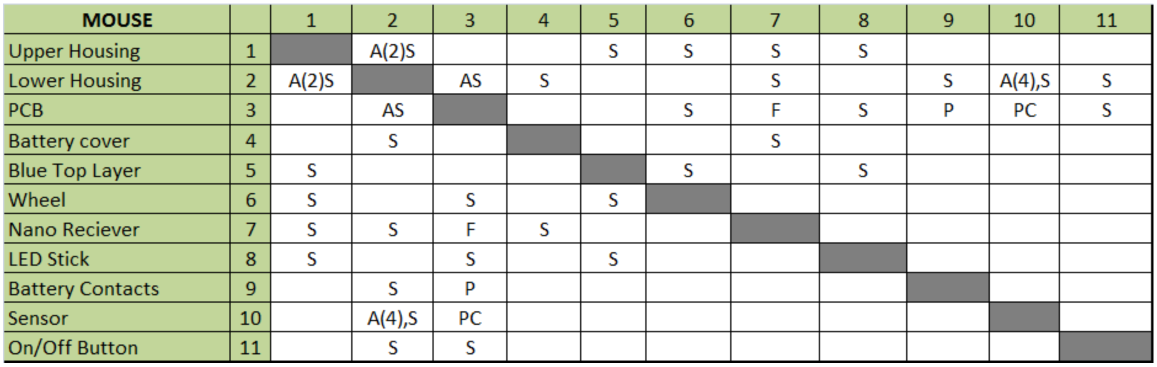
A.11. Oven Interface DSM

OVEN		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Handle1	1			A(3),S																									
Handle2	2			A(5),S																									
Cover Panel	3	A(3),S	A(5),S				S		A(4),S	A(4),S																			
Side Panel 1	4						A(2),S		A(3),S	A(4),S		A(4),S							A(2),S			A(4),S	S	S	S	S	S		
Side Panel 2	5						A(2),S	A(2),S	A(3),S	A(3),S		A(4),S											AS	S	S	S	S		
Front Panel	6			S	A(2),S	A(2),S		AS		A(2),S																			
Door	7					A(2),S	AS			A(3)																			A(2),S(D
Back Panel	8			A(4),S	A(3),S	A(3),S				A(3),S			A	A															
Under Panel	9			A(4),S	A(4),S	A(3),S	A(2),S	A(3)	A(3),S																			S	
Control Panel	10														S	S	S	S		A(2),S	A(2),S								
Top Panel	11				A(4),S	A(4),S																							
Feet1	12									A																			
Feet2	13									A																			
Temperature Button	14										S								S										
Function Button	15										S									S									
Time Button	16										S										S								
LED	17										S										P							P	
Temperature Switch	18				A(2),S										S										P			P	
Function Switch	19										A(2),S					S					PC	C				P			
Time Switch	20										A(2),S						S	P		PC				P				P	
Rotisserie Rotor	21				A(4),S															C			T					P	
Rotisserie End	22				S	AS																T							
Ceramic Rod1	23				S	S															P				P				
Ceramic Rod2	24				S	S													P					P			P		
Ceramic Rod3	25				S	S														P							P		
Ceramic Rod4	26				S	S																			P	P			
Power	27									S								P	P		P	P							
Door Handle	28							A(2),S(D)																					

A.12. Heater Interface DSM

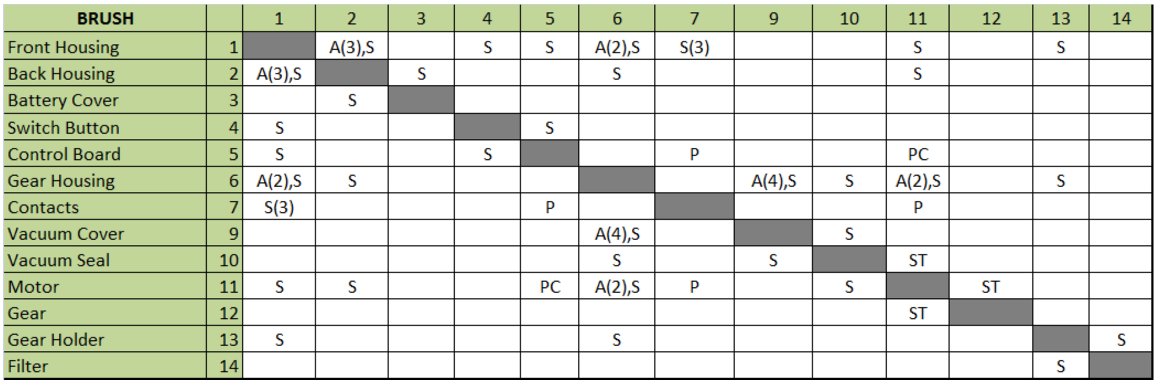
HEATER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Top Cover	1		S	S		A(4)																				
Surround Control Switch	2	S			S	S																				
Metal Outer Mesh	3	S						A(2),S					S													
Surround Metals	4		S			S																				
Inner Metal Housing	5	A(4)	S		S																					
Plastic side housing (fron	6							S	S	A(2),S		A(2),S	A(2),S					S		S	S		AS			
Plastic side housing (back	7			A(2),S			S		AS			A(2),S	S					S								
Control Pannel	8						S	AS												AS		A(2)S				
Lower Housing	9						A(2),S				A(8),S															
Lower Housing Cover	10									A(8),S																
Heating Unit Housing	11						A(2),S	A(2),S					A(3),S	A(2),S	S	S	S	S	A						A(2),S	
H.U Housing Cover	12			S			A(2),S	S				A(3),S														
Fan/Motor	13											A(2),S			P	P		S			PC			P	P	
Heating Elements1	14											S		P		P					PC				P	
Heating Elements2	15											S		P	P						PC				P	
Heating Spacersx2	16											S														
Fan Mesh	17						S	S				S		S												
Controller2 Cover	18											A													S	
Thermostat	19						S		AS													S		P	PF	P
Rotary Switch	20						S							PC	PC	PC							S	P		P
Thermostat Knob	21								A(2)S											S						
Rotary Switch Knob	22						AS														S					
LED	23													P						P	P					P
Controller2	24											A(2),S		P	P	P			S	PF						
Power	25																			P	P			P		

A.13. Mouse Interface DSM



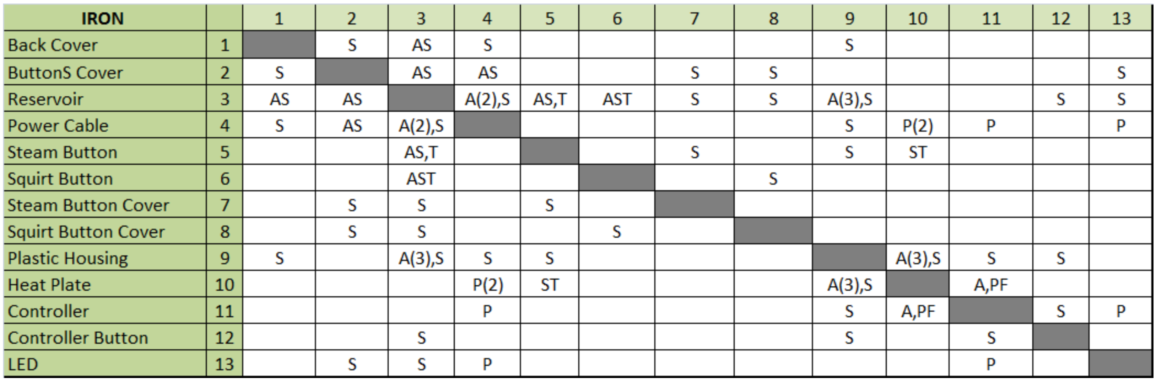
MOUSE		1	2	3	4	5	6	7	8	9	10	11
Upper Housing	1	▨	A(2)S			S	S	S	S			
Lower Housing	2	A(2)S	▨	AS	S			S		S	A(4),S	S
PCB	3		AS	▨			S	F	S	P	PC	S
Battery cover	4		S		▨			S				
Blue Top Layer	5	S				▨	S		S			
Wheel	6	S		S		S	▨					
Nano Reciever	7	S	S	F	S			▨				
LED Stick	8	S		S		S			▨			
Battery Contacts	9		S	P						▨		
Sensor	10		A(4),S	PC							▨	
On/Off Button	11		S	S								▨

A.14. Electric Brush Interface DSM



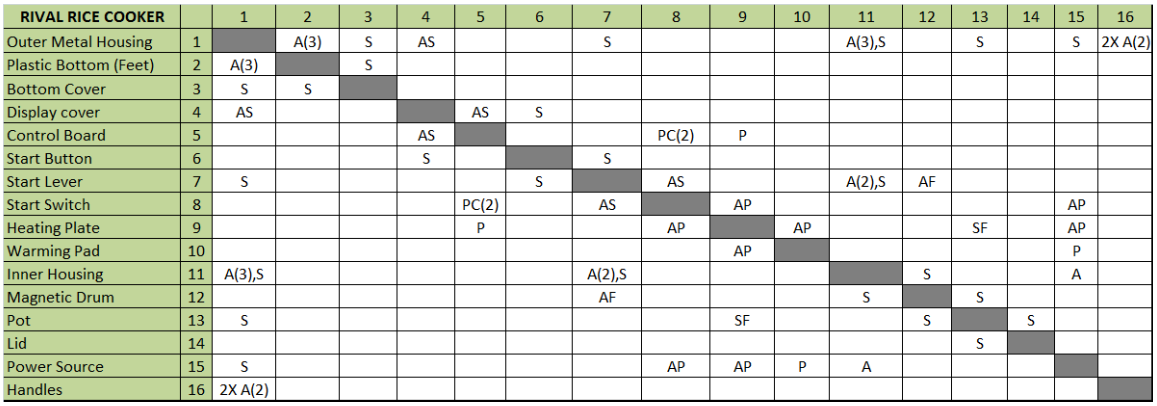
BRUSH		1	2	3	4	5	6	7	9	10	11	12	13	14
Front Housing	1	▨	A(3),S		S	S	A(2),S	S(3)			S		S	
Back Housing	2	A(3),S	▨	S			S				S			
Battery Cover	3		S	▨										
Switch Button	4	S			▨	S								
Control Board	5	S			S	▨		P			PC			
Gear Housing	6	A(2),S	S				▨		A(4),S	S	A(2),S		S	
Contacts	7	S(3)				P		▨			P			
Vacuum Cover	9						A(4),S		▨	S				
Vacuum Seal	10						S		S	▨	ST			
Motor	11	S	S			PC	A(2),S	P		S	▨	ST		
Gear	12										ST	▨		
Gear Holder	13	S					S						▨	S
Filter	14												S	▨

A.15. Iron Interface DSM



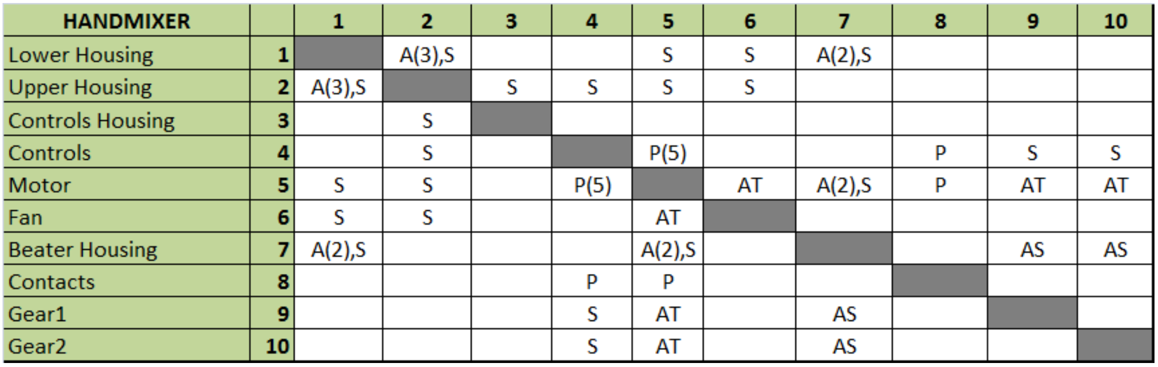
IRON		1	2	3	4	5	6	7	8	9	10	11	12	13
Back Cover	1		S	AS	S					S				
ButtonS Cover	2	S		AS	AS			S	S					S
Reservoir	3	AS	AS		A(2),S	AS,T	AST	S	S	A(3),S			S	S
Power Cable	4	S	AS	A(2),S						S	P(2)	P		P
Steam Button	5			AS,T				S		S	ST			
Squirt Button	6			AST					S					
Steam Button Cover	7		S	S		S								
Squirt Button Cover	8		S	S			S							
Plastic Housing	9	S		A(3),S	S	S					A(3),S	S	S	
Heat Plate	10				P(2)	ST				A(3),S		A,PF		
Controller	11				P					S	A,PF		S	P
Controller Button	12			S						S		S		
LED	13		S	S	P							P		

A.16. Rival Rice Cooker Interface DSM



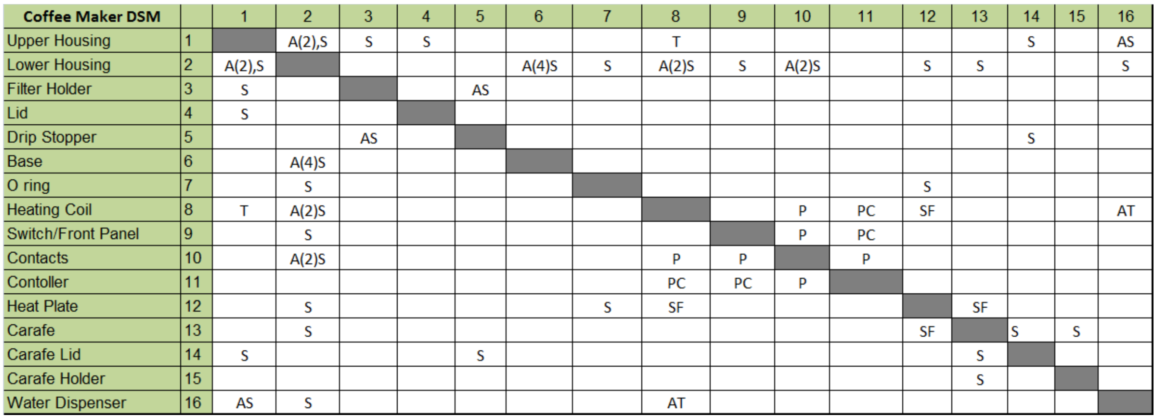
RIVAL RICE COOKER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Outer Metal Housing	1		A(3)	S	AS			S				A(3),S		S		S	2X A(2)
Plastic Bottom (Feet)	2	A(3)		S													
Bottom Cover	3	S	S														
Display cover	4	AS				AS	S										
Control Board	5				AS				PC(2)	P							
Start Button	6				S			S									
Start Lever	7	S					S		AS			A(2),S	AF				
Start Switch	8					PC(2)		AS		AP						AP	
Heating Plate	9					P			AP		AP			SF		AP	
Warming Pad	10									AP						P	
Inner Housing	11	A(3),S						A(2),S					S			A	
Magnetic Drum	12							AF				S		S			
Pot	13	S								SF			S		S		
Lid	14													S			
Power Source	15	S							AP	AP	P	A					
Handles	16	2X A(2)															

A.17. Handmixer Interface DSM



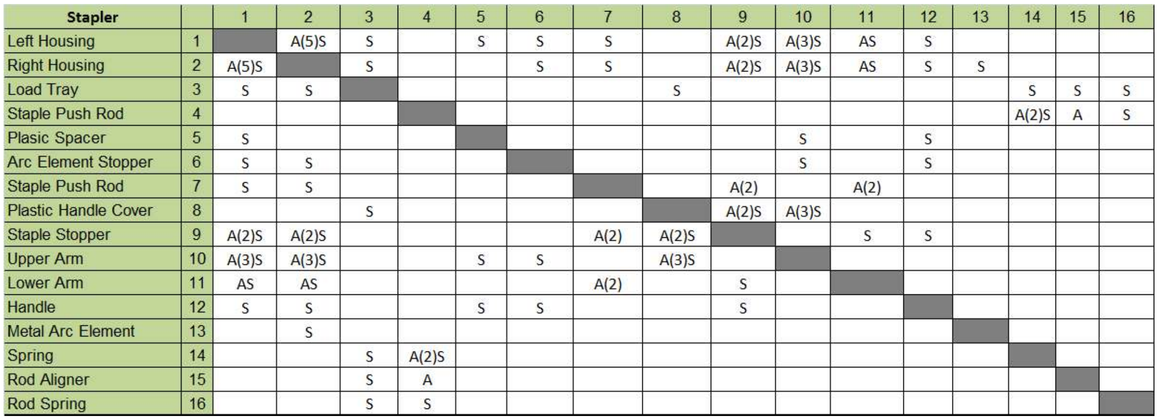
HANDMIXER		1	2	3	4	5	6	7	8	9	10
Lower Housing	1	■	A(3),S			S	S	A(2),S			
Upper Housing	2	A(3),S	■	S	S	S	S				
Controls Housing	3		S	■							
Controls	4		S		■	P(5)			P	S	S
Motor	5	S	S		P(5)	■	AT	A(2),S	P	AT	AT
Fan	6	S	S			AT	■				
Beater Housing	7	A(2),S				A(2),S		■		AS	AS
Contacts	8				P	P			■		
Gear1	9				S	AT		AS		■	
Gear2	10				S	AT		AS			■

A.18. Coffee Maker Interface DSM



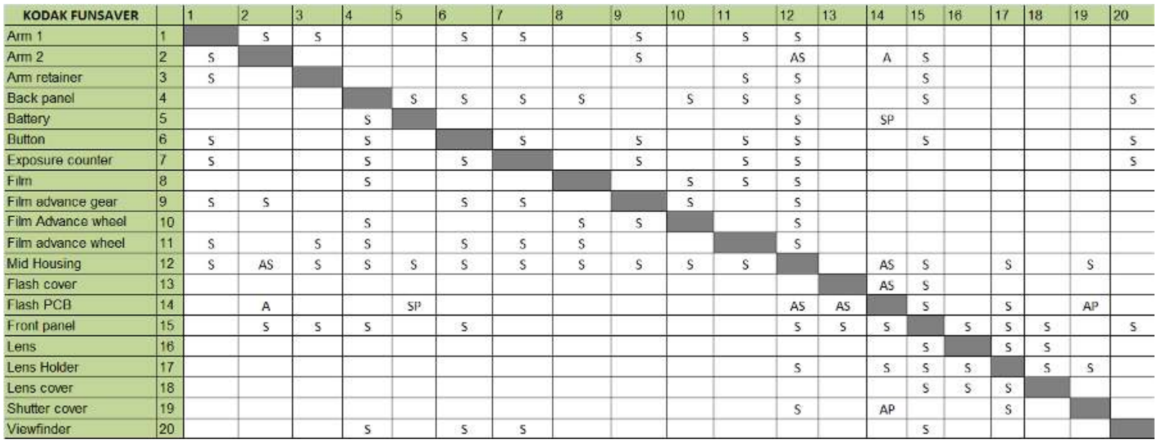
Coffee Maker DSM		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Upper Housing	1	■	A(2),S	S	S				T						S		AS
Lower Housing	2	A(2),S	■				A(4)S	S	A(2)S	S	A(2)S		S	S			S
Filter Holder	3	S		■		AS											
Lid	4	S			■												
Drip Stopper	5			AS		■									S		
Base	6		A(4)S				■										
O ring	7		S					■					S				
Heating Coil	8	T	A(2)S						■		P	PC	SF				AT
Switch/Front Panel	9		S							■	P	PC					
Contacts	10		A(2)S						P	P	■	P					
Contoller	11								PC	PC	P	■					
Heat Plate	12		S					S	SF				■	SF			
Carafe	13		S										SF	■	S	S	
Carafe Lid	14	S				S								S	■		
Carafe Holder	15													S		■	
Water Dispenser	16	AS	S						AT								■

A.19. Stapler Interface DSM



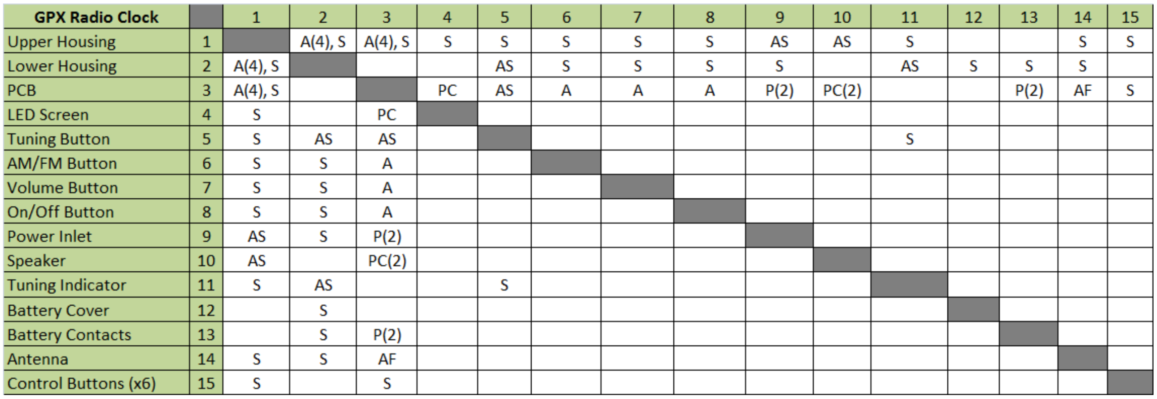
Stapler		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Left Housing	1	■	A(5)S	S		S	S	S		A(2)S	A(3)S	AS	S				
Right Housing	2	A(5)S	■	S			S	S		A(2)S	A(3)S	AS	S	S			
Load Tray	3	S	S	■					S						S	S	S
Staple Push Rod	4				■										A(2)S	A	S
Plasic Spacer	5	S				■					S		S				
Arc Element Stopper	6	S	S				■				S		S				
Staple Push Rod	7	S	S					■		A(2)		A(2)					
Plastic Handle Cover	8			S					■	A(2)S	A(3)S						
Staple Stopper	9	A(2)S	A(2)S					A(2)	A(2)S	■		S	S				
Upper Arm	10	A(3)S	A(3)S			S	S		A(3)S		■						
Lower Arm	11	AS	AS					A(2)		S		■					
Handle	12	S	S			S	S			S			■				
Metal Arc Element	13		S											■			
Spring	14			S	A(2)S										■		
Rod Aligner	15			S	A											■	
Rod Spring	16			S	S												■

A.20. Single-Use Camera Interface DSM



KODAK FUNSAVER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Arm 1	1	■	S	S			S	S		S		S	S								
Arm 2	2	S	■							S			AS		A	S					
Arm retainer	3	S		■								S	S			S					
Back panel	4				■	S	S	S	S		S	S	S			S					S
Battery	5				S	■							S		SP						
Button	6	S			S		■	S		S		S	S			S					S
Exposure counter	7	S			S		S	■		S		S	S								S
Film	8				S				■		S	S	S								
Film advance gear	9	S	S				S	S		■	S		S								
Film Advance wheel	10				S				S	S	■		S								
Film advance wheel	11	S		S	S		S	S	S			■	S								
Mid Housing	12	S	AS	S	S	S	S	S	S	S	S	S	■		AS	S		S		S	
Flash cover	13													■	AS	S					
Flash PCB	14		A			SP							AS	AS	■	S		S		AP	
Front panel	15		S	S	S		S						S	S	S	■	S	S	S		S
Lens	16															S	■	S	S		
Lens Holder	17												S		S	S	S	■	S	S	
Lens cover	18															S	S	S	■		
Shutter cover	19												S		AP			S		■	
Viewfinder	20				S		S	S								S					■

A.21. Radio Clock Interface DSM

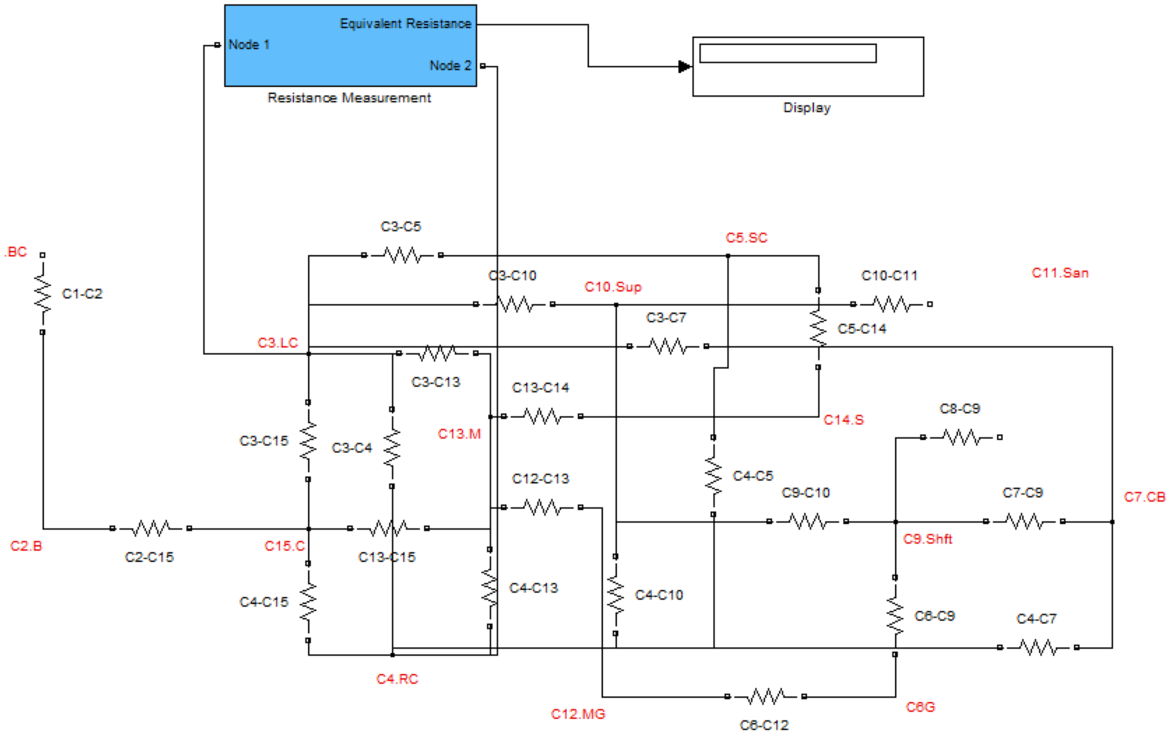


GPX Radio Clock		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Upper Housing	1		A(4), S	A(4), S	S	S	S	S	S	AS	AS	S			S	S
Lower Housing	2	A(4), S				AS	S	S	S	S		AS	S	S	S	
PCB	3	A(4), S			PC	AS	A	A	A	P(2)	PC(2)			P(2)	AF	S
LED Screen	4	S		PC												
Tuning Button	5	S	AS	AS								S				
AM/FM Button	6	S	S	A												
Volume Button	7	S	S	A												
On/Off Button	8	S	S	A												
Power Inlet	9	AS	S	P(2)												
Speaker	10	AS		PC(2)												
Tuning Indicator	11	S	AS			S										
Battery Cover	12		S													
Battery Contacts	13		S	P(2)												
Antenna	14	S	S	AF												
Control Buttons (x6)	15	S		S												

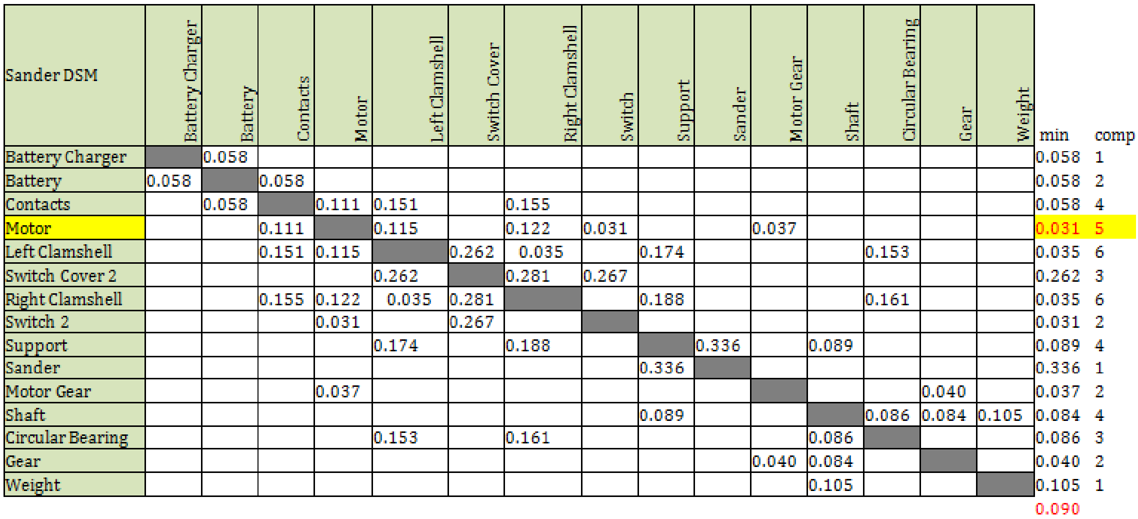
APPENDIX B

ELECTRICAL CIRCUIT MODELS AND OUTPUTS OF ANALYZED PRODUCTS

B.1. Sander Model

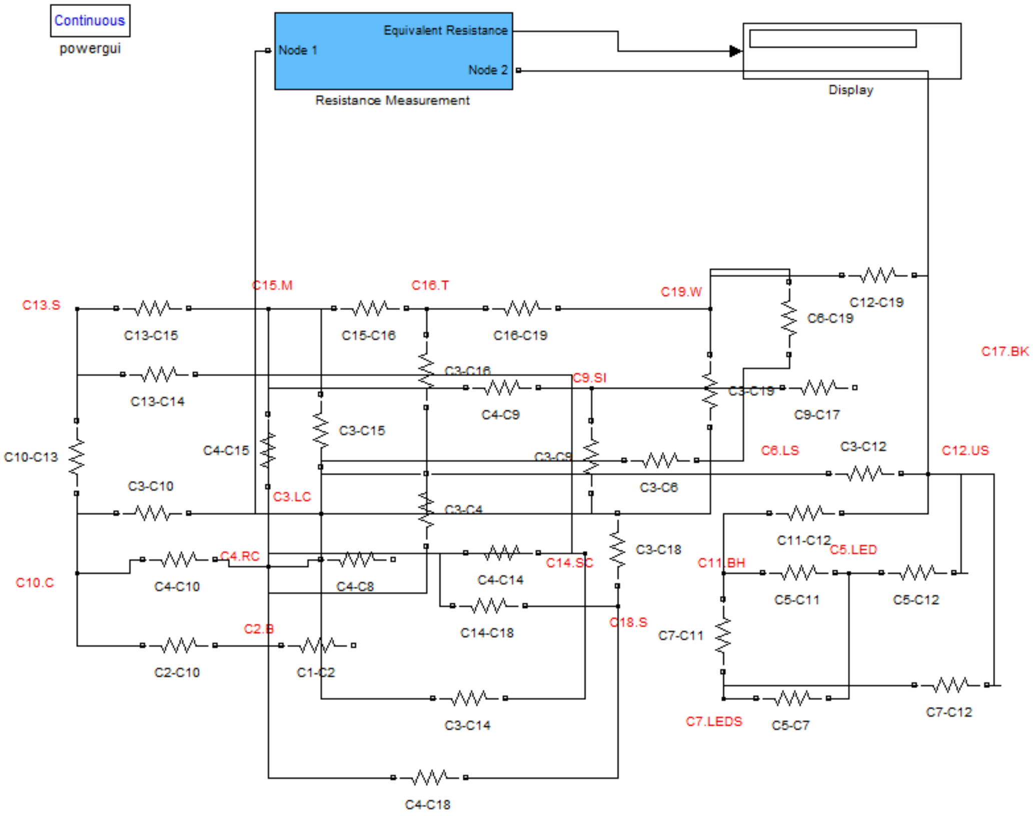


B.2. Sander Output

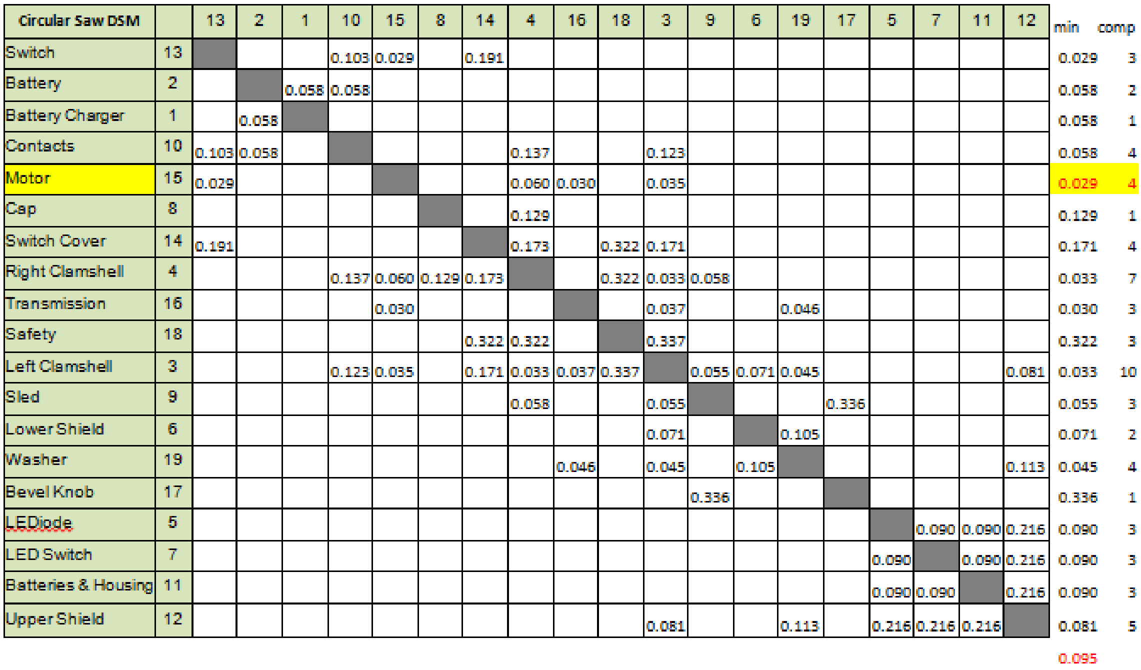


Sander DSM	Battery Charger	Battery	Contacts	Motor	Left Clamshell	Switch Cover	Right Clamshell	Switch	Support	Sander	Motor Gear	Shaft	Circular Bearing	Gear	Weight	min	comp
Battery Charger		0.058														0.058	1
Battery	0.058		0.058													0.058	2
Contacts		0.058		0.111	0.151		0.155									0.058	4
Motor			0.111		0.115		0.122	0.031			0.037					0.031	5
Left Clamshell			0.151	0.115		0.262	0.035		0.174				0.153			0.035	6
Switch Cover 2					0.262		0.281	0.267								0.262	3
Right Clamshell			0.155	0.122	0.035	0.281			0.188				0.161			0.035	6
Switch 2				0.031		0.267										0.031	2
Support					0.174		0.188			0.336		0.089				0.089	4
Sander									0.336							0.336	1
Motor Gear				0.037										0.040		0.037	2
Shaft									0.089				0.086	0.084	0.105	0.084	4
Circular Bearing					0.153		0.161					0.086				0.086	3
Gear											0.040	0.084				0.040	2
Weight												0.105				0.105	1
																0.090	

B.3. Circular Saw Model

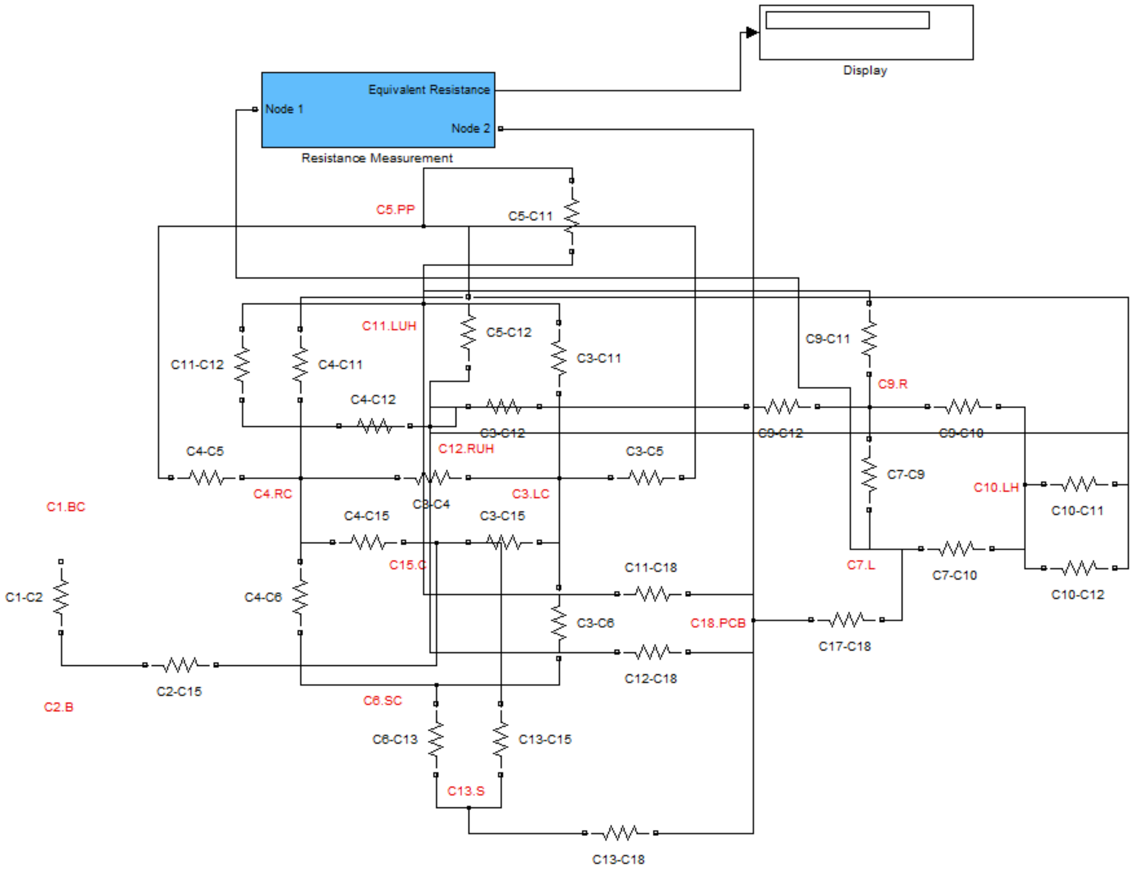


B.4. Circular Saw Output

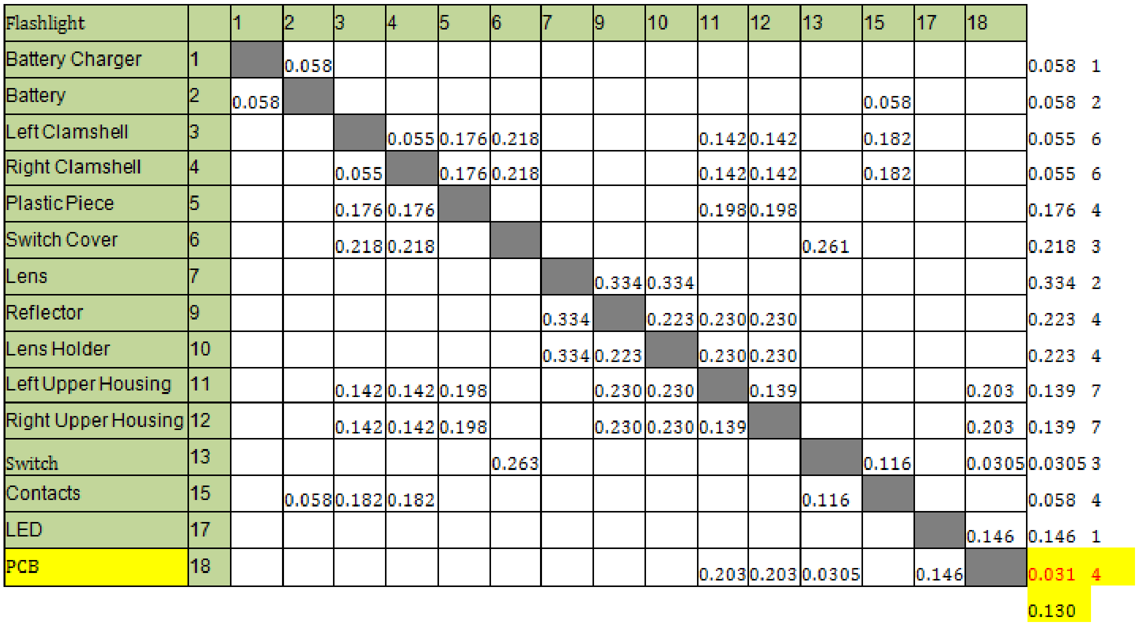


Circular Saw DSM		13	2	1	10	15	8	14	4	16	18	3	9	6	19	17	5	7	11	12	min	comp
Switch	13	■			0.103	0.029		0.191													0.029	3
Battery	2		■	0.058	0.058																0.058	2
Battery Charger	1		0.058	■																	0.058	1
Contacts	10	0.103	0.058		■				0.137			0.123									0.058	4
Motor	15	0.029				■			0.060	0.030		0.035									0.029	4
Cap	8						■		0.129												0.129	1
Switch Cover	14	0.191						■	0.173		0.322	0.171									0.171	4
Right Clamshell	4				0.137	0.060	0.129	0.173	■		0.322	0.033	0.058								0.033	7
Transmission	16					0.030				■		0.037			0.046						0.030	3
Safety	18							0.322	0.322		■	0.337									0.322	3
Left Clamshell	3				0.123	0.035		0.171	0.033	0.037	0.337	■	0.055	0.071	0.045					0.081	0.033	10
Sled	9								0.058			0.055	■			0.336					0.055	3
Lower Shield	6											0.071		■	0.105						0.071	2
Washer	19									0.046		0.045		0.105	■					0.113	0.045	4
Bevel Knob	17												0.336			■					0.336	1
LEDiode	5																■	0.090	0.090	0.216	0.090	3
LED Switch	7																0.090	■	0.090	0.216	0.090	3
Batteries & Housing	11																0.090	0.090	■	0.216	0.090	3
Upper Shield	12											0.081			0.113		0.216	0.216	0.216	■	0.081	5
																					0.095	

B.5. Flashlight Model

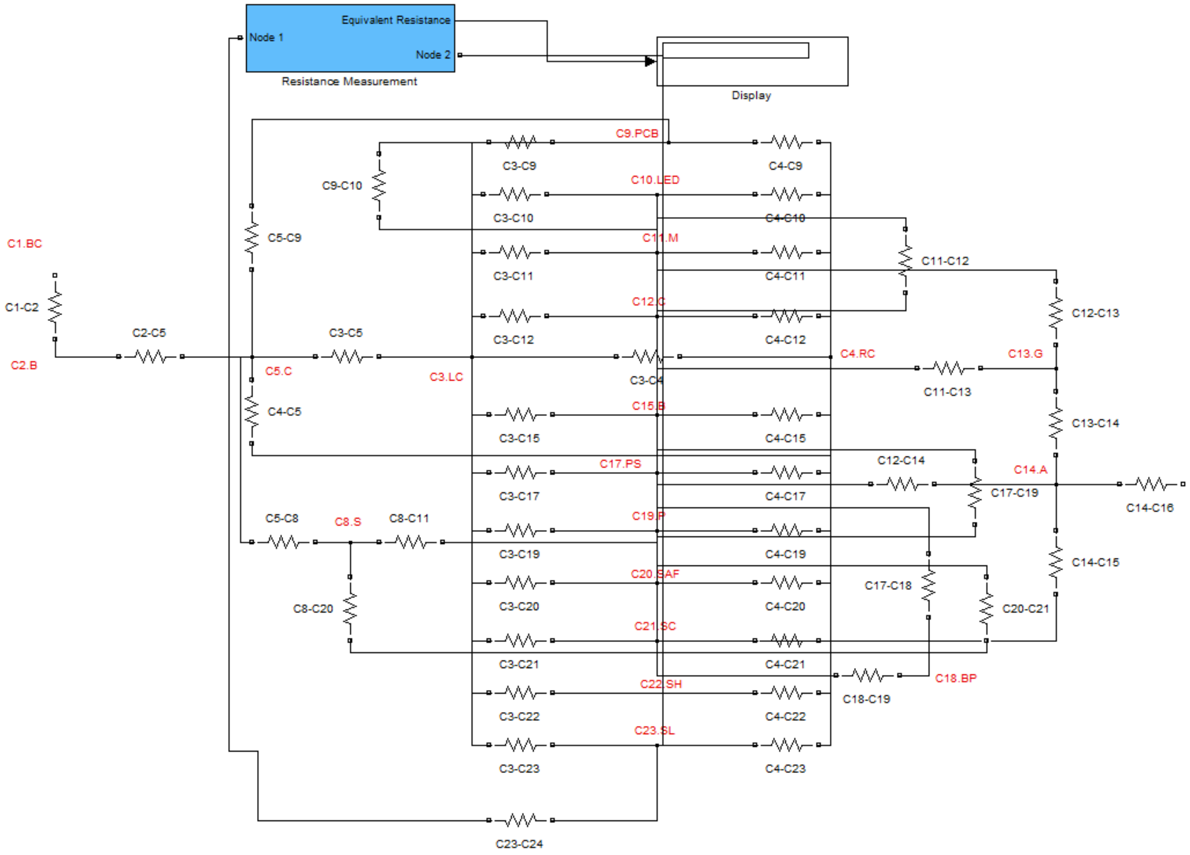


B.6. Flashlight Output

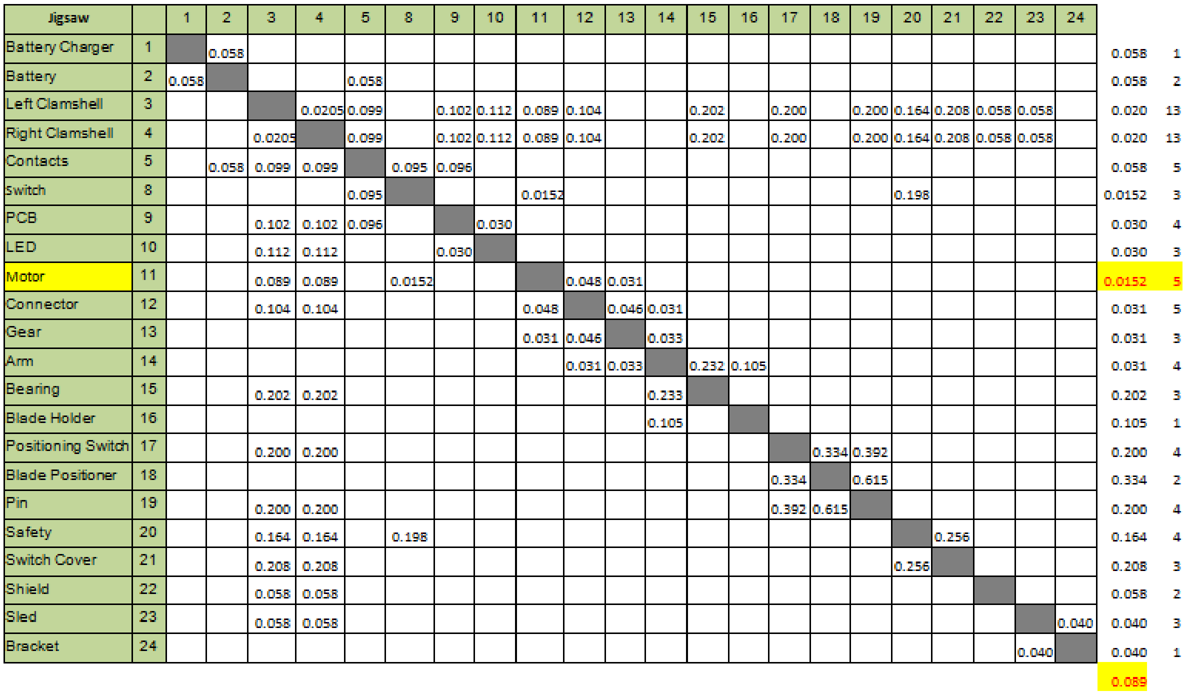


Flashlight		1	2	3	4	5	6	7	9	10	11	12	13	15	17	18		
Battery Charger	1		0.058														0.058	1
Battery	2	0.058												0.058			0.058	2
Left Clamshell	3				0.055	0.176	0.218				0.142	0.142		0.182			0.055	6
Right Clamshell	4			0.055		0.176	0.218				0.142	0.142		0.182			0.055	6
Plastic Piece	5			0.176	0.176						0.198	0.198					0.176	4
Switch Cover	6			0.218	0.218								0.261				0.218	3
Lens	7								0.334	0.334							0.334	2
Reflector	9							0.334		0.223	0.230	0.230					0.223	4
Lens Holder	10							0.334	0.223		0.230	0.230					0.223	4
Left Upper Housing	11			0.142	0.142	0.198			0.230	0.230		0.139				0.203	0.139	7
Right Upper Housing	12			0.142	0.142	0.198			0.230	0.230	0.139					0.203	0.139	7
Switch	13						0.263							0.116		0.0305	0.0305	3
Contacts	15		0.058	0.182	0.182								0.116				0.058	4
LED	17															0.146	0.146	1
PCB	18										0.203	0.203	0.0305		0.146		0.031	4
																	0.130	

B.7. Jigsaw Model

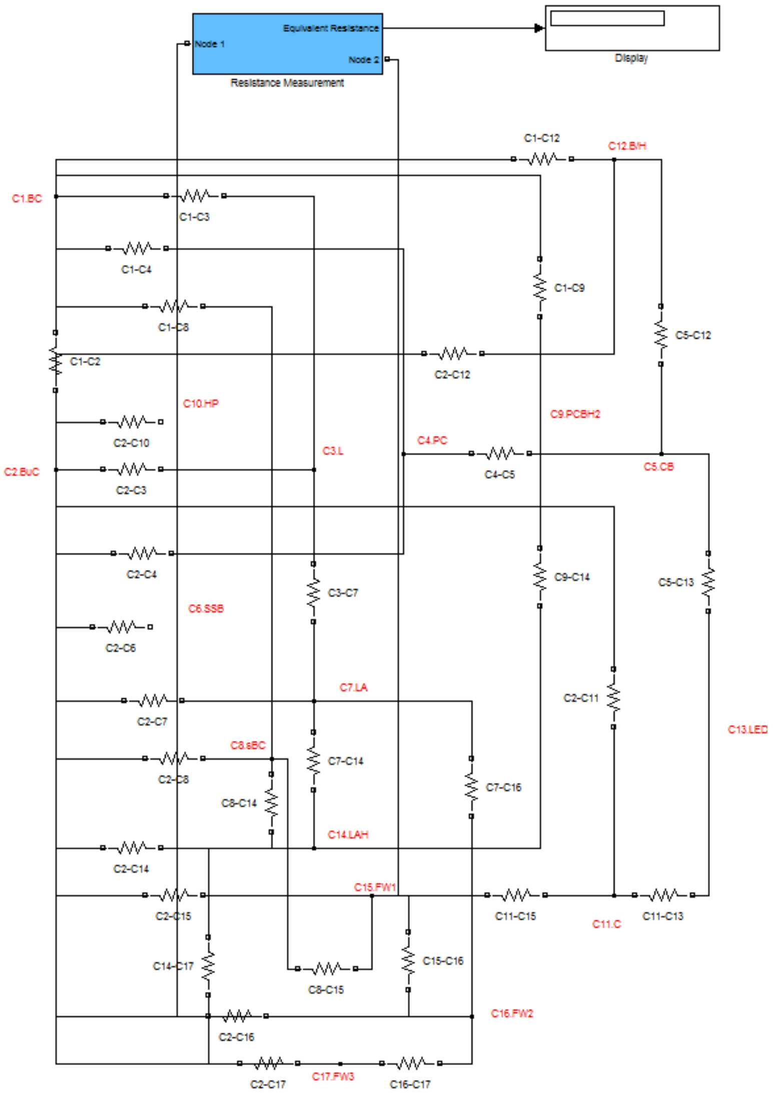


B.8. Jigsaw Output

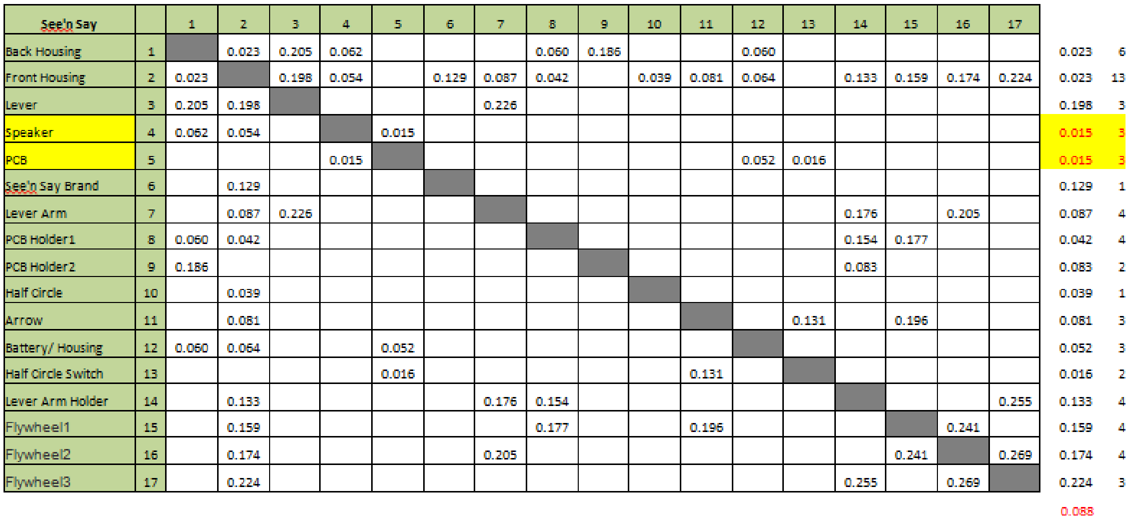


Jigsaw		1	2	3	4	5	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
Battery Charger	1	■	0.058																					0.058	1
Battery	2	0.058	■			0.058																		0.058	2
Left Clamshell	3			■	0.0205	0.099		0.102	0.112	0.089	0.104			0.202		0.200		0.200	0.164	0.208	0.058	0.058		0.020	13
Right Clamshell	4			0.0205	■	0.099		0.102	0.112	0.089	0.104			0.202		0.200		0.200	0.164	0.208	0.058	0.058		0.020	13
Contacts	5		0.058	0.099	0.099	■	0.095	0.096																0.058	5
Switch	8					0.095	■			0.0152									0.198					0.0152	3
PCB	9			0.102	0.102	0.096		■	0.030															0.030	4
LED	10			0.112	0.112			0.030	■															0.030	3
Motor	11			0.089	0.089		0.0152			■	0.048	0.031												0.0152	5
Connector	12			0.104	0.104					0.048	■	0.046	0.031											0.031	5
Gear	13									0.031	0.046	■	0.033											0.031	3
Arm	14										0.031	0.033	■	0.232	0.105									0.031	4
Bearing	15			0.202	0.202								0.233	■										0.202	3
Blade Holder	16												0.105		■									0.105	1
Positioning Switch	17			0.200	0.200											■	0.334	0.392						0.200	4
Blade Positioner	18															0.334	■	0.615						0.334	2
Pin	19			0.200	0.200											0.392	0.615	■						0.200	4
Safety	20			0.164	0.164		0.198												■	0.256				0.164	4
Switch Cover	21			0.208	0.208														0.256	■				0.208	3
Shield	22			0.058	0.058																■			0.058	2
Sled	23			0.058	0.058																	■	0.040	0.040	3
Bracket	24																					0.040	■	0.040	1
																								0.089	

B.9. See’n Say Model

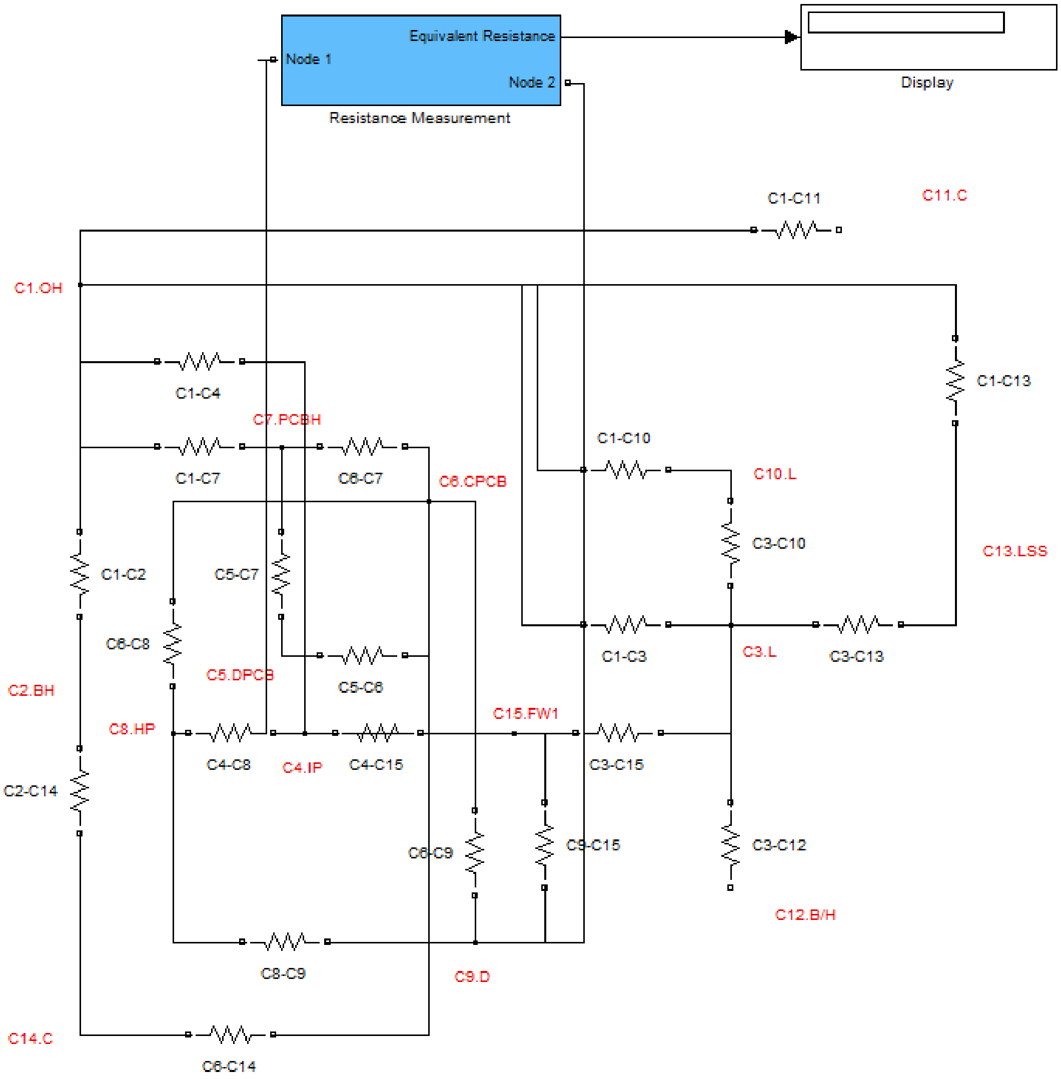


B.10. See'n Say Output

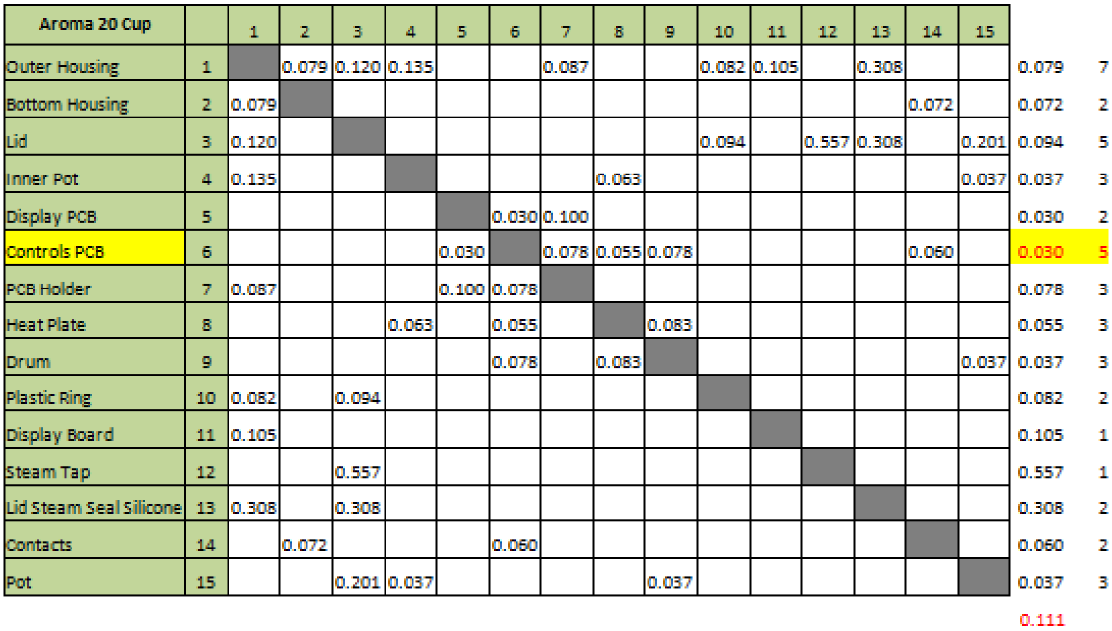


See'n Say		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
Back Housing	1		0.023	0.205	0.062				0.060	0.186			0.060						0.023	6
Front Housing	2	0.023		0.198	0.054		0.129	0.087	0.042		0.039	0.081	0.064		0.133	0.159	0.174	0.224	0.023	13
Lever	3	0.205	0.198					0.226											0.198	3
Speaker	4	0.062	0.054			0.015													0.015	3
PCB	5				0.015								0.052	0.016					0.015	3
See'n Say Brand	6		0.129																0.129	1
Lever Arm	7		0.087	0.226											0.176		0.205		0.087	4
PCB Holder1	8	0.060	0.042												0.154	0.177			0.042	4
PCB Holder2	9	0.186													0.083				0.083	2
Half Circle	10		0.039																0.039	1
Arrow	11		0.081											0.131		0.196			0.081	3
Battery/ Housing	12	0.060	0.064			0.052													0.052	3
Half Circle Switch	13					0.016						0.131							0.016	2
Lever Arm Holder	14		0.133					0.176	0.154									0.255	0.133	4
Flywheel1	15		0.159						0.177			0.196					0.241		0.159	4
Flywheel2	16		0.174					0.205								0.241		0.269	0.174	4
Flywheel3	17		0.224												0.255		0.269		0.224	3
																			0.088	

B.11. Aroma Rice Cooker (20 Cup) Model

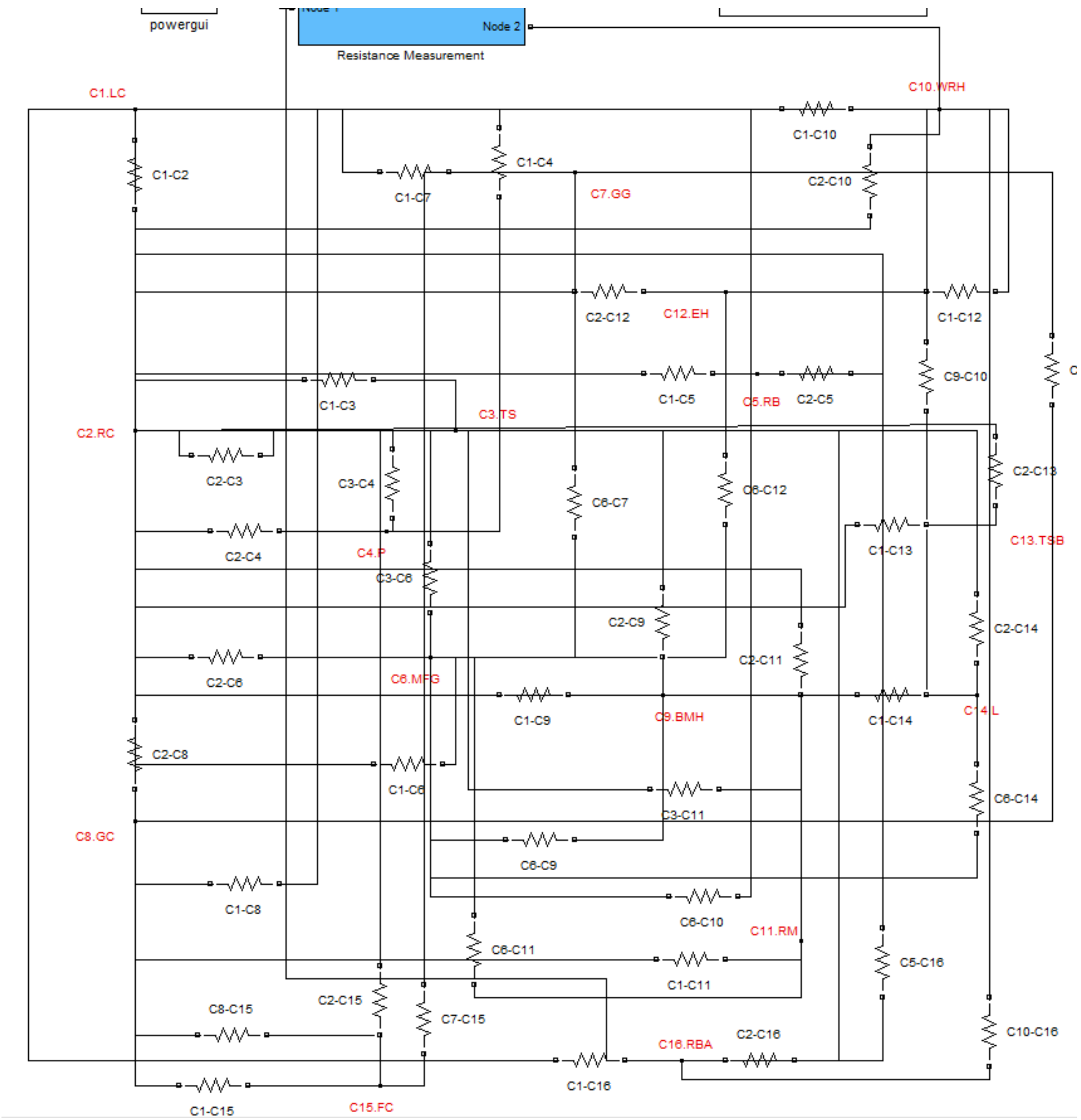


B.12. Aroma Rice Cooker (20 Cup) Output

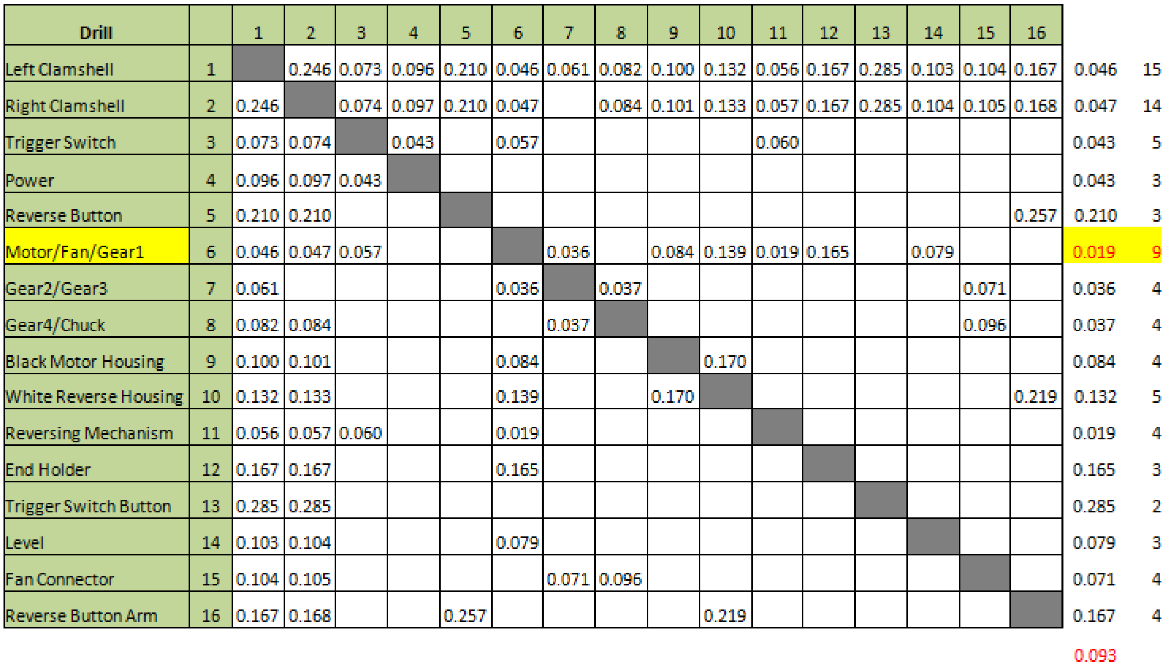


Aroma 20 Cup		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Outer Housing	1	■	0.079	0.120	0.135			0.087			0.082	0.105		0.308			0.079	7
Bottom Housing	2	0.079	■												0.072		0.072	2
Lid	3	0.120		■							0.094		0.557	0.308		0.201	0.094	5
Inner Pot	4	0.135			■				0.063							0.037	0.037	3
Display PCB	5					■	0.030	0.100									0.030	2
Controls PCB	6					0.030	■	0.078	0.055	0.078					0.060		0.030	5
PCB Holder	7	0.087				0.100	0.078	■									0.078	3
Heat Plate	8				0.063		0.055		■	0.083							0.055	3
Drum	9						0.078		0.083	■						0.037	0.037	3
Plastic Ring	10	0.082		0.094							■						0.082	2
Display Board	11	0.105										■					0.105	1
Steam Tap	12			0.557									■				0.557	1
Lid Steam Seal Silicone	13	0.308		0.308										■			0.308	2
Contacts	14		0.072				0.060								■		0.060	2
Pot	15			0.201	0.037					0.037						■	0.037	3
																	0.111	

B.13. Drill Model

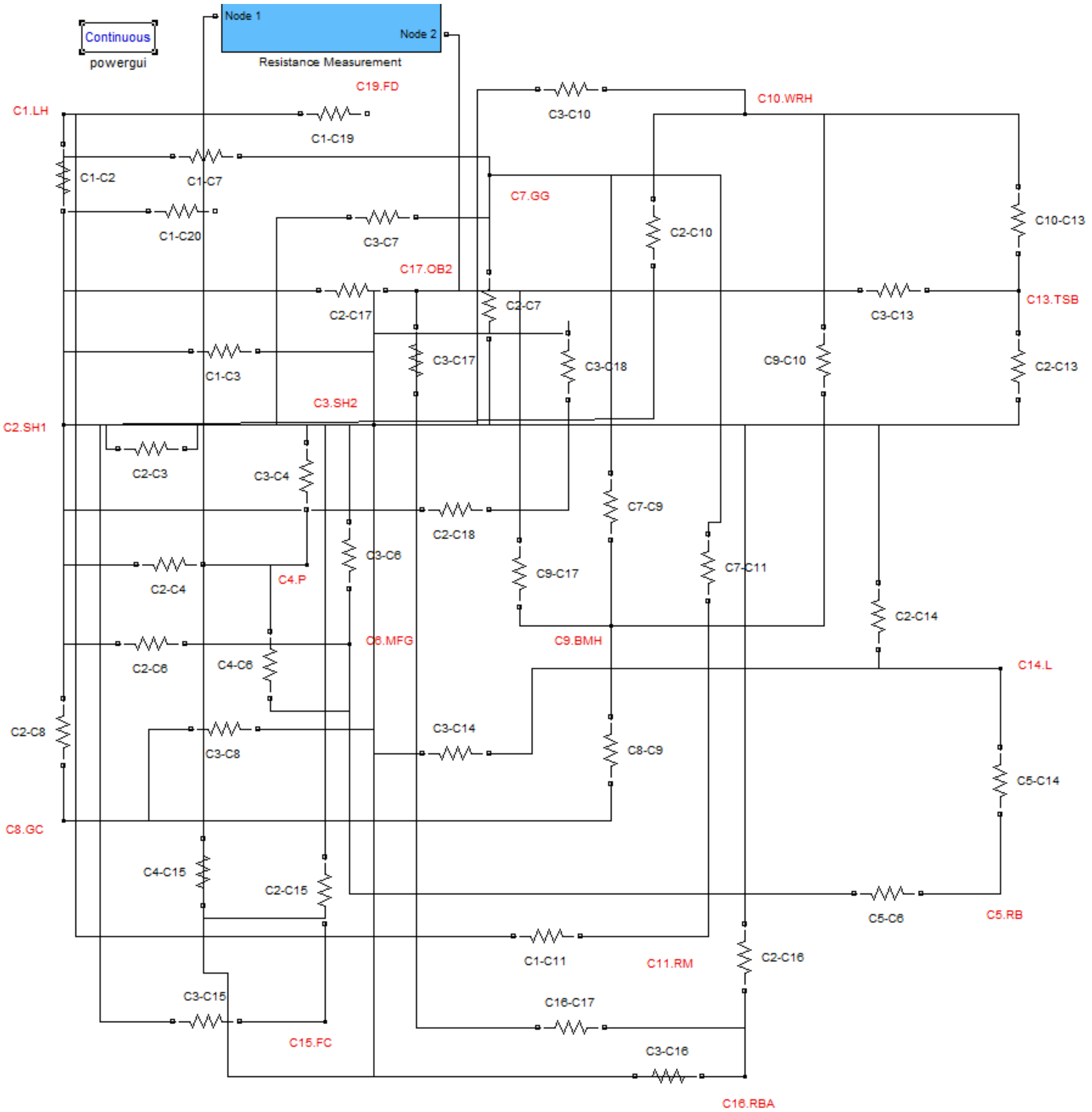


B.14. Drill Output

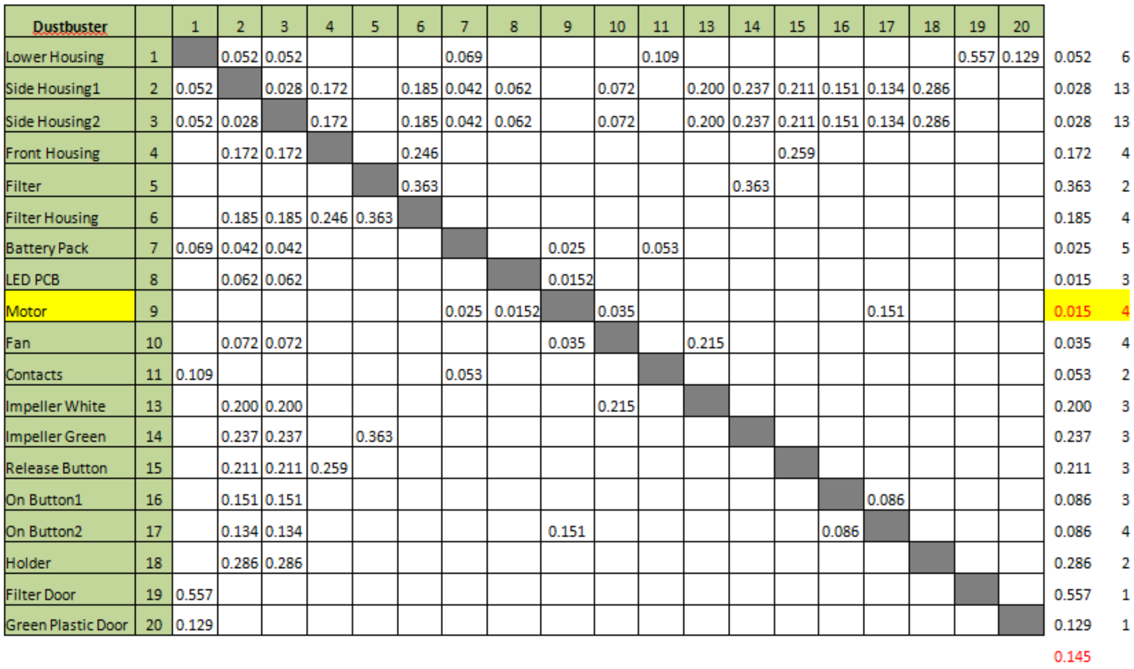


Drill		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
Left Clamshell	1	■	0.246	0.073	0.096	0.210	0.046	0.061	0.082	0.100	0.132	0.056	0.167	0.285	0.103	0.104	0.167	0.046	15
Right Clamshell	2	0.246	■	0.074	0.097	0.210	0.047		0.084	0.101	0.133	0.057	0.167	0.285	0.104	0.105	0.168	0.047	14
Trigger Switch	3	0.073	0.074	■	0.043		0.057					0.060						0.043	5
Power	4	0.096	0.097	0.043	■													0.043	3
Reverse Button	5	0.210	0.210			■											0.257	0.210	3
Motor/Fan/Gear1	6	0.046	0.047	0.057			■	0.036		0.084	0.139	0.019	0.165		0.079			0.019	9
Gear2/Gear3	7	0.061					0.036	■	0.037							0.071		0.036	4
Gear4/Chuck	8	0.082	0.084					0.037	■							0.096		0.037	4
Black Motor Housing	9	0.100	0.101				0.084			■	0.170							0.084	4
White Reverse Housing	10	0.132	0.133				0.139			0.170	■						0.219	0.132	5
Reversing Mechanism	11	0.056	0.057	0.060			0.019					■						0.019	4
End Holder	12	0.167	0.167				0.165						■					0.165	3
Trigger Switch Button	13	0.285	0.285											■				0.285	2
Level	14	0.103	0.104				0.079								■			0.079	3
Fan Connector	15	0.104	0.105					0.071	0.096							■		0.071	4
Reverse Button Arm	16	0.167	0.168			0.257					0.219						■	0.167	4
																		0.093	

B.15. Dustbuster Model

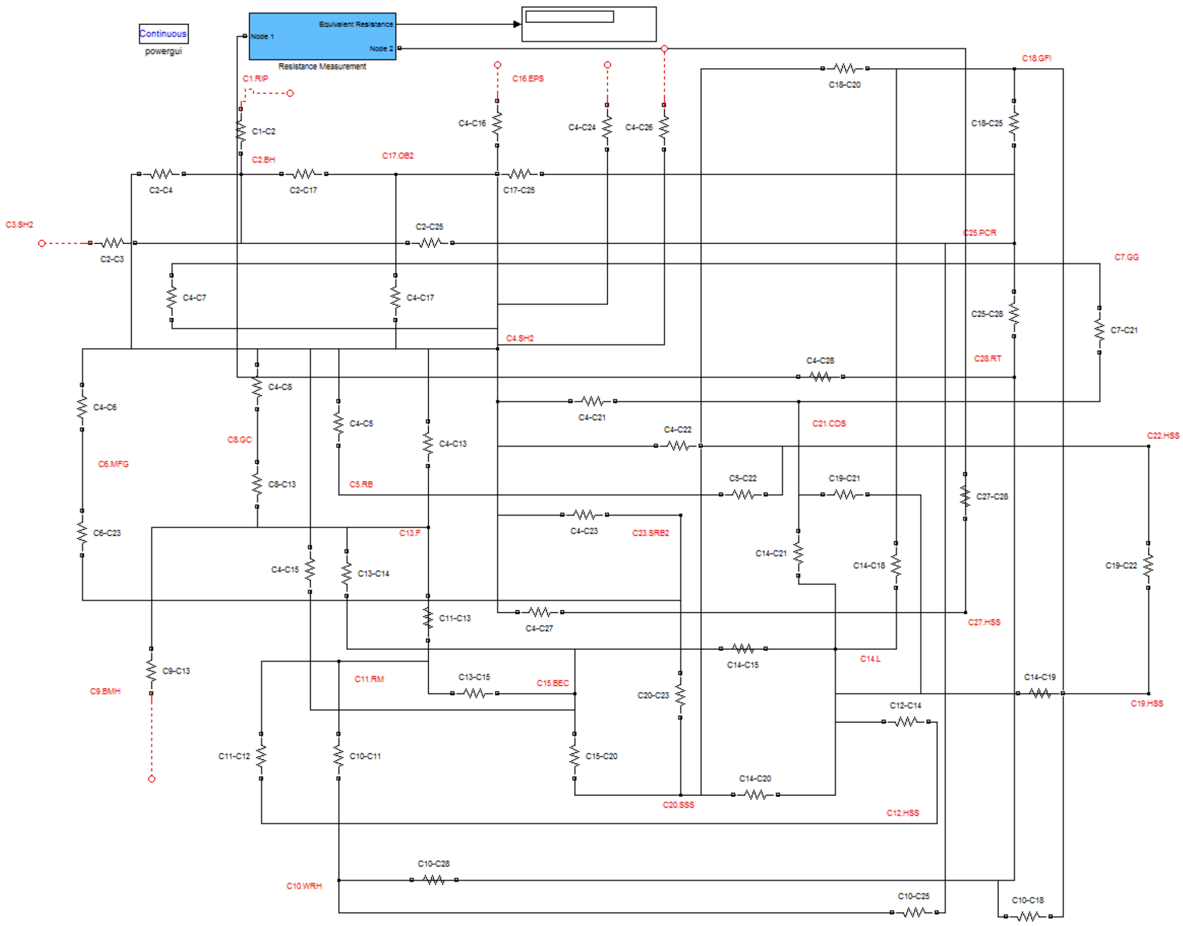


B.16. Dustbuster Output

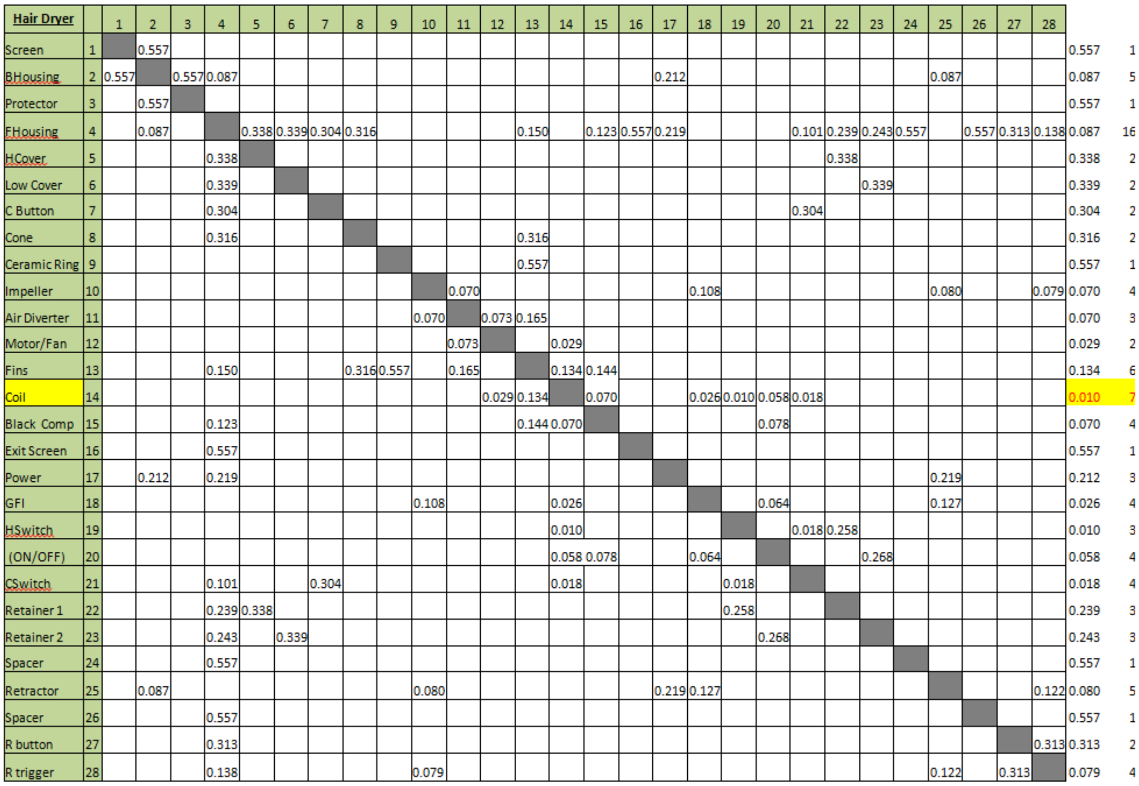


Dustbuster		1	2	3	4	5	6	7	8	9	10	11	13	14	15	16	17	18	19	20		
Lower Housing	1	▨	0.052	0.052				0.069				0.109							0.557	0.129	0.052	6
Side Housing1	2	0.052	▨	0.028	0.172		0.185	0.042	0.062		0.072		0.200	0.237	0.211	0.151	0.134	0.286			0.028	13
Side Housing2	3	0.052	0.028	▨	0.172		0.185	0.042	0.062		0.072		0.200	0.237	0.211	0.151	0.134	0.286			0.028	13
Front Housing	4		0.172	0.172	▨		0.246								0.259						0.172	4
Filter	5					▨	0.363							0.363							0.363	2
Filter Housing	6		0.185	0.185	0.246	0.363	▨														0.185	4
Battery Pack	7	0.069	0.042	0.042				▨		0.025		0.053									0.025	5
LED PCB	8		0.062	0.062					▨	0.0152											0.015	3
Motor	9							0.025	0.0152	▨	0.035						0.151				0.015	4
Fan	10		0.072	0.072						0.035	▨		0.215								0.035	4
Contacts	11	0.109						0.053				▨									0.053	2
Impeller White	13		0.200	0.200							0.215		▨								0.200	3
Impeller Green	14		0.237	0.237		0.363								▨							0.237	3
Release Button	15		0.211	0.211	0.259										▨						0.211	3
On Button1	16		0.151	0.151												▨	0.086				0.086	3
On Button2	17		0.134	0.134						0.151						0.086	▨				0.086	4
Holder	18		0.286	0.286														▨			0.286	2
Filter Door	19	0.557																	▨		0.557	1
Green Plastic Door	20	0.129																		▨	0.129	1
																					0.145	

B.17. Hairdryer Model

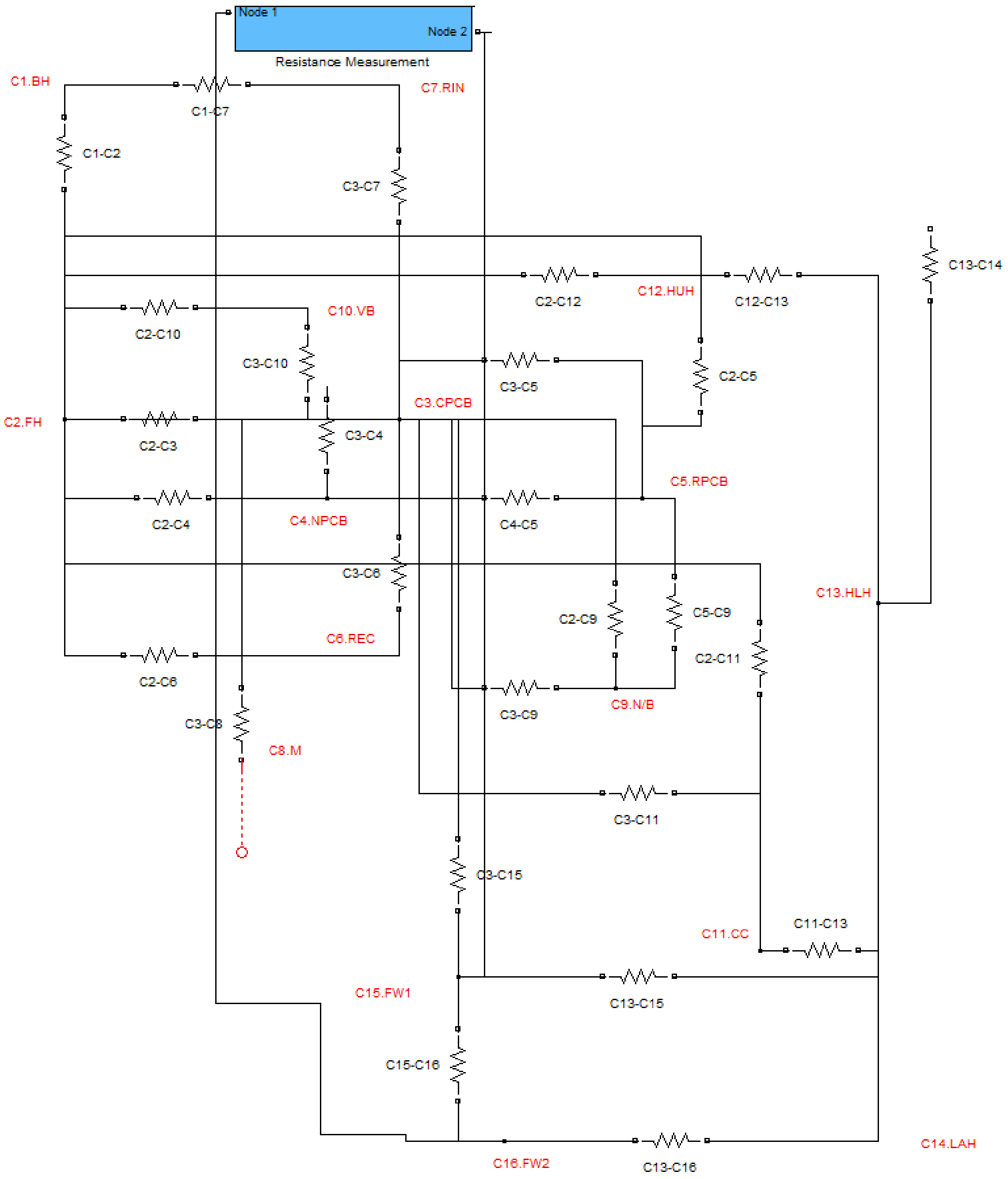


B.18. Hairdryer Output

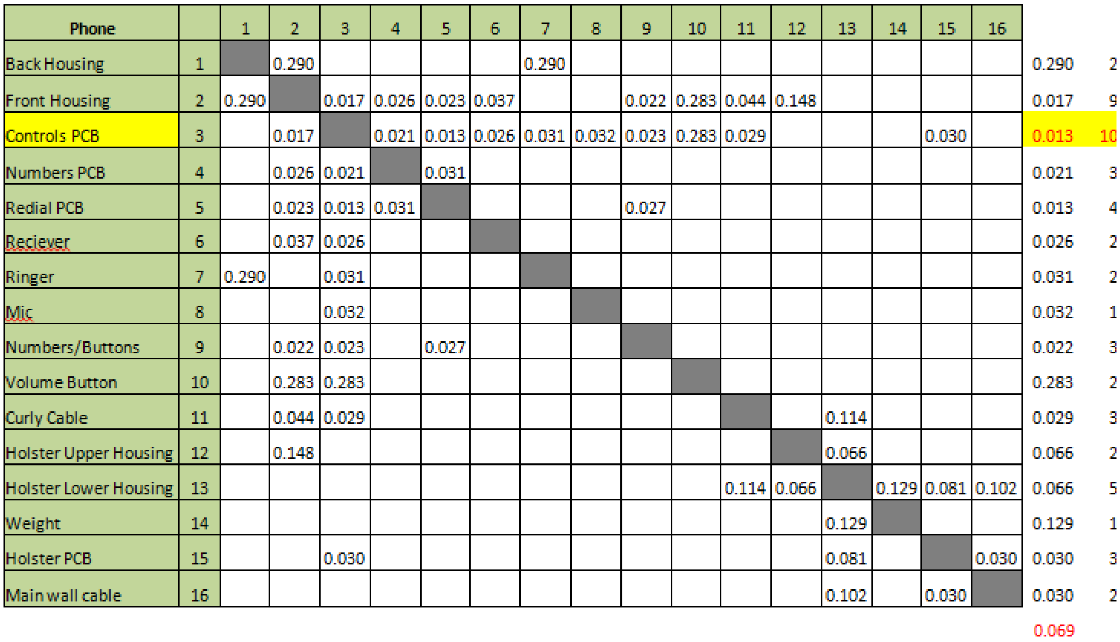


Hair Dryer		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		
Screen	1	▨	0.557																											0.557	1
BHousing	2	0.557	▨	0.557	0.087													0.212								0.087				0.087	5
Protector	3		0.557	▨																										0.557	1
FHousing	4		0.087		▨	0.338	0.339	0.304	0.316					0.150		0.123	0.557	0.219				0.101	0.239	0.243	0.557		0.557	0.313	0.138	0.087	16
HCover	5				0.338	▨																	0.338							0.338	2
Low Cover	6				0.339		▨																	0.339						0.339	2
C Button	7				0.304			▨														0.304								0.304	2
Cone	8				0.316				▨					0.316																0.316	2
Ceramic Ring	9									▨				0.557																0.557	1
Impeller	10										▨	0.070							0.108							0.080			0.079	0.070	4
Air Diverter	11										0.070	▨	0.073	0.165																0.070	3
Motor/Fan	12											0.073	▨		0.029															0.029	2
Fins	13				0.150				0.316	0.557		0.165		▨	0.134	0.144														0.134	6
Coil	14												0.029	0.134	▨	0.070			0.026	0.010	0.058	0.018								0.010	7
Black Comp	15				0.123									0.144	0.070	▨					0.078									0.070	4
Exit Screen	16				0.557												▨													0.557	1
Power	17		0.212		0.219													▨								0.219				0.212	3
GFI	18										0.108				0.026				▨		0.064					0.127				0.026	4
HSwitch	19														0.010					▨		0.018	0.258							0.010	3
(ON/OFF)	20														0.058	0.078			0.064		▨			0.268						0.058	4
CSwitch	21				0.101			0.304							0.018					0.018		▨								0.018	4
Retainer 1	22				0.239	0.338														0.258			▨							0.239	3
Retainer 2	23				0.243		0.339														0.268			▨						0.243	3
Spacer	24				0.557																				▨					0.557	1
Retractor	25		0.087								0.080							0.219	0.127							▨			0.122	0.080	5
Spacer	26				0.557																						▨			0.557	1
R button	27				0.313																							▨	0.313	0.313	2
R trigger	28				0.138						0.079															0.122		0.313	▨	0.079	4

B.19. Phone Model

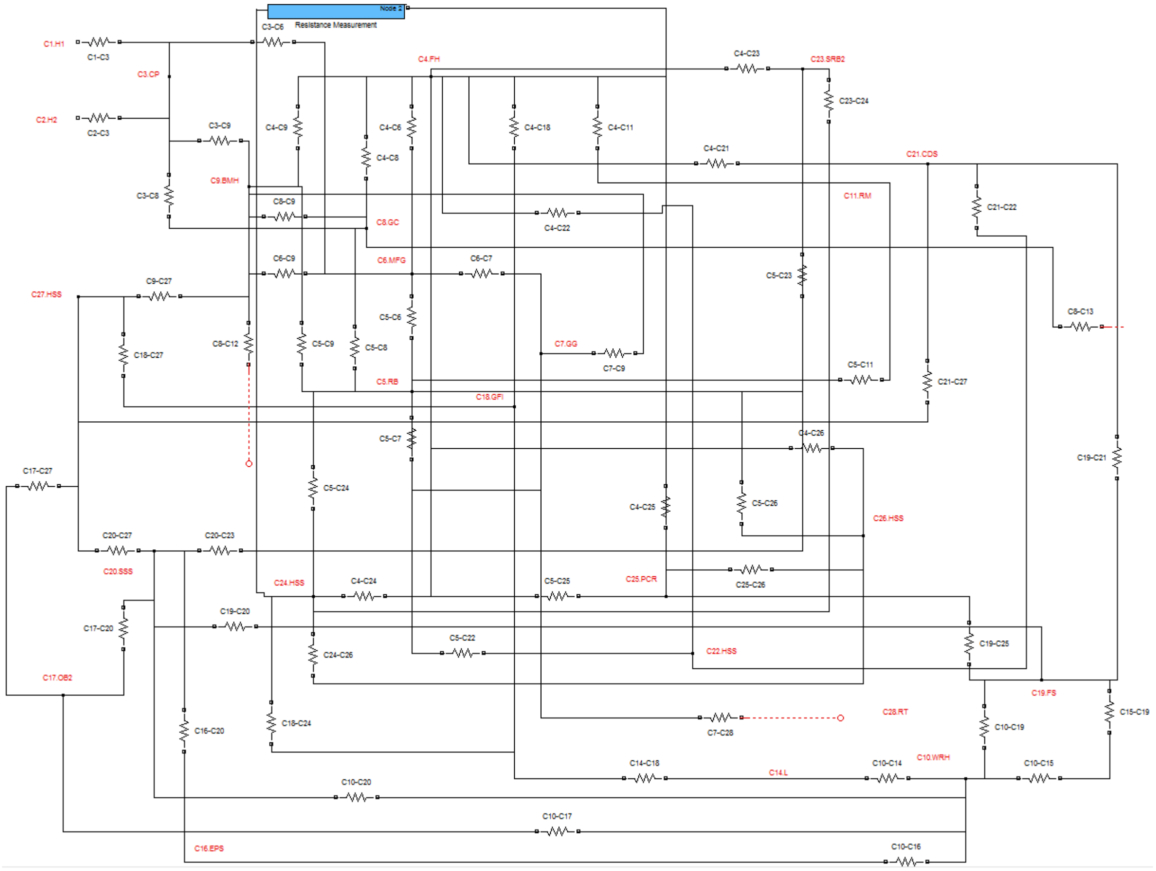


B.20. Phone Output

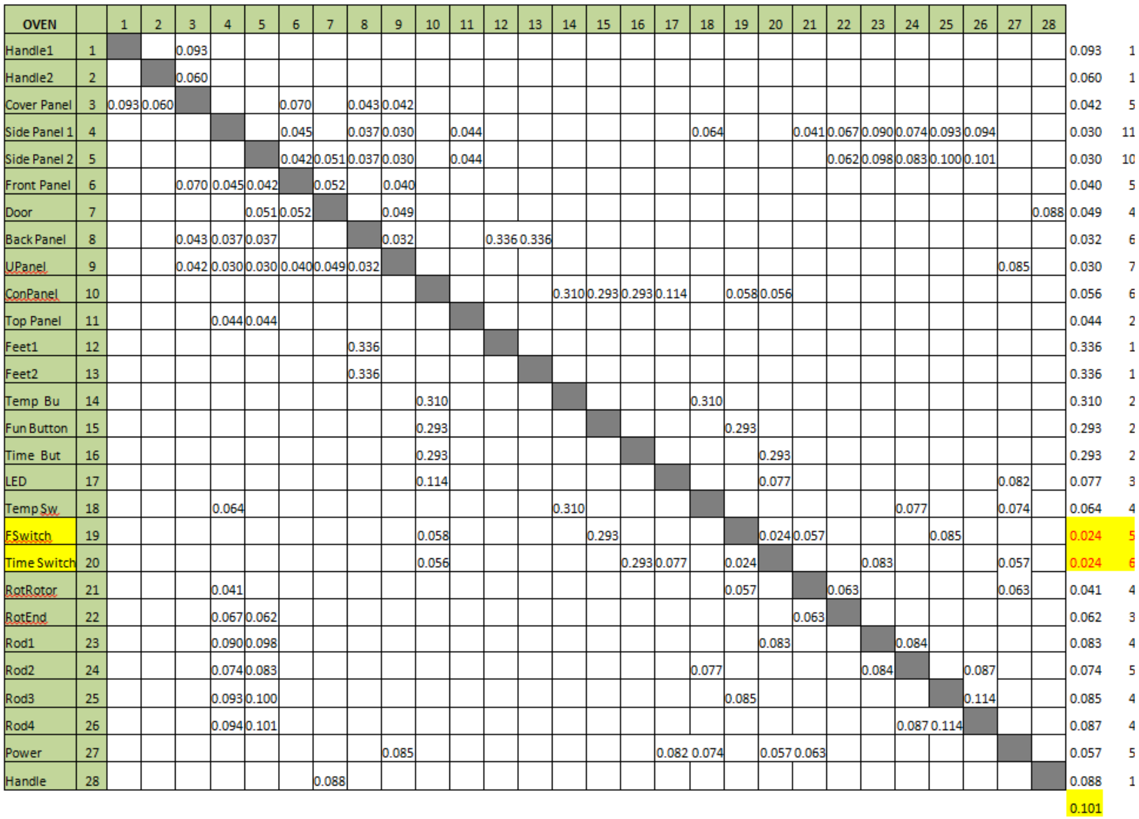


Phone		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
Back Housing	1	■	0.290					0.290										0.290	2
Front Housing	2	0.290	■	0.017	0.026	0.023	0.037			0.022	0.283	0.044	0.148					0.017	9
Controls PCB	3		0.017	■	0.021	0.013	0.026	0.031	0.032	0.023	0.283	0.029				0.030		0.013	10
Numbers PCB	4		0.026	0.021	■	0.031												0.021	3
Redial PCB	5		0.023	0.013	0.031	■				0.027								0.013	4
Reciever	6		0.037	0.026			■											0.026	2
Ringer	7	0.290		0.031				■										0.031	2
Mic	8			0.032					■									0.032	1
Numbers/Buttons	9		0.022	0.023		0.027				■								0.022	3
Volume Button	10		0.283	0.283							■							0.283	2
Curly Cable	11		0.044	0.029								■		0.114				0.029	3
Holster Upper Housing	12		0.148										■	0.066				0.066	2
Holster Lower Housing	13											0.114	0.066	■	0.129	0.081	0.102	0.066	5
Weight	14													0.129	■			0.129	1
Holster PCB	15			0.030										0.081		■	0.030	0.030	3
Main wall cable	16													0.102		0.030	■	0.030	2
																		0.069	

B.21. Oven Model

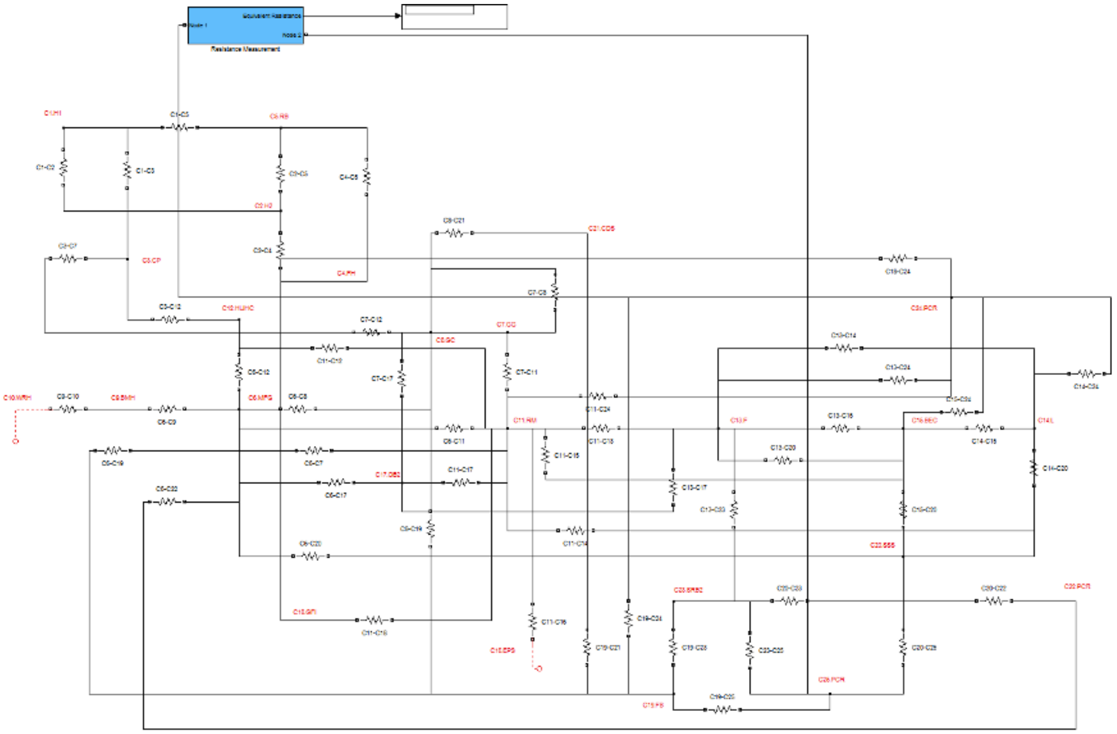


B.22. Oven Output

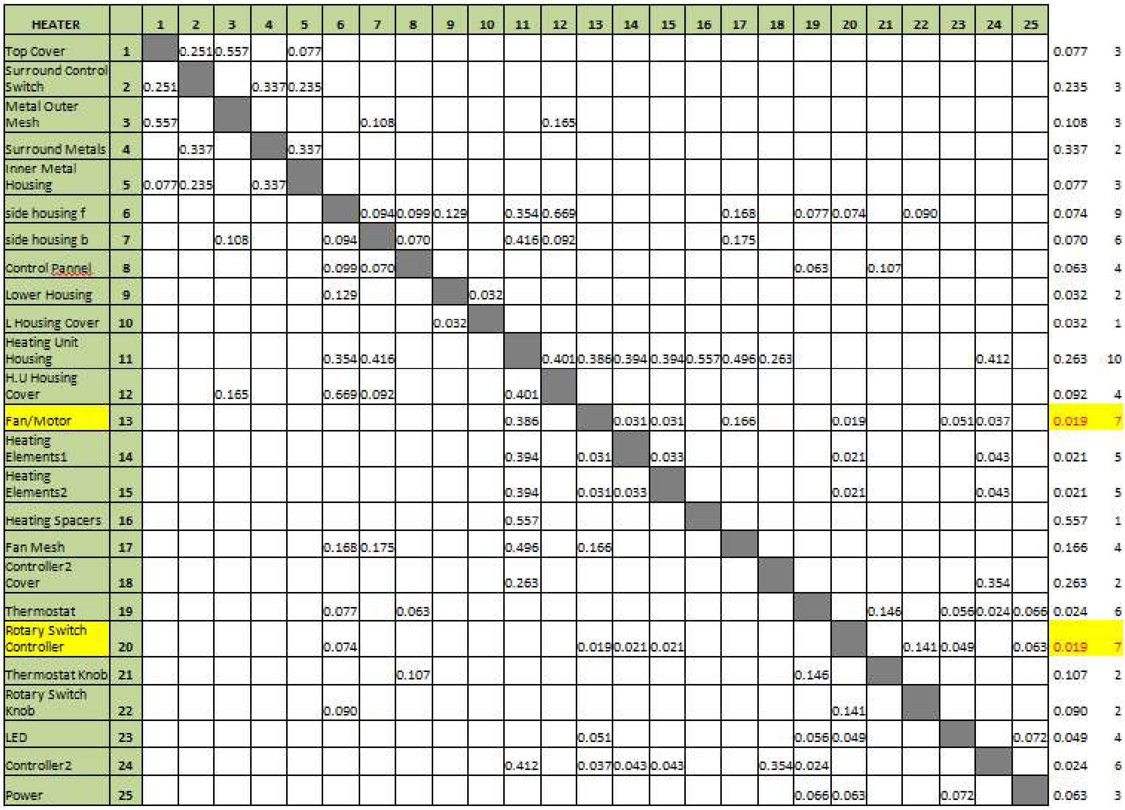


OVEN		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		
Handle1	1	■		0.093																										0.093	1
Handle2	2		■	0.060																										0.060	1
Cover Panel	3	0.093	0.060	■			0.070		0.043	0.042																				0.042	5
Side Panel 1	4				■		0.045		0.037	0.030		0.044							0.064			0.041	0.067	0.090	0.074	0.093	0.094			0.030	11
Side Panel 2	5					■	0.042	0.051	0.037	0.030		0.044											0.062	0.098	0.083	0.100	0.101			0.030	10
Front Panel	6			0.070	0.045	0.042	■	0.052		0.040																				0.040	5
Door	7					0.051	0.052	■		0.049																			0.088	0.049	4
Back Panel	8			0.043	0.037	0.037			■	0.032			0.336	0.336																0.032	6
UPanel	9			0.042	0.030	0.030	0.040	0.049	0.032	■																		0.085		0.030	7
ConPanel	10										■				0.310	0.293	0.293	0.114		0.058	0.056									0.056	6
Top Panel	11				0.044	0.044						■																		0.044	2
Feet1	12								0.336				■																	0.336	1
Feet2	13								0.336					■																0.336	1
Temp Bu	14										0.310				■				0.310											0.310	2
Fun Button	15										0.293					■				0.293										0.293	2
Time But	16										0.293						■				0.293									0.293	2
LED	17										0.114							■			0.077							0.082		0.077	3
Temp Sw	18				0.064										0.310				■						0.077			0.074		0.064	4
FSwitch	19										0.058					0.293				■	0.024	0.057				0.085				0.024	5
Time Switch	20										0.056						0.293	0.077		0.024	■			0.083				0.057		0.024	6
RotRotor	21				0.041															0.057		■	0.063					0.063		0.041	4
RotEnd	22				0.067	0.062																0.063	■							0.062	3
Rod1	23				0.090	0.098															0.083			■	0.084					0.083	4
Rod2	24				0.074	0.083													0.077					0.084	■		0.087			0.074	5
Rod3	25				0.093	0.100														0.085						■	0.114			0.085	4
Rod4	26				0.094	0.101																			0.087	0.114	■			0.087	4
Power	27									0.085								0.082	0.074		0.057	0.063						■		0.057	5
Handle	28							0.088																					■	0.088	1
																														0.101	

B.23. Heater Model

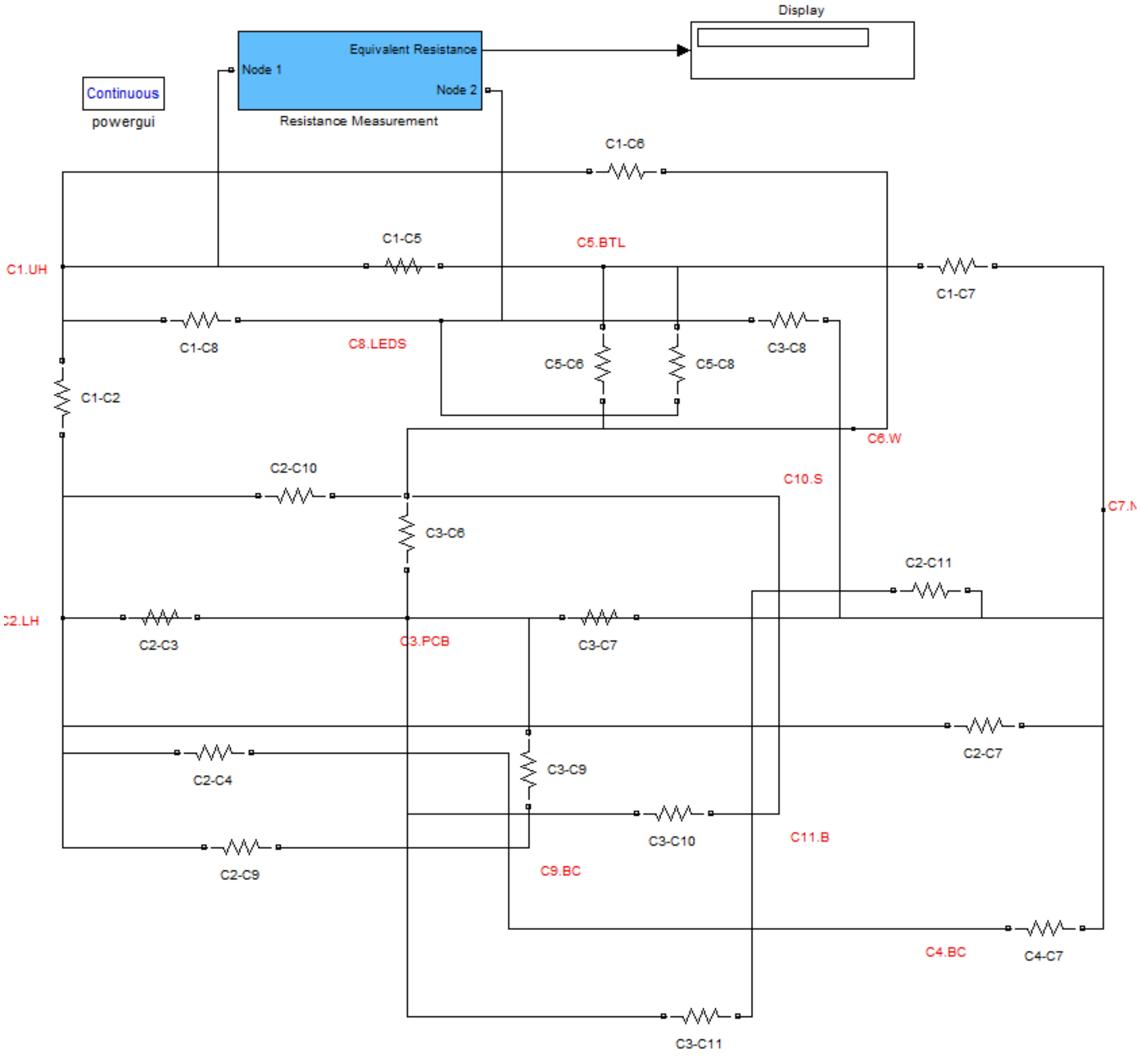


B.24. Heater Output

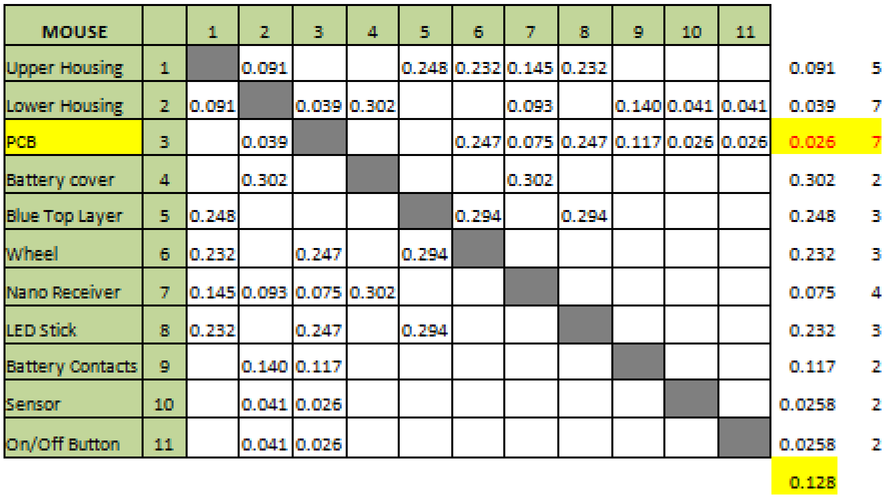


HEATER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
Top Cover	1	▨	0.251	0.557		0.077																					0.077	3
Surround Control Switch	2	0.251	▨		0.337	0.235																					0.235	3
Metal Outer Mesh	3	0.557		▨				0.108					0.165														0.108	3
Surround Metals	4		0.337		▨	0.337																					0.337	2
Inner Metal Housing	5	0.077	0.235		0.337	▨																					0.077	3
side housing f	6						▨	0.094	0.099	0.129		0.354	0.669					0.168		0.077	0.074		0.090				0.074	9
side housing b	7			0.108			0.094	▨	0.070			0.416	0.092					0.175									0.070	6
Control Pannel	8						0.099	0.070	▨											0.063		0.107					0.063	4
Lower Housing	9						0.129			▨	0.032																0.032	2
L Housing Cover	10									0.032	▨																0.032	1
Heating Unit Housing	11						0.354	0.416				▨	0.401	0.386	0.394	0.394	0.557	0.496	0.263						0.412		0.263	10
H.U Housing Cover	12			0.165			0.669	0.092				0.401	▨														0.092	4
Fan/Motor	13											0.386		▨	0.031	0.031		0.166			0.019			0.051	0.037		0.019	7
Heating Elements1	14											0.394		0.031	▨	0.033					0.021				0.043		0.021	5
Heating Elements2	15											0.394		0.031	0.033	▨					0.021				0.043		0.021	5
Heating Spacers	16											0.557					▨										0.557	1
Fan Mesh	17						0.168	0.175				0.496		0.166				▨									0.166	4
Controller2 Cover	18											0.263							▨						0.354		0.263	2
Thermostat	19						0.077		0.063											▨		0.146		0.056	0.024	0.066	0.024	6
Rotary Switch Controller	20						0.074							0.019	0.021	0.021					▨		0.141	0.049		0.063	0.019	7
Thermostat Knob	21								0.107											0.146		▨					0.107	2
Rotary Switch Knob	22						0.090														0.141		▨				0.090	2
LED	23													0.051						0.056	0.049			▨		0.072	0.049	4
Controller2	24											0.412		0.037	0.043	0.043			0.354	0.024					▨		0.024	6
Power	25																			0.066	0.063			0.072		▨	0.063	3

B.25. Mouse Model

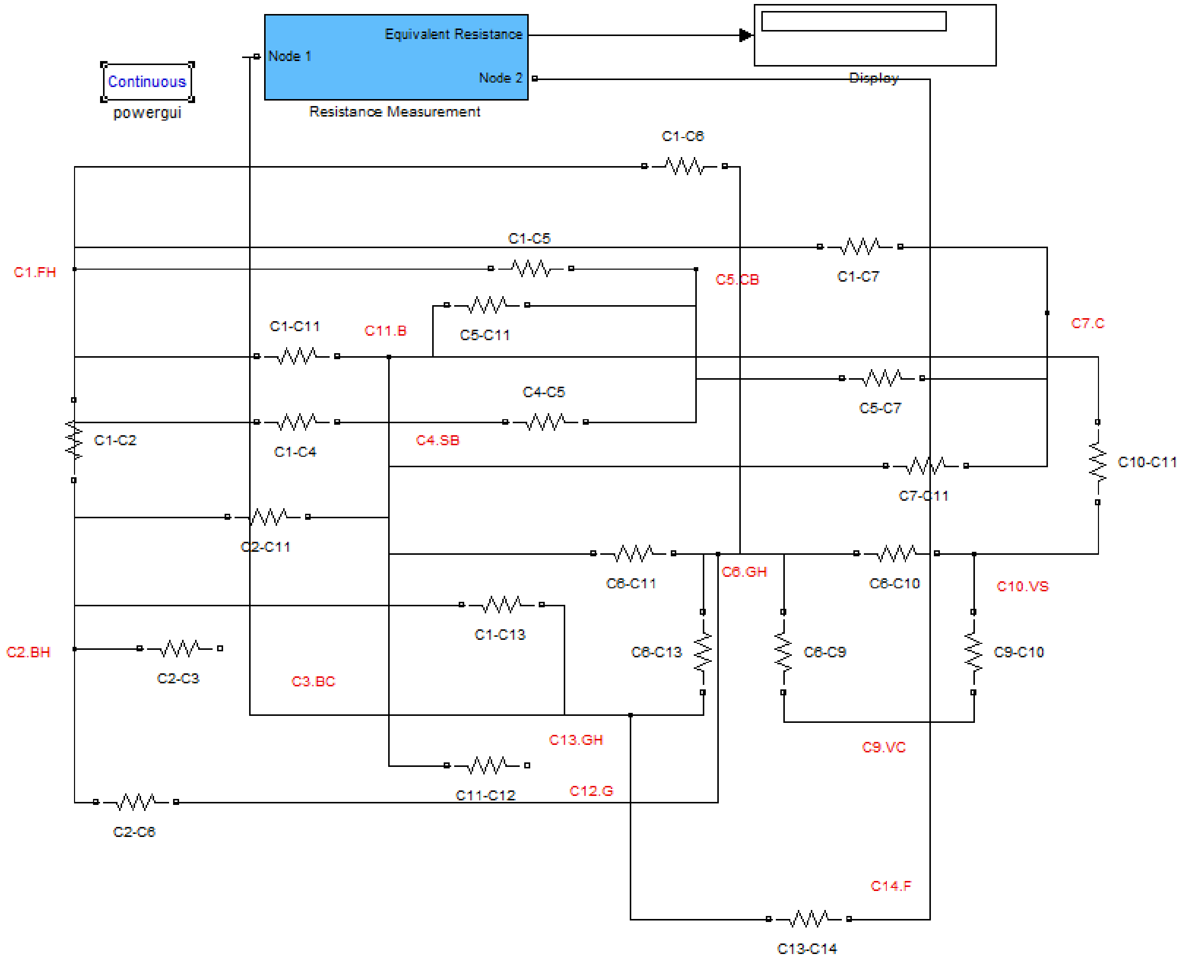


B.26. Mouse Output

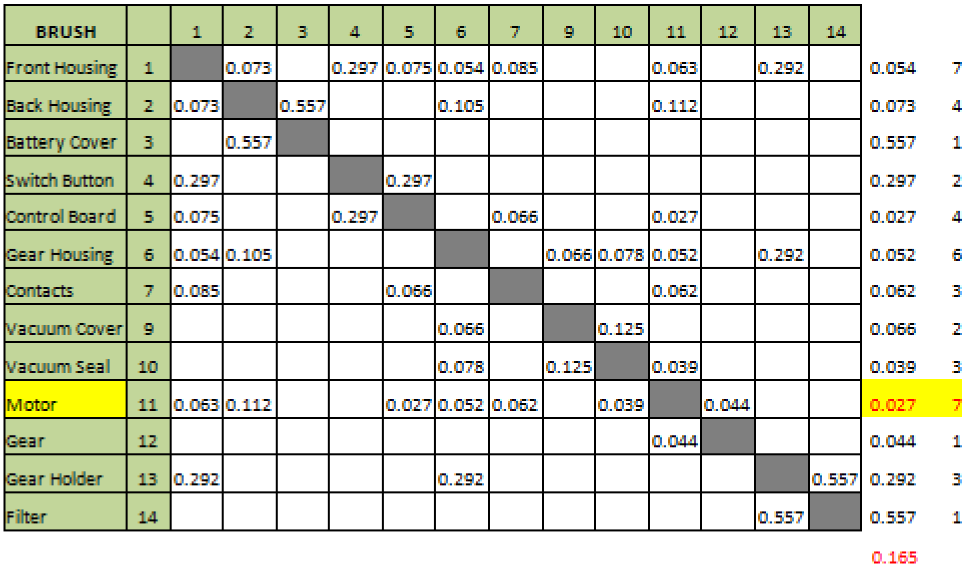


MOUSE		1	2	3	4	5	6	7	8	9	10	11		
Upper Housing	1		0.091			0.248	0.232	0.145	0.232				0.091	5
Lower Housing	2	0.091		0.039	0.302			0.093		0.140	0.041	0.041	0.039	7
PCB	3		0.039				0.247	0.075	0.247	0.117	0.026	0.026	0.026	7
Battery cover	4		0.302					0.302					0.302	2
Blue Top Layer	5	0.248					0.294		0.294				0.248	3
Wheel	6	0.232		0.247		0.294							0.232	3
Nano Receiver	7	0.145	0.093	0.075	0.302								0.075	4
LED Stick	8	0.232		0.247		0.294							0.232	3
Battery Contacts	9		0.140	0.117									0.117	2
Sensor	10		0.041	0.026									0.0258	2
On/Off Button	11		0.041	0.026									0.0258	2
													0.128	

B.27. Electric Brush Model

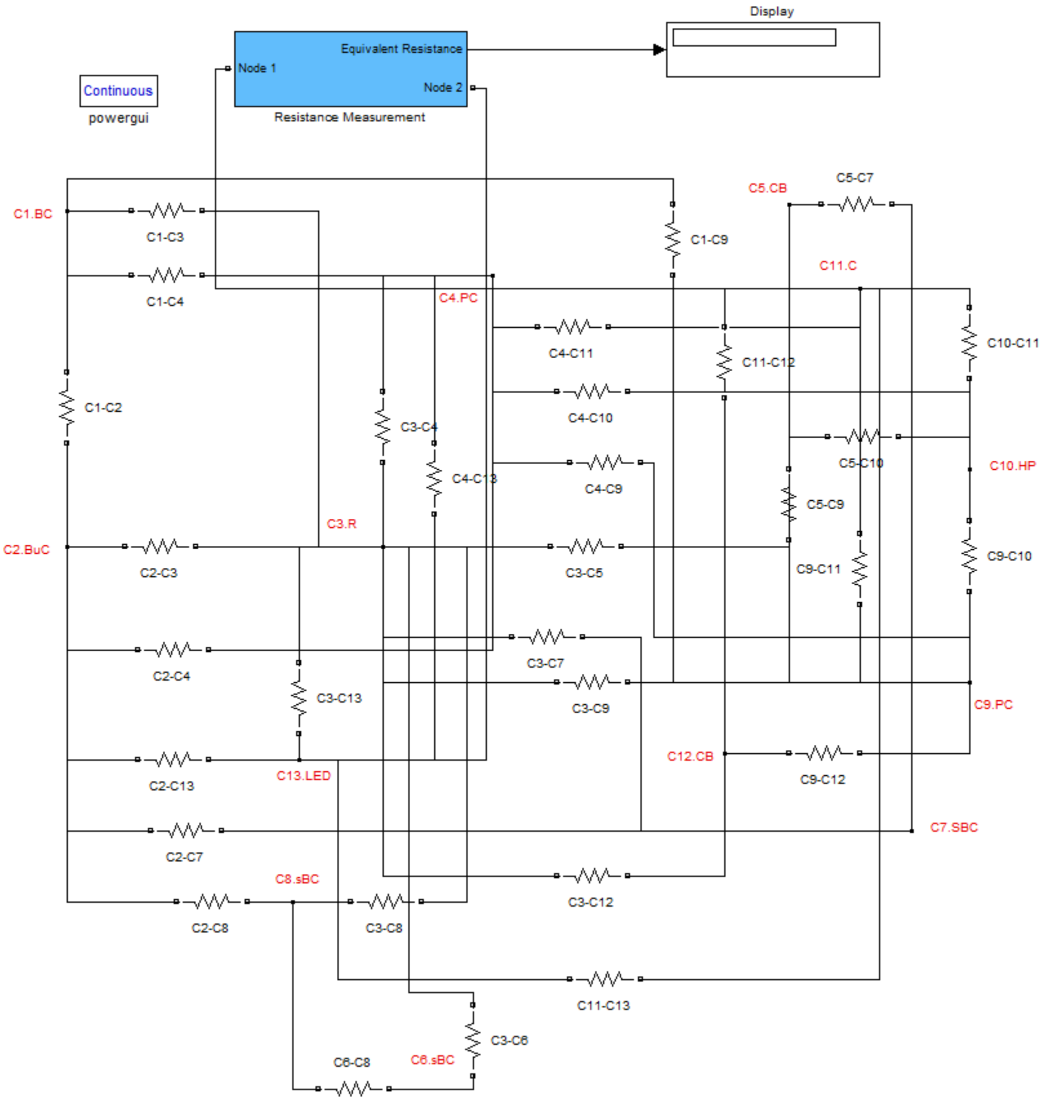


B.28. Electric Brush Output

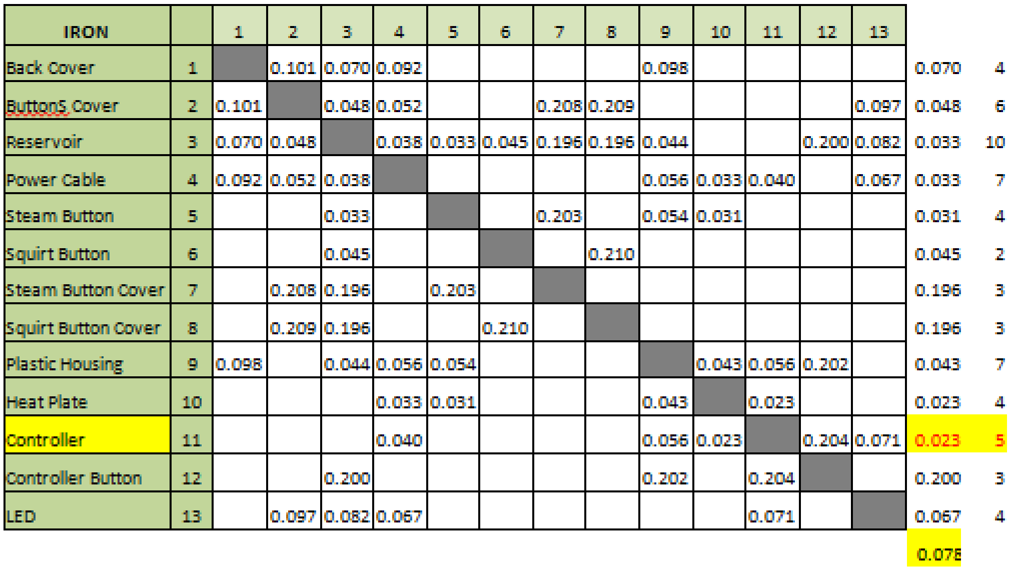


BRUSH		1	2	3	4	5	6	7	9	10	11	12	13	14		
Front Housing	1		0.073		0.297	0.075	0.054	0.085			0.063		0.292		0.054	7
Back Housing	2	0.073		0.557			0.105				0.112				0.073	4
Battery Cover	3		0.557												0.557	1
Switch Button	4	0.297				0.297									0.297	2
Control Board	5	0.075			0.297			0.066			0.027				0.027	4
Gear Housing	6	0.054	0.105						0.066	0.078	0.052		0.292		0.052	6
Contacts	7	0.085				0.066					0.062				0.062	3
Vacuum Cover	9						0.066			0.125					0.066	2
Vacuum Seal	10						0.078		0.125		0.039				0.039	3
Motor	11	0.063	0.112			0.027	0.052	0.062		0.039		0.044			0.027	7
Gear	12										0.044				0.044	1
Gear Holder	13	0.292					0.292							0.557	0.292	3
Filter	14												0.557		0.557	1
															0.165	

B.29. Iron Model

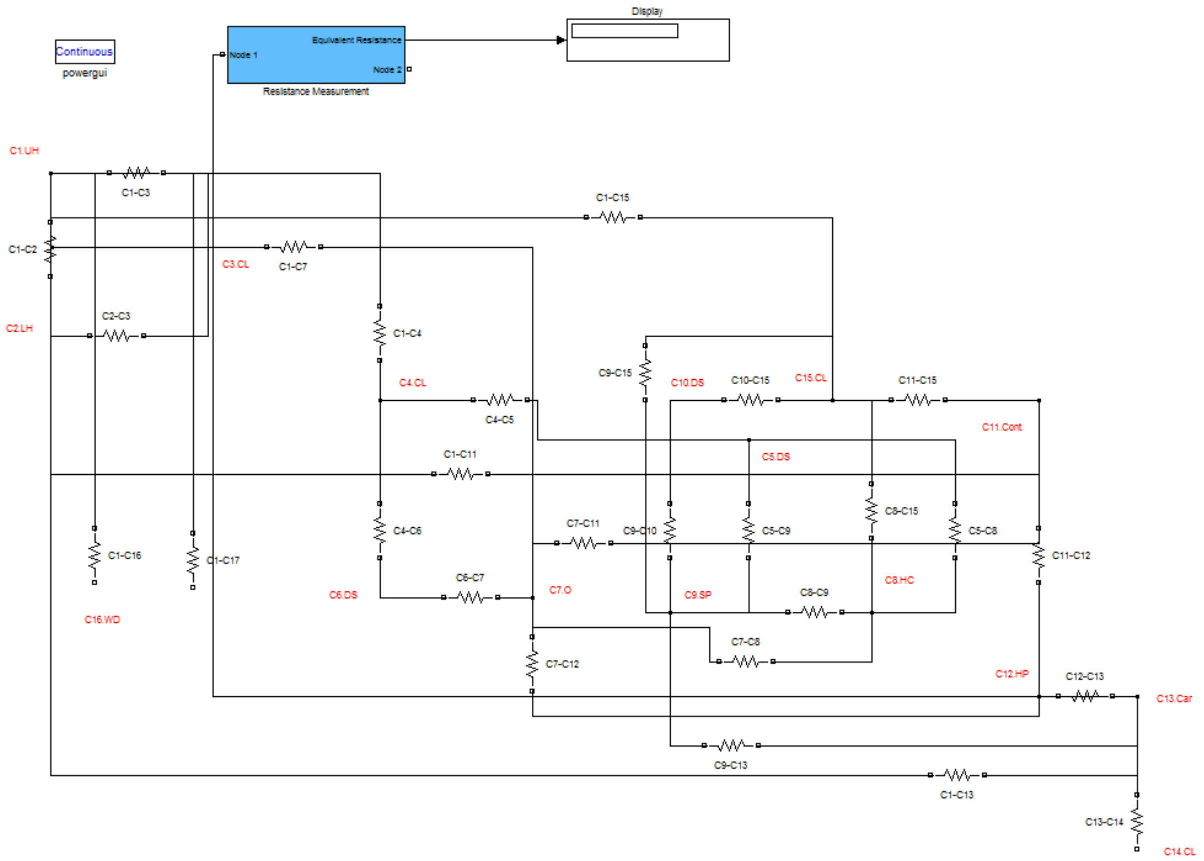


B.30. Iron Output

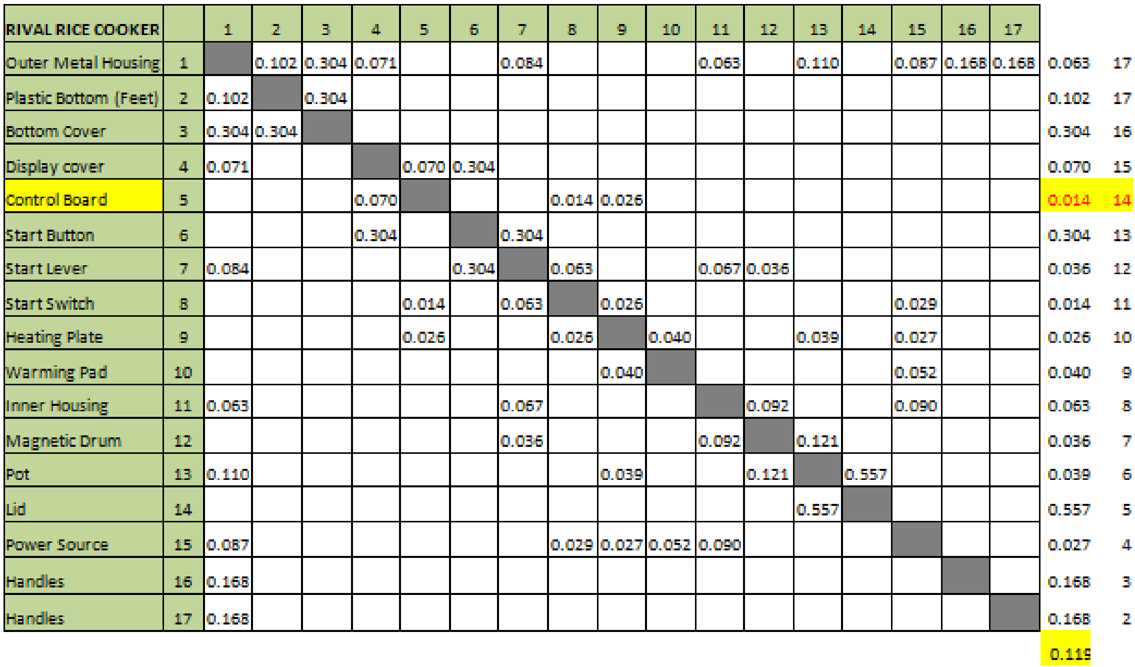


IRON		1	2	3	4	5	6	7	8	9	10	11	12	13		
Back Cover	1		0.101	0.070	0.092					0.098					0.070	4
ButtonS Cover	2	0.101		0.048	0.052			0.208	0.209					0.097	0.048	6
Reservoir	3	0.070	0.048		0.038	0.033	0.045	0.196	0.196	0.044			0.200	0.082	0.033	10
Power Cable	4	0.092	0.052	0.038						0.056	0.033	0.040		0.067	0.033	7
Steam Button	5			0.033				0.203		0.054	0.031				0.031	4
Squirt Button	6			0.045					0.210						0.045	2
Steam Button Cover	7		0.208	0.196		0.203									0.196	3
Squirt Button Cover	8		0.209	0.196			0.210								0.196	3
Plastic Housing	9	0.098		0.044	0.056	0.054					0.043	0.056	0.202		0.043	7
Heat Plate	10				0.033	0.031				0.043		0.023			0.023	4
Controller	11				0.040					0.056	0.023		0.204	0.071	0.023	5
Controller Button	12			0.200						0.202		0.204			0.200	3
LED	13		0.097	0.082	0.067							0.071			0.067	4
															0.078	

B.31. Rival Rice Cooker (5 Cup) Model

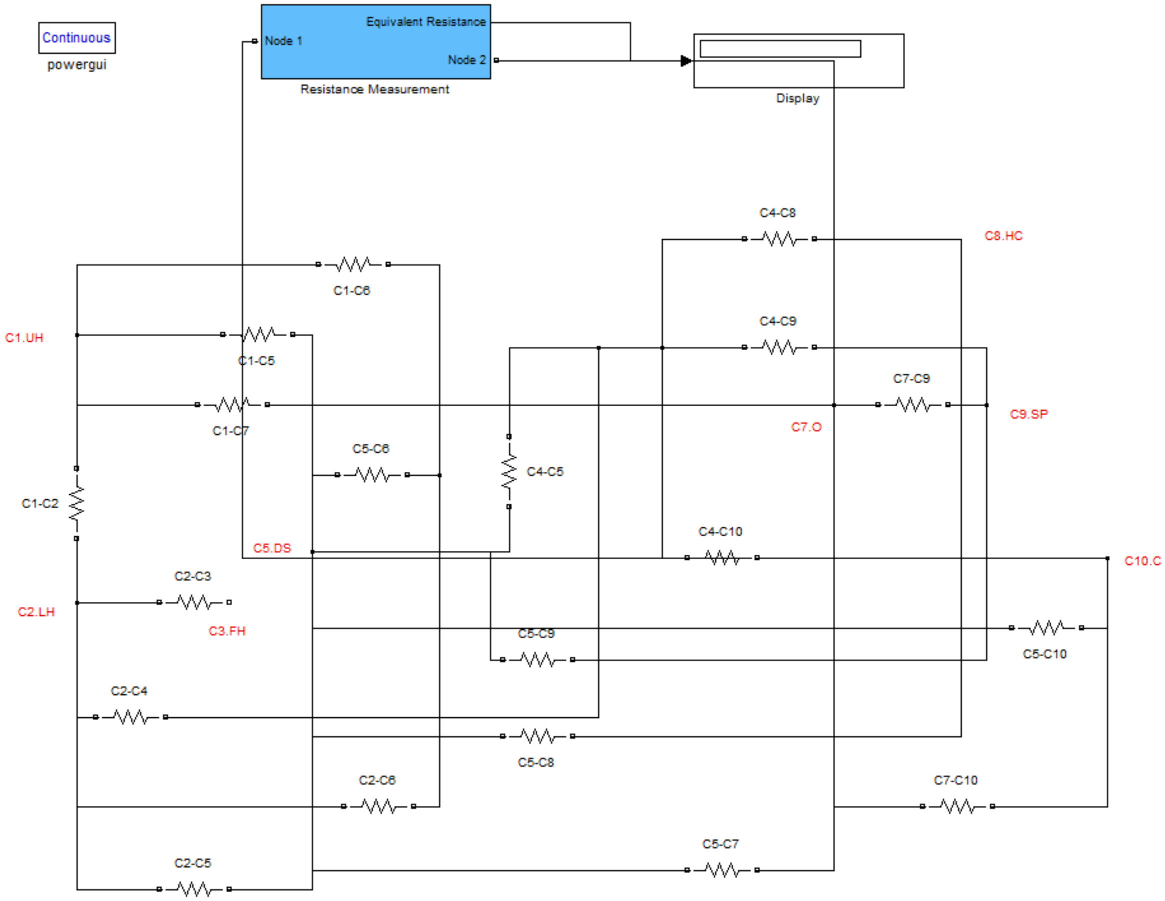


B.32. Rival Rice Cooker (5 Cup) Output

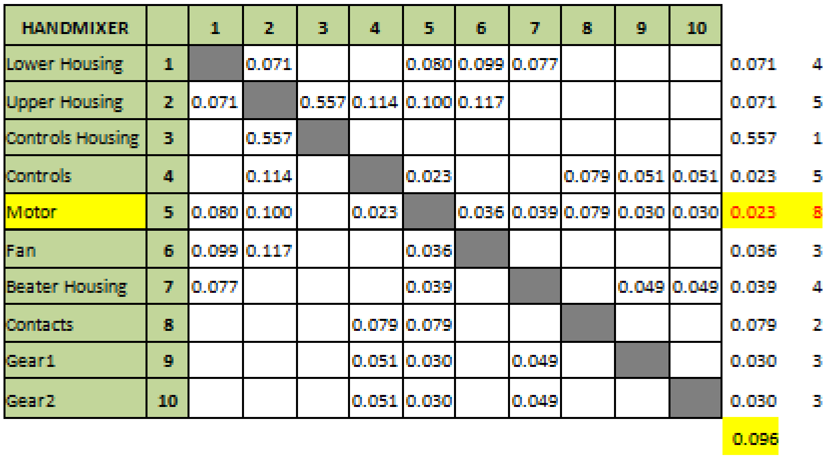


RIVAL RICE COOKER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
Outer Metal Housing	1		0.102	0.304	0.071			0.084				0.063		0.110		0.087	0.168	0.168	0.063	17
Plastic Bottom (Feet)	2	0.102		0.304															0.102	17
Bottom Cover	3	0.304	0.304																0.304	16
Display cover	4	0.071				0.070	0.304												0.070	15
Control Board	5				0.070				0.014	0.026									0.014	14
Start Button	6				0.304			0.304											0.304	13
Start Lever	7	0.084					0.304		0.063			0.067	0.036						0.036	12
Start Switch	8					0.014		0.063		0.026						0.029			0.014	11
Heating Plate	9					0.026			0.026		0.040			0.039		0.027			0.026	10
Warming Pad	10									0.040						0.052			0.040	9
Inner Housing	11	0.063						0.067					0.092			0.090			0.063	8
Magnetic Drum	12							0.036				0.092		0.121					0.036	7
Pot	13	0.110								0.039			0.121		0.557				0.039	6
Lid	14													0.557					0.557	5
Power Source	15	0.087							0.029	0.027	0.052	0.090							0.027	4
Handles	16	0.168																	0.168	3
Handles	17	0.168																	0.168	2
																			0.119	

B.33. Handmixer Model

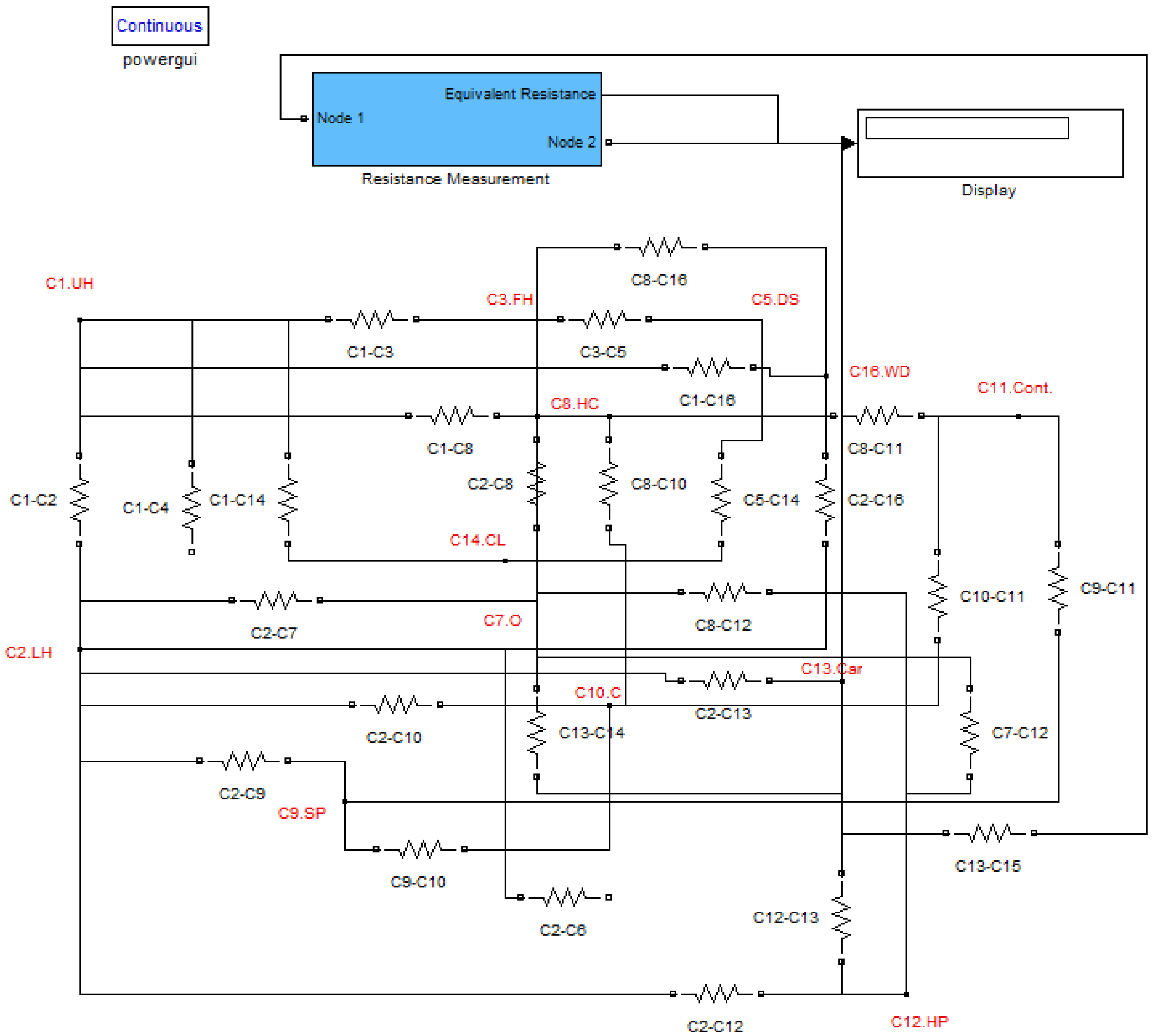


B.34. Handmixer Output

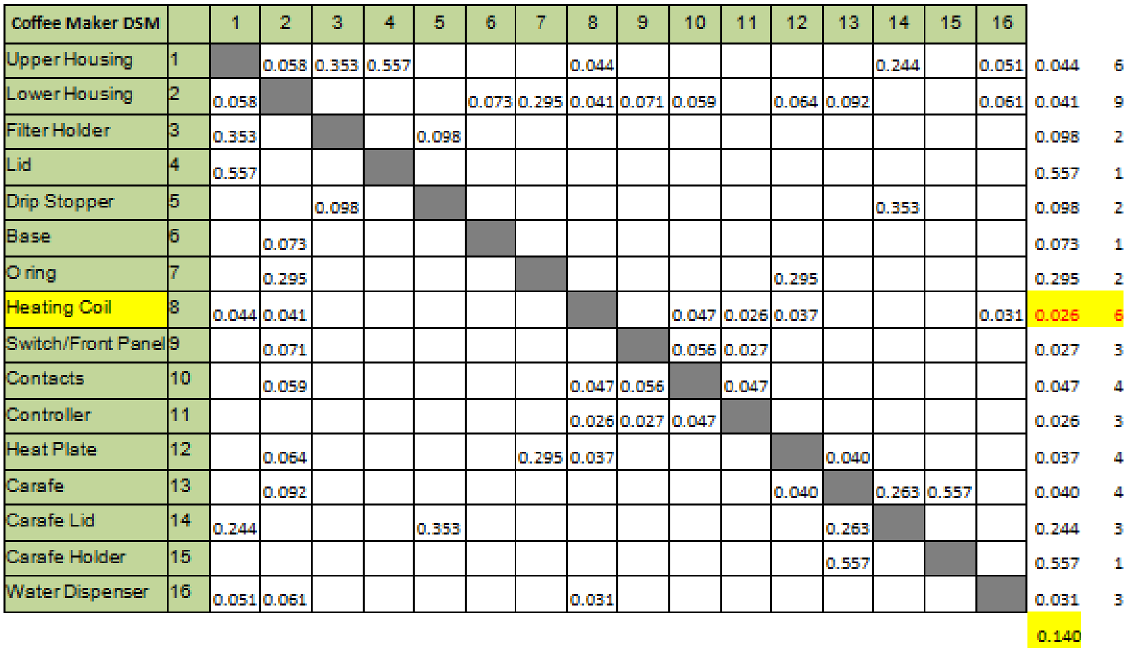


HANDMIXER		1	2	3	4	5	6	7	8	9	10		
Lower Housing	1		0.071			0.080	0.099	0.077				0.071	4
Upper Housing	2	0.071		0.557	0.114	0.100	0.117					0.071	5
Controls Housing	3		0.557									0.557	1
Controls	4		0.114			0.023			0.079	0.051	0.051	0.023	5
Motor	5	0.080	0.100		0.023		0.036	0.039	0.079	0.030	0.030	0.023	8
Fan	6	0.099	0.117			0.036						0.036	3
Beater Housing	7	0.077				0.039				0.049	0.049	0.039	4
Contacts	8				0.079	0.079						0.079	2
Gear1	9				0.051	0.030		0.049				0.030	3
Gear2	10				0.051	0.030		0.049				0.030	3
												0.096	

B.35. Coffee Maker Model

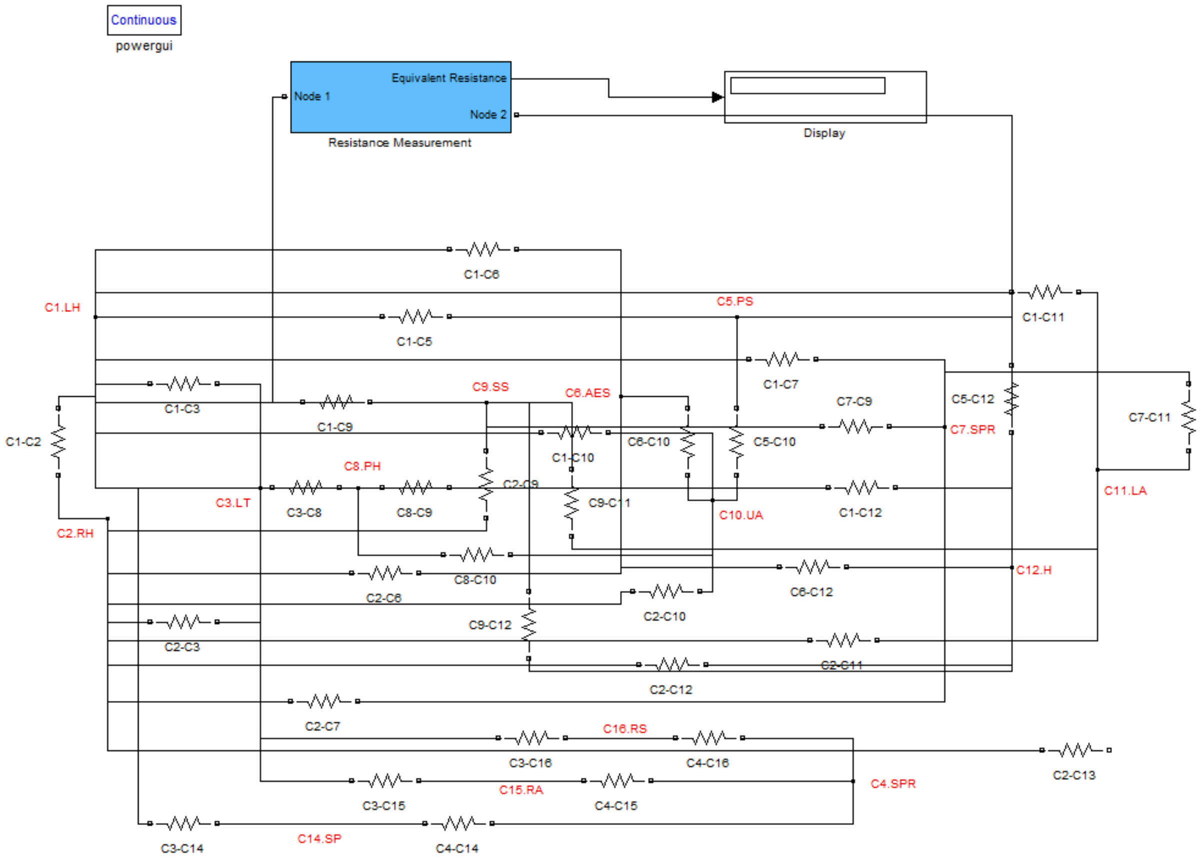


B.36. Coffee Maker Output

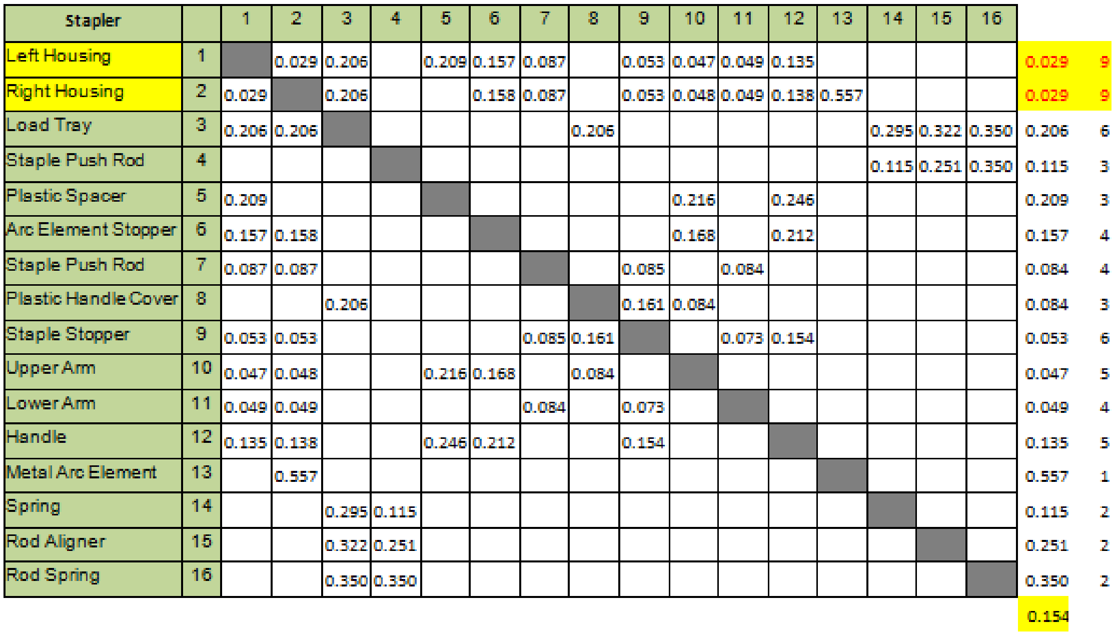


Coffee Maker DSM		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
Upper Housing	1	■	0.058	0.353	0.557				0.044						0.244		0.051	0.044	6
Lower Housing	2	0.058	■				0.073	0.295	0.041	0.071	0.059		0.064	0.092			0.061	0.041	9
Filter Holder	3	0.353		■		0.098												0.098	2
Lid	4	0.557			■													0.557	1
Drip Stopper	5			0.098		■									0.353			0.098	2
Base	6		0.073				■											0.073	1
O ring	7		0.295					■					0.295					0.295	2
Heating Coil	8	0.044	0.041						■		0.047	0.026	0.037				0.031	0.026	6
Switch/Front Panel	9		0.071							■	0.056	0.027						0.027	3
Contacts	10		0.059						0.047	0.056	■	0.047						0.047	4
Controller	11								0.026	0.027	0.047	■						0.026	3
Heat Plate	12		0.064					0.295	0.037				■	0.040				0.037	4
Carafe	13		0.092										0.040	■	0.263	0.557		0.040	4
Carafe Lid	14	0.244				0.353								0.263	■			0.244	3
Carafe Holder	15													0.557		■		0.557	1
Water Dispenser	16	0.051	0.061						0.031								■	0.031	3
																		0.140	

B.37. Stapler Model

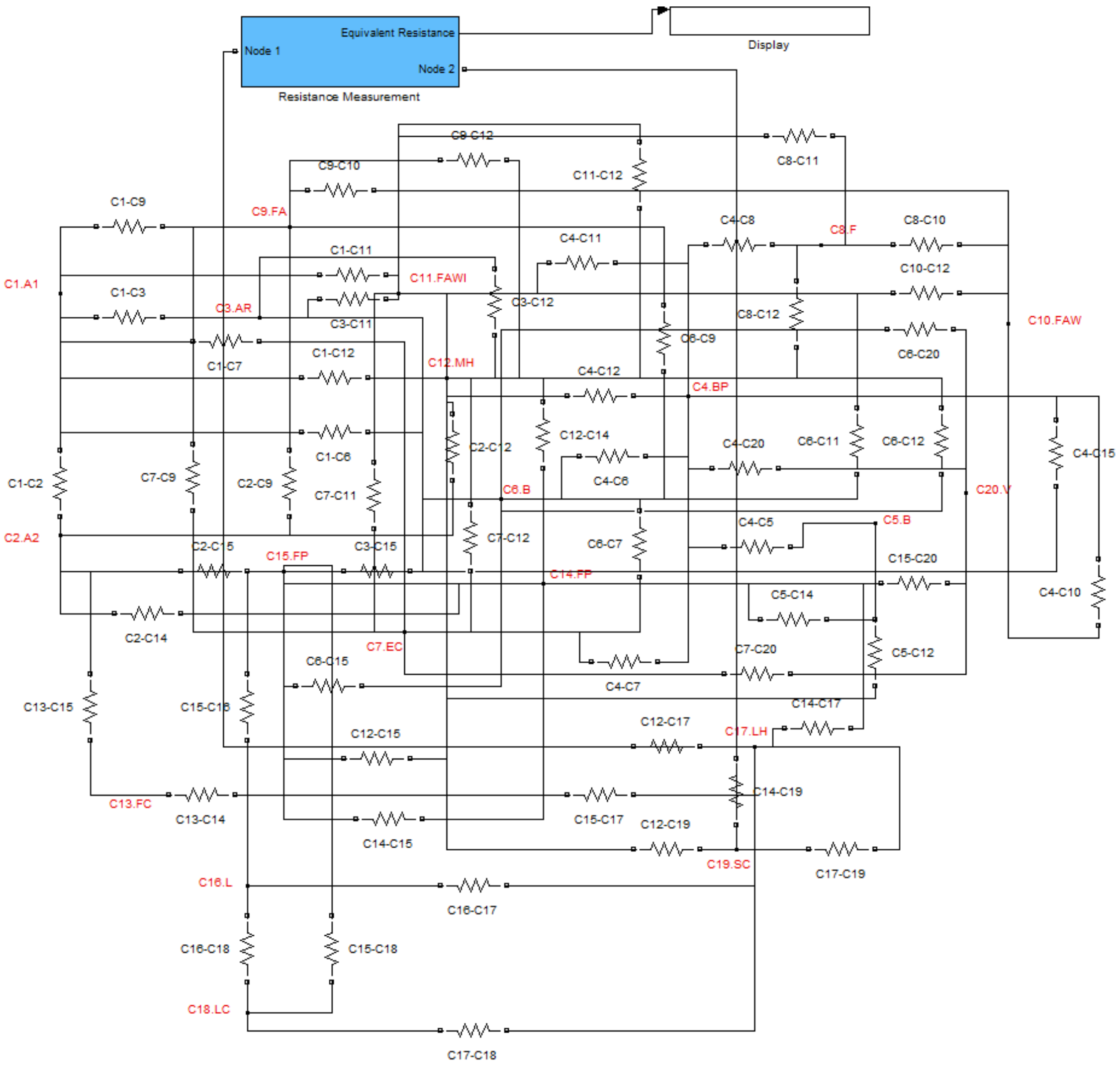


B.38. Stapler Output

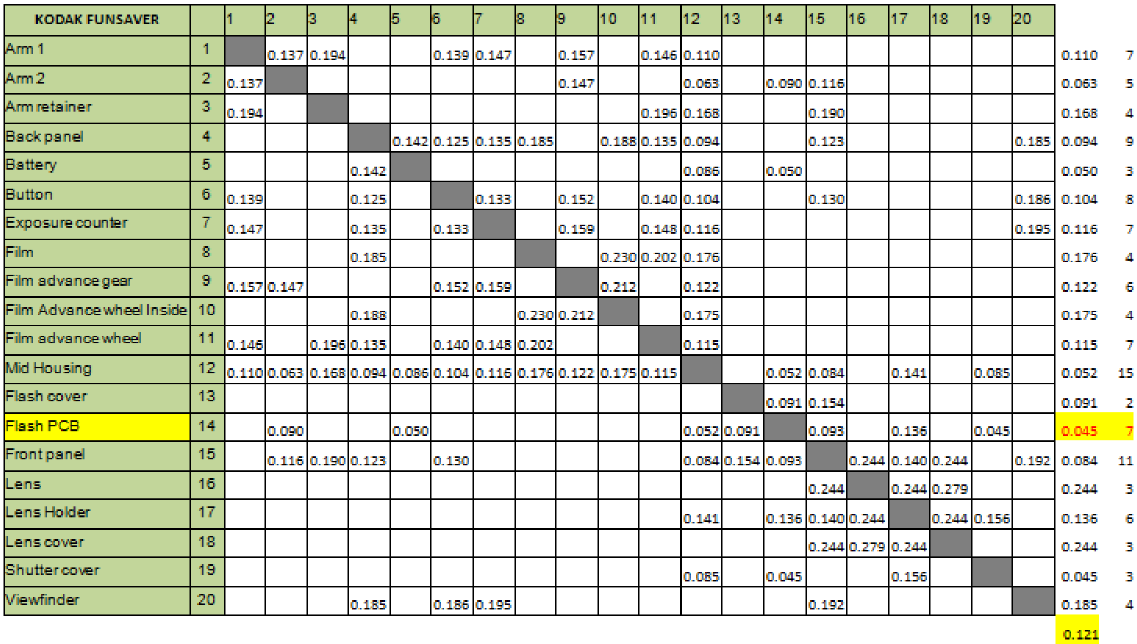


Stapler		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
Left Housing	1		0.029	0.206		0.209	0.157	0.087		0.053	0.047	0.049	0.135					0.029	9
Right Housing	2	0.029		0.206			0.158	0.087		0.053	0.048	0.049	0.138	0.557				0.029	9
Load Tray	3	0.206	0.206						0.206						0.295	0.322	0.350	0.206	6
Staple Push Rod	4														0.115	0.251	0.350	0.115	3
Plastic Spacer	5	0.209									0.216		0.246					0.209	3
Arc Element Stopper	6	0.157	0.158								0.168		0.212					0.157	4
Staple Push Rod	7	0.087	0.087							0.085		0.084						0.084	4
Plastic Handle Cover	8			0.206						0.161	0.084							0.084	3
Staple Stopper	9	0.053	0.053					0.085	0.161			0.073	0.154					0.053	6
Upper Arm	10	0.047	0.048			0.216	0.168		0.084									0.047	5
Lower Arm	11	0.049	0.049					0.084		0.073								0.049	4
Handle	12	0.135	0.138			0.246	0.212			0.154								0.135	5
Metal Arc Element	13		0.557															0.557	1
Spring	14			0.295	0.115													0.115	2
Rod Aligner	15			0.322	0.251													0.251	2
Rod Spring	16			0.350	0.350													0.350	2
																		0.154	

B.39. Single-Use Camera Model

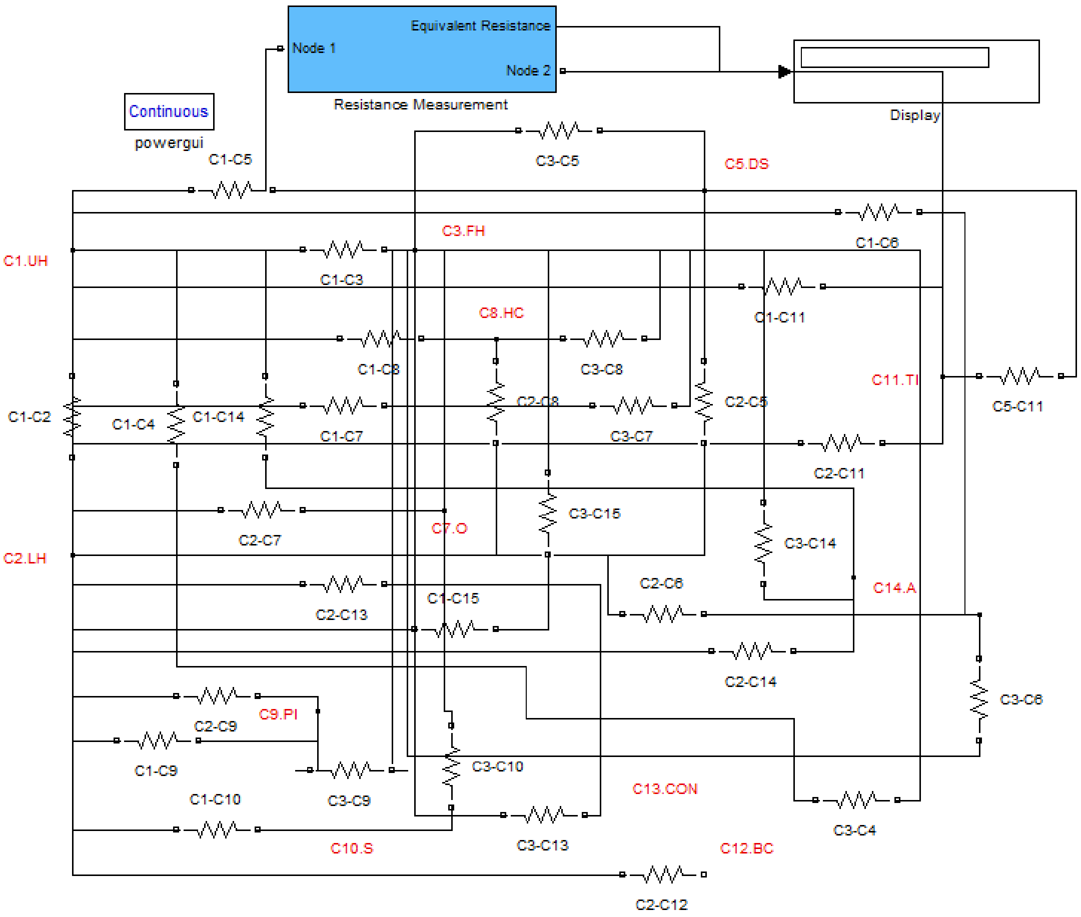


B.40. Single-Use Camera Output

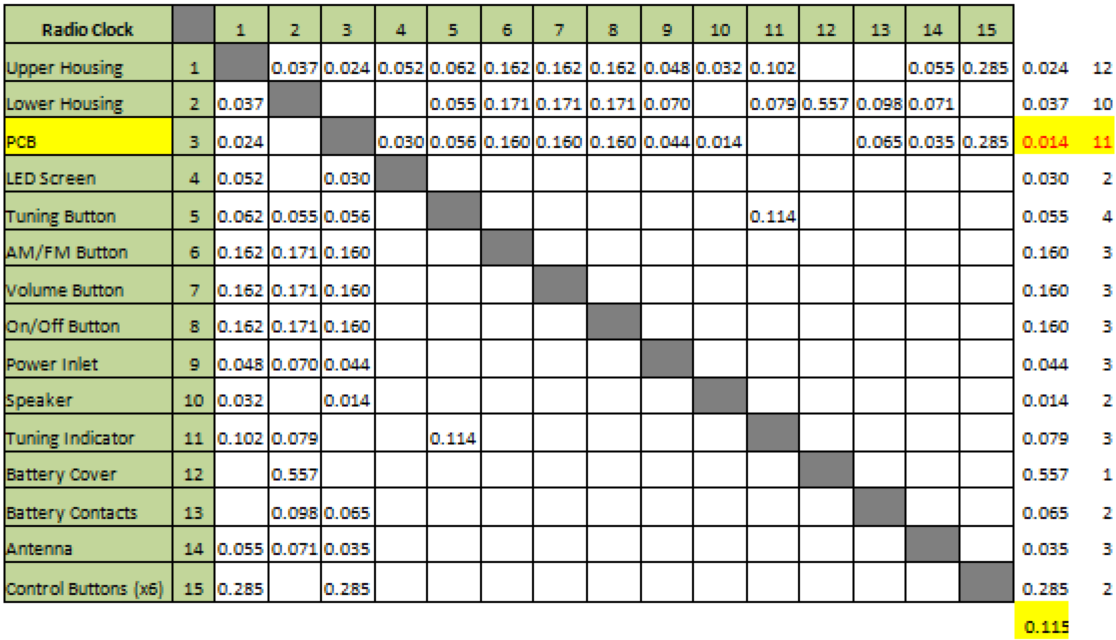


KODAK FUNSAVER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
Arm 1	1		0.137	0.194			0.139	0.147		0.157		0.146	0.110									0.110	7
Arm 2	2	0.137								0.147			0.063		0.090	0.116						0.063	5
Arm retainer	3	0.194										0.196	0.168			0.190						0.168	4
Back panel	4					0.142	0.125	0.135	0.185		0.188	0.135	0.094			0.123					0.185	0.094	9
Battery	5				0.142								0.086		0.050							0.050	3
Button	6	0.139			0.125			0.133		0.152		0.140	0.104			0.130					0.186	0.104	8
Exposure counter	7	0.147			0.135		0.133			0.159		0.148	0.116								0.195	0.116	7
Film	8				0.185						0.230	0.202	0.176									0.176	4
Film advance gear	9	0.157	0.147				0.152	0.159			0.212		0.122									0.122	6
Film Advance wheel Inside	10				0.188				0.230	0.212			0.175									0.175	4
Film advance wheel	11	0.146		0.196	0.135		0.140	0.148	0.202				0.115									0.115	7
Mid Housing	12	0.110	0.063	0.168	0.094	0.086	0.104	0.116	0.176	0.122	0.175	0.115			0.052	0.084		0.141		0.085		0.052	15
Flash cover	13														0.091	0.154						0.091	2
Flash PCB	14		0.090			0.050							0.052	0.091		0.093		0.136		0.045		0.045	7
Front panel	15		0.116	0.190	0.123		0.130						0.084	0.154	0.093		0.244	0.140	0.244		0.192	0.084	11
Lens	16															0.244		0.244	0.279			0.244	3
Lens Holder	17												0.141		0.136	0.140	0.244		0.244	0.156		0.136	6
Lens cover	18															0.244	0.279	0.244				0.244	3
Shutter cover	19												0.085		0.045			0.156				0.045	3
Viewfinder	20				0.185		0.186	0.195								0.192						0.185	4
																						0.121	

B.41. Radio Clock Model



B.42. Radio Clock Output



Radio Clock		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Upper Housing	1		0.037	0.024	0.052	0.062	0.162	0.162	0.162	0.048	0.032	0.102			0.055	0.285	0.024	12
Lower Housing	2	0.037				0.055	0.171	0.171	0.171	0.070		0.079	0.557	0.098	0.071		0.037	10
PCB	3	0.024			0.030	0.056	0.160	0.160	0.160	0.044	0.014			0.065	0.035	0.285	0.014	11
LED Screen	4	0.052		0.030													0.030	2
Tuning Button	5	0.062	0.055	0.056								0.114					0.055	4
AM/FM Button	6	0.162	0.171	0.160													0.160	3
Volume Button	7	0.162	0.171	0.160													0.160	3
On/Off Button	8	0.162	0.171	0.160													0.160	3
Power Inlet	9	0.048	0.070	0.044													0.044	3
Speaker	10	0.032		0.014													0.014	2
Tuning Indicator	11	0.102	0.079			0.114											0.079	3
Battery Cover	12		0.557														0.557	1
Battery Contacts	13		0.098	0.065													0.065	2
Antenna	14	0.055	0.071	0.035													0.035	3
Control Buttons (x6)	15	0.285		0.285													0.285	2
																	0.115	

APPENDIX C

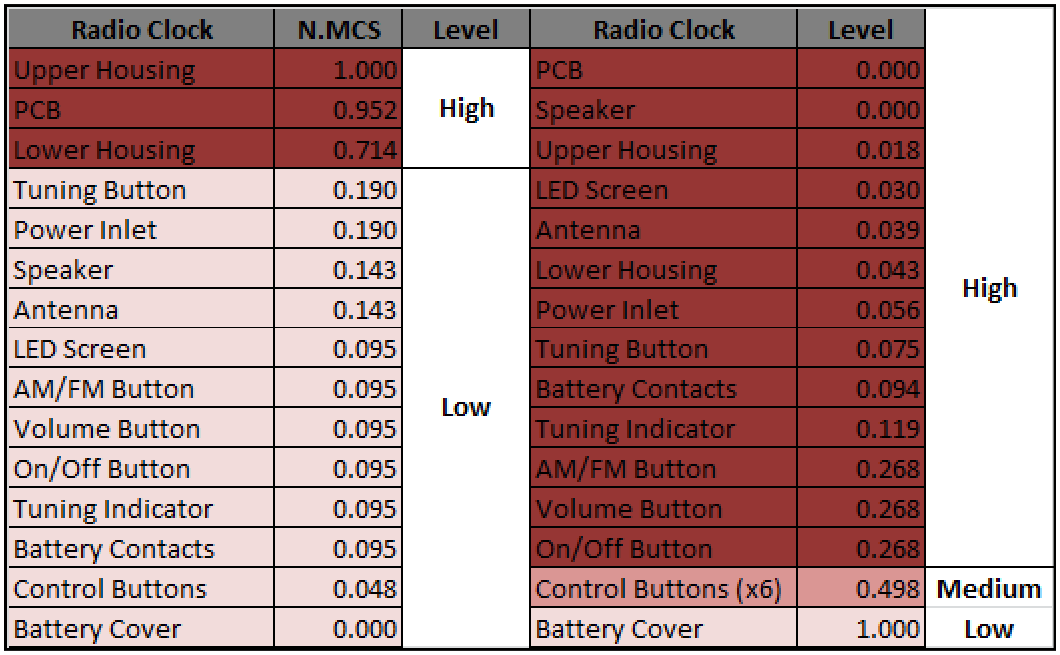
COMPARISON OF RESULTS

C.1. Comparison of Normalized Average Min. C.R values to Normalized MCS Model for the Phone

Phone	N. MCS	Level
Front Housing	1.000	High
Controls PCB	0.811	
Numbers/Buttons	0.811	
Redial PCB	0.351	Medium
Holster Lower Housing	0.297	Low
Numbers PCB	0.216	
Holster PCB	0.135	
Holster Upper Housing	0.108	
Receiver	0.081	
Curly Cable	0.054	
Ringer	0.027	
Weight	0.027	
Main wall cable	0.027	
Back Housing	0.000	
Mic	0.000	
Volume Button	0.000	

Phone	N. CR	Level
Controls PCB	0.000	High
Redial PCB	0.000	
Front Housing	0.016	
Numbers PCB	0.032	
Numbers/Buttons	0.032	
Receiver	0.049	
Curly Cable	0.059	
Holster PCB	0.062	
Main wall cable	0.064	
Ringer	0.067	
Mic	0.069	
Holster Upper Housing	0.192	
Holster Lower Housing	0.192	
Weight	0.419	Medium
Volume Button	0.973	Low
Back Housing	1.000	

C.2. Comparison of Normalized Average Min. C.R values to Normalized MCS Model for the Radio Clock



Radio Clock	N.MCS	Level	Radio Clock	Level	
Upper Housing	1.000	High	PCB	0.000	High
PCB	0.952		Speaker	0.000	
Lower Housing	0.714		Upper Housing	0.018	
Tuning Button	0.190	Low	LED Screen	0.030	
Power Inlet	0.190		Antenna	0.039	
Speaker	0.143		Lower Housing	0.043	
Antenna	0.143		Power Inlet	0.056	
LED Screen	0.095		Tuning Button	0.075	
AM/FM Button	0.095		Battery Contacts	0.094	
Volume Button	0.095		Tuning Indicator	0.119	
On/Off Button	0.095		AM/FM Button	0.268	
Tuning Indicator	0.095		Volume Button	0.268	
Battery Contacts	0.095		On/Off Button	0.268	
Control Buttons	0.048		Control Buttons (x6)	0.498	Medium
Battery Cover	0.000		Battery Cover	1.000	Low

VITA

Öykü Aşıkoğlu was born in Samsun, Turkey on 18 February 1982. She grew up and attended college in Istanbul. She received her Bachelor of Science degree in Geological Engineering in 2004 from Istanbul Technical University. Following two years of industry experience in supply chain management, she joined the Industrial and Manufacturing Engineering Department at The Pennsylvania State University as a Ph.D. student. Over the course of her studies at Penn State, Öykü took part in numerous projects and taught in different departments including The Energy Institute and the College of Information Science and Technology. She conducted her recent research in the Engineering Design and Optimization Group (EDOG) under supervision of Dr. Timothy W. Simpson. She also helped teach two undergraduate level classes on product design (in the Department of Industrial Engineering and in the Department of Mechanical and Nuclear Engineering). Upon graduation, she has accepted an Associate Consultant position. Her academic interests are in the areas of change propagation and flexibility in product architectures.