

**IMECE2004-62312**

## **APPLYING FUNCTIONAL MODELING AS A UNIFYING BASIS FOR DESIGN FOR SIX SIGMA EXECUTION**

**Ryan S. Hutcheson, University of Missouri-Rolla**

**Joseph A. Donndelinger, General Motors Research  
and Development**

**Daniel A. McAdams, University of Missouri-Rolla**

**Robert B. Stone, University of Missouri-Rolla**

### **ABSTRACT**

This paper explores the applicability of the most recently developed methods in functional modeling to Design for Six Sigma transfer function development and requirements flowdown. An example created during a collaborative research project between the General Motors R&D Center and the University of Missouri – Rolla is used to demonstrate the benefits of using standardized functional modeling during conceptual design. The proposed standard for creating the functional models is the Functional Basis. The Functional Basis is a list of function and flow terms that can be used to describe electro-mechanical systems. The example presented in this paper is based on the parking brake system of a passenger car. Module heuristics, function-based rules for partitioning systems, were used to define the sub-systems during the requirements flowdown example. The functional modeling techniques used in this example provide a standard method of capturing current engineering design knowledge while allowing additional knowledge to be discovered.

### **INTRODUCTION**

Six sigma methodologies were introduced by Motorola in the 1980s (Brue and Launsby, 2003) to reduce variation and to improve the quality of their products. Since then, the basic methods have become widely used in a large number of industries. More recently, the philosophies of six sigma are being extended toward the front end of product cycling in attempt to perform Design for Six Sigma (DFSS) (Creveling et al., 2003). There are many challenges in designing something that possesses six sigma level quality. Foremost, no common method exists for creating six sigma designs. Several methodologies have been proposed and are currently being used in the engineering industry. Our attempt is to improve the potential application of Design for Six Sigma by adding a formal function-based structure that assists the development of transfer functions, requirements flowdown and the creation of models that describe system performance. The methods presented in this paper are not limited to DFSS

projects and can be applied to most engineering design projects.

This paper is divided into four sections with the first section defining DFSS and introducing currently applied DFSS methods. The Identify-Design-Optimize-Verify (IDOV) method, probably the most generally accepted method of performing DFSS activities, is also covered. In addition, this section discusses shortcomings with current DFSS methods and a proposal on where functional modeling assists the current DFSS framework. Functional modeling, the Functional Basis and module heuristics are also reviewed in this section. These tools are used to produce and partition a standard functional model.

Section two presents a blueprint for applying function-based design techniques to DFSS projects. The application of functional modeling to each phase of DFSS is outlined and the overall process of conducting function-based DFSS projects is explained.

The third section contains the parking brake example and begins with the application of functional modeling to DFSS transfer function development. This section details the creation of the black-box and functional model of the system as well as the creation of transfer functions based on the functional model. The application of functional modeling to requirements flowdown in the parking brake system is also explored in this section. The parking brake system is partitioned using module heuristics. The resulting sub-system boundaries are used to investigate the flow of requirements from system to sub-system levels and to create a set of sub-system models.

The final section lists the next steps to be performed. Currently, the opportunity to apply functional modeling techniques to a new product design is being investigated. Also, other areas of application such as engineering knowledge repositories, component selection, technical memory and Failure Mode and Effect Analysis will be introduced.

## NOMENCLATURE

Variable	Description
$\delta_{FCI}$	Displacement of front cable in
$\delta_{FCO}$	Displacement of front cable out
$\delta_{RCI}$	Displacement of rear cable in
$\delta_{RCO}$	Displacement of rear cable out
$\delta_s$	Spring deflection
$\epsilon_{EQ}$	Travel efficiency of equalizer
$\epsilon_{FC}$	Travel efficiency of front cable
$\epsilon_{LVR}$	Travel efficiency of lever
$\epsilon_{RC}$	Travel efficiency of rear cable
$\mu_{SHOE}$	Static coefficient of friction between hat and shoe
$\theta$	Angle of slope
$\theta_{HI}$	Input angle of parking brake handle
$\theta_{HO}$	Output angle of parking brake handle
$\xi_{EQ}$	Force efficiency of equalizer
$\xi_{FC}$	Force efficiency of front cable
$\xi_{LVR}$	Force efficiency of lever
$\xi_{RC}$	Force efficiency of rear cable
$F_{FCI}$	Force into front parking brake cable
$F_{FCO}$	Force out from front parking brake cable
$F_{HI}$	Force into parking brake handle
$F_{HO}$	Force out of parking brake handle
$F_{PRE}$	Preload in spring
$F_{RBL}$	Force into rear parking brake lever
$F_{RCI}$	Force into rear parking brake cable
$F_{RCO}$	Force out from rear parking brake cable
$g$	Acceleration due to gravity
$K_s$	Spring coefficient
$L_{HI}$	Input length of parking brake lever
$L_{HO}$	Output length of parking brake lever
$L_{PBL}$	Moment arm length of brake lever
$L_{SHOE}$	Moment arm length of shoe
$m$	Mass of vehicle
$M_{HAT}$	Moment generated by the brake shoe on the hat
$M_{HO}$	Moment out of parking brake handle
$M_{PBL}$	Moment on parking brake lever
$M_{SHOE}$	Moment acting on brake shoe
$M_W$	Moment generated from vehicle being on incline
$R_{HAT}$	Radius of brake hat
$R_W$	Radius of the rear tire
$X_{PBM}$	Multiplication factor of park brake mechanism

## 1 RELATED WORK

The two basic design methods we are proposing to add to the DFSS toolkit are a formalized functional modeling framework with associated definitions for functions and a set of heuristics used to create modules or system partitions. These methods are outlined briefly in the following sections. For a complete review of these methods, relevant references are given. First, however, we review the state of the art in DFSS and function-based design.

### 1.1 DFSS

Design For Six Sigma (DFSS) is a popular framework for new product development. DFSS may be succinctly defined as “a rigorous process for defining products, services, and/or processes to reduce delivery time, [reduce] development cost, increase effectiveness, and better satisfy the customers” (Brue and Launsby, 2003). An ideal six sigma design exhibits fewer than 3.4 defects per every million opportunities. To design a system for this level of quality, quality issues must be addressed before prototypes are made. As a result, the statistical models used to improve the product must be based on mathematical models rather than experimental results.

The idea of creating six sigma products began at Motorola in the 1980s. After reducing labor costs to nearly half of their previous levels and reducing scrap by around 65%, Motorola quickly became more competitive as a result of six sigma. Other companies began to adopt six sigma philosophies after observing Motorola’s success. These early six sigma efforts focused on improving manufacturing processes. In the 1990s, it became apparent that in order to produce true six sigma products, quality had to be designed into a product starting in its early design stages (Creveling et al., 2003). Thus, Design for Six Sigma was born. Extending six sigma methods back to the early stage of design is currently an active area of research. Thus, there are currently several methodologies that are being applied by various companies. These methods include PIDOV (Plan-Identify-Design-Optimize-Validate), DMADV (Define-Measure-Analyze-Design-Verify) (Brue and Launsby, 2003), I<sup>2</sup>DOV (Invent and Innovate-Develop-Optimize-Verify) and CDOV (Concept Development-Design-Optimization-Verification) (Creveling et al., 2003). The most prevalent method for DFSS implementation in industry is the IDOV method, consisting of the four distinct phases described below:

- **Identify:** Select the best design concept based on the Voice of the Customer
- **Design:** Build a thorough base of knowledge about the design
- **Optimize:** Achieve a balance of quality, cost, and time to market
- **Verify:** Demonstrate that the design meets its requirements

A number of well-known tools are typically applied throughout the execution of the IDOV method. Development of alternative design concepts in the Identify phase is commonly conducted using the Theory of Inventive Problem Solving (TIPS), also commonly known by its original Russian

acronym of TRIZ, (Altshuller, 1984) along with various brainstorming methods (Chowdhury, 2002). Requirements flowdown and system performance modeling are conducted using transfer functions that may be derived from a variety of sources, including first-principles relationships from the physical sciences as well as regressions, response surfaces, simulation models, or even finite element models. Failure Modes and Effects Analysis (FMEA) is often applied in DFSS and may be used as early as the Identify phase to guide definition of design alternatives or as late as the Optimize phase to estimate failure rates and to identify means of reducing them. Collectively, these form a powerful, albeit loosely integrated, suite of tools for DFSS execution.

The additional DFSS techniques mentioned above are similar to the IDOV method. PIDOV consist of the four steps in the IDOV method with the addition of a Plan step. During this planning phase, all vital steps of the project are mapped out. The DMADV process includes the following five steps (Brue and Launsby, 2003):

- **Define:** Determine project goals and customer requirements
- **Measure:** Quantify customer needs and specifications
- **Analyze:** Explore methods of meeting customer requirements
- **Design:** Develop a process to meet customer requirements
- **Verify:** Insure that the design has met customer requirements

The I<sup>2</sup>DOV method consists of four steps and is similar to the IDOV method. The first step is Invention and Innovation. In this phase, business goals and markets are defined, technological trends are identified and technological roadmaps are created. In the Develop stage, technology concepts are generated based on customer information. The Optimize stage consists of increasing the robustness of a design and tuning adjustment factors. Verification, the final phase, involves the integration and validation of sub-systems as well as the complete product. CDOV is similar to the IDOV method but replaces the Identify phase with a customer needs-based Concept development phase (Creveling et. al., 2003).

DFSS techniques focus on creating efficient and high quality designs. However, as reviewed above, no single method has emerged as the definitive DFSS approach. The transfer of DFSS method from experts to practitioners in industry is typically completed through projects. These projects are performed in several different ways but generally are based on a real project of interest to the client. The projects are spaced out over a period of months to enable clients to apply the methods before returning for more training. As a result, there is little consistency between the various implementations of DFSS throughout the industry.

One of the core needs – and core difficulties – of executing DFSS is the development of performance models. The difficulties are compounded in most DFSS methodologies as form and function are often modeled together creating ambiguity between form, function, and the performance model that describes how the form fulfills the function. Also, by including form in the early stages of product development (or

redevelopment) the designers are biased by an existing or assumed form solution and the range of potentially robust concepts is limited.

Another practical challenge of implementing DFSS in large organizations is the communication required between many different groups and the total time spent on the projects. To facilitate better communication between DFSS, systems, manufacturing, and management groups, a clear and consistent methodology is needed in the earliest stages of design. Also, consistency is needed in representations so that prior knowledge can be reused reducing the time spent on each DFSS project.

## 1.2 FUNCTION-BASED DESIGN

The use of a functional framework as a basis for designing systems is not a new idea. Several function-based methods for conduction conceptual design have recently been proposed. What we offer to the function-based design community is a standard framework for applying these tools. The Functional Basis provides an exhaustive and linearly independent (Hirtz et al., 2002) way to represent both dynamic and static systems. Modular heuristics allow designers to partition systems based on functional models created using the Functional Basis. Combined, these tools can provide structure to existing design processes and assist in the application of other function-based design tools.

One such function-based tool is Bracewell and Sharpe's Schemebuilder. Schemebuilder is a set of design tools that uses a "bond graph based ontology of rigorously defined physical and information functions" to assist a designer during the conceptual and embodiment phases of design (Bracewell and Sharpe, 1996). Bond graphs are an energy-based method of representing dynamic systems. Schemebuilder is an attempt to create a computational method to relate the function of a system to its form. Schemebuilder attempts to create potential solutions to a design problem by assembling fragments of bond graphs based on the desired functionality of the system. Due to its computational nature, the functional components of a system must exist within Schemebuilder's predefined knowledge base. In addition, potential solutions are limited to systems that can be represented with a bond graph. These limitations result from Schemebuilder's reliance on bond graph models as a framework.

Rather than focus on a solution method and work back to a design tool, we propose that a functional framework for the entire design process should be defined then design tools adapted to fit within this framework. Using the proposed Functional Basis design framework, a tool such as the bond graph-based Schemebuilder would fit in as a method of translating the functional model's of certain systems to performance models.

## 1.3 FUNCTIONAL MODELING

Functional modeling is a form-independent method of representing electro-mechanical systems (Hundal, 1990; Hubka and Eder, 1984; Murdock et al., 1997; Lai and Wilson, 1989; Iwasaki et al., 1995; Umeda and Tomiyama, 1997; Miles, 1972; Akiyama, 1991; VAI, 1993; Collins et al., 1976; Modarres, 1997; Amoussou et al., 1997; Vicarini, 1995; Szykman et al., 1999). A functional model consists of the

energy, material and signal flows into and out of a system and the functions that are performed on these flows to transform them from an input to a desired output state.

In order to produce functional models that are understood by all members of a DFSS team, it is recommended that the Functional Basis be used. The Functional Basis is a list of function and flow terms, verbs and nouns respectively, that has been developed in a joint effort between NIST, The University of Missouri-Rolla, and The University of Texas-Austin (Hirtz et al., 2002; Stone & Wood, 2000). The functions and flows are broken into three categories: primary, secondary and tertiary. Primary functions and flows are generally used in black-box models. Secondary terms are more specialized and are used in the functional models themselves. Tertiary terms offer an additional level of specification if more detail is required when creating models. The Functional Basis consists of three primary flows, twenty secondary flows, eight primary functions and twenty-one secondary functions.

Creating a functional model represented in the Functional Basis involves five steps. The first step (1) is to identify flows that address customer needs. These needs can be identified through customer surveys and past design efforts (Otto and Wood, 2001; Ulrich and Eppinger, 1995; Ullman, 2003). Once the needs are identified, they are mapped to the inputs and outputs of the system. These inputs and outputs are stated in the Functional Basis. The next step (2) is to create a black-box model of the system. This model contains all the inputs and outputs of the system along with an overall function that describes the system. Function chains are then created to represent the operations performed on a flow to transform it from an input to an output (step 3). These chains are then aggregated to produce an overall functional model (step 4). The next step (5) is to verify that all customer needs are met within the functional model. Functions are added to the model to address any needs that have not been satisfied. The result is a model that describes the function of a system separate from its form in a standardized language (Stone & Wood, 2000; Otto & Wood 2001).

#### 1.4 MODULE HEURISTICS

Performing a flowdown of system level requirements to sub-system level requirements is important for any design method including DFSS. Overall nominal performance expectations and associated variability requirements imply nominal performance and variability expectations for sub-systems and components. As teams work on sub-system design and the associated DFSS projects, clear performance objectives are needed in order to ensure overall system performance. In complex systems, performing requirements flowdown is a challenging task. By partitioning a system into hierarchical chunks (Ulrich and Eppinger, 1995), a requirements flowdown can be performed in a consistent, stepwise manner.

To facilitate requirements flowdown, module heuristics are used to create modules (Stone et al., 2000). The module heuristics provide a method for identifying the sub-systems (modules) of a system based on rules that are empirical in nature yet scientifically proven. The module heuristics depend on the construction of a functional model

and specifically the usage of the Functional Basis to define the functions.

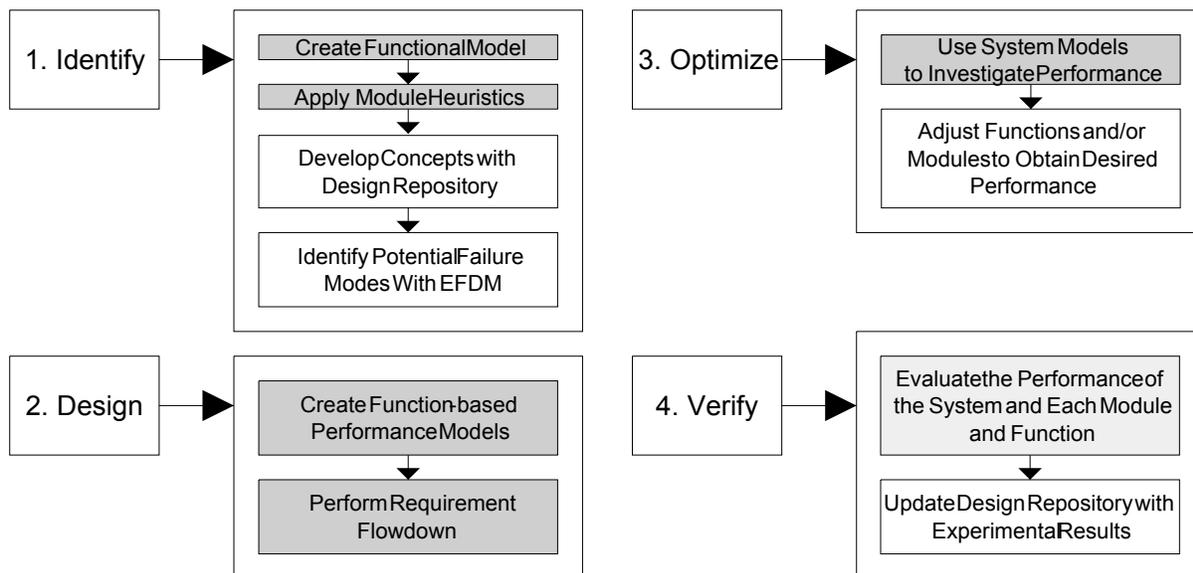
The three module heuristics are the Dominant Flow, Branching Flow and Conversion-Transmission rules. A module created using the Dominant Flow rule includes the functions performed on a single flow from its input through being output or changed into a different flow. Modules are created from the Branching Flow rule by identifying parallel flows that have branched from a single function. Each of these flows is considered a module. The final rule considered, Conversion-Transmission, identifies a module as the functions from the conversion of a flow to a different flow (such as convert human energy to rotational energy) to the transmission of a flow. A complete discussion of the module heuristics and clarifying examples can be found in Stone et al. (2000).

## 2 BLUEPRINT FOR A STRUCTURED PROCESS IN EARLY DFSS

To overcome the inconsistencies between current DFSS methods, we advocate that standard methods of creating system performance models from customer needs be implemented during the Identify, Design and Optimize stages. As part of the approach, the knowledge gained from DFSS projects must be stored independent of user and form. Form independence allows engineering design knowledge to be transferred between designs with similar functionality. User independence limits the ambiguity of system functional descriptions and allows design knowledge to be interpretable by future engineers.

A standard method of translating customer needs to system performance modeling is developed using functional modeling techniques. The Functional Basis is used to represent product or sub-system functionality (Hirtz et al., 2002). A standard method, based on the module heuristics, is also developed for partitioning systems and analyzing the flowdown of requirements between the customer, system and sub-system level. The function-based NIST-inspired Design Repository (Bohm et al., 2003), function-based system performance models and the Elemental Function-Failure Design Method (EFDM) (Stock et al., 2003) round out the techniques employed to create a powerful, integrated tool suite for execution of the IDOV method.

Our proposed, structured approach to DFSS is shown schematically in Fig. 1. In the Identify stage, we incorporate the steps of creating a functional model and applying the module heuristics to *identify* aspects of the desired system of interest based on the Voice of the Customer. Combined, these two steps establish the system architecture. As a complement to other concept generation methods, the Design Repository is then used to generate alternative design concepts by matching suitable design artifacts to the functions in the functional model. Similarly, the EFDM is applied in the Identify phase to identify potential product failures based solely on the system functions specified in the functional model. Additionally, this final step may drive changes to the system's functionality or to its embodiment through design artifacts that eliminate failure modes or to reduce their frequency and severity.



**Figure 1 - A Structured Process for the Early Stages of DFSS.**

In the Design stage, the function-based system performance models are derived and requirements flowdown are applied to the identified sub-systems. The functional model established during the Identify phase is used as the basis for creating these system performance models. For each function in the functional model, a transfer function is created. These transfer functions are then assembled according to the flows in the functional model. The result is a function-based system performance model. Module heuristics are then used to identify sub-system boundaries. The system's performance models can then be partitioned along these boundaries to investigate the flow of requirements between the system and sub-system levels. The completion of these two steps achieves system integration and the specification of design parameters. Alternatively, these two steps may be carried out during the Identify phase as an aid in design concept selection.

During the Optimize phase, the function-based system performance models can be used to investigate the performance of a system. The transfer functions and modules developed during the Design stage serve as the basis for adjusting the performance of the system. Parameters within the system models of individual functions or sub-systems can be adjusted to obtain the required level of performance. Using a function-based representation for the system allows knowledge gained from the Optimize stage to be applied to the design of future systems with similar functionality.

The performance models for each function, each module and the system as a whole are validated in the Verify stage. The function and module boundaries provide a clear framework for verifying the performance of a system. Rather than scrapping an entire model after a series of unsuccessful tests, errors can be traced to the performance models of individual functions and/or modules. These models can then be refined until the system model accurately reflects the actual performance of the system. The knowledge gained from the validation of each function and module should then be stored in the engineering knowledge base. This insures that the lessons learned from a design can be applied to future designs that feature similar functions or modules.

The structured DFSS approach shown in Figure 1 supports the systematic application of function-based design techniques to DFSS in order to overcome existing challenges in functional representation, the construction of system performance models, knowledge reuse, and - perhaps the most challenging of all - generating initial concepts that will be robust and have six sigma type performance. In this paper, we will focus on the specific step of generating system performance models from functional models. Developing illustrative examples for each of the steps is beyond the scope of this paper and is left as future work.

### 3 PARKING BRAKE EXAMPLE

To explore the applicability and impact of structured functional modeling using the Functional Basis and the module heuristics on DFSS, the methods are applied to a hand operated parking brake system. A DFSS group at General Motors (GM) had conducted a DFSS project on parking brake systems to develop a generalized analytical method for designing future parking brake systems with similar functionality; thus significant detail about both the braking system and the DFSS methods used were available.

#### 3.1 SYSTEM PERFORMANCE MODEL GENERATION

The system performance model of the parking brake system was created based on the functionality of the system. To develop the performance model, a functional model was first created. Transfer functions were then associated with each function in the functional model. These transfer functions were then combined according to the flows in the functional model to produce the system performance model.

##### 3.1.1 DEVELOPMENT OF THE FUNCTIONAL MODEL

The first step in creating the functional model for the parking brake system was to identify the critical customer needs. In the DFSS project, the handle force, travel, and the slope the car was capable of holding were identified as the primary customer needs. The relationship between these needs represents the performance of the system and was the basis for creating the functional model. The next step was to

map these needs to the input and output flows of the system. The handle force and travel were identified as input flows. The moment generated at the rear wheels from the vehicle being on an incline also acted as an input. To create a complete functional model, all the input and output flows from a system must be identified. Lists of the input and output flows for the parking brake system appear in Tables 1 and 2 respectively. These flows are represented using Functional Basis flow terms.

Once the input and output flows of the system have been identified, a black-box model of the system is generated. This model contains the input and output flows along with the overall function of the system represented in the Functional Basis language. For the parking brake system, the black box function is stop rotational energy. The boundary of the system is the interface between the rear wheels and the parking brake drum. The moment generated at the wheels by the vehicle resting on an incline or attempting to move is transmitted to the system from the parking brake drum. With this boundary, the wheel is an input to the system rather than a component in the system.

**Table 1 – Parking Brake System Input Flows**

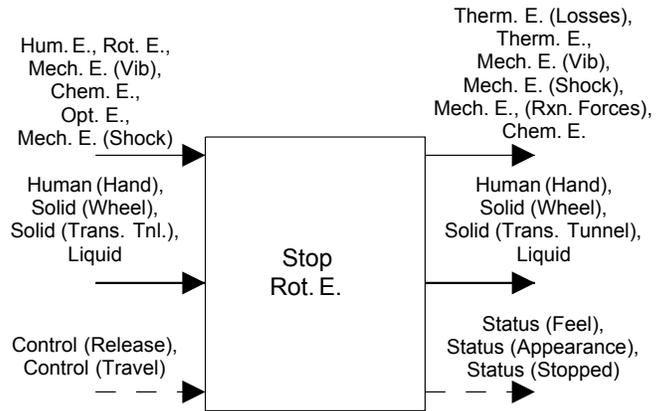
Input from DFSS Project	Type	Input translated to the Functional Basis
Handle force	Energy	Human Energy
Moment from rear wheels	Energy	Rotational Energy
Vibrations	Energy	Mechanical Energy (Vibration)
Corrosives	Energy	Chemical Energy
Light	Energy	Optical Energy
Shock forces	Energy	Mechanical Energy (Shock)
Driver's hand	Material	Human (Hand)
Rear wheels	Material	Solid (Wheel)
Transmission tunnel	Material	Solid (Transmission Tunnel)
Spills	Material	Liquid
Release	Signal	Control (Release)
Driver's hand	Signal	Status (Travel)

**Table 2 – Parking Brake System Output Flows**

Output from DFSS Project	Type	Output translated to the Functional Basis
Force and Travel losses	Energy	Thermal Energy (Losses)
Heat	Energy	Thermal Energy
Vibrations	Energy	Mechanical Energy (Vibration)
Shock forces	Energy	Mechanical Energy (Shock)
Reaction forces	Energy	Mechanical Energy (Reaction Forces)
Rear wheels	Material	Solid (Wheel)
Corrosives	Material	Chemical Energy
Driver's hand	Material	Human (Hand)
Transmission tunnel	Material	Solid (Transmission Tunnel)
Spills	Material	Liquid
Feel	Signal	Status (Feel)
Appearance	Signal	Status (Appearance)
Stopped Vehicle	Signal	Status (Stopped)

The primary performance output of the system is a status signal that indicates if the vehicle is stopped or not.

This is a binary signal that is true if the stopping moment generated by the parking brake system is greater than the moment created from the vehicle being on an incline. It is false if the incline moment is greater than the stopping moment. The overall function of the system and the flows were combined to produce the black-box model for the parking brake system. The resulting black-box model appears in Figure 2. Energy flows are represented using thin arrows, material flows by thick arrows and signals by dashed arrows.



**Figure 2 – Parking Brake System Black-Box Model**

Once the inputs and outputs for the system were found, the functions that must operate on a given input flow to produce the desired output flow are determined. In applying this process to the parking brake system, the first flow analyzed was the operator's hand. The hand is first imported into the system and then energy from the human operator is used to rotate the parking brake lever. Then, the hand is exported from the system. The required functional chain for this flow is import hand, import human energy, and export hand. The human energy flow from the human operator to the stopping force generated by the parking brake system is modeled similarly. The human energy is first imported into the system, this energy is then used to rotate the parking brake lever, and the rotating lever is then used to pull on the end of the front parking brake cable. The cable is attached to an equalizer that puts equal tension on both rear brake cables. The force from the rear cables is then transmitted to the brake levers at the rear corners. This force is used to press the parking brake shoe onto the drum. This creates a frictional force that resists rotation of the rear wheels. The resultant functional flow chain is import human energy, convert human energy to rotational energy, convert rotational energy to translational energy, guide translational energy, distribute translational energy, guide translational energy, change translational energy, convert translational energy to rotational energy, convert rotational energy to mechanical energy, and stop rotational energy.

Other functions and flows that have less impact on the primary performance of the system were included to produce the complete functional model. These functions and flows include the transmission of reaction forces, resisting corrosion, preventing liquids from penetrating the system, preventing shock loads from damaging the system, resisting damage from light, damping vibrations, and signaling the appearance of the system. The reaction forces are needed to

find the stresses in components and at system boundaries. Corrosion degrades the performance of the system over time by increasing losses and increasing the possibility of a failure. Liquid flows can lead to corrosion and electrical shorting problems. Shock loads from installation and accidental impacts must also be handled by the system. To reduce the complexity of the performance models, these flows were not included in the modeling of the system. When actually designing a system, all input and output flows must be investigated and modeled. The functional chains for the parking brake system appear in Figure 3.

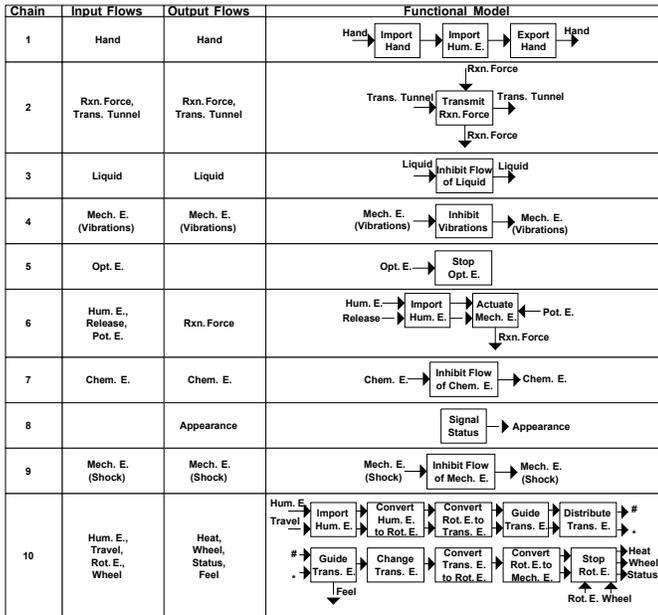


Figure 3– Parking Brake System Functional Chains

The next step was to aggregate the functional chains into a complete functional model. This is accomplished by joining chains with common flows and/or functions. Some chains appear in multiple places in the model. For instance, the import hand, import human energy and export hand chain appears twice, once to engage the system and once to release it. Some functions do not directly interface with primary flows but are necessary to include for completeness of the model. These functions include the corrosion resistance, fade resistance, and resistance to shock loads. The complete functional model for the system appears in Figure 4.

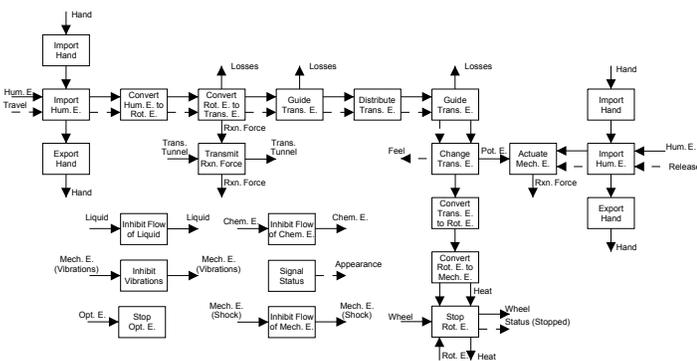


Figure 4 – Parking Brake System Functional Model

To create the system performance model, each function will be replaced by a transfer function. These transfer functions will be chained together to produce the overall system model. To determine the transfer function for each function, knowledge of the physical component that solves the function is required. With the functional model completed, the components that satisfy each function are mapped to the functional model. The import human energy function is satisfied by the handle of the parking brake lever. The lever itself turns the human energy into translational energy, which is amplified by the quadrant (or sector). The quadrant places tension on the front cable, which guides the translational energy to the equalizer. The equalizer performs the distribute translational energy function by transmitting equal force to the rear cables. Each rear cable guides the translational energy to the brake levers. These brake levers convert the translational energy from the cables into rotational energy. This rotational energy is then converted into a frictional force (mechanical energy) by the brake shoe. The frictional force from the brake shoe then acts on the drum to prevent it from rotating. The component model appears in Figure 5.

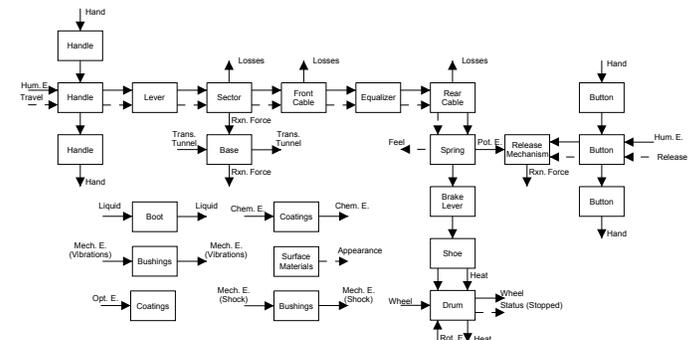


Figure 5 – Parking Brake System Component Model

### 3.1.2 PARKING BRAKE EXAMPLE – TRANSFER FUNCTION DEVELOPMENT

After the functional model and component correlation for a system are found, transfer functions are developed. For each function-component combination, a set of transfer functions exists that relates the inputs to the outputs. These transfer functions can take on many forms including closed-form equations, bond graphs, finite element models and statistical models generated from experiments. Any mathematical relationship between the output of a function and its input can be used. In this example, closed-form equations are used. This example was chosen because the results of an existing DFSS project on the parking brake system were available. Many other examples have been completed with varying degrees of modeling complexity. To create the transfer functions for the parking brake system, the input and output states for each function must be defined. For example, the distribute translational energy function of the parking brake system has two input flows and two output flows. From the translational energy input, the force on the front parking brake cable is identified as an input state. The corresponding output state is the force on each of the rear parking brake cables. The travel of the front cable is identified as an input state from the input of the travel signal flow. Correspondingly, the travel of each rear cable is

identified as an output state from the output of the travel signal flow. Once these states have been found for every function, a mathematical model is then created that relates each function's output states to its input states.

In the parking brake system both the forces in the systems and the travel of the handle are important to satisfying customer needs, two parallel flows of transfer functions need to be expressed to capture the performance of the system. Table 3 shows the Function-Component-Equation correlation for the forces. The Function-Component-Equation relationship for the travel of the parking brake system was similarly modeled but the results are not included in this paper for reasons of brevity.

**Table 3 – Parking Brake System Force Function-Component-Equation Correlation**

Function	Component	Equation
Import Human Energy	Handle	$F_{HO}=F_{HI}$
Convert Hum. E. to Rot. E.	Lever	$M_{HO}=F_{HO} \cdot L_{HI} \cdot \xi_{LVR}$
Convert Rot. E. to Trans. E.	Sector	$F_{FCI}=M_{HO}/L_{HO}$
Guide Trans. E.	Front Cable	$F_{FCO}=F_{FCI} \cdot \xi_{FC}$
Distribute Trans. E.	Equalizer	$F_{RCI}=F_{FCO} \cdot \xi_{EQ}/2$
Guide Trans. E.	Rear Cable	$F_{RCO}=F_{RCI} \cdot \xi_{RC}$
Change Trans. E.	Spring	$F_{RBL}=F_{RCO} \cdot K_S \cdot \delta_S - F_{PRE}$
Convert Trans. E. to Rot. E. / Convert Rot. E. to Mech. E.	Brake Lever/ Shoe	$M_{SHOE}=F_{RBL} \cdot L_{PBL} \cdot X_{PBM}$
Stop Rotational Energy	Drum	$M_{HAT}=M_{SHOE} \cdot \mu_{SHOE} \cdot R_{HAT} / L_{SHOE}$
Rotational Energy	Wheel	$M_W=1/2 \cdot m \cdot g \cdot \sin(\theta) \cdot R_W$

When combined, these equations produce the system performance models that appear in Table 4. Equation 1 relates the input force from the human operator to the maximum moment resisted by one corner of the parking brake system. Equation 2 uses the angle of incline of the vehicle to determine the moment generated at each rear wheel. The equation for the travel of the parking brake must be solved and inserted into equation (1) to find the stopping moment.

**Table 4– Parking Brake System Performance Models**

Flow	System Performance Model
Stopping Moment	$M_{HAT} = \frac{\left( \frac{F_{HI} \cdot L_{HI} \cdot \xi_{LVR} \cdot \xi_{FC} \cdot \xi_{EQ} \cdot \xi_{RC}}{L_{HO} \cdot 2} - K_S \cdot \delta_S - F_{PRE} \right) \cdot L_{PBL} \cdot X_{PBM} \cdot \mu_{SHOE} \cdot R_{HAT}}{L_{SHOE}} \quad (1)$
Moment from Incline	$M_W = \frac{m \cdot g \cdot \sin(\theta) \cdot R_W}{2} \quad (2)$

### 3.1.3 IMPACT OF FUNCTIONAL MODELING AND COMPARISON TO EXISTING TOOLS

During many DFSS projects, a function diagram is the primary aid to developing transfer functions. Sub-systems (Cable System), components (Tire), functions (Force Conversion, Tension to Compression), and requirements (Hold Vehicle Until Released) can all appear on the same function diagram. Mixing these different factors can lead to

confusion in the modeling process as well as a very complex model if each factor – component, function, and requirements – is modeled with detail. Also, a casually constructed function diagram represents poor modeling content from a design perspective. The functionality represents what a system must do to achieve it overall black box function or functionally meet some requirement. The components are how the system achieves its functionality and the overall system requirements. To model these items simultaneously is mixing different levels of abstraction, information, and design intent.

By modeling the functionality of a system separately from its form and requirements, it is possible to describe the performance of a system more completely. In the current approach, function diagrams are used strictly as a means to generate the transfer functions used to model the system. As a result, there is no consistency between the models developed in each DFSS project. Also, because no standard methodology exists for creating the models, different engineers can create different models for the same system. By applying functional modeling techniques along with the Functional Basis it is possible to replace these function diagrams with form and user independent functional models. These functional models are then used as a framework for the entire design process in addition to assisting performance model development.

The performance models generated using the functional model can then be used during the Optimize stage of a DFSS project to evaluate the performance of system configurations. By using functional methods of describing the parking brake system, the transfer functions for each component of the system can be analyzed individually or aggregated to produce an overall model. This method of representing systems allows designers to substitute components that would not be considered using other design methodologies for components that perform the same function. As long as the function, input flows and output flows match, the components can be interchanged. The result is a flexible design method that is not platform or component specific. The parking brake functional model can be used to design similar future system or can be modified to assist in designing completely new systems.

This design framework allows more freedom in application than other current function-based design methods such as bond graph-based tools. A bond graph represents only a small subset of the allowable modeling techniques within the proposed functional framework. A functional model represented with the Functional Basis can be created for any electro-mechanical system. The transfer functions used to create performance equations for each function can be represented with any mathematical relationship.

### 3.2 PARKING BRAKE EXAMPLE – REQUIREMENTS FLOWDOWN

Once a functional model has been created for a system, it can be used to assist requirements flowdown during the Design phase of a DFSS project. Functional modeling assists requirements flowdown by providing clear lines to partition a system into sub-systems and identify flows at the boundaries. The input and output flows from the functional model can be mapped to system and sub-system requirements. The performance models generated during the initial DFSS

modeling can then be partitioned along the same lines as the functional model. By relating the input and output flows to the system with the flows at the sub-system boundaries it is possible to investigate the flow of requirements between the customer, vehicle, system and sub-system level. Creating a complete functional model for a system also makes it possible to identify all requirements for a system using the input and output flows.

### 3.2.1 PARKING BRAKE EXAMPLE – REQUIREMENTS DEVELOPMENT

To demonstrate the usefulness of functional modeling to the requirements flowdown process, the parking brake functional model was used to generate a list of requirements. The first step was to determine the overall requirements based on the black-box model. Next, the system was partitioned into sub-systems across lines of functionality. The sub-systems were then isolated and the requirements were found at their boundaries. Transfer functions were then developed for each sub-system. These transfer functions were then used to show how higher-level requirements such as the slope on which the vehicle can remain stationary can be mapped to lower-level requirements such as the tension in the front parking brake cable.

The black-box model was used to find the overall system requirements. Each input and output flow of the system represents one or more of these requirements. The human energy input to the system can be split into two requirements, the force required to move the handle and the force required to release the brake. The maximum slope on which the vehicle can rest was derived from the rotational energy input to the system (the moment generated by the rear wheels from the vehicle being on an incline). The chemical energy flow into the system contains the corrosion resistance requirements for the system. Material inputs to the system determine the interface requirements. The parking brake handle must be comfortable to the driver; this requirement is contained within the hand input to the system. The transmission tunnel and wheel inputs to the system contain the requirements for the physical interfaces.

By using functional methods to identify requirements, it is possible to capture all requirements of a system before it has been completely designed. This prevents costly oversights in detecting requirements. For example, the optical energy flow into the system necessitates a requirement to prevent fading of the parking brake handle. Such a requirement does not exist in the current list of sub-system technical specifications. If this requirement was overlooked by a supplier, the parking brake system that was supposed to be designed for best in class quality would eventually have a handle that is a different color than the rest of the interior. Identifying requirements based on a standard set of flows prevents such oversights. If all the flows into and out from a system are identified, all requirements can be captured. The Functional Basis is the key to identifying these flows. An engineer can quickly look through the Basis terms to find requirements that haven't been identified

### 3.2.2 PARKING BRAKE EXAMPLE – MODULE HEURISTICS

Module heuristics were applied to the parking brake system in order to determine sub-system boundaries and requirements. The first step in this process was to look for dominant flows in the functional model. For example, the hand material flow travels through the system without being converted to a different flow or branched, this represents a dominant flow. This module of the parking brake system was labeled *hand module 1* and includes the import hand, import human energy and export hand functions. *Hand module 2* is a similar module and was also identified using the Dominant Flow rule. The next module identified was *convert human energy to translational energy*. This module was identified using the Conversion-Transmission rule by following the human energy flow until it was converted to a different form of energy and transmitted through the system. This module includes the convert human energy to rotational energy, convert rotational energy to translational energy and transmit reaction force functions. This module converts the input of human energy to the translational energy needed to operate the parking brake. It also transmits the human energy to the transmission tunnel as a reaction force. The *guide translational energy* module was the next to be identified. This module follows the dominant flow of translational energy from the guide translational energy function to the change translational energy function. Two more modules were then identified using the Branching Flow rule from the split in the energy flow at the change translational energy function. Some of the energy leaves as potential energy which enters the actuate mechanical energy function. This single function exists between two modules and does not belong to either, as a result, it becomes a single-function module identified by the branching energy flow. This module was labeled *actuate mechanical energy*. The remaining energy from the change translational energy function enters the convert translational energy to rotational energy function. This represents the beginning of a conversion-transmission module. The flow of energy was followed until its termination at the stop rotational energy function to identify the *convert translational energy to mechanical energy module*.

The module boundaries appear in the functional model shown in Figure 6.

Once the modules have been identified, they can be used to investigate the flowdown of requirements between the system and sub-system (module) levels. The transfer functions that were created for each function during the DFSS system performance modeling can be assembled according to the modular boundaries to produce sub-system models. By using these sub-system models, it is possible to determine the requirements at the boundaries of the modules. For example, the human energy input to the system can be related to the translational energy going into the *guide translational energy* module via the sub-system model for the *convert human energy to translational energy* module. The result of this technique is a standard method of defining sub-system boundaries and identifying the requirements at the boundaries of these sub-systems.

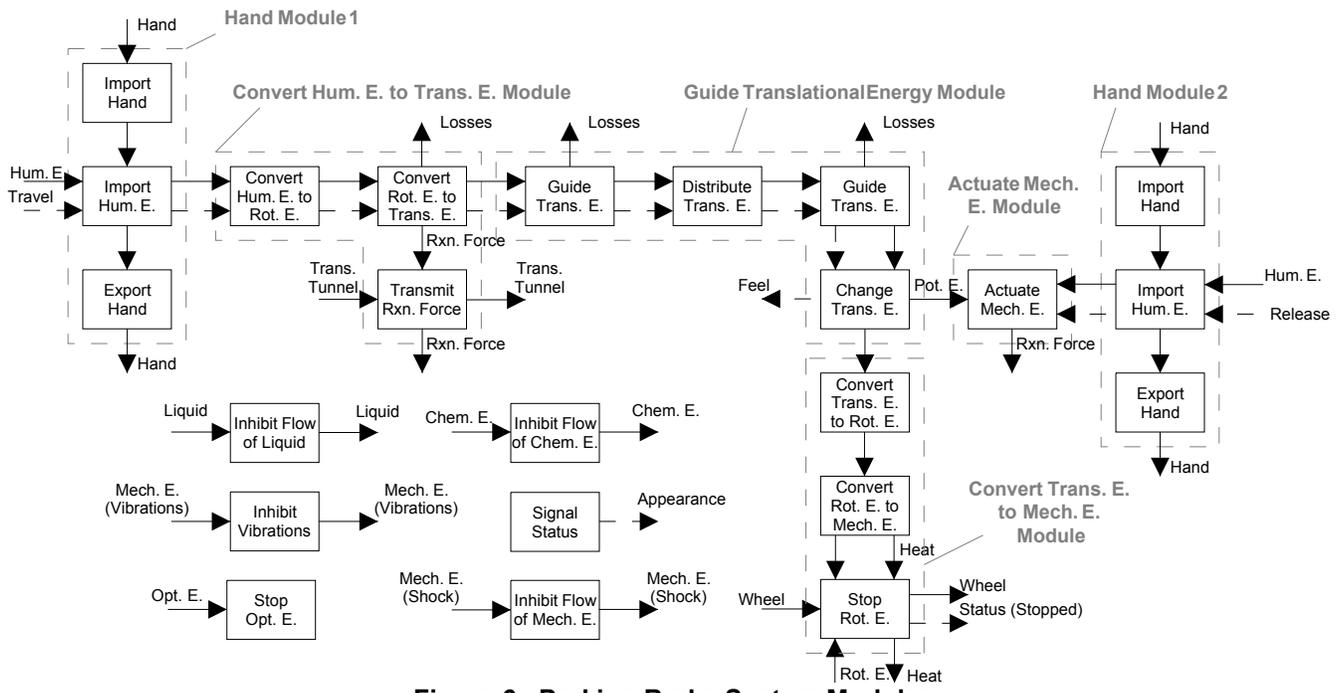


Figure 6– Parking Brake System Modules

Function	Component	Equation
Import Human Energy	Handle	$F_{HO} = F_{HI}$
Convert Hum. E. to Rot. E.	Lever	$M_{HO} = F_{HO} \cdot L_{HI} \cdot \xi_{LVR}$
Convert Rot. E. to Trans. E.	Sector	$F_{FCI} = M_{HO} / L_{HO}$
Guide Trans. E.	Front Cable	$F_{FCO} = F_{FCI} \cdot \xi_{FC}$
Distribute Trans. E.	Equalizer	$F_{RCI} = F_{FCO} \cdot \xi_{EQ} / 2$
Guide Trans. E.	Rear Cable	$F_{RCO} = F_{RCI} \cdot \xi_{RC}$
Change Trans. E.	Spring	$F_{RBL} = F_{RCO} \cdot K_S \cdot \delta_S - F_{PRE}$
Convert Trans. E. to Rot. E./ Convert Rot. E. to Mech. E.	Brake Lever/Shoe	$M_{SHOE} = F_{RBL} \cdot L_{PBL} \cdot X_{PBM}$
Stop Rotational Energy	Drum	$M_{HAT} = M_{SHOE} \cdot \mu_{SHOE} \cdot R_{HAT} / L_{SHOE}$

**Hand Module 1**

$$F_{HO} = F_{HI}$$

**Convert Hum. E. to Trans. E. Module**

$$F_{FCI} = \frac{F_{HO} \cdot L_{HI} \cdot \xi_{LVR}}{L_{HO}}$$

**Guide Trans. E. Module**

$$F_{RBL} = \frac{F_{FCI} \cdot \xi_{RC} \cdot \xi_{EQ} \cdot \xi_{FC}}{2} - K_S \cdot \delta_S - F_{PRE}$$

**Convert Trans. E. to Mech. E. Module**

$$M_{HAT} = \frac{F_{RBL} \cdot L_{PBL} \cdot X_{PBM} \cdot \mu_{SHOE} \cdot R_{HAT}}{L_{SHOE}}$$

Figure 7 – Sub-system Performance Models

**3.2.3 PARKING BRAKE EXAMPLE – SUB-SYSTEM MODEL DEVELOPMENT**

The next step was to develop transfer functions for the sub-systems. Since the release mechanism was not included in the DFSS analysis, it was excluded. To develop the sub-system transfer functions, the transfer functions for each function in the assembly are chained together. The transfer functions from the DFSS example were used to create the sub-system models. These models appear in Figure 7.

The *hand module 1* sub-system model relates the force going into the *convert human energy to translational energy module* to the force on the parking brake handle. The *convert human energy to translational energy module* sub-system model converts this force to the tension entering the *guide translational energy module*. The other inputs to this

model are the lengths of the handle and sector and force efficiency of the handle. The *guide translational energy module* sub-system model relates the force going into the *convert translational energy to mechanical energy module* to the input tension on the cable. This model contains the force efficiencies of the front and rear cables, the spring constant of the return spring, the displacement of the rear brake lever (which must be found from the travel model) and the preload of the return spring. The *convert translational energy to mechanical energy module* model relates the force on the rear parking brake lever to the maximum moment that can be resisted by each rear corner. This model contains the length of the rear parking brake lever, the radius of the brake drum and the properties of the brake shoe. The moment generated by the *convert translational energy to mechanical energy module* must be greater than the incoming rotational energy.

By using these system and sub-system models, it is possible to relate vehicle, customer, system and sub-system requirements. For example, the vehicle-level requirement Mass is an input to the parking brake system model and the *convert translational energy to mechanical energy module* sub-system model. The customer requirement Slope is also an input to the *convert translational energy to mechanical energy module* sub-system model. The Force Efficiency requirement for the system is broken down into individual sub-system Force Efficiency requirements.

Understanding the flow of requirements through a system is crucial to the Identify, Design and Optimize phases of DFSS. During the Identify phase, module identification is necessary to determine which sub-systems (if not all) of a product must be changed. Once these modules have been identified, they can be used to perform requirements flowdown in the Design phase to determine what parameters can be changed in the sub-systems to affect the overall system performance. During the Optimize phase, the modules and requirements flowdown assist in determining how much these parameters can be changed in order to obtain the desired level of performance.

#### 4 CONCLUSIONS AND NEXT STEPS

To truly *design* for six sigma, no more than 3.4 out of every one million products can fail to satisfy a customer. However most DFSS projects simply focus on improved quality rather than a set goal of six standards of deviation. Regardless of the quality targets, the high levels of quality required during a DFSS project clearly presents many challenges. To achieve this level of success, methods need to be extended to the earliest stages of design. DFSS methods must be customer focused, produce concepts that in principle can satisfy up to 99.99966% of customers and execute these concepts successfully. Integrating functional modeling into a DFSS effort can have a significant impact on connecting both concepts and models to customer needs thus keeping the design effort focused on satisfying 99.99966% of customers rather than being limited to focusing on having 99.99966% of the products produced operate as intended. Thinking in terms of customer satisfaction rather than product failure is a subtle but important difference. Design for Six Sigma needs to include all aspects of design. Functional modeling and the Functional Basis make a fundamental contribution to DFSS through allowing structured methods to be employed at the earliest stages of design as well as providing a continuous connection from concept generation to requirements flowdown, to parametric design and optimization, to knowledge reuse.

Several shortcomings have been identified in current DFSS practices. No standard framework exists for creating six sigma designs. This lack of structure hinders communications within DFSS teams and the storage of design knowledge. In addition, during many DFSS design efforts, function and form are modeled together. This inhibits creative design and complicates transfer function development. We propose that functional modeling can overcome these problems by providing a standard function-based framework. Once a functional model has been created for a system, it can be used as a framework for the entire DFSS process. During the Identify phase, a functional model

should be created. This modeling process directly relates the functionality of the product to the customer's needs. Once this model has been created, it can be used during the Design phase to create transfer functions, partition the system and investigate requirements flowdown. The transfer functions and modules developed during Design can then be used in the Optimize stage to investigate the performance of system configurations. During the Verify stage, the partitions generated using the functional model can be used to validate the performance of a system's modules. The parking brake example demonstrates the applicability of functional modeling to DFSS. The complete process for conducting a functional modeling based DFSS project appears below.

##### 1. Identify

- Create the functional model
- Apply module heuristics
- Develop concepts using the design repository
- Identify potential failure modes with EFDM

##### 2. Design

- Create function-based performance models
- Perform requirements flowdown

##### 3. Optimize

- Use system models to investigate performance
- Adjust function and/or module parameters to obtain desired performance

##### 4. Verify

- Evaluate the performance of the overall system as well as each function and module
- Update the design repository with the experimental results

To support the application of functional modeling during systems engineering, function-based design repositories are also being explored. Such a repository would provide a method of representing systems, sub-systems and components based on functionality. This is a key enabler for such engineering design practices as component selection during product development cost management processes and technical memory. The uses of functional modeling techniques in these areas of application and other areas of system engineering will be explored next. In addition, function-based methods of extracting, storing and applying sensitivity information are being investigated. The function-based system performance model generation technique outlined in this paper has been applied to several systems with varying levels of modeling complexity. These models have included non-linear equation systems, mechatronic systems, systems with nominal output values as well as systems with nominal output ranges.

Methods of identifying and preventing failure modes based on functionality rather than experience are also being researched. The Elemental Function-Failure Design Method

allows failures to be predicted and prevented using a functional model. Also, it provides a standard vocabulary for representing such failures.

## REFERENCES

- (Value Analysis Incorporated), V. A. I. (1993), Value Analysis, Value Engineering, and Value Management, Clifton Park, New York.
- Akiyama, K. (1991), Function Analysis: Systematic Improvement of Quality Performance, Productivity Press.
- Altshuller, G. (1999), The Innovation Algorithm: TRIZ, Systematic Innovation and Technical Creativity, Technical Innovation Center.
- Amoussou, G., Vicarini, M., Rohmer, S., and Barros, L. (1997), "Application of TROPOS Functional Model to a Maintenance System of a Nuclear Plant," Proceedings of the Fifth International Workshop on Functional Modeling of Complex Technical Systems, Paris-Troyes, France.
- Bohm, M., Stone, R. and Szykman, S. (2003), "Enhancing Virtual Product Representations for Advanced Design Repository Systems," Proceedings of DETC2003, DETC2003/CIE-48239, Chicago, IL.
- Bracewell, R. and Sharp, J. (1996), "Functional Descriptions Used in Computer Support for Qualitative Scheme Generation – 'Schemebuilder'," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10(4), pp. 333-345.
- Brue, G., Launsby, R. (2003), Design for Six Sigma, New York, McGraw-Hill.
- Chowdhury, S. (2002), Design for Six Sigma, Chicago, Dearborn Trade.
- Collins, J. A., Hagan, B. T., and Bratt, H. M. (1976), "The Failure-Experience Matrix: A Useful Design Tool," Journal of Engineering for Industry, 98, pp. 1074-1079.
- Creveling, C., Slutsky, J. and Antis, D. (2003), Design for Six Sigma in Technology and Product Development, New Jersey, Prentice-Hall.
- Hirtz, J., Stone, R., McAdams, D., Szykman, S. and Wood, K. (2002), "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Research in Engineering Design, 13(2): 65-82.
- Hubka, V. and Ernst Eder, W. (1984), Theory of Technical Systems, Springer-Verlag, Berlin.
- Hundal, M. (1990), "A Systematic Method for Developing Function Structures, Solutions and Concept Variants," Mechanism and Machine Theory, 25, pp. 243-256.
- Iwasaki, Y., Vescovi, M., Fikes, R., and Chandrasekaran, B. (1995), "Casual Functional Representation Language with Behavior-Based Semantics," Applied Artificial Intelligence, 9.
- Lai, K. and Wilson, W. (1989), "FDL - A Language for Function Description and Rationalization in Mechanical Design," Journal of Mechanics, Transmissions, and Automation in Design, 111, pp. 117-123.
- Miles, L. (1972), Techniques of Value Analysis Engineering, McGraw-Hill, New York.
- Modarres, M. (1997), "A Functional Classification Based on Conservation Principles," Proceedings of the Fifth International Workshop on Functional Modeling of Complex Technical Systems, Paris-Troyes.
- Murdock, J., Szykman, S., and Sriram, R. (1997), "An Information Modeling Framework to Support Design Databases and Repositories," Proceedings of DETC'97, DETC97/DFM-4373, Sacramento, CA.
- Otto, K. and Wood, K. (2001), Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development, New York, Prentice-Hall.
- Stock, M., Stone, R. and Tumer, I. (2003), "Going Back in Time to Improve Design: The Elemental Function-Failure Design Method," Proceedings of DETC2003, DETC2003/DTM-48638, Chicago, IL.
- Stone, R. and Wood, K. (2000), "Development of a Functional Basis for Design," Journal of Mechanical Design, 122(4):359-370.
- Stone, R., Wood, K. and Crawford, R. (2000), "A Heuristic Method for Identifying Modules for Product Architectures," Design Studies, 21(1): 5-31.
- Szykman, S., Racz, J., and Sriram, R. (1999), "The Representation of Function in Computer-Based Design," Proceedings of the ASME Design Theory and Methodology Conference, DETC99/DTM-8742, Las Vegas, NV.
- Ullman, D. (2002), *The Mechanical Design Process 3rd Edition*, McGraw-Hill, Inc., New York, NY.
- Ulrich, K. and Eppinger, S. (1995), *Product Design and Development*, McGraw-Hill, New York, NY.
- Umeda, Y. and Tomiyama, T. (1997), "Functional Reasoning in Design," IEEE Expert Intelligent Systems and Their Applications, 12, pp. 42-48.
- Vicarini, M. (1995), "Mesurer pour simplifier," L'Informatique Professionnelle, Paris, France.