FUNCTION-BASED DESIGN TOOLS TO STREAMLINE DESIGN FOR SIX SIGMA EXECUTION AND ADDRESS UNCERTAINTY DURING EARLY ENGINEERING DESIGN

by

RYAN SCOTT HUTCHESON

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

2005

Approved by

PUBLICATION THESIS OPTION

This thesis consists of the following two articles that have been submitted for publication as follows:

Paper 1 has been published in the Proceedings of the ASME International Mechanical Engineering Congress and RD&D Expo, IMECE2005/62312, Anaheim, CA.

Paper 2 is intended for submission to the Proceedings of IDETC/CIE 2005 ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2005, Long Beach, CA.

ABSTRACT

The objective of the research presented in these two papers was to develop a set of function-based tools to aide the early design of engineering systems. The first paper proposes a function-based framework to streamline and augment the application of Design for Six Sigma to the design of systems. In this paper, 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 functional modeling techniques used in this example provide a standard method of capturing current engineering design knowledge while allowing additional knowledge to be discovered. The second paper introduces the FUNdesign concept and outlines the sensitivity measures and knowledge storage process required to implement such a concept. FUNdesign, a function-based method for addressing uncertainty during early engineering design, represents a set of tools for obtaining and storing sensitivity information of functions from previous designs as well as tools for applying this information to designing new systems. To store the sensitivity information, functional models created using the Functional Basis are first created for previous designs. Sensitivity information and associated performance models are then stored according to functionality in an engineering design repository. This information is then used to aide design and modeling resources during the design of a system with similar functionality. Together, these papers outline a set of tools that increase the knowledge available during the conceptual design of a system. These tools also illustrate the need for designers to store the knowledge gained throughout the engineering design process.

ACKNOWLEDGMENTS

I would like to thank the various groups of people who made this research possible. Foremost, my advisors at UMR, Drs. Daniel McAdams and Robert Stone. Over the last two years they have provided support, knowledge and insight during my research efforts. Additionally, they have presented me with multiple opportunities to advance my academic career through internships with corporate research facilities.

As sponsor of the research, I would also like to thank the men and women at General Motors who supported and advised me throughout my research. Assistance from Joseph Donndelinger, Bob Lust and several others in GM R&D and product development was invaluable during the course of my research.

I would also like to thank the researchers who laid the foundation for the work presented in these papers. Those who pioneered functional modeling, the Functional Basis, module heuristics and Design for Six Sigma all deserve credit for making this research possible.

Finally, I would like to thank my family for being patient and understanding with their only engineer.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

PAPER I

PAPER II

LIST OF TABLES

PAPER I

PAPER II

PAPER I

Proceedings of 2004 IMECE:
2004 ASME International Mechanical Engineering Congress and RD&D Expo
November 13-19, 2004 Anaheim, California

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
δ_{FCI}	Displacement of front cable in
δ_{FCO}	Displacement of front cable out
δ_{RCI}	Displacement of rear cable in
δ_{RCO}	Displacement of rear cable out
δ_{S}	Spring deflection
ε_{EQ}	Travel efficiency of equalizer
ε_{FC}	Travel efficiency of front cable
ε_{LVR}	Travel efficiency of lever
ε_{RC}	Travel efficiency of rear cable
μ_{SHOE}	Static coefficient of friction between hat and shoe
θ	Angle of slope
θ_{HI}	Input angle of parking brake handle
θ_{HO}	Output angle of parking brake handle
ξ_{EQ}	Force efficiency of equalizer
ξ_{FC}	Force efficiency of front cable
ξ_{LVR}	Force efficiency of lever
ξ_{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^2DOV (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:

- **I**dentify: Select the best design concept based on the Voice of the Customer
- **D**esign: Build a thorough base of knowledge about the design
- **O**ptimize: Achieve a balance of quality, cost, and time to market
- **V**erify: 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):

- **D**efine: Determine project goals and customer requirements
- **M**easure: Quantify customer needs and specifications
- **A**nalyze: Explore methods of meeting customer requirements
- **D**esign: Develop a process to meet customer requirements
- **V**erify: Insure that the design has met customer requirements

The I^{2}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 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.

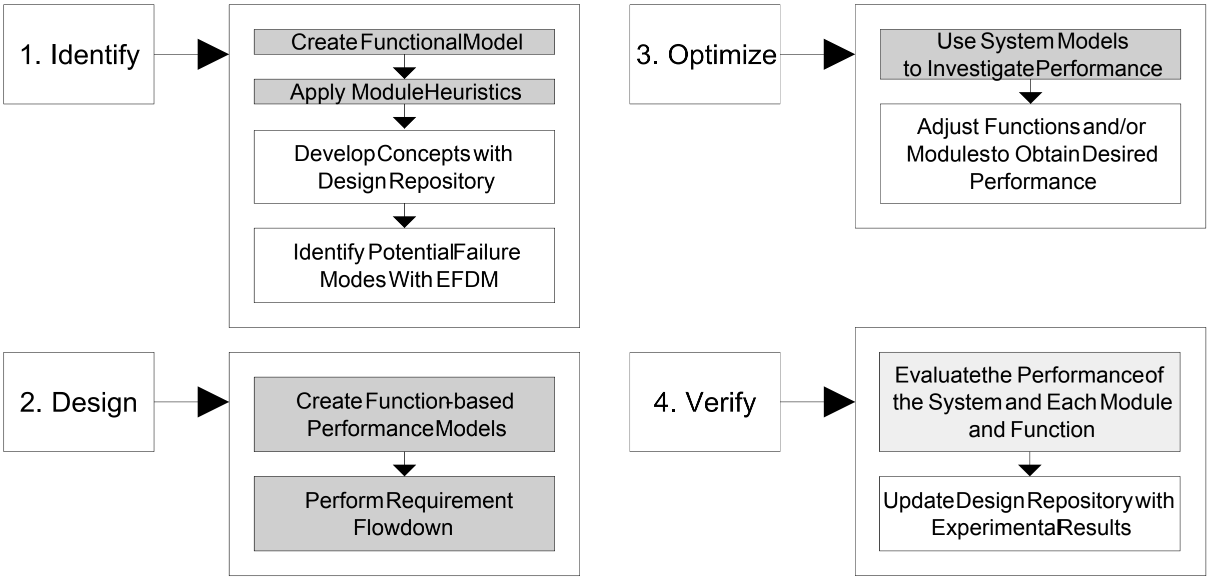


Figure 1 - A Structured Process for the Early Stages of DFSS.

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.

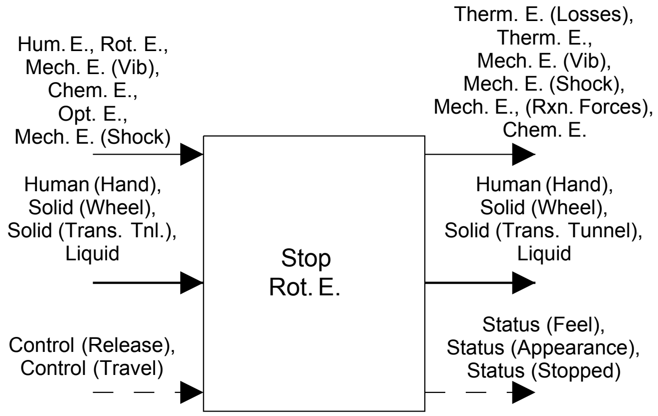


Figure 2 – Parking Brake System Black-Box Model

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.

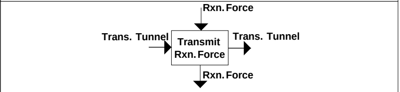
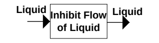
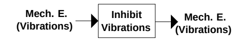
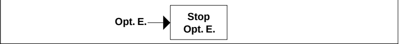
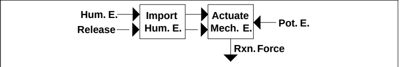
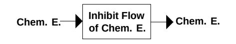
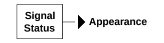
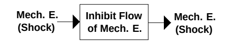
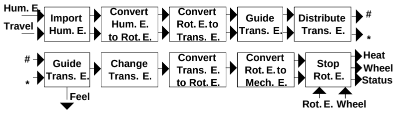
Chain	Input Flows	Output Flows	Functional Model
1	Hand	Hand	
2	Rxn. Force, Trans. Tunnel	Rxn. Force, Trans. Tunnel	
3	Liquid	Liquid	
4	Mech. E. (Vibrations)	Mech. E. (Vibrations)	
5	Opt. E.		
6	Hum. E., Release, Pot. E.	Rxn. Force	
7	Chem. E.	Chem. E.	
8		Appearance	
9	Mech. E. (Shock)	Mech. E. (Shock)	
10	Hum. E., Travel, Rot. E., Wheel	Heat, Wheel, Status, Feel	

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.

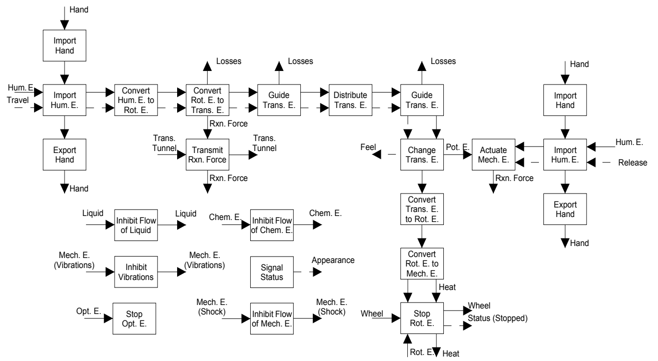


Figure 4 – Parking Brake System Functional Model

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.

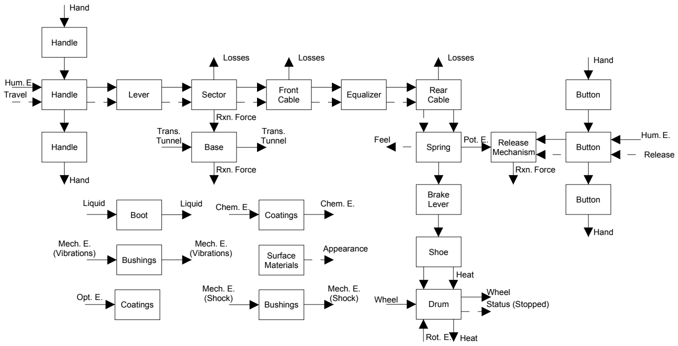


Figure 5 – Parking Brake System Component Model

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

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

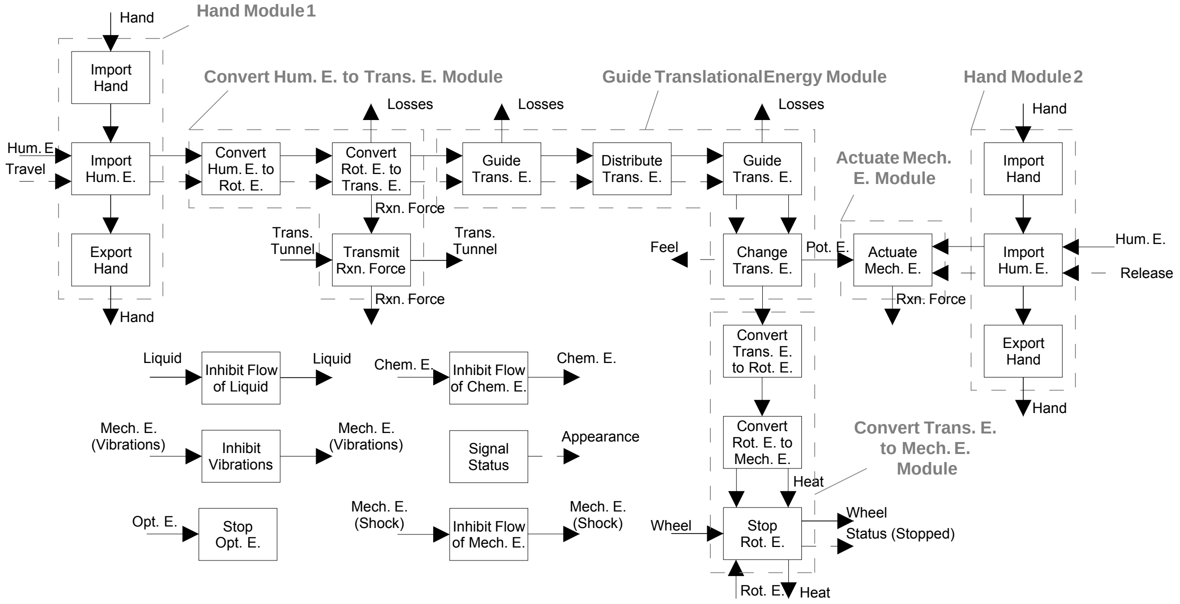
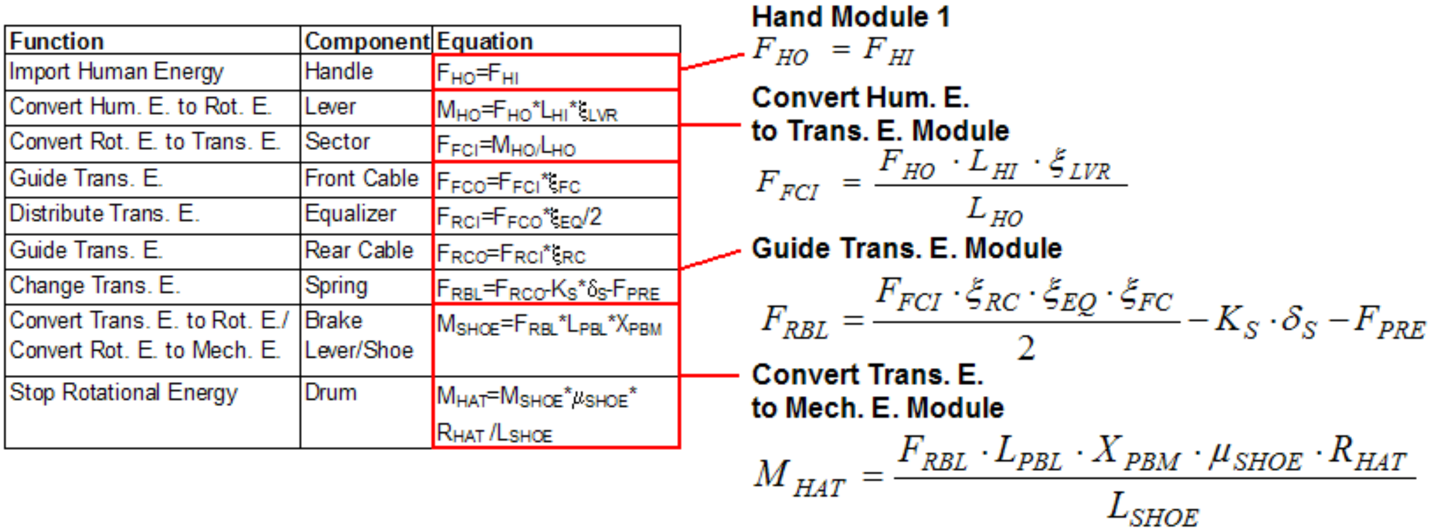


Figure 6– Parking Brake System Modules



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

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, McGaw-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.

Paper II

Submitted to Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California

IDETC/DTM

FUNDESIGN – A FUNCTION-BASED METHOD FOR ADDRESSING UNCERTAINTY DURING THE DESIGN OF ENGINEERING SYSTEMS: REPRESENTATIONS AND KNOWLEDGE STORAGE

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 outlines a function-based method for addressing design parameter uncertainty during the conceptual design phase of an engineering system. The method is given the name FUNdesign and represents a set of tools for obtaining and storing sensitivity information of functions from previous designs as well as tools for applying this information to designing new systems. To store the sensitivity information, functional models created using the Functional Basis are first created for previous designs. Performance models for each function in the model are then identified and a sensitivity analysis performed on each model with respect to each design parameter. This sensitivity information and its associated performance models are then stored according to functionality in an engineering design repository. The information stored in the repository is then used to aide the allocation of design and modeling resources during the design of a system with similar functionality. The specific focus areas of this paper are the sensitivity parameters and methods required to store the sensitivity information.

INTRODUCTION

During the design of engineering systems, uncertainty in the values of the design parameters of a system can greatly affect the system's performance. In order to quantify and reduce the effect of this uncertainty during the design process, resources must be allocated to accurately identify and model the major effects of uncertainty in a system. Identifying these effects as early as the conceptual design process allows better resource allocation throughout the entire design process. However, often during the conceptual design of a system, little is known about the potential physical forms of the solution. Without this information, it is difficult to predict which particular sub-systems or components are the most affected by uncertainty in design parameter values.

The objective of this research is to develop a well-defined method for addressing the problem of identifying the areas of a system that are more susceptible to uncertainty as early as the conceptual design process. The proposed solution is a function-based method that uses sensitivity information from previous design efforts to identify major sources of sensitivity to variation in new designs. This **F**unction-based method for addressing the **UN**certainty of system parameters during conceptual **design** will be referred to as FUNdesign in the context of this paper. FUNdesign consists of the following major steps:

1) Identify sources of significant sensitivity to variation in previous designs
2) Relate this sensitivity information to the functionality of the investigated systems
3) Store this information in a design repository
4) Create a functional model of the system to be designed
5) Use the knowledge stored in the repository to allocate modeling resources during the design of a new system based on common functionality

This paper focuses on the methods required to extract and store the sensitivity knowledge for the FUNdesign process (steps 1 through 3). The research is presented in four sections. The first section outlines the enabling technologies and contains a review of existing research in this area. Section two examines the identification and application of potential sensitivity measures for the FUNdesign process. The third section is an overview of the application of the FUNdesign knowledge storage process to a human-powered flashlight example. The final section concludes the paper and presents future work.

1 ENABLING TECHNOLOGIES AND LIT. REVIEW

Three technologies are key enablers of the FUNdesign process. The first is standardized functional modeling. Creating functional models using a standard process and taxonomy allows design knowledge to be shared between designs based on common functionality. The next key technology is function-based system performance modeling. Function-based performance modeling breaks down the mathematical modeling process of a system into a series of smaller modeling tasks for each function in the system. The final enabling technology is the function-based sensitivity analysis.

1.1 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; Akiyama, 1991; Collins et al., 1976; Modarres, 1997; 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.

The FUNdesign method requires a standard list of functions and flow terms in order to capture and reuse sensitivity knowledge. To this end, 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 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, 2002). 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).

1.2 FUNCTION-BASED SYSTEM PERFORMANCE MODELING

In order to perform a function-based sensitivity analysis, performance models must be associated with each function in the system. These models are created using the function-based system performance modeling process. This process involves the following steps (Hutcheson et al., 2004):

1) Define overall performance goals for the system
2) Create a functional model for the system
3) Define input and output states for each function
4) Create a model for each function that relates its output to its input
5) Aggregate these models to create a system performance model

This process allows performance models to be created for each function in a system in addition to a model for the system as a whole. The performance models for each function can then be used to find the sensitivity of each function to design parameter uncertainty. This allows the engineering knowledge gained during sensitivity analysis to be mapped back to individual functions.

1.3 SENSITIVITY ANALYSES AND UNCERTAINTY

The final enabling technology is the function-based sensitivity analysis. In order to classify the sensitivity of a function to parametric uncertainty, a sensitivity history must be established. To generate this history, the sensitivity of functions must be identified from previous design efforts. To form a complete data set, the sensitivity of a function must be analyzed across multiple domains and well as multiple solutions within specific domains. Additionally, sensitivity parameters must be identified to codify and store the sensitive information so that it can be applied to new design efforts. The parameters must also be broadly applicable to various domains as well and model types. An attempt to identify such parameters is presented next.

2 SENSITIVITY MEASURES

The FUNdesign process requires the storage of sensitivity information associated with functionality. To accomplish this, well-defined sensitivity measures with broad application are required. The first step in identifying or developing applicable measures is to define the sources of variation. In the FUNdesign process, two sources of variation are considered, input variables and design variables. Figure 1 shows how these sensitivities fit within the function-based design framework. Variation in the input variables is a result of noisy functions upstream in the functional flow and noisy inputs to the overall system. Variation in design variables is a result of variation in the physical parameters that govern a system's performance. The nominal values for these design variables as well as their tolerances are identified during design. Both sources of variation affect the performance of the system. Uncontrollable input variation causes undesirable performance fluctuations in the output while controllable design variation allows an engineer to tune the performance output. In both cases, it is desirable for the sensitivity to be well behaved (predictable and controllable). The focus of this paper is generally on the effect of uncertainty in design parameters on the performance of a system since these parameters fall under the control of the designer.

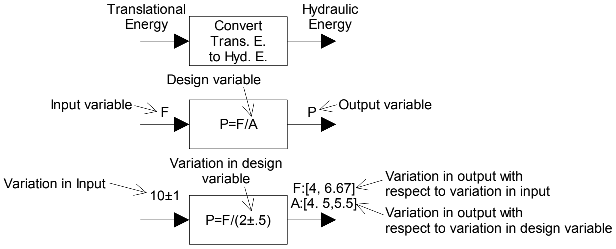


Figure 1 – Sensitivity and Functions

2.1 NOMINAL VALUE MEASURE

Typically, the procedure used to find the sensitivity of a performance output is to find the first partial derivative of the output with respect to the variable of interest at some nominal value, multiply this by the expected change in the variable of interest and then normalize the result to the value of the performance metric at the nominal value. The result is a unitless measure that expresses how sensitive a function is to a particular variable at a nominal value. The formula for a standard sensitivity analysis appears in Equation 1. An example of this equation in use appears in Figure 2.

$$\left.\frac{\partial F}{\partial x_i}\right|_{x_i,nom} \cdot \frac{\delta x}{F(x_i, nom)} \quad (1)$$

Function:	**Sensitivity (x):**	**Sensitivity (a):**
$F(x) = ax^3$	$\left.\frac{\partial F}{\partial x}\right\|_{x,nom} = 3ax^2 = 60$	$\left.\frac{\partial F}{\partial a}\right\|_{x,nom} = x^3 = 8$
Nominal Values: $a = 5 \quad \delta x = .01$ $x = 2 \quad \delta a = .03$	$\left.\frac{\partial F}{\partial x}\right\|_{x,nom} \cdot \frac{\delta x}{F(x,nom)} = .015$	$\left.\frac{\partial F}{\partial a}\right\|_{x,nom} \cdot \frac{\delta a}{F(x,nom)} = .006$

Figure 2– Nominal Value Sensitivity Analysis

2.2 RANGED MEASURES

This approach works well for systems that have nominal output values, but is insufficient for systems whose output must vary over a nominal range. An example of such a system is an automobile suspension system. The performance outputs of a suspension (camber, caster, bump steer, etc.) must be "well-behaved" over a range of input values (the travel of the suspension up and down). Using the sensitivity measure shown above, several nominal values must be selected and many sensitivity analyses must be done to capture the sensitivity information required during a concept selection process. In order to represent this information in a more concise manner, and provide a well-defined procedure for analyzing the sensitivity of systems with nominal output ranges, an average sensitivity measure is required.

To develop this sensitivity measure, several methods were investigated. The underling theory to all of these approaches was to find an average value for the sensitivity across the nominal output range and normalize this to produce a unitless single-value measure to represent the sensitivity of a function with respect to each input and design variable. The non-normalized sensitive average will be labeled as S for the remainder of this paper. Two methods are presented for normalizing the average sensitivity. The first involves dividing the average sensitivity by the range of the sensitivity. This normalized measure is labeled H and is a measure of the spread of the sensitivity. In addition, the coefficient of variation (normalizing to the standard deviation) was identified as a useful statistical measure for storing sensitivity information. This measure, labeled Cv, is found by dividing the standard deviation of the sensitivity by its average value.

The first step in the calculation of these parameters is to find the sensitivity of the performance function throughout the nominal range of the input variable. The performance model is represented as a function of two variables, i and x. The variable i represents the input to the function while the set x contains the design variables for a system. To find the sensitivity of the function, the first partial derivative of the function with respect to the variable of interest must be calculated. This can be done analytically (Equations 2 and 3) or numerically using partial differencing. To analytically find the average value of the partial derivative, an integral approach is suggested. The average value of a function can be found using Equation 4.

$$s_i = \frac{\partial f(i, x_n)}{\partial i} \ (2) \qquad s_{x_n} = \frac{\partial f(i, x_n)}{\partial x_n} \ (3)$$

$$\overline{f}(x) = \frac{\int_a^b f(x)\,dx}{b-a} \ (4)$$

For the case of the input variable, the average sensitivity reduces to the form shown in Equation 5. For design variables, this average cannot be reduced and is found using Equation 6.

$$S_i = \frac{\int_{i_0}^{i_f} \frac{\partial f(i, x_n)}{\partial i}\,di}{i_f - i_0} = \frac{f(i_f, x_n) - f(i_0, x_n)}{i_f - i_0} \ (5)$$

$$S_{x_n} = \frac{\int_{i_0}^{i_f} \frac{\partial f(i,x_n)}{\partial x_n} di}{i_f - i_0} \quad (6)$$

The value of S must then be normalized to produce a unitless measure of sensitivity over a range. Early in the research, it was chosen to normalize the sensitivity to the difference in extreme values of the first partial derivative (maximum minus minimum). The resulting measure, labeled H, related the average value of the sensitivity to the range of the sensitivity. The formulations of H for the input variable and design variables appear in Equations 7 and 8 respectively. H is a measure of linearity across a range of performance. A value of H greater than one means that the average value of the sensitivity is greater than the range of H, the opposite is true when H is less than one. H can take on values between zero and positive infinity. Close to zero, the range of the sensitivity is much greater than its average value. At very large values of H, the average value of the sensitivity is much greater than its range. This corresponds to a proportional relationship between the variable of interest and the output of the system.

$$H_i = \frac{S_i}{\left.\frac{\partial f(i,x_n)}{\partial i}\right|_{\max} - \left.\frac{\partial f(i,x_n)}{\partial i}\right|_{\min}} = \frac{f(i_f,x_n) - f(i_0,x_n)}{\left(i_f - i_0\right)\left(\left.\frac{\partial f(i,x_n)}{\partial i}\right|_{\max} - \left.\frac{\partial f(i,x_n)}{\partial i}\right|_{\min}\right)} \quad (7)$$

$$H_{x_n} = \frac{\int_{i_0}^{i_f} \frac{\partial f(i,x_n)}{\partial x_n} di}{\left(i_f - i_0\right)\left(\left.\frac{\partial f(i,x_n)}{\partial x_n}\right|_{\max} - \left.\frac{\partial f(i,x_n)}{\partial x_n}\right|_{\min}\right)} \quad (8)$$

The H sensitivity parameter was applied to several design examples including a clutch actuation system and the included human-powered flashlight example. During these examples, it was found that functions that exhibit near constant first derivatives produce values of H that vary greatly. Small changes or inaccuracies resulting from machine precision would have order of magnitude changes in the value of H. It was often difficult to make conclusions from the value of H without past experience in applying the measure. To this end, additional research was conducted to identify a suitable measure for representing ranged sensitivity information that produced a consistent interpretation of results.

The result of this research was the identification of the coefficient of variation (CV) as a useful measure for representing ranged variation. The value of CV is determined by dividing the standard deviation of the sensitivity by its mean value. The CV can be found for both discrete and continuous sensitivities using commonly used formulas. For CV values close to zero, the mean value of the function is much greater than the standard deviation. This situation occurs for functions that exhibit a proportional relationship between the output and variable of interest. Increasing values of CV represent larger variation in the sensitivity. An example of the application of CV to a ranged sensitivity problem is presented next.

2.3 EXAMPLE

To demonstrate how the S and CV parameters are calculated and used, three functions are considered: $F(i,x)=xi^3$, $F(i,x)=xi$ and $F(i,x)=xi+x*\sin(2*pi*i)$. The variable *x* is considered a design variable while *i* is considered an input variable. The domain of each function is [0,5]. The value of *x* for each function was chosen such that the range for each function is [0,125]. Figure 3 shows graphs of each function. Note that even though the functions share start and end points, how each function gets from start to end is drastically different.

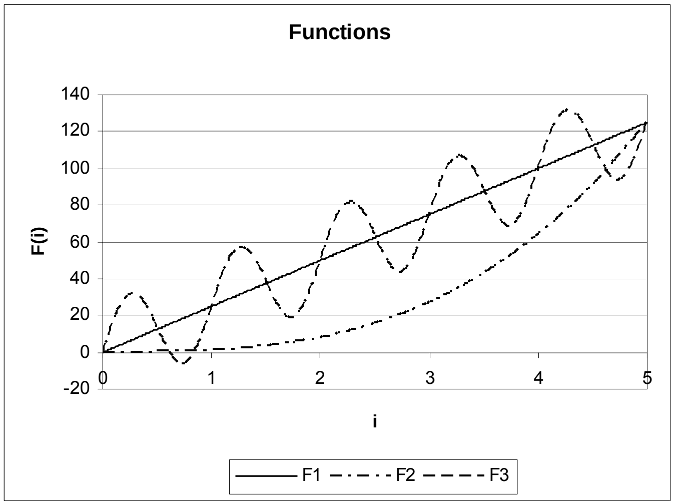


Figure 3-Function Values

The next figure 4, shows the sensitivity of the functions with respect to the input parameter *i*. For the first function, the sensitivity is a constant 25. The second function's sensitivity is one leg of a parabola. The sensitivity of the third function is a sin wave with an amplitude of 25*4*pi that is shifted up by 25 units. Using the formula for S, the average values for the sensitivities were found to be 25 for all functions. This results from all functions beginning at (0,0) and ending at (5,125). The coefficients of variation for the three functions are 0%, 89.63% and 439.66% respectively. As the value of the CV for the sensitivity increases, the variation from the mean increases. The value of the CV for first function's sensitivity represents the most consistent effect of input on output. The third function's sensitivity varies the greatest.

An engineer can determine this same result by simply looking at a side-by-side comparison of the graphs. However, this presents a completely different problem when a computer is trying to compare multiple solutions to a function. By calculating the CV of sensitivity for each function, it is a simple task for a computer to calculate and store the sensitivity and associated variation of a performance model with respect to a variable.

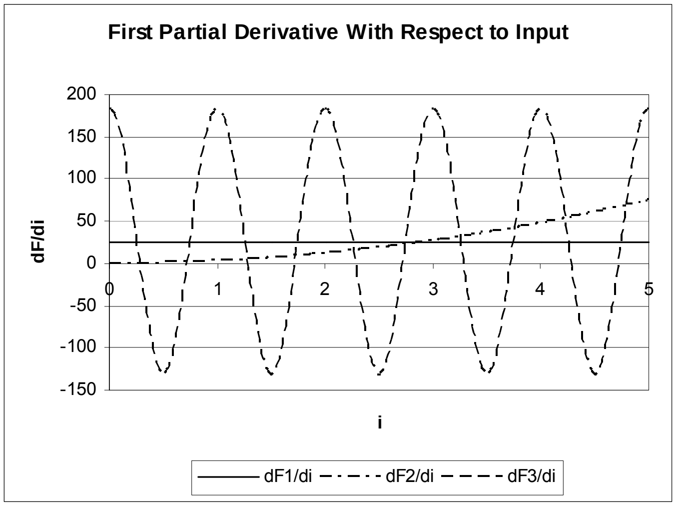


Figure 4-Sensitivity With Respect to Input

Figure 5 shows the graphs for the sensitivity analyses of the functions with respect to the design variable *x*. As seen in the graph, the first and third functions are less sensitive to changes in x than the second function. The average values for the sensitivity of the functions with respect to x are: 2.5, 31.3 and 2.5 respectively. The values of CV for the functions are 57.91%, 113.60% and 60.37% respectively. The values of S and H for the design variable provide two important pieces of information about the sensitivity of the function. The low value of S for the first and third functions relative to the second shows that the second function is much more sensitive to changes in the value of *x*. The higher value of the CV for the second function shows that this function exhibits larger variation in sensitivity across the range of the input. Changing the value of x for the second function has a greater effect on the output than the other functions but this effect varies greatly over the range of the input. The results of this example appear in Table 1.

Table 1-Sensitivity Analysis Results

	$\partial F_1/\partial i$	$\partial F_1/\partial x$	$\partial F_2/\partial i$	$\partial F_2/\partial x$	$\partial F_3/\partial i$	$\partial F_3/\partial x$
S	25.0	2.5	25.0	31.3	25.0	2.5
Cv	0.0%	57.9%	89.6%	113.6%	439.7%	60.4%

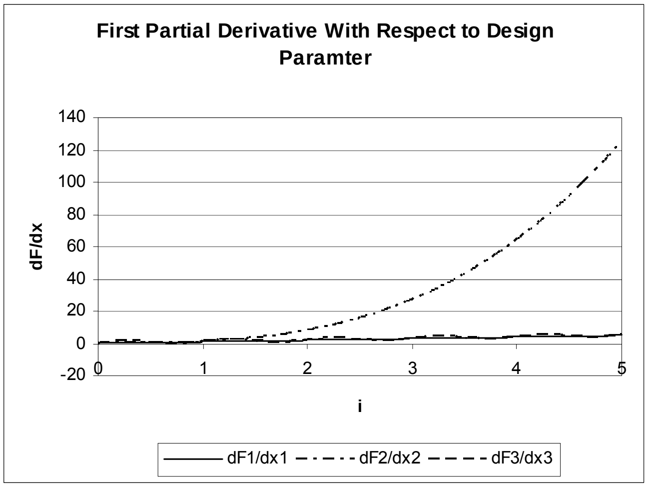


Figure 5-Sensitivity With Respect to Design Parameter

2.4 SENSITIVITY MEASURES AND FUNDESIGN

By applying these sensitivity measures to functions from previous designs, it is possible to extract and store design knowledge for use in future designs of systems with similar functionality. To accomplish this task, performance models must be associated with specific functions. The sensitivity of these models to the various design parameters in the model must then be found and stored in a design repository along with the performance model. This represents the FUNdesign knowledge storage process. When designing a new system, a conceptual functional model must be created to identify functions required in the design. The repository should then be searched to identify performance models and associated sensitivities for each function in the conceptual design. This information allows a designer to evaluate specific solutions to function based on their sensitivity to design parameters. Additionally, this information provides a means of allocating resources during future designs efforts by identifying specific functions and/or parameters than are sensitive to variation. These steps represent the application of FUNdesign to a new system.

3 HUMAN-POWERED FLASHLIGHT EXAMPLE

To illustrate the application of FUNdesign sensitivity knowledge storage, an example was conducted for the energy storage system of a human-powered flashlight. Several commercially available units are currently being offered. Three such flashlights were used to generate sensitivity information for use in the design of a new flashlight. Each flashlight satisfies the same overall function, to store human energy in the form of electrical energy and then convert this stored energy to light. However, the lower-level functions used to solve the overall function differ between the flashlight concepts. Identifying the individual functions in each concept that have the greatest effect on the performance of the flashlight as a whole can provide insight into which functions should have more design time and resources allocated. In addition, information regarding the sensitivity of performance with respect to design parameters can be used to identify parameters whose variation must be better controlled. Through FUNdesign, the problematic functions and design parameters were identified and the results of the analysis used to store knowledge that can be used for designing a robust flashlight concept or other system with similar functionality.

3.1 DESIGN GOALS

The goals of the human-powered flashlight design are to store as much human energy as possible is a given period of time and to store this energy proportional to time. The reasoning for the first goal is obvious; more energy stored in the flashlight allows more energy to be released as light. To accomplish this goal, the efficiency of each function in the design must be optimized to develop an efficient overall design. The FUNdesign process was used to determine which functions and associated design parameters had the greatest effect on the overall energy storage efficiency of the flashlight.

The second goal in the design of the flashlight is a result of the use of human energy to power the light. When someone is charging the flashlight, they are not going to charge it for an exact predetermined amount of time. It the energy storage rate is linear, a 15 second charge will store half the energy of a 30 second charge. The sensitivity parameters introduced earlier provide a convenient way of quantifying the satisfaction of these goals. S, the average sensitivity, is a measure of how much energy is stored versus time on average. The coefficient of variation of the sensitivity is used to quantify the behavior of the sensitivity across an input range. Combined, the S and Cv measures were used to represent how changes in design parameters affect the performance of the system both in average magnitude and behavior.

3.2 FUNCTIONAL MODELS

The first step in the FUNdesign knowledge storage process of the flashlight was to create a functional model for the system. As reviewed earlier, this process begins with the selection of a black box function. Since the purpose of the human-powered flashlight is to use human energy to create light, the overall function was chosen to be Convert Human Energy to Optical Energy. This function is represented in Figure 6.

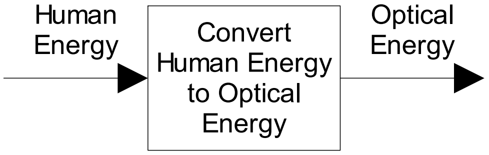


Figure 6 – Human-powered Flashlight Black Box

The next step in the functional modeling process was to create conceptual function chains. For the flashlight, only one chain was considered (one input and one output were modeled in the black box). This chain represents the transformation and storage of the human energy entering the system. This energy must first be imported into the system. This is accomplished with the Import Human Energy function. Next, this energy should be converted into electrical energy. This conversion is labeled Convert Human Energy to Electrical Energy. The electrical energy is then stored via the Store Electrical Energy function. To release this energy, a Supply Electrical Energy function is then needed. An Actuate Electrical Energy function is then used to turn the flashlight on and off. Finally, the Convert Electrical Energy to Optical Energy function is used to convert the electricity into light. This chain of functions is listed in Figure 7.

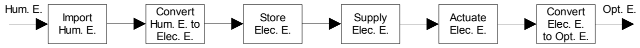


Figure 7 – Human-powered Flashlight Conceptual Model

Once a conceptual functional model has been created, lower-level process specific models can be developed for each concept. For the flashlight example, three different concepts were considered. The first concept analyzed used a crank and a gearset linked to a rotary generator to produce electrical energy. The energy was stored in a capacitor. To represent this concept functionally, the transformation of input energy to stored energy must be analyzed. After energy is input into the system by the human operator, it is converted into rotational energy by the crank. The gearset then changes the rotational energy by increase the rotational velocity while decreasing the transmitted torque. The rotational energy is then turned into electrical energy via a rotary DC generator. This sequence of functions is represented with the Functional Basis in the following function chain:

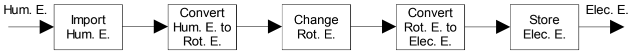


Figure 8 –Crank Flashlight Model

The next concept considered was a flashlight that converted a shaking motion into electrical energy. The force of shaking the flashlight was converted into electrical energy by a linear generator. This energy was also stored in a capacitor. A functional model for the shake flashlight concept appears in Figure 9.

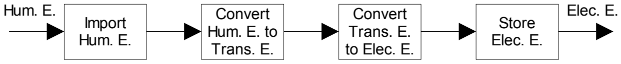


Figure 9 –Shake Flashlight Model

The final concept considered was a variant of the crank flashlight. This flashlight stored the rotational energy from the crank with a constant force spring. Once the spring was fully wound, the energy stored within it was transmitted through a gearset to a rotary DC generator. As with the preceding concepts, this energy was stored in a capacitor. The functional model for this concept appears in Figure 10.

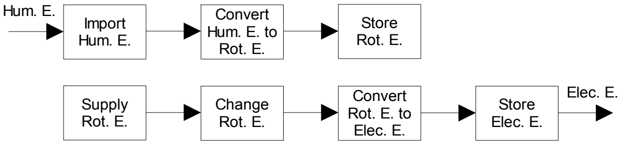


Figure 10 –Crank-spring Flashlight Model

3.3 PERFORMANCE MODELS

The next step in the FUNdesign process for the flashlight was to create performance models for the concepts. To create these models, performance equations for each function were created and then aggregated to produce an overall performance model. Since the energy stored in the flashlight is a function of time, differential equations were used to model each function. Simulink and Matlab were used to graphically represent and solve these differential equations for each flashlight. The crank model appears in Figure 11 along with a model for the Convert Human Energy to Rotational Energy function in Figure 12.

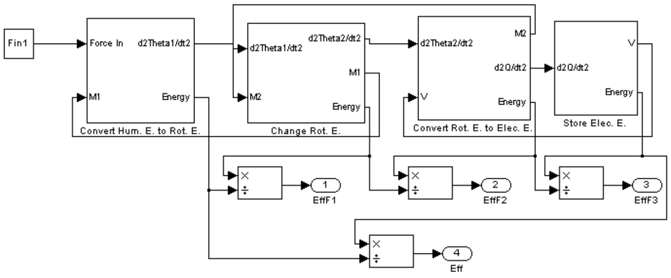


Figure 11 -Crank Flashlight Simulink Model

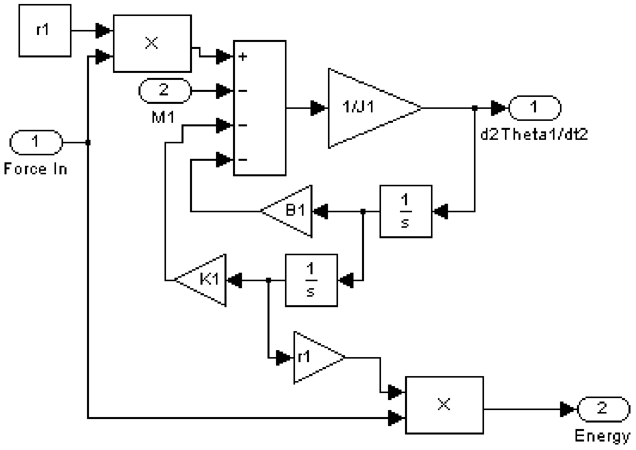


Figure 12 –Convert Human Energy to Rotational Energy Simulink Model

Parameter values for the models were obtained based on available parts or empirically chosen to produce the desired output. To insure that the concepts exhibited equal nominal performance, a goal of 150J of stored energy after 30 seconds of charging was selected. This amount of energy will power a 150 mA Xenon bulb for five minutes. For each concept, the design variables were chosen to result in 150J of stored energy from 180J of input energy after 30 seconds. An example of the efficiently versus time relationship appears in Figure 13. The values of the design parameters for each system appear in Tables 2, 3 and 4 for the Crank, Shank and Crank-Spring concepts respectively.

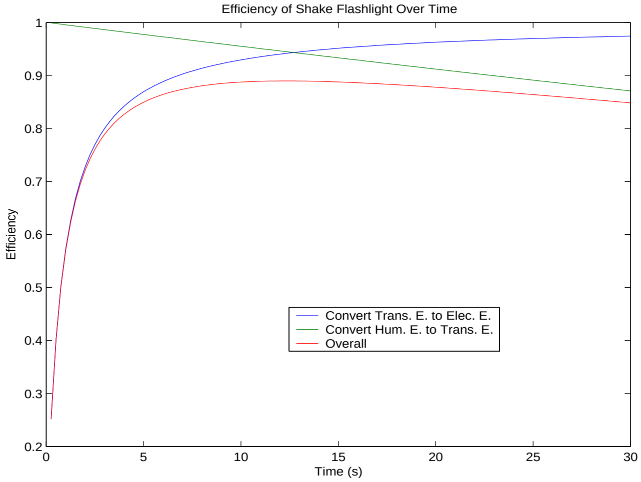


Figure 13 – Shake Flashlight Efficiency versus Time

Table 2 – Crank Flashlight Variables

Variables	Value	Unit	Description
F_{in1}	10	N	Input force
r_1	0.2	m	Radius of crank
J_1	0.001	kg/m2	Moment of inertia of crank
B_1	0.0001	N*m*s/rad	Friction of crank
X_2	0.0212	Unitless	Gear ratio
J_3	0.0005	kg/m2	Moment of inertia of generator
B_3	0.00001	N*m*s/rad	Friction of generator
K_3	0.2		Back EMF and Torque constant
L_3	0.1	H	Inductance of generator
R_3	5	Ohm	Resistance of generator
C_4	0.1	Farad	Capacitance

Table 3 – Shake Flashlight Variables

Variables	Value	Unit	Description
m_1	0.4	kg	Mass of flashlight
B_1	0.01	N*s/m	Friction between handle and magnet
K_2	0.29		Back EMF and Torque constant
L_2	0.0796	H	Inductance of generator
R_2	0.0011	Ohm	Resistance of generator
C_3	0.0796	Farad	Capacitance

Table 4 – Crank-Spring Flashlight Variables

Variables	Value	Unit	Description
F_{in1}	10	N	Input force
r_1	0.2	m	Radius of crank
J_1	0.01	kg/m2	Moment of inertia of crank
B_1	0.0333	N*m*s/rad	Friction of crank
M_2	1.9	N*m	Moment of constant force spring
J_2	0.01		Moment of supply reel
B_2	0.0333		Friction of supply reel
X_3	0.0265	Unitless	Gear ratio
J_4	0.0002	kg/m2	Moment of inertia of generator
B_4	0.00001	N*m*s/rad	Friction of generator
K_4	0.25		Back EMF and Torque constant
L_4	0.1	H	Inductance of generator
R_4	4	Ohm	Resistance of generator
C_5	0.1	Farad	Capacitance

3.4 SENSITIVITY ANALYSIS

After the nominal performance was equalized for each concept a sensitivity analysis was performed for each function in the concepts. The objective of this analysis was to find which functions were more affected by uncertainty in the values of the design parameter. To determine the sensitivity of each function, a normalized time-averaged sensitivity was used. This value was computed by finding the first partial derivative of the efficiency with respect to the variable of interest at each time step in the solution. The average value of the sensitivity was then found and normalized with respect to a 1% change in the design parameter. For example, in Table 5 a 1% increase in the value of the variable X3 results in an average .0655% increase in overall efficiency. The coefficient of variation of each sensitivity curve was also calculated and appears in the tables below.

Table 5 – Change Rotational Energy Sensitivity

Crank Spring	Change Rot. E.	X_3
	S (1%)	0.0655%
	Cv	54.98%

The values of S and Cv were calculated for each design variable within each function for all three concepts. The results of this sensitivity analysis were then used to identify various classes of functions. Tables 6 and 7 show functions that are on average more sensitivity and less sensitivity respectively. The lowest sensitivity to variation in the Convert Rot. E. to Elec. E. function is two orders of magnitude higher than the most sensitive variable in the Convert Hum. E. to Rot. E. function. It is also shown in Table 6 and 7 that these two functions have comparative sensitivities across the different concepts. The values for the average sensitivity of similar design variables (the values of R for example) are similar.

Table 6 – Convert Rotational Energy to Electrical Energy Sensitivity

Crank	Rot. E. /Elec. E.	J_3	B_3	K_3	R_3
	S (1%)	-0.106800%	-0.0231%	0.27%	0.09%
	Cv	13.13%	57.62%	19.20%	79.57%
Crank Spring	Rot. E. /Elec. E.	J_4	B_4	K_4	R_4
	S (1%)	-0.025000%	-0.0136%	0.143%	0.069%
	Cv	7.88%	56.44%	36.75%	79.57%

Table 7 – Convert Human Energy to Rotational Energy Sensitivity

Crank	Convert Hum. E. to Rot. E.	J_1	B_1
	S (1%)	-0.00010%	-0.00010%
	Cv	11.57%	57.60%
Crank Spring	Convert Hum. E. to Rot. E.	J_1	B_1
	S (1%)	0%	0%
	Cv		

During the sensitivity analysis it was also found that some functions exhibited large fluctuations in sensitivity across concepts. Table 8 shows the sensitivity analysis results for the Store Elec. E. function. The largest value of sensitivity is three orders of magnitude greater than the smallest value. In addition, the values of the coefficient of variation for the capacitance is much greater than one for the Crank and Crank-Spring systems and less than one for the Shake system.

Table 8 – Store Electrical Energy Sensitivity

Crank	Store Elec. E.	C_4
	S (1%)	0.04%
	Cv	216.43%
Shake	Store Elec. E.	C_3
	S (1%)	-1.40%
	Cv	73.67%
Crank Spring	Store Elec. E.	C_5
	S (1%)	0.0017%
	Cv	4776.00%

The overall effect of each function on the efficiency of a concept is shown in Table 9. This table contains the sensitivity analysis for each function in the crank concept. The large relative values of S for the design variables in the Convert Rot. E. to Elec. E. function shows that this function deserves the most attention during the design process. Uncertainty in the design parameters of this function result in large changes in the overall efficiency of the system. This also means that tweaking these parameters results is the largest gains in overall efficiency. The low relative values of Cv for the J3 and K3 parameters show that adjusting these parameters has a more consistent effect on efficiency across the 30 second charging time.

Table 9 – Crank Concept Sensitivity

Hum. E. / Rot. E.	**J_1**	**B_1**		
S (1%)	-0.00010%	-0.00010%		
Cv	11.57%	57.60%		
Change Rot. E.	**X_2**			
S (1%)	0%			
Cv	32.34%			
Rot. E. / Elec. E.	**J_3**	**B_3**	**K_3**	**R_3**
S (1%)	-0.106800%	-0.0231%	0.27%	0.09%
Cv	13.13%	57.62%	19.20%	79.57%
Store Elec. E.	**C_4**			
S (1%)	0.04%			
Cv	216.43%			

3.5 DISCUSSION OF RESULTS

The results of the sensitivity analysis of the three flashlight concepts can be used to provide key information when designing a new flashlight or a device with similar functionality. The high sensitivity of the Convert Rot. E. to Elec. E. function means that great care should be taken when using this function to reduce variation in design variables. This also means that the greatest gains in efficiency can be found by optimizing this function. The values of Cv for the design parameters allow a designer to choose which variables to modifying based on how their effect on the overall performance varies with time. Since the probability that a human powered flashlight is going to be charged for exactly 30 seconds is small, a consistent effect on performance over time is desirable. Keying the results of the sensitivity analysis to functionality allows this knowledge to be directly applied during the design of a new flashlight or any system that requires similar functionality. To store this knowledge, an engineering design knowledge repository is used.

4 FUNDESIGN

The flashlight example shows the sensitivity of functions to uncertainty in design variables can be classified during the design process and stored for use in future design efforts. Certain functions in a system, such as the Convert Rot. E. to Elec. E. function, exhibit much higher sensitivity than other functions the system. With these functions, certain design variables exhibit more a consistent effect on the performance of the system. Additionally, functions that exhibit large variation in sensitivity across applications in different concepts can be identified. These functions can be classified as noisy and their use can be reduced or modified in order to satisfy design goals. To enable FUNdesign for new systems, the results of the sensitivity analyses should be stored in an engineering design repository.

By performing these function-based sensitivity analyses and storing the results, significant knowledge about the sensitivity to design variable uncertainty can be retained and reused though common functionality. To apply this knowledge, a functional model must be created for the system to be designed. For each function in the model, the design repository should be searched for similar functions. If solutions containing similar functions are found, the knowledge associated with these functions should be applied to the new design to prioritize modeling and resource allocation. If no matching functions are found in the repository, the information must be generated as shown in the flashlight example. Generating and applying sensitive information is the basis for the FUNdesign technique (Figure

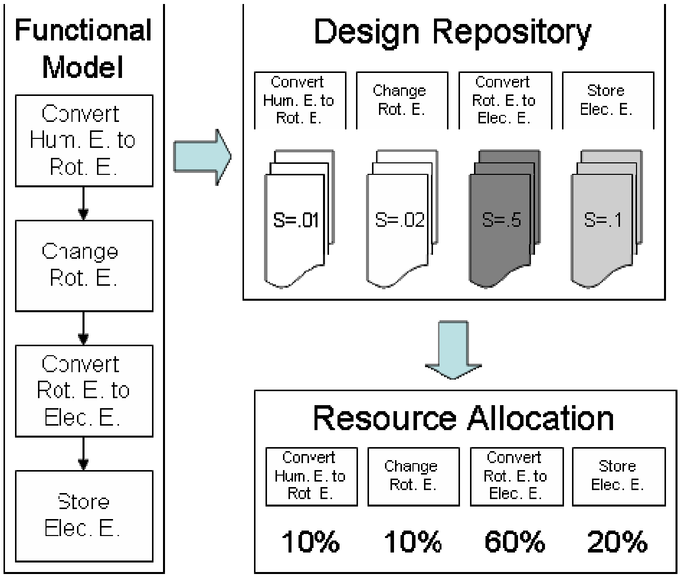


Figure 14 –FUNdesign

The next steps for this research include populating a design repository with performance models and associated sensitivities for a variety of functions as well as multiple solutions for particular functions and applying the results from the knowledge storage phase of FUNdesign to the design of a new system. The first task requires the identification of a set of existing design solutions that possess a variety of functionality in addition to multiple common functions. In addition to the presented human-powered flashlight solutions, several other human-powered devices, such as a human-powered radio or certain types of watches, exist that could be used as sources for design knowledge. Specific solutions for function could also be used to increase the knowledge stored in the repository. For example, a chemical battery could be modeled for the Store Electrical Energy function instead of a capacitor or a flywheel could be used to store rotational energy before being converted to electrical energy.

The second task, applying the stored knowledge, involves identifying a design problem that requires specific functions that are represented in the knowledge base. This could involve a design for a product that doesn't exist or an improvement on a specific solution that occurs in the repository.

REFERENCES

Akiyama, K. (1991), Function Analysis: Systematic Improvement of Quality Performance, Productivity Press.

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.

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.

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.

Hutcheson, R., Donndelinger, J., McAdams, D. and Stone R. (2004), “Applying Functional Modeling as a Unifying Basis for Design for Six Sigma Execution,” Proceedings of the ASME International Mechanical Engineering Congress and RD&D Expo, IMECE2005/62312, Anaheim, CA.

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.

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.

Stone, R. and Wood, K. (2000), "Development of a Functional Basis for Design," Journal of Mechanical Design, 122(4):359-370.

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.

VITA

Ryan Scott Hutcheson was born April 12, 1981 in Texas County, Missouri. He completed his Bachelor's Degree in Mechanical Engineering at the University of Missouri-Rolla in August 2003 and will graduate in May 2005 with a Master's Degree in Mechanical Engineering. During his academic career at UMR, Ryan was as member of the Formula SAE team for five years and a helpdesk consultant for three. During his time on the Formula team, he served as Chief Engineer in 2003 and Team Secretary in 2002. Ryan has interned with General Motors Research and Development and North American Product Development and is currently contracted to NASA's Ames Research Center in Mountain View, California through August 2005 at which time he will begin pursuing a PhD in Mechanical Engineering at the University of Missouri-Rolla.