Function-based behavioural modelling

Ryan S. Hutcheson

Department of Mechanical and Aerospace Engineering,
Missouri University of Science and Technology,
Rolla, MO 656409, USA
Fax: 573-341-4086
E-mail: rhutch@mst.edu

Daniel A. McAdams*

Department of Mechanical Engineering,
Texas A&M University,
MS3123, College Station, TX 77843, USA
Fax: 979-845-3081
E-mail: dmcadams@tamu.edu
*Corresponding author

Irem Y. Tumer

School of Mechanical, Industrial, and Manufacturing Engineering,
Oregon State University,
Corvallis, OR 97331-6001, USA
Fax: 541-737-2600
E-mail: irem.tumer@oregonstate.edu

Abstract: Using mathematical behaviour models to determine system performance is crucial to the design of complex systems. Nevertheless, during the early stages of design, it is often difficult to create behavioural models as component solutions have not been fully identified. What is known is information about the desired functionality of the system is known. Particularly for complex system design, functional-based representations are important during the early stages of design. The objective of the work presented in this paper is to develop a functional model-based method for creating behavioural models to facilitate early behaviour model creation, model reuse, and concept evaluation. Significant contributions of this work include establishing the method and the formal addition of quantitative flow modelling to existing qualitative functional modelling methods. This quantitative flow modelling method enables a natural and consistent extension of qualitative functional modelling to quantitative behaviour modelling. An extensive example application of the method based on the dynamic modelling of a Society of Automotive Engineers Formula competition racecar is included to illustrate and verify the method.

Keywords: complex system modelling; vehicle design; simulation-based design; function-based design; model reuse; function-based behavioural modelling.

Reference to this paper should be made as follows: Hutcheson, R.S., McAdams, D.A. and Tumer, I.Y. (2012) 'Function-based behavioural modelling', *Int. J. Computer Aided Engineering and Technology*, Vol. 4, No. 3, pp.250–271.

Biographical notes: Ryan S. Hutcheson is currently an Assistant Teaching Professor of Mechanical Engineering at Missouri University of Science and Technology. He received his PhD from Texas A&M University in 2009. His research interests focus on complex system modelling, functional modelling, product design, and simulation-based design with focus on automotive applications.

Daniel A. McAdams is currently an Associate Professor of Mechanical Engineering at Texas A&M University. He received his PhD from the University of Texas at Austin in 1999. His research interests are in the area of design theory and methodology with specific focus on functional modelling; innovation in concept synthesis; biomimetic design methods; universal design; and complex system model construction and selection for design as applied to product design. His research has been funded by various organisation including US Government research agencies and industry.

Irem Y. Tumer is an Associate Professor of Mechanical Engineering at Oregon State University. Her research focuses on the overall problem of designing highly complex and integrated systems with reduced risk of failures, developing formal methodologies and mathematical frameworks to help understand and enhance complex system design. Prior to accepting a faculty position at OSU, she led the Complex Systems Design and Engineering Group in the Intelligent Systems Division at NASA Ames Research Center, where she worked from 1998 through 2006.

1 Introduction

Behavioural modelling, the process of using quantitative models to investigate the performance of a system relative to requirements and specifications, is a basic part of the systems engineering process. Currently, the behavioural modelling process is essentially component driven: the intended functionality of a system is described, component solutions are identified for this functionality, and then behavioural models are created based on the identified components. During a typical behavioural modelling process, the functional description of the system is largely ignored. In this work, a function driven behavioural modelling approach is introduced as a fundamental component towards the tying together of functional modelling, concept generation, concept selection, and behavioural modelling.

This function-driven behavioural modelling approach is described in four sections. Section 2 outlines current systems engineering practices and illustrates how the function-based behavioural modelling (FBBM) approach integrates within this work. Section 3 describes the modelling approach in detail. A complete case study for a large system is then presented in Section 4. Finally, conclusions and future work are presented in Section 5.

2 Design of systems

The behavioural modelling framework developed here complements and integrates with formal design processes such as those used to design complex systems. Formal design processes for large complex systems can be classified as the 'European' or traditional approach and 'systems engineering' approach, perhaps more associated with the evolution of ideas that originated primarily in The US National Aeronautics and Space Administration (NASA).

The European school of mechanical design is exemplified by the work of Pahl and Beitz (1996) and Pahl et al. (2007). Other research into design methodologies within this body of work define similar processes involving more or less the same four steps (Otto and Wood, 2001a; Suh, 1998; Ullman, 2002). A comprehensive review of systems engineering can be found in a text by Sage and Armstrong (2000). Both schools of engineering design promote the same basic order of operations during the initial stages of system design: identify what needs to be done at a functional level, find solutions that can accomplish this functionality, compare the solutions through the use of models, and make a decision about which solutions to investigate further. It is this exact sequence of engineering activities that FBBM aims to assist by augmenting currently recommended systems engineering practices with formal functional modelling and FBBM.

2.1 Functional modelling during design

The formal functional decomposition of a complex design problem is promoted in both traditional engineering design (Pahl et al., 2007) and systems engineering (Sage and Armstrong, 2000). Several methods exist for performing this decomposition. System engineering texts often use the function-flow block diagram (FFBD) approach (Morgan, 2006; Blanchard and Fabrycky, 2006). These models include the desired functional elements of a system along with a logic-based control flow connecting these functions. FFBDs evolved within the systems engineering community (Blanchard and Fabrycky, 2006).

A similar functional decomposition approach evolved within the traditional engineering design approach beginning in the early 1960s (Miles, 1972). This functional analysis approach focused on the use of verb-noun pairs along with an explicit breakdown of flows into energies, materials, and signals. This flow-based approach has been extensively researched and formalised with the creation of a function lexicon known as the functional basis in the work of Hirtz et al. (2002). The benefits of this approach over the FFBD approach include the use of a standard modelling lexicon and an emphasis on the energy and material flow interaction of the functions.

Independent researchers have critiqued and verified the functional basis lexicon and associated flow-based modelling methodology for engineering design applications (Gietka and Verma, 2002; Ahmed and Wallace, 2003). The functional basis continues to gain increasing acceptance as the preferred representation and lexicon for functional modelling (Dieter and Schmidt, 2009). Other researchers are extending it with rules and grammars for functional model creation (Caldwell and Mocko, 2008). Other extensions related to the specific use here include a sensitivity-based analysis method for comparing concepts during concept selection (Hutcheson and McAdams, in press). For a complete survey of this branch of functional analysis see Nagel et al. (2008). For the remainder of this paper, a functional model will refer to a formal, flow-based functional decomposition of a system using the functional basis approach (Hirtz et al., 2002).

The functional decomposition of a system allows a complex design problem to be broken down into smaller elements that can then be analysed in detail. In most formal system design methods, the functional decomposition is also recommended as the starting point for identifying potential solutions to the design problem. The primary reason for this recommendation is that functional models allow designers to represent a system with minimal dependence on a particular physical, form-based solution. Functional abstraction decouples the task of representing what a system needs to do from how it is going to do it. From morphological charts (Otto and Wood, 2001b) to current knowledge driven concept generation algorithms (Kurtoglu and Campbell, 2008), significant research has been conducted into expanding the ability to explore a problem solution space by using function-based methods.

The importance of functional abstraction and the associated decoupling of function and form is demonstrated in the following example. If asked to design an open reservoir system that extends a single piston (with an external return force) with single manual control input actuation, a hydraulics designer would likely start by creating a schematic like the one shown in Figure 1. Though the diagram in Figure 1 is an abstraction, it is an abstraction of specific component families that implies significant assumptions about the form of the solution. In design cases where the technologies of the solution space are not yet established, a less form-specific representation should be used during the early stages of design. As a contrast, the same basic functionality of the system, represented with minimal assumptions about form, appears in the functional model in Figure 2.

Figure 1 A hydraulic system schematic

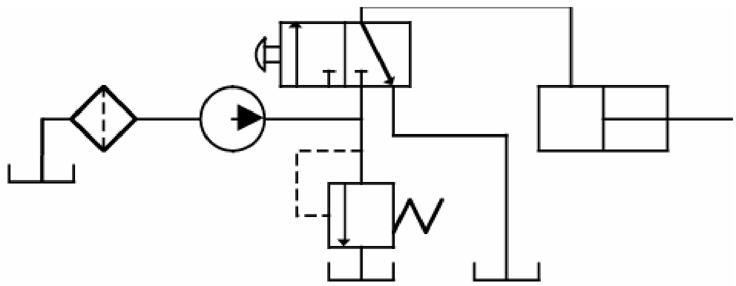
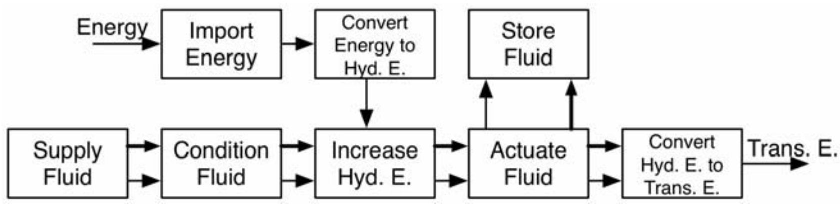
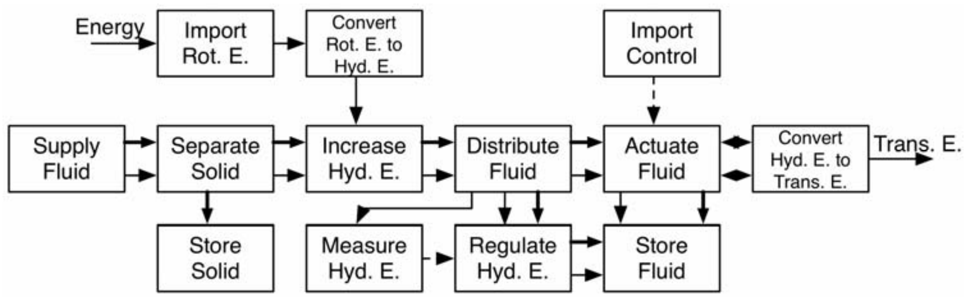


Figure 2 A functional model of a hydraulic system



A model such as the one in Figure 2 enables engineers to investigate potential design solutions that may not have been considered if a physical form and technology had already been selected. Similar to the detail implied in Figure 1, functional models can be further decomposed into form-specific models once component-based solutions have been identified. This is done by updating the original functional model with functions specific to the selected components as exemplified by the functional model in Figure 3.

Figure 3 A component specific functional model of a hydraulic system



2.2 Behavioural modelling during design

Mathematical models are useful evaluation tools throughout the design process. During the planning phase, energy balance equations can be used to identify size and performance requirements, cost models can be used to estimate potential product cost, time-to-market models can be made, etc. During conceptual design, models are used to estimate performance and other aspects of a potential concept such as weight and size. These models can in turn be used to compare concepts during the concept selection process. During embodiment design, behavioural models are used for detailed analysis of components such as a finite element analysis or entire system simulations. Additionally, during embodiment design, behavioural models for auxiliary functionality are made. These more detailed models can then be used to further evaluate and optimise a design.

Because models are used for varying tasks through different phases of design, multiple modelling paradigms are often used during the design of a single system. In general, different problems require different modelling approaches. Algebraic equations might be suitable for some modelling efforts. Partial differential equations can appear when creating high-fidelity models resulting from analysing stresses and fluid flows. Design-of-experiments (DOE) models can be used to analyse the results of experiments on prototypes. Ordinary differential equations (ODEs) and differential algebraic equations (DAEs) are frequently encountered during systems modelling tasks. The wide range of modelling paradigms used in the design of complex systems has resulted in an effort to create tools and frameworks to better create design models.

Generic component-based system modelling languages such as Modelica (The Modelica Association, 2007) have been developed as a means to model complex systems across multiple domains. Simulations of systems (The MathWorks Inc., 2007; Laminar Research, 2007; Mechanical Car Simulation Company, 2007) are also useful for designing complex systems that incorporate multiple domains, but are generally limited to a few well-understood cases. Dynamic systems can be represented using bond graphs (Karnopp et al., 2000; Bracewell and Sharpe, 1996). Computational fluid dynamics and finite element analyses are used to model domain specific problems such as stress analysis or flow over aerodynamic surfaces. Discrete event models appear frequently in complex systems engineering problems and information intense systems.

What these methods all have in common is that they are mathematical representations of the transformations of inputs to outputs. They are model representation formats and each has its own method of solution and development process.

2.3 Opportunities to improve upon current practices

Currently, behavioural modelling is considered more of an art than a science (Cross and Moscardini, 1985; Giordano, 1985; Fawkes and Mahony, 1994). The creation of behavioural models is based primarily on an engineer's personal experience and the resultant problem specific judgement and insight. The personal experience is generally tied to a component that fulfils some function, not the function itself. As a result, the decomposition of complex behavioural modelling tasks is generally performed at the component level. Thus, if different sets of components are used to solve the same problem, a new model decomposition and development must be done. Because of this, it is often difficult to create detailed behavioural models early in the design of a system when there are multiple potential conceptual solutions and detailed component information may not be completely available.

Functional models provide an opportunistic starting point for creating behavioural models. A functional model contains encapsulations of desired transformations, explicit definitions of energy, material and signal flows, and flow routing information. In related work, functional modelling techniques have successfully been used to drive the mathematical modelling of systems in limited applications (Chandrasekaran et al., 1993; Tomiyama et al., 1993; Yekula et al., 2003). The use of functional models to assist model-based design of systems using discrete state behavioural models is demonstrated in Bhatta et al. (1994). In this work, the functions in the functional models are represented as a schema that specifies the behavioural state the function takes as input, the behavioural state it gives as output, and a link to internal causal behaviour that identifies the behaviour that enables the function (Bhatta and Goel, 1994).

The method developed here differs from earlier proposed FBBM approaches in that it uses the functional basis flow-based functional modelling method and does not prescribe a fixed behavioural model format (such as bond graphs).

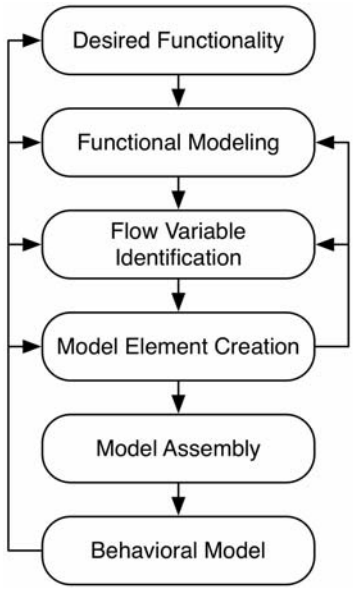
3 Function-based behavioural modelling

The process of FBBM consists of four basic steps along with iterations between the steps. These steps and their iteration loops are outlined in Figure 4 and are explained in detail in this section.

3.1 Functional modelling

The first step in the FBBM process is creating a functional model of the system. The functional model graphically captures both the overall and detailed functionality of the system. The objective of this step is to model all of the important transformations of energies, materials, and signals. To promote model reuse in subsequent steps, a standard taxonomy such as the functional basis is used when creating functional models.

Figure 4 FBBM approach



To create a functional model for a system, a black box model is first created. This model captures the overall functionality of the system along with external energy, material, and signal flows. These external flows are used later to create detailed chains of internal functions. The overall function is selected by determining the most important aspect of a product's required functionality based on the needs of the user.

Once the black box model has been created, a detailed functional model is made. The detailed functional model captures the sequence of transformations that must occur on external input flows in order for them to become the desired output flows. The objective of this step is to determine exactly *what* needs to be done by the system in order to satisfy the user's needs. *How* these functions are satisfied is addressed during the behavioural modelling steps. The general process for creating such a functional model involves selecting an input flow from the black box and identifying the chains of functions that must occur in sequence to transform the input flow into one or more of the output flows. The chains are then aggregated and combined to create a complete functional model.

3.2 Quantitative flow variable identification

The functional model for a system abstractly and qualitatively captures what a system must do in order to satisfy the user's needs. The behavioural model quantitatively expresses how these needs are fulfilled by the functions. The next step in the FBBM process is to express the qualitative material, energy, and signal flows with quantitative descriptors termed quantitative flow variables. The quantitative flow variables represent numeric quantities that are operated on by the behavioural model. Representing all qualitative flows with quantitative variables is necessary to model the impact of each function on the flow as it sequences through the function chain.

Before appropriate quantitative flow variables can be selected, the behavioural model type needs to be considered. Selection of a model type depends on the information desired from the model as well as the information that is available regarding the specific solutions to functions. If the behavioural model is being used very early in the design of a system, there might not be much information available regarding potential solutions. As a result, a lower-fidelity, less solution-specific model is appropriate. Such models include basic energy and mass conservation equations along with simple signal processing models. The quantitative flow variables required for such models include energies expressed as power flows, mass flows (kg/s, etc.), and continuous or discrete signals.

For higher fidelity models, the form of the model becomes more specific to the physical form of identified solutions. These models include FEA stress models of components, CFD flow models, simulations, etc. In such models, quantitative flow variables reflect energy flows. Frequently, they are broken down into formal power flow and power effort components such as force with displacement, pressure with flow rate, or voltage with current (Karnopp et al., 2000).

3.3 Model element identification

The next step in the FBBM process is to identify behavioural model elements. The objective of this step is to find a mathematical relationship between the input and output flow variables for each function in the functional model. If the model is being created early in the design process, the model elements are created as form-independently as possible. If component solutions have been identified, the models can be more detailed to include component-specific information. If the model is to be used for component selection, then various models of the different potential component solutions for each function should be made. These models may be reused from prior solutions with similar functions (and correspondingly similar input and output flows) or can be derived using first principles. The type of model used for each function depends on the overall objectives of the model. FBBM is not a model representation format, but rather a method for organising and structuring the model development process. Thus, selection of a suitable model representation format is dependent on the modelling objectives and is left to the designer.

During the model element identification step, the designer may notice overlooked flow variables and needed system functionality. If this is the case, these elements are added to the model and their impact on the existing model elements accommodated. The benefits of using a functional model for organising the behaviour modelling task are apparent here. The original copies of the model can be saved and the updated versions tracked along with any other information discovered during the modelling process. This information can be associated with functionality, stored in a design repository, and retrieved when designing systems with similar functions.

3.4 Model solution

The particular solution method required for the behavioural model depends on the type of the model. For certain classes of problems, general solutions are available. For combinations of linear, non-DAEs, matrix methods can be used. For linear differential equations, state-space representation and solution can be used. General systems of non-linear algebraic equations can be found with n-dimensional root finding algorithms. If the behavioural model elements are Simulink blocks or Modelica code, then Simulink or a Modelica compiler can be used to solve the problem. As with the previous step, during model solution, missing functionality or quantitative flow variables as well as improper model elements may be identified. If this is the case, then iterations returning to prior steps should be done. Once again, copies of the model elements and changes should be tracked and associated with functionality for reuse.

3.5 Discussion

The process of constructing a behavioural model from a functional model is done in four steps: functional model creation, quantitative flow variable identification, model element identification, and model solution – along with any iterations required to arrived at the desired solution. The result of this process is a mathematical model of a system that is tied to its functionality. The FBBM modelling method includes five important contributions to system modelling practice:

1. it links behavioural model decomposition with functional decomposition
2. it creates a framework to reuse behavioural model elements based on common functionality
3. it creates a framework that enables swapping of model elements with varying fidelity
4. it provides a framework for linking function-based component identification with mathematical component combination evaluation
5. it links assumptions made during mathematical modelling to their effects on the functionality of the product and vice-versa.

Contribution 1 is enabled by the creation of a functional model early in the design process. This model is then used as the starting point for the behavioural modelling tasks that occur later on. The second contribution is enabled through the storage of behavioural model elements based on their associated function. Once a complete behavioural model has been created for a system, model elements can be stored, along with their associated functions and components plus the assumptions made during the model creation process, in a design repository. When creating behavioural models for new systems, the functional model of the new system can be used to look up model elements and component solutions from prior designs based on common functionality.

The third contribution stems from the use of functional model flows for quantitative flow variable identification. The flow variables that are evaluated by the behavioural model are tied to the energy and material flows in and out of its associated functions. As long as a behavioural model operates on the same input and output flow variables and contains the same number of independent relationships, the model can be used in the overall solution. Thus, the underlying model for a single function can be developed at different levels of fidelity and swapped in or out of the system model. Such a framework allows a detailed exploration of solutions for some function while leaving other functions modelled at coarse level of fidelity.

If a large set of existing model elements is available for the functions in a functional model and the behavioural model elements for each function have compatible input and output flow variables, the method allows mathematical evaluation of multiple concepts. Through enumerated or guided searches, the behavioural model elements from prior solutions to the functions in the system being designed can be assembled according to the flow connectivity in the functional model. The result is a well-defined (and potentially automated) method for creating a mathematical evaluation model for multiple concept solutions to a functional model.

The fifth contribution is implemented by associating the functional and behavioural model elements. When a change is made to a behavioural model, its effect on the functional model should be investigated. The opposite should also be considered. When a change is made to the functionality of a system, its impact on the behavioural models should be explored. By keeping track of these changes and their effects, this information can be preserved for use in designing future similar systems.

The method presented here has some apparent similarities to component-based modelling methods such as bond graphs and block diagrams. However, it has a key fundamental difference. The approach presented here is a method for associating a mathematical model to a set of functions rather than a mathematical model representation method. A major objective of the work is to not limit the approach to a specific model representation format and/or solution method. In the following example, multiple modelling techniques are used to create a complete automotive simulation including linear dynamic equations, non-linear algebraic equations, and a complex experimental tire model.

4 Automotive model example

To illustrate FBBM, the process is used here to create a dynamic model of a Formula SAE racecar. Formula SAE is a collegiate design competition that requires students to design, build, test and, compete with an open-wheeled racecar (The Society of Automotive Engineers, 2007). The objective of the model created here is to create a platform for evaluating suspension and tire design and setup decisions focused on high-speed performance and handling. Consistent with the philosophy of design modelling, the goal is to make the decisions as accurately as possible without having to build prototypes or conduct costly testing. Based on the vehicle dynamic performance focus, the functional and behavioural modelling aspects of this example are limited to those that directly affect vehicle dynamic performance. Each of the four-step behaviour modelling steps, along with results, is presented in this section.

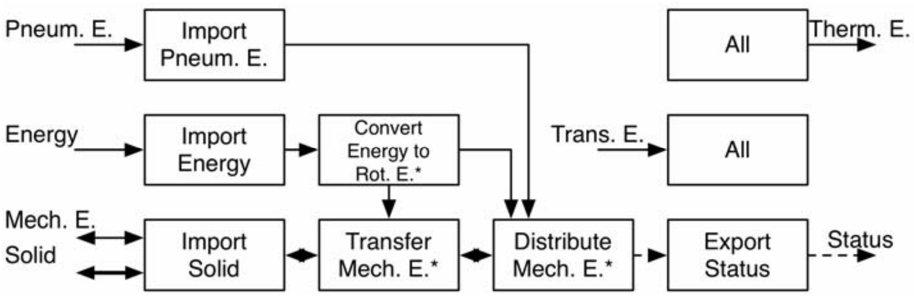
4.1 Functional modelling

The first step in creating a functional model for the racecar is to create a black box model. The overall function of the car is to transport the driver (*human*) around a racetrack. The next step is to find the overall flows in and out of the system. Since the model objective is to determine the suspension and handling aspects of the car, only input and output flows that impact vehicle dynamics are considered. These flows include the ground (*solid*) and the *mechanical energy* transmitted to it as it passes through the system. Additionally, there are flows of *pneumatic energy* (from aerodynamic forces), *translational energy* from gravity and a *rotational energy* flow representing the energy input from the engine. The driver's inputs to the system are modelled as throttle, steering, and braking *control signals*. Two signal outputs are included for the system: the state of the engine and the motion of the car. The overall function and flows are shown in the black box functional model shown in Figure 5.

Figure 5 Black box functional model

Figure 6 shows a conceptual level functional model. Such a functional model is consistent with one created during a conceptual design project in which there are few constraints or assumptions about the use of the vehicle. As the goal here is to create a quantitative vehicle dynamics model from a functional model, the functional model development extends with this specific focus.

Figure 6 Conceptual vehicle functional model

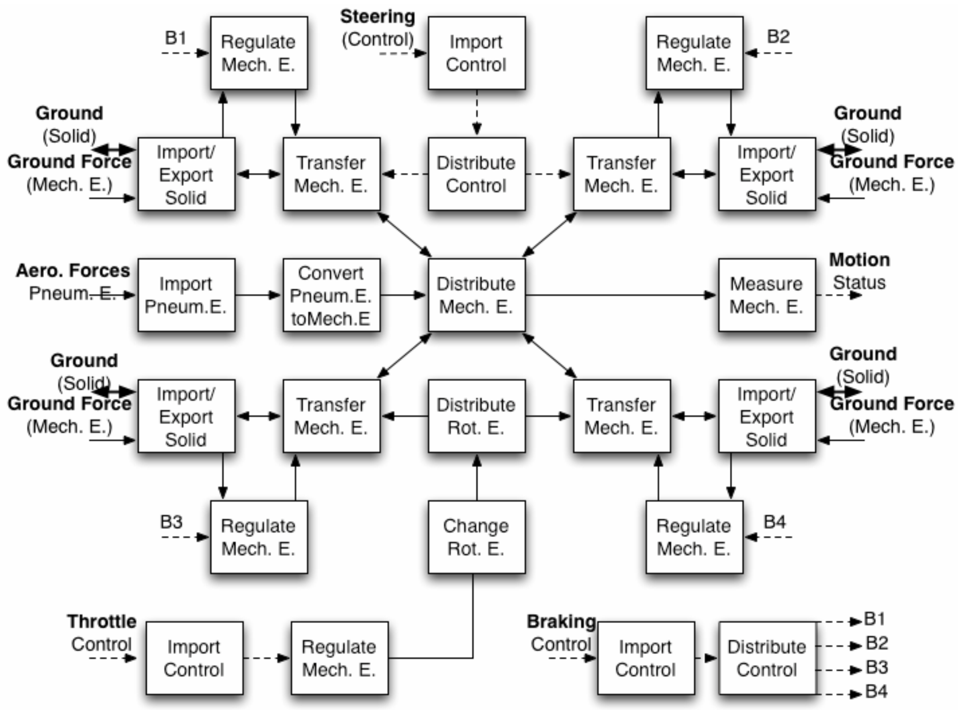


Note: *Driver input control passes into these functions

Continuing, the flows in the black box model are now used to create a complete functional model. The complete model is created by selecting an input flow and identifying the chain of functions that must occur on the flow in order to transform it to one or more of the system’s outputs. For example, the *solid* flow of the ground is *imported* into the system. The *mechanical energy* associated with this flow is then *transferred* through the system by the suspension of the car. Next, this energy is *distributed* through the frame of the car before being *transferred* through the other suspension corners in the system and finally *exported* from the system. Since a Formula SAE car must contain four wheels, four of these flow chains are considered. Chains are also created for the other flows in the black box. These chains are then aggregated to produce a complete functional model. A functional model that focuses on important functions for vehicle dynamics is shown in Figure 7. Again, two-way arrows are used in this model to simplify the model representation (as opposed to showing function chains for each direction). The use of two-way arrows versus one-way arrows has no explicit impact on causality in the behavioural models, it is simply a way of tidying up the model. Additionally, carrier flow relationships (Nagel et al., 2007) are used to simplify the importation of *mechanical energy* from the ground. The mechanical *energy* is carried into the system via the *solid* flow of the ground.

The functional models for the vehicle shown in Figures 6 and 7 are consistent with accepted functional modelling methods (Dieter and Schmidt, 2009; Otto and Wood, 2001a; Nagel et al., 2008). Specifically, these models use the functional basis for functional representation and the associated flow-based method in which the product functions operate exclusively on the flows and not on the product itself. Nevertheless, the functional model in Figure 7 represents an extension beyond typical current functional modelling practice. The model here has been created with a focus on behavioural modelling of vehicle dynamics: the product functions crucial to dynamic performance are modelled. Thus, required vehicle functionality such as *import energy* for bringing fuel into the vehicle, and shown in Figure 6 conceptual level functional model, is not included.

Figure 7 Vehicle dynamics functional model



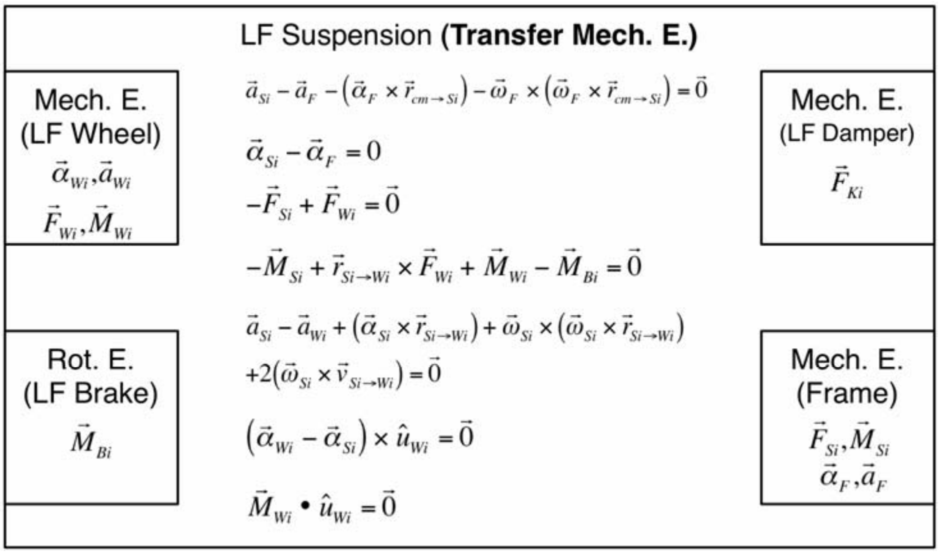
4.2 *Quantitative flow variable identification*

The next step in creating the behavioural model for the car is to identify the quantitative flow variables to be explicitly modelled. To complete this step, each qualitative energy or material flow in the model is broken down into a number of quantitative variables to be constrained through mathematical models. For example, the mechanical energy flow between the *distribute mechanical energy* function (the car's frame) and the *transfer mechanical energy* functions (the suspension corners) consists of both *rotational energy* and *translational energy*. The *rotational energy* is modelled as a three-dimensional moment vector along with an angular acceleration. Similarly, the *translational energy* component is modelled as a 3D force vector along with a translational acceleration. The result is four descriptors each with three vector components resulting in twelve quantitative flow variables.

In this case, the total number of quantitative flow variables exceeds the number of variables minimally required to model this aspect of the system (12 variables versus one actual degree of freedom). From a computational point of view, the explicit representation of these quantitative flow variables increases the computational solution time required for the complete model. Traditional methods for dynamics modelling such as a virtual work approach as presented in (Shi and McPhee, 2000) combined with the correct selection of generalised coordinates allows a modeller to reduce the number of variables in the problem and produce a more computationally efficient model for solution. Comparing the two approaches, traditional methods require extra work and expertise on the part of the modeller. The method used in this example requires extra work on the part of the computer during model assembly. Additionally, traditional modelling methods do not enable the model flexibility and element reuse feature inherent in the FBBM approach presented here.

A detailed example of developing quantitative flow variables from qualitative energy or material flows is show in Figure 8. In this case, the function is *transfer mechanical energy* which, in vehicle terminology, represents the left front suspension of the vehicle. Qualitatively, the flows are mechanical energy. The quantitative flow variables are specific values that represent accelerations, forces, and moments of this energy flow.

Figure 8 Model elements for left front suspension (*Transfer Mech. E.*)



Another example of selecting quantitative flow variables comes from the four *regulate mechanical energy* functions attached to each suspension corner. These functions represent the functionality of the car's brakes which *regulate* the flow of energy from the spinning wheels to the suspension corners. This flow is controlled by a signal imported from the driver. The *control signal* input from the driver is assigned a single continuous quantitative flow variable C_B. This signal is distributed to the four corners of the car by a *distribute control* function. Each of these distributed flows is given an additional quantitative flow variable called C_{Bi} where *i* is a number that corresponds to each of the four corners (1 – front left, 2 – front right, 3 – rear left, 4 – rear right). Each *regulate mechanical energy* function takes this quantitative flow variable as an input and uses it to regulate the moment and angular acceleration quantitative flow variables from the *import/export solid* function (the car's wheels) to the *transfer mechanical energy* function (the suspension corners).

Proceeding with each function, and the flows between them, the quantitative flow variables are developed in this fashion. For the vehicle example here, quantitative flow variables are listed for all of the flows in the functional model in Table 1. Assumptions used in developing this behavioural model and selecting the associated quantitative flow variables include a non-dynamic engine model, forceless control inputs, and a linear-slider suspension.

Table 1 Quantitative flow variable table

Origin function	*Destination function*	*Qualitative flow*	*Quantitative flow variable*	*Description*
External	Import control	Control	C_B	External braking input
Import control	Distribute control	Control	C_B	Braking control
Distribute control	Regulate Mech. E.	Control	C_{Bi}	Braking control (per wheel)
Regulate Mech. E.	Transfer Mech. E.; import/export solid	Mech. E.	$\vec{M}_{Bi}$	Braking moment
External	Import control	Control	C_S	External steering input
Import control	Distribute control	Control	C_S	Steering control
Distribute control	Transfer Mech. E.	Control	C_{Si}	Steering control (per front wheel)
External	Import control	Control	C_T	External throttle control
Import control	Regulate Rot. E.	Control	C_T	Throttle control
Regulate Rot. E.	Change Rot. E.	Rot. E.	M_E, ω_E	Engine moment, speed
Change Rot. E.	Distribute Rot. E.	Rot. E.	M_T, ω_T	Transmission moment, speed
Distribute Rot. E.	Transfer Mech. E.	Rot. E.	M_{Di}, ω_{Di}	Half-shaft moment, speed
Distribute Mech. E.	Measure Mech. E.	Mech. E.	$\vec{\alpha}_F, \vec{a}_F$	Acceleration of frame
Measure Mech. E.	External	Status	$\vec{v}_F, \vec{\omega}_F$	Velocity of frame
Measure Mech. E.	External	Status	$\vec{q}_F, \vec{r}_F$	Orientation, position of frame
Distribute Mech. E.	Transfer Mech. E.	Mech. E.	$\vec{\alpha}_F, \vec{a}_F$	Acceleration of frame
Distribute Mech. E.	Transfer Mech. E.	Mech. E.	$\vec{F}_{Si}, \vec{M}_{Si}$	Suspension reactions

Table 1 Quantitative flow variable table (continued)

Origin function	*Destination function*	*Qualitative flow*	*Quantitative flow variable*	*Description*
Distribute Mech. E.	Import/export ground	Mech. E.	$\dot{F}_{Ki}$	Damper force
Transfer Mech. E.	Import/export ground	Mech. E.	$\vec{\alpha}_{Wi}, \vec{a}_{Wi}$	Wheel acceleration
Transfer Mech. E.	Import/export ground	Mech. E.	$\vec{F}_{Wi}, \vec{M}_{Wi}$	Wheel reactions
External	Import/export ground	Mech. E.	$\vec{F}_{Gi}, \vec{M}_{Gi}$	Ground forces
External	Import/export ground	Solid	$\hat{u}_{Gi}, \vec{r}_{Gi}, \vec{v}_{Gi}$	Ground orientation, position, velocity
External	Import Pneum. E.	Pneum. E.	v_A	Forward wind speed
Import Pneum. E.	Convert Pneum. E. to Mech. E.	Mech. E.	$\vec{F}_A$	Aerodynamic force

4.3 Model identification

Once quantitative flow variables are identified for each function, behavioural model elements are developed that relate the input and output quantitative flow variables for each function. The objective of the model is to create a dynamic simulation of the car. As a result, most of the models are 3D kinematic constraints and 3D dynamic equations of motion.

For some functions, specific component solutions are known and constrained. These constrained elements of the system solution are used to assist the model development task. For example, it is assumed that a rigid frame with four corners would be the solution used, so a rigid-body model with four suspension attachments is modelled. Of note, a benefit of using the functional representation is that a compliant frame model could be developed later and substituted for the rigid frame model used here without altering any other aspects of the model (the input and output flows are the same). Thus, a designer can explore the impact of frame compliance on vehicle handling and suspension design.

For some functions, such as those representing the suspension, the specific form solution is not known in this example and multiple component solutions could be used. As a general abstraction here, a simple linear slider is used to model any potentially selected suspension configuration. However, if the impact of suspension solution on performance is to be investigated, additional models can be made to represent other suspensions types (short-long arm, strut, solid-axle, etc.). These model elements, if made based on the same functional decomposition and quantitative flow variables, will have the same inputs and outputs. Thus, these specific form solution concepts can be swapped in and out of the complete system model.

Figure 8 above provides a graphical explanation of the connection of behavioural models to the quantitative flow variables. As is illustrated, the functional behavioural model relates the input and output quantitative flow variables through a set of equations. These function level equations are then combined together to create the overall system level model.

Continuing the explanation of the development of function level behavioural models, Table 2 contains the models for the *convert pneumatic energy to mechanical energy* function. This function represents the aerodynamic aspects of the car. The input to the model is the forward speed of the car. The outputs are forces in the downward and backward directions. These forces represent a simplified aerodynamic model using lift and drag coefficients.

Table 2 Convert pneumatic energy to mechanical energy models

Function: Convert Pneum. E. to Mech. E.	*Models*
Solution: Body	$F_{AX} + \frac{1}{2} C_D A_B \sigma_A {v_A}^2 = 0$ $F_{AZ} - \frac{1}{2} C_L A_B \sigma_A {v_A}^2 = 0$

Table 3 contains the powertrain models for the car. The *regulate rotational energy* function outputs the torque produced by the engine based on throttle control and engine speed inputs. This power train model is implemented as a lookup table for torque as a function of throttle and speed. The *change rotational energy* function relates the moment and speed of the engine to the output of the transmission. The models for the *distribute mechanical energy* function relate the moment and speed at the two rear wheels to the moment and speed coming out of the transmission. For these functions, component solutions are known due to regulations of FSAE rules. The car must have an internal combustion engine and as a result requires some kind of transmission.

Table 3 Powertrain models

Function: Regulate Rot. E.	*Models*
Solution: Engine	$M_E - f(C_T, \omega_E) = 0$
Function: Change Rot. E.	*Models*
Solution: Engine	$M_T - G \cdot M_E = 0$ $\omega_E - G \cdot \omega_T = 0$
Function: Distribute Rot. E.	*Models*
Solution: Open differential	$M_{Di} - \frac{M_T}{2} = 0$ $\omega_T - \frac{\omega_{D1} + \omega_{D2}}{2} = 0$

Table 4 contains the dynamic models for the car. The models for the *distribute mechanical energy* function relate the forces, moments, and accelerations (both translational and rotational) of the frame and four suspension corners. The *transfer mechanical energy* models relate the forces, moments, and accelerations of the wheels to that of the suspension. Additionally, these models constrain the suspension to move along the axis of the slider and constrain the wheel to spin about its axis. The models for the *import/export ground* function relate the forces, moments, and accelerations of the wheel to the forces generated at the contact patch of the tire. The tire force is found using a Pacejka '96 tire model (Pacejka, 2005) that includes longitudinal and lateral forces and a linear spring/damper model for the normal component.

Table 4 Dynamic models

Function: Distribute Mech. E.	*Models*
Solution: Frame	$\sum_{i=1}^{4}\left(\vec{F}_{Si}+\vec{F}_{Ki}\right)+\vec{F}_A+m_F\vec{g}-m\vec{a}_F=\vec{0}$
	$\sum_{i=1}^{4}\left(\vec{r}_{F\to Si}\times\vec{F}_{Si}+\vec{r}_{F\to Ki}\times\vec{F}_{Ki}+\vec{M}_{Si}\right)-[I_F]\cdot\vec{\alpha}_F-\vec{\omega}_F\times[I_F]\vec{\omega}_F=\vec{0}$
Function: Transfer Mech. E.	*Models*
Solution: Suspension	$\vec{a}_{Si}-\vec{a}_F-\left(\vec{a}_F\times\vec{r}_{cm\to Si}\right)-\vec{\omega}_F\times\left(\vec{\omega}_F\times\vec{r}_{cm\to Si}\right)=\vec{0}$
	$\vec{\alpha}_{Si}-\vec{\alpha}_F=0$
	$-\vec{F}_{Si}+\vec{F}_{Wi}=\vec{0}$
	$-\vec{M}_{Si}+\vec{r}_{Si\to Wi}\times\vec{F}_{Wi}+\vec{M}_{Wi}-\vec{M}_{Bi}=\vec{0}$
	$\vec{a}_{Si}-\vec{a}_{Wi}+\left(\vec{a}_{Si}\times\vec{r}_{Si\to Wi}\right)+\vec{\omega}_{Si}\times\left(\vec{\omega}_{Si}\times\vec{r}_{Si\to Wi}\right)+2\left(\vec{\omega}_{Si}\times\vec{v}_{Si\to Wi}\right)=\vec{0}$
	$\left(\vec{\alpha}_{Wi}-\vec{\alpha}_{Si}\right)\times\hat{u}_{Wi}=\vec{0}$
	$\vec{M}_{Wi}\bullet\hat{u}_{Wi}=\vec{0}$
Function: Import/export ground	*Models*
Solution: Wheel	$-\vec{F}_{Wi}-\vec{F}_{Ki}+\vec{F}_{Gi}+m_{Wi}\cdot\vec{g}-m\cdot\vec{a}_{Wi}=\vec{0}$
	$\vec{r}_{Wi\to Gi}\times\vec{F}_{Gi}+\vec{M}_{Wi}+\vec{M}_{Bi}+\vec{M}_{Di}-[I_W]\cdot\vec{\alpha}_W-\vec{\omega}_W\times[I_W]\vec{\omega}_W=\vec{0}$
	$\vec{F}_{Gi}-f\left(\hat{n}_{Gi},\vec{r}_{Wi\to Gi},\vec{v}_{Wi\to Gi}\right)=\vec{0}$

This set of models formed the most significant portion of the dynamic simulation for the car. There were several additional models used for the other functions in the system model. These models, along with an additional set of quantitative flow variables are not included here for brevity.

4.4 Model solution

The resulting model of the vehicle is a set of DAEs that relate forces and moments to accelerations. Additionally, kinematic constraints were formulated to be linear systems of equations as well through differentiating them to produce linear acceleration constraints. This method of solution is called index reduction. The solution method converts the system models from a set of DAEs with an index of 2 to a set of ODEs. There are known methods for solving DAEs that produce a more accurate solution than index reduction (Ascher and Petzold, 1998). However, our objective was to use the model solution as part of a real time simulator on a specific computing platform. Thus, speed of solution was more important than numerical accuracy.

Some of the other models, such as the aerodynamic models, are non-linear algebraic equations but are not functions of the highest-order derivatives (in this case, the accelerations). As a result, the models for the car can be solved by aggregating the models into one large linear system of equations which are then solved as a set of ODEs. Terms using lower-order derivatives (positions and velocities) are calculated based on initial values. The accelerations, forces, and moments are then solved using Gaussian elimination. The accelerations are then integrated to find the lower-order derivatives using a 2nd order integrator. Quaternions are used to represent the rotation of the frame; single scalar values are used to represent the rotations of each wheel and the translations of each suspension slider. Both choices reduce the accumulation of error over time and ensure that the mechanisms stay properly configured despite the round off and truncation errors that result from the numerical solution and integration. This solution approach is chosen due to its fixed solution time during each time step (no iteration) and long-term stability (both required aspects of a real time simulation) rather than absolute numerical accuracy.

4.5 Model results

The dynamic model resulting from the FBBM method was used as the basis for creating a simulation programme for the racecar. The dynamic model was integrated with a simulated racetrack and visualisation system. Driver input to the simulation was taken from a driving wheel and set of pedals intended for video game applications.

Expert opinion was used as a qualitative verification of model validity. Experienced FSAE drivers drove the simulated car and compared it to actual driving experience. In these tests, as many known parameters as possible were taken from an existing FSAE car. The dynamic model stayed constrained and solvable for the majority of testing. Model breakdown occurred if a driver managed to flip the car upside down or drive off the pre-defined racecourse.

The objective of the modelling process was successfully completed. A complete, solvable and highly useful full-vehicle behavioural model was created based on the vehicle's functional model. The qualitative functional model flows are extended to quantitative flow variables. Each individual function is modelled to operate on the flows input and output of that function. These functions are then combined into an overall system model that simulates vehicle dynamics.

A screen capture from the simulation along with plots of the acceleration flow variables and ground speed during a simulation run appear in Figures 9, 10 and 11.

Figure 9 Simulation programme screen capture (see online version for colours)

Figure 10 Lateral/longitudinal GG scatter plot (see online version for colours)

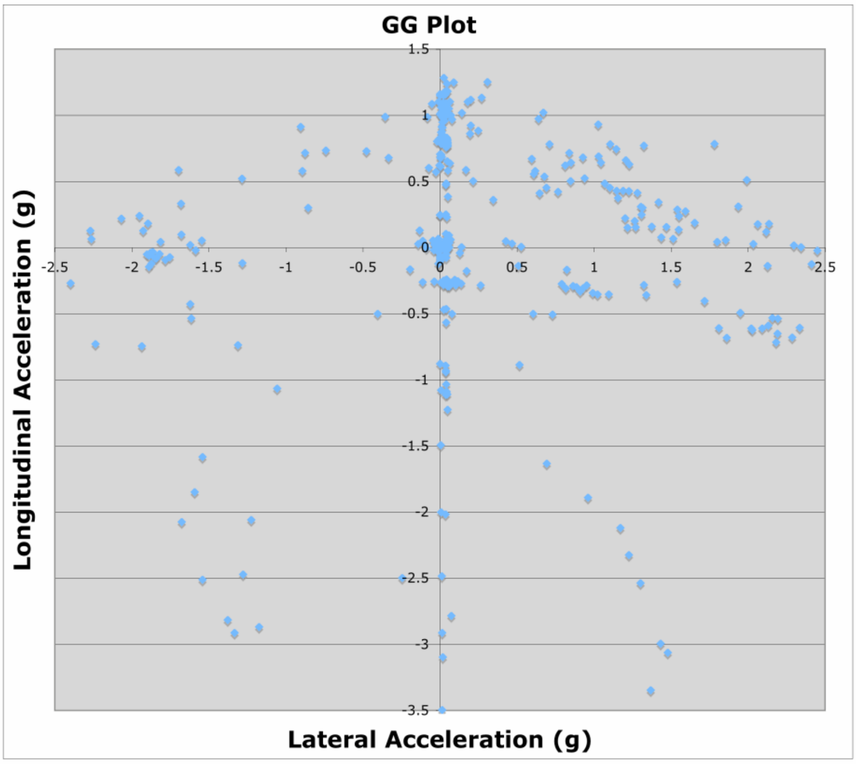
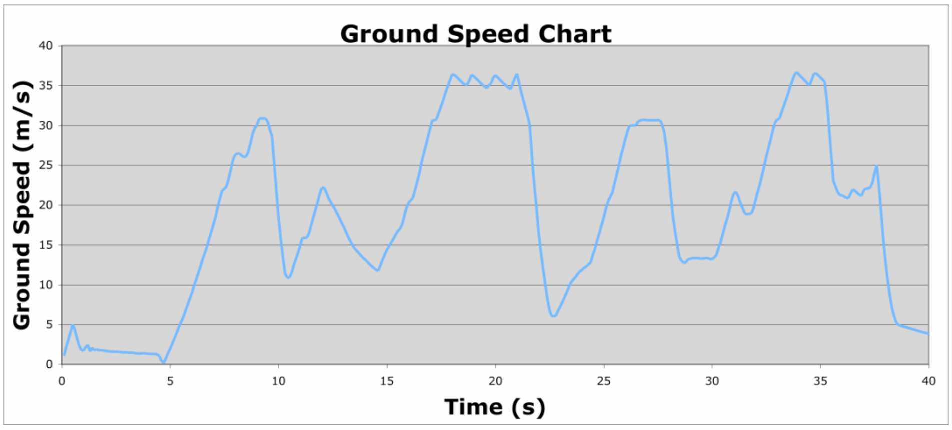


Figure 11 Speed versus time plot (see online version for colours)



5 Conclusions and future work

The objective of this research was to create a method for developing a complete behavioural model for a system based on its functional model. Specific benefits of the work include the assistance of concept selection through the use of a functional model as a unifying starting point for concept identification and evaluation and the natural extension of qualitative functional model flows to quantitative flow variables. Based on the functional system representation and flow connectivity, the individual function level behaviour models integrate smoothly into an overall system level simulation model.

The framework developed provides groundwork for the improved storage and reuse of behavioural model elements by association of behavioural models with individual functionality and integration of behavioural modelling decomposition with functional decomposition. Additionally, the method facilitates model element fidelity swapping and the storage of modelling knowledge through consistent flow representation.

The proposed method begins with functional modelling and includes quantitative flow variable identification, model element identification, and model solution along with internal iterations between these steps. In the functional modelling step, the overall functionality and flows are identified along with a complete description of the system's detailed functionality. During the quantitative flow variable identification step, potential model types are investigated and quantitative flow variables are identified based on the flows in the functional model. Next, behavioural model elements are identified for each function in the functional model. The boundaries of these model elements correspond to those of the functions to which they are associated. The models themselves relate the input and output flow variables as identified from the flows in the functional model. Once model elements have been identified for each function in the functional model, the model elements are combined to produce a complete, solvable model. The model is then solved using an appropriate solution method. Iterations within the method are performed until the functional model, behavioural model elements, and the complete model accomplish the overall objective for the modelling process. A complete example of the approach applied to a Formula SAE racecar is shown. This example results in a successful full-vehicle dynamic simulation model.

This work extends qualitative functional modelling methods into a framework for quantitative behavioural modelling. As such, it provides a continuous representation to compliment the transition from conceptual design, to concept selection, to the initial stages of embodiment design. Though not explored here, and remaining as future work, this continuous framework provides promise to facilitate a coherent requirements flow down from customer needs, to function, to form, to parametric realisation. Nevertheless, the method does not remove the opportunity for the model constructor to exercise their skill and judgement. As developed here, the method shares the iterative nature of engineering model construction in general: models are created, critiqued, and revised. As the designer iterates between the quantitative and qualitative, this modelling enables the designer to better ensure the creation of physically feasible functional representation.

The work presented in this paper offers several additional paths for future research. For certain classes of problems, the creation of a more general behavioural model solver is possible. Existing component-based model solutions could form the basis for such a solver. A general purpose solver could be used to quickly integrate and mathematically evaluate potential concept solutions given a functional model. Because concepts can be mathematically evaluated, optimisation of a concept's performance can also be investigated. It is the ultimate goal of this work to enable such activities. Additionally, work is being planned to replace certain model elements in the racecar dynamic model with higher fidelity counterparts to investigate the specific methods needed to realise model fidelity swapping and model element replacement using this method.

References

Ahmed, S. and Wallace, K. (2003) 'Evaluating a functional basis', *ASME Design Engineering Technical Conference*, Chicago, IL, ASME.

Ascher, U. and Petzold, L. (1998) 'Computer methods for ordinary differential equations and differential-algebraic equations', *SIAM: Society for Industrial and Applied Mathematics*.

Bhatta, S. and Goel, A. (1994) 'Innovations in analogical design: a model-based approach', *Third Annual Conference on Artifical Intelligence in Design (AID-94)*, Lausanne, Switzerland.

Blanchard, B.S. and Fabrycky, W.J. (2006) *Systems Engineering and Analysis*, Prentice-Hall, Upper Saddle River, NJ.

Bracewell, R.H. and Sharpe, J.E.E. (1996) 'Function descriptions used in computer support for qualitative scheme generation – schemebuilder', *AIEDAM*, Vol. 10, pp.333–346.

Caldwell, B.W. and Mocko, G.M. (2008) 'Towards rules for functional composition', *Proceedings of the 2008 ASME Design Engineering Technical Conference*, New York, New York.

Chandrasekaran, B., Goel, A.K. and Iwasaki, Y. (1993) 'Functional representation as design rationale', *Computer*, Vol. 26, pp.48–56.

Cross, M. and Moscardini, A.O. (1985) *Learning the Art of Mathematical Modeling*, Ellis Horwood.

Dieter, G.E. and Schmidt, L.C. (2009) *Engineering Design*, McGraw-Hill, St. Louis.

Fawkes, N.D. and Mahony, J.J. (1994) *An Introduction to Mathematical Modeling*, Wiley.

Gietka, P. and Verma, M. (2002) 'Functional modeling, reverse engineering, and design reuse', *ASME Design Engineering Technical Conference, Design Theory and Methodology Conference*, Montreal, CA.

Giordano, F. (1985) *An Introductory Course in Mathematical Modelling*, Brooks Cole.

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*, Vol. 13, pp.65–82.

Hutcheson, R.S. and McAdams, D.A. (in press) 'A hybrid sensitivity analysis for use in early design', *The ASME Journal of Mechanical Design*.

Karnopp, D.C., Margolis, D.L. and Rosenberg, R.C. (2000) *System Dynamics*, New York, Wiley-Interscience.

Kurtoglu, T. and Campbell, M. (2008) 'Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping', *Journal of Engineering Design*, Vol. 20, pp.83–104.

Laminar Research (2007) 'X-Plane', available at http://www.x-plane.com.

Mechanical Car Simulation Company (2007) 'CarSim', available at http://www.carsim.com.

Miles, L. (1972) *Techniques of Value Analysis Engineering*, McGraw-Hill, New York.

Morgan, G. (2006) *Images of Organization*, Sage Publications, Inc., Thousand Oaks, CA.

Nagel, R.L., Bohm, M.R., Stone, R.B. and McAdams, D.A. (2007) 'A representation of carrier flows for functional design', *International Conference on Engineering Design*, Paris, France.

Nagel, R.L., Hutcheson, R.S., Stone, R., McAdams, D. and Donndelinger, J. (2008) 'Function design framework (FDF): integrated process and function modeling for complex system design', *ASME International Design Engineering Technical Conference*, DETC2008-49369.

Otto, K. and Wood, K. (2001a) *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice-Hall, Upper Saddle River, New Jersey.

Otto, K. and Wood, K. (2001b) *Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development*, Prentice-Hall, New York.

Pacejka, H.B. (2005) *Tire and Vehicle Dynamics*, SAE International.

Pahl, G. and Beitz, W. (1996) *Engineering Design: A Systematic Approach*, Springer-Verlag, London, UK.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.H. (2007) *Engineering Design: A Systematic Approach*, Springer Verlag, London.

Sage, A. and Armstrong, J. (2000) *Introduction to Systems Engineering*, Wiley-Interscience, New York.

Shi, P. and McPhee, J. (2000) 'Dynamics of flexible multibody systems using virtual work and linear graph theory', *Multibody Systems Dynamics*, pp.355–381.

Suh, N. (1998) *The Principles of Design*, Oxford University Press, New York.

The MathWorks Inc. (2007) 'The MathWorks Simulink® – simulation and model-based design', available at http://www.mathworks.com/products/simulink/.

The Modelica Association (2007) 'Modelica and the Modelica Association', available at http://www.modelica.org.

The Society of Automotive Engineers (2007) *SAE International Formula Series*, available at http://students.sae.org/competitions/formulaseries/.

Tomiyama, T., Umeda, Y. and Yoshikawa, H. (1993) 'A CAD of functional design', *Annals of the CIRP*, Vol. 42.

Ullman, D.G. (2002) *The Mechanical Design Process*, 3rd ed., McGraw-Hill, Inc., New York.

Yekula, R., McAdams, D. and Stone, R. (2003) 'Functional and mathematical equivalence of mechanisms: a novel approach to integrating synthesis and design analysis', *Proceedings of DETC2003*, Chicago, IL.