Proceedings of the ASME 2014 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DETC2014-34971

IDENTIFYING AND MAPPING EXCESS RELATIONSHIPS IN COMPLEX ENGINEERED SYSTEMS

Ethan Z. Cansler
Graduate Research Assistant
North Carolina State University
Raleigh, NC 27695 USA

Scott M. Ferguson[1]
Assistant Professor
North Carolina State University
Raleigh, NC 27695 USA

Christopher A. Mattson
Associate Professor
Brigham Young University
Provo, UT 84602 USA

ABSTRACT

The design of complex engineered systems is one of the great challenges currently facing designers. Beyond addressing the obvious difficulties stemming from system complexity, designers must also consider that such systems will likely evolve within their service lifetime. As future environments are often unknown, designers must create systems capable of evolving in-service to meet unforeseen requirements. Previous research exploring the concept of service-phase evolvability has indicated that design excess is a critical factor enabling such change. This paper explores how information available from current techniques in the design literature that focus on system change can be expanded and synthesized to map excess within a component and within a system. Examples are presented where information from High-Definition Design Structure Matrices and functional models are used to complete this mapping. The goal of this paper is to serve as the foundation for quantifying design excess in future work.

1 INTRODUCTION

Complex Engineered Systems (CES) are a mainstay of today's engineering design world. Well-known examples include the F-35 Joint Strike Fighter [1], the Nimitz-class aircraft carrier [2], and the new Boeing 787 Dreamliner [3]. Not only are CES challenging to initially design, they are often expected to have long service lives with durations on the order of decades due to the significant investment that their design and manufacturing costs represent. However, it is unlikely for designers to have foreknowledge of the environment that CES will face over the entire course of their service lifetimes. Consequently, service-phase evolvability, defined as the ability of a system to physically transform from one configuration to a more desirable configuration while in service, is key to the success of CES.

Prior research by Tackett et al. [4] analyzed 210 engineered systems and found that excess in CES is a critical factor in service-phase evolvability. Excess, in the context of this work, is defined as the surplus in a component or system once the necessities of the object have been met. Subsequent research, using two classes of US Navy aircraft carrier as examples, developed mathematical relationships between excess and evolvability [5]. Specifically, evolvability was demonstrated to be a function of two factors: excess and evolvability gain per unit excess. However, selecting appropriate excesses was dependent on the institutional experience reflected in [6,7]. While the importance of excess is currently recognized by the authors' work, no standardized method exists to map relevant excesses throughout a system, and consequently to inform the value of gains per unit excess. As a result, the usefulness of the evolvability metric is only as good as the excess information which it receives. Further, different designers might produce divergent descriptions of a system's excess in the current absence of guidelines. In [5], four system-level excesses were considered: Displacement (LT), Volume (ft^3), Stability (ft), and Electrical Power (MW). These were sourced directly from statements by the US Navy and Department of Defense [6, 7]; they are unquestionably relevant but by no means the only types of excess in an aircraft carrier. This paper explores how excess identification can be generalized for all systems when institutional design experience is not available.

The primary challenge of modeling excess throughout a system is determining which types to consider and how it should be

---

[1] Corresponding Author

represented. To illustrate this challenge, consider a common heat gun as depicted in Figure 1. Though the system in question is orders of magnitude less complex than an aircraft carrier, the exercise of identifying component excesses is no different. A heat gun can be decomposed into a few main subsystems: the body, power cord, fan, resistive heating coils, and controller.

Figure 1: Heat Gun

Considering the purposes that the body fulfills, it could be recognized to have excess that includes the dimensions of:

- Volume
- Thermal energy dissipation
- Load tolerance
- Torque tolerance

However, excess in the body sub-system could be expanded to include the:

- Depth of mounting screw holes
- Amount of deflection at steady-state operating temperature
- Mass air flow permitted by the motor vent

Going to an even finer level of detail, the fastening screws could be considered to have excess in the tolerances of their machined threads, or in the thickness of zinc plating used to prevent corrosion. As this example shows, it is generally possible to describe excess in a system to an ever-increasing resolution. However, similar to how a designer can often assume bulk properties for materials instead of accounting for microscale properties, a designer must also be able to consider excess only up to a relevant degree of granularity.

Another challenge of considering excess in the design process is that it is, as a property of defined components, solution-dependent in terms of the designed system. This means that excess cannot be considered in the design process until the system architecture has been specified. However, the purpose of evolvability, and hence excess in a design, is to enable inherently unknown future solutions. Therefore, excess must also be solution-independent of possible future states. This motivates the development of a method capable of synthesizing the architectural knowledge embodied in a detailed design with an abstracted view of the components themselves, thereby producing a way to consider solely the effect of excess on component interactions. The flow of design information embodied by this work is shown in Figure 2 below.

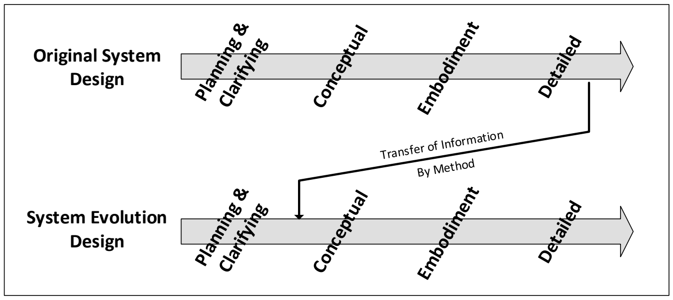


Figure 2: Information Flow when Mapping Excess

This work aims to give designers a tool with which to map and quantify excess within a system. It accomplishes this by a synthesis of elements from existing methodologies in the literature and defining a working 'Excess Basis' to classify sources of excess within a system. With such a tool, designers will be able to objectively consider the amount of excess present in a given design and compare it to the excess present in another design. Additionally, this work explores the idea of what constitutes relevant excess for design consideration.

Though this paper does not explicitly explore a way to define gains per unit excess, it is believed that designers implementing the method will be able to extract beneficial knowledge regarding the behavior of excess in a system, thereby furthering the overall goal of the authors' research to enable designers to treat future evolvability as a design criterion.

2 BACKGROUND

While research has been published with empirically-derived design guidelines to enable future evolvability [7,8], no analytical method exists in the literature with the ability to map or quantify excess in an engineered system. Such a method is viewed by this work as a precursor to an analytical approach to designing directly for evolvability. Despite the lack of an extant analytical framework to treat excess, it is recognized that the concept of evolvability is intrinsically tied to change within a system, a topic that has received significant attention in the literature. A review of works pertaining to change and change propagation found methods with traits that are also applicable to the task of analytical mapping of excess throughout an engineered system.

2.1 CHANGE PROPAGATION

Change propagation has been defined in the literature as "the process by which a change to one part or element of an existing system configuration or design results in one or more additional changes to the system, when those changes would not otherwise have been required" [10]. Change propagation analysis has received increased attention in engineering design research because of the value associated with efficient change management in the design process for complex systems. Similarly, the motivation for evolvability research is based on the belief that systems capable of service phase evolution possess greater value over their lifespan than those that are not [11]. It has been shown by Braha and Bar-Yam that for CES, the success of design tasks, which typically correspond to components, are generally insensitive to random perturbations but are highly sensitive to perturbations targeted at specific tasks [12]. These results imply that certain components within CES are crucial to their overall success; in the context of system evolution, excess within these components will likely be critical to the ability of CES to evolve in service.

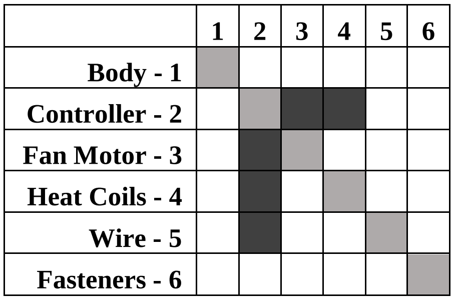
In the literature, Eckert et al. [13] developed four classifications for components with respect to how they interact with change: constants, absorbers, carriers, and multipliers. Constants are unaffected by change and have no interaction with change whatsoever. Absorbers can absorb more changes than they cause and diminish change complexity. Carriers cause and absorb changes in roughly equal measure, and do not affect change complexity. Multipliers cause more changes than they absorb, and increase change complexity.

Since service-phase evolution, by its definition, involves change to a system, it is believed by the authors that the work developed within the change propagation analysis field may also bear utility in the future consideration of evolvability. The Change Prediction Method developed by Clarkson et al. [14] attempts to objectively describe the overall risk of a change affecting components within a system based on combining designer-sourced direct likelihood and impact information. It is anticipated that a similar scheme might help inform the strategic placement of excess throughout a system in future work.

2.2 HIGH-DEFINITION DESIGN STRUCTURE MATRICES

Design Structure Matrices (DSM's) were originally developed to address change propagation in the most direct sense. They have been used for decades in the design field, and are typically used to document where components are interconnected, though they do not traditionally specify the domain or magnitude of the connection [15]. However, DSMs are not limited to component-based change propagation; other works have expanded their use to explore interactions between design tasks encountered in developing CES. Smith and Eppinger [16] demonstrated an extension of DSMs to model design iteration in engineering tasks within large projects, capable of identifying those tasks which pose the greatest impediment to project completion by generation of iterative rework. Yassine and Braha [17] carry the application of DSMs to the modeling of CES design tasks even further, using them to arrange and partition design tasks in such a way as to address four problems in the development of CES: iteration, overlapping, decomposition and integration, and convergence. These works illustrate the versatility of DSMs, as well as their applicability to CES design.

In general for a DSM, a square matrix is used to indicate the dependency relationship between one component and another. Each identified subsystem is numbered and given the corresponding row and column. The matrix is read such that the column indicates the initiating subsystem and the row indicates the dependent subsystem, i.e. the method incorporates the concept of directionality. Since each component clearly affects itself, the diagonal of a DSM bears no useful information and is typically marked out. In the context of flexibility analysis for future evolvability, Tilstra et al. [18] developed the High-Definition Design Structure Matrix (HD-DSM) methodology to account for direct change propagation potential throughout a multi-domain engineered system. The HD-DSM methodology increases the information resolution of a traditional DSM by mapping changes to the domain to which they correspond, e.g. mechanical energy or liquid flow. It accomplishes this by making the DSM three-dimensional, such that each face applies to a particular domain. The domains are largely sourced from the functional basis defined by Hirtz et al. [19] in the context of formalizing functional modeling. Figure 3 shows a sample face of an HD-DSM for a heat gun in the electrical energy domain. The marked blocks (apart from the diagonal) indicate interactions between components in the specified domain. For example, the diagram indicates that Component 4 (the heating coils) interacts with Component 2 (the controller).



	1	**2**	**3**	**4**	**5**	**6**
Body - 1	▨					
Controller - 2		▨	■	■		
Fan Motor - 3		■	▨			
Heat Coils - 4		■		▨		
Wire - 5		■			▨	
Fasteners - 6						▨

Figure 3: Sample HD-DSM Face for Heat Gun in Electrical Energy Domain

An important strength of the HD-DSM method is its ability to track change across multiple domains. As excess across a system may occur in any domain, and is therefore inherently multi-domain, this is a trait which needs to be embodied in the developed method for mapping excess. Further, since an HD-DSM is a matrix, it could receive numerical excess information and thereby systematically record excess, likely adding a bottom row to account for the total excess available from each component.

However, despite the technique's beneficial qualities, HD-DSMs are not directly suitable for this work's purpose. While the information regarding change flow across multiple domains is encoded in an HD-DSM, tracking excess throughout a system would be a daunting task for a designer, as components often convert one type of flow (and hence one type of excess) into another. For example, this is the case with a motor converting electrical energy into mechanical energy. As a result, tracking excess flow throughout a system would require moving between several faces of a matrix rather than being able to track excess using a single field of view.

Prior work by Tackett et al. [5] expressed gains and benefits of excess in terms of the high-level objective functions of the system. In the same manner, this work is mindful of the ultimate goal to relate excess to gains in overall system performance, which will entail the inclusion of dependency relationships. An HD-DSM cannot practically incorporate the dependency relationships inherent in excess interactions between components with respect to overall system objective functions. It would be technically possible to place equations into cells of a spreadsheet to describe excess flow relationships, but performing this task for a system with more than even a small number of components would be laborious, prone to user error, and even in the ideal case, result in a technically correct but very unintuitive model for the designer. As a result of the shortcomings of the HD-DSM methodology for excess modeling, other techniques were explored in the interest of addressing the multi-domain nature of excess and representing excess flows in an intuitive manner.

2.3 FUNCTIONAL MODELS

Functional models are a well-established method in the design field of representing a system in terms of its functions and flows, rather than by the properties of its components. Block diagrams are used where the blocks represent the functions of the system (rather than individual components or subsystems) and the arrows that pass between the blocks are labeled with the flow they represent. This approach enables a designer to step back from a hardware-focused view of individual components and consider the high-level architecture of a system. In an effort to standardize functional model nomenclature, [19] developed a reconciled functional basis that addressed both the functional flows between components and the functional vocabulary pertaining to the individual components' operations.

Considering, as in the case of the heat gun, that excess will often occur in terms of base flows (energy, mass, or signal), it is intrinsically tied to the functional flows of a component. Therefore, pertinent to this work is the set of functional flows described by Hirtz. This is divided between the three base types of flow—signal, energy, and mass—and sub-divided into specific types such as control signal, electrical energy, thermal energy, liquid flow, etc. Absent are geometric parameters relevant to excess, but the involved flows are the same and already standardized in design nomenclature. As a reference, a selected portion of a traditional functional diagram for a heat gun is shown here in Figure 4.

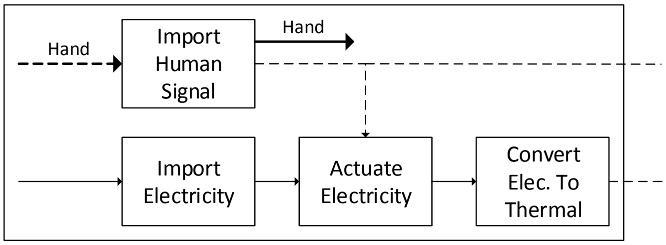


Figure 4: Portion of Heat Gun Functional Diagram

While the flows of excess observed in a system do not always belong to the energy, mass, or signal flow classes denoted in a functional diagram, the method's approach to flows does hold value for this work. Specifically, using arrows to map flows through a block-diagram based representation of a system, coupled with numerical quantification, is considered applicable to modeling excess in a system.

3 THEORY

The preceding section detailed approaches from works in the literature pertaining to change that also held value for treating excess. Namely, this work posits that a multi-domain approach to modeling excess is necessary and that a block diagram based model is a useful degree of abstraction. However, not all excess behavior is equivalent to change; this section details the underlying theory particular to the nature and treatment of excess in this work. This material serves as the foundation of the methodology derived in Section 4.

3.1 TWO CLASSES OF EXCESS

Consideration of the flows within a functional diagram and their implication in the physical domain will quickly suggest that excess can occur in any functional flow, i.e. any component may either produce more of a flow or be capable of receiving more of a flow. However, not all excess is in the form of a flow. A common and relatable example of a non-flow derived excess is volume. Volume is a component property consumed by anything placed within the component. Another common example is the maximum load tolerated by a component. This work posits that all types of excess in a system belong in one of two classes: Storage or Flow. Prior examples from [5] implicitly acknowledged that both types of excess exist, but the nomenclature is formally defined here and fully developed in the next section.

3.2 WORKING EXCESS BASIS

To effectively communicate in terms of system excess, a unified vocabulary is necessary. This work defines a standard 'Excess Basis' to account for all possible types of excess present in an engineered system. In this interest of maintaining consistency with the established literature, the reconciled functional basis flow set developed in [19] was used as the core of the working excess basis and then extended where necessary. This was used because most excess types encountered are either in the form of a functional flow or a stored flow. To describe all excesses, two ‘categorical’ extensions and six ‘type’ extensions were needed.

The first categorical extension was Geometric, with associated type entries of Volume, Area, and Length. These extensions were necessary because, as geometric characteristics may not literally flow between components, there was no provision for them in the extant flow set. However, the ability to describe geometric properties is crucial for characterizing excess. In a heat gun, geometric excess could be described in the screw lengths, the area available in the grip for the operator controls to protrude through, or the volume present in the case for internal components such as the fan or heating coils to occupy.

The second categorical extension was Structural, with associated type entries of Load, Torque, and Pressure. The Structural category was added because structural characteristics were absent from the extant flow set due to their non-flow nature. In the context of excess, however, any designer is familiar with the concept of structural excess, wherein a demand to tolerate a greater force or moment than currently required may be made of an object. As a result, Load and Torque were added to the excess basis. Pressure was also added so that a designer using the method will not have to decompose a pressure excess into a load and/or area excess, but rather may treat objects such as pressure vessels. In a heat gun, structural excess could be described in the weight capacity of the body or the capacity of the body to exert a reaction torque on the fan motor.

Geometric and Structural excess comprise the set of excesses which may exist solely within the Storage class; i.e. neither volume nor load may flow. Inversely, any measured excess which can be classified as a Flow may also exist as a Storage, since anything that flows can also be stored. The word storage was chosen because it concisely represents what is conceptually occurring. Flows, or the ability to produce them, are literally being stored; likewise, geometric considerations imply the ability to contain surplus. The inclusion of structural parameters becomes necessary if one considers an alternative view of forces and moments, in which they cause strain energy to develop within objects. In this interpretation, an object may only develop a certain amount of strain energy before failure; hence, the Structural category may be viewed as a storage of strain energy. The working excess basis is shown in Table 1.

3.3 RESOLUTION OF EXCESS FLOWS

As earlier noted, excess may be described to an almost infinite resolution. However, as with any engineering model, the resolution must be finite. This work posits that a useful strategy for selecting relevant excess flows is similar to the idea of a Taylor Series Expansion (TSE). In the field of mathematics, a TSE is based on the observation that a complex function’s behavior may be locally approximated by a much simpler function with only mildly diminished accuracy, generally of an order of magnitude equal to the first neglected term [20]. Stated another way, a TSE neglects higher-order terms that add complexity but little significant information to a model. This work promotes the same mentality, and this principle is demonstrated in the case study. As a general example, consider an aircraft carrier. Top level excesses such as displacement or power generation are unquestionably relevant to the overall excess present in the system. However, the screws affixing speakers to bulkheads are irrelevant in terms of system excess, as they do not directly contribute to the primary functions of the system, are insignificant to replace if needed, and most importantly for this method, would constitute an enormous burden to model across the entire system for a designer. As such, this method advocates describing only excesses that directly contribute to the function of a system.

Table 1: Working Excess Basis

Class	*Category*	*Type*	*Abbr.*
Flow or Storage	Signal	Status	S-S
		Control	S-C
	Material	Human	M-H
		Gas	M-G
		Liquid	M-L
		Solid	M-S
		Plasma	M-P
		Mixture	M-M
	Energy	Human	E-H
		Acoustic	E-A
		Biological	E-B
		Chemical	E-C
		Electrical	E-E
		Electromagnetic	E-EM
		Hydraulic	E-Hy
		Magnetic	E-Mag
		Mechanical	E-M
		Pneumatic	E-P
		Radioactive	E-R
		Thermal	E-T
Storage	Geometric	Length	G-1
		Area	G-2
		Volume	G-3
	Structural	Load	S-L
		Torque	S-T
		Pressure	S-P

3.4 NATURE OF EXCESS FLOW

In this work, a flow from one component to another indicates that the supplying component is addressing a requirement of the receiving component. This, as noted in the two class definitions, may or may not comprise a literal flow. Technically, excess is not continually flowing; rather, they are flows in which surplus is possible. However, since the only inter-component flows depicted in this method are those deemed significant from a system excess perspective, it is intuitive and meaningful to use the term 'excess flow.' As a consequence of the requirement-oriented view of excess flows, directionality in the excess domain does not necessarily correspond to that experienced in the physical domain.

In a heat gun, the fan motor consumes excess of the type Storage-Structural-Torque from the body, and the arrow would be drawn pointing to the fan motor. However, in the physical domain the body experiences the reaction torque imposed by the fan motor.

The foregoing discussion leads to a distinction between inbound and outbound flows at the primary block level of a system. Excess which flows into a primary block may be termed "compatibility" excess and that which the component supplies is termed "functional" excess. This is depicted in Figure 5. Described another way, compatibility excess occurs in the inbound flows required by a component. Functional excess occurs in outbound flows that the component supplies to other components. At the level of an engineered system, almost all excess can be viewed as compatibility excess, since each component's functional excess becomes compatibility excess for the next component that it interacts with. Therefore, at a system level, no distinction is drawn between compatibility and functional excess. However, this leads to a significant observation concerning the nature of excess across a system. Namely, excesses agglomerate from lower level components to ultimately produce desired performance characteristics for the entire system.

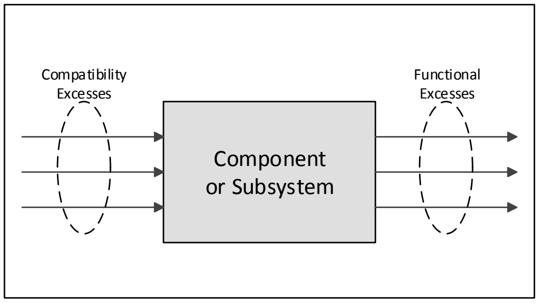


Figure 5: Compatibility and Functional Excess Flows

3.5 COMPONENT CLASSIFICATION SCHEME

The categorization of components with respect to change propagation developed by Eckert [13] is viewed as a useful mentality for considering excess. This work proposes a component classification scheme similar to that developed by Clarkson consisting of three groups: consumers, carriers, and transformers. The component categories are:

- *Consumers*, which consume more types of excess than they produce. An example would be an electric motor, which consumes electrical energy, volume, load, and torque from its housing, and produces mechanical energy.
- *Carriers*, which consume roughly as many types of excess as they produce. An example would be a power cord which consumes electrical energy and outputs electrical energy.
- *Transformers*, which produce more types of excess than they consume. An example would be a case for handheld equipment which consumes human energy and produces volume, load, and torque excess.

Constant, a classification present in the initial change propagation scheme, was discarded for the purpose of excess analysis as there is never a situation in which a system component, wholly unaffected by excess, would be included in an excess model. This scheme is expected to aid designers in systematically determining the relationship between excess and evolvability gains within a system, and will be further explored in future work.

4 SYNTHESIZED METHOD

Building upon the consideration that excess may be treated similarly to change, this work first considered adapting the HD-DSM method of [18] as they effectively document dependency relationships across multiple domains between subsystems or components. However, for reasons previously documented they are not directly suitable. Hence, the block diagram structure and abstraction techniques from functional diagrams were also utilized, as they better capture in a designer-accessible manner the flows between components in a system.

This section presents guidelines for building a model that maps excess relationships between components within a system. As a result of the considerations described in the motivation and in Section 2, the decision was made to create a type of quantified block diagram model.

In this method, the components are abstracted as blocks with the pertinent flows attached. This allows the inner workings of the component to be treated as a black box. Of note is that this scheme is flexible in terms of system granularity, such that if a component block (which is actually representing a subsystem) needs to be further decomposed, it can be replaced by a new set of blocks with more finely mapped excess flows without requiring rework elsewhere in the model, as demonstrated in Figure 6.

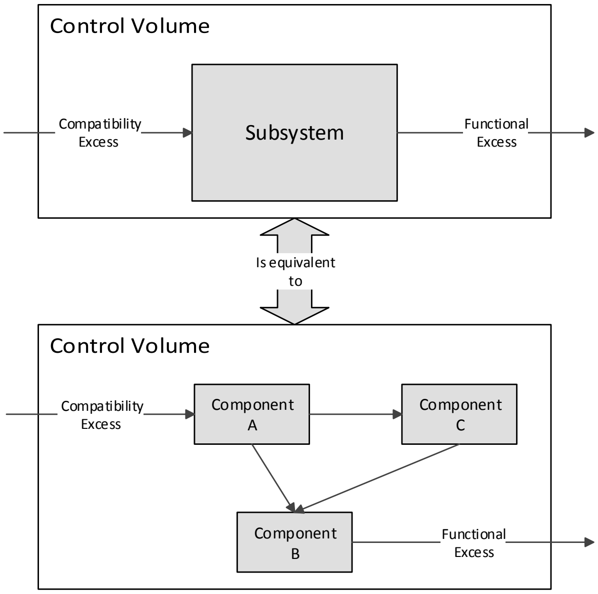


Figure 6: Decomposition Equivalency of Components

4.1 CONSTRUCTING THE MODEL

The model consists mainly of an assembly of three items: components, excess types, and flows. An Environment block is also present to account for the flows originating outside the system's control volume.

The first task for a designer, however, is defining a control volume that identifies system boundaries. This requires defining which flows come from the environment and what output flows are expected from the system. Once the control volume has been defined, the model is populated by a representation of the system constructed from primary blocks (components), excesses and flows, as depicted in Figure 7. The following subsections describe each in detail. Section 5 contains a fully developed example of the method.

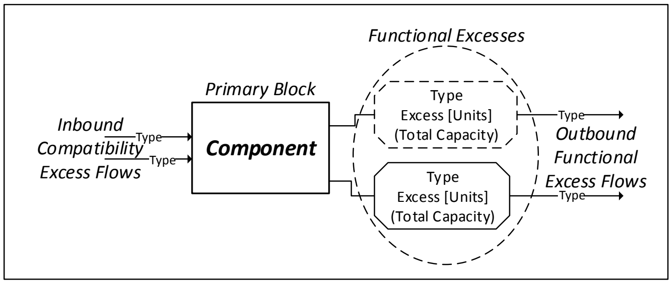


Figure 7: General Architecture

4.2 PRIMARY BLOCKS (COMPONENTS)

Primary blocks may represent either individual components or subsystems. They are comprised of square-edged rectangles and labeled with the name of the abstracted entity. Assuming that the excess flows are mapped correctly, expanded and condensed representations of subsystems are equivalent, as depicted in Figure 6 previously.

A suggested starting point for system decomposition is to use the main subsystems—those components whose functions and flows are relatively self-contained—as the primary blocks and to subsequently decompose them where greater resolution is deemed necessary by the designer. It is impossible to give a set of definitive guidelines for further decomposition, as this choice is inescapably dependent on the designer's discretion as to which components are likely to influence the system's ability to evolve in future.

As a note, there can arise situations in which it can be advantageous to condense components as a subsystem rather than expand them. As an example, consider a component which must be physically placed within a housing composed of two bolted (asymmetrical) halves. With an expanded model, excess volume would flow from each half of the housing and some relationship ensuring that the excesses from each half summed to that required by the housed component would be necessary. However, with a condensed representation of a housing subsystem, only one excess flow would be required.

4.3 EXCESS TYPES

Excess stemming from a component is identified by a snipped-edge rectangle, attached by a line to the originating primary block. The total amount of the parameter in question is to be indicated within the block, along with its units and the remaining excess. An important consideration when implementing this work's mapping methodology is the types of excess to include. Assuming that compatibility excess requirements have been correctly ascribed to components, a definition of minimal sufficiency is that no component has an unconnected compatibility excess point.

In general, it is suggested to confine descriptions of excess to those pertaining to the core functions of the components in question, as these excesses are most likely to have an effect throughout the system. For example, an electric motor's excess power capacity is likely relevant, whereas its excess in mounting hole diameter is likely superfluous.

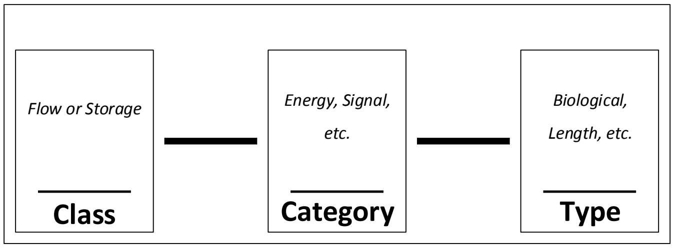


Figure 8: Naming Scheme for Excess Types

4.4 FLOWS

Flow arrows are to originate from an excess block and terminate on the left side of the consuming component block. They are to be labeled along their path with the abbreviation from the excess basis pertaining to the parameter which they represent; this is to aid in the coherency of complex models, and the hyphens may be omitted for brevity. The numerical quantity which they represent is to be shown at both the origin and terminus. Additionally, the units of excess are to be indicated at the terminus.

Flows from the environment block are also to be labeled with the maximum possible value, if one is known and/or applicable, as a feasibility check for the designer. In general, once all of the functional excess flows have been mapped to the appropriate components, the remaining required compatibility flows should reveal where flows originating from the Environment are necessary. If a compatibility excess requirement remains that the environment is known to be unable to address, the functional excesses of the components must be reevaluated to ensure correctness.

5 CASE STUDY

As a practical example of the method, the heat gun working example was used. It represents a moderately complex system with enough types of excess interaction to illustrate the points of this work's methodology. Two models of a heat gun are presented.

5.1 HEAT GUN BASE MODEL

The first, Figure 9, is of the heat gun decomposed into its primary subsystems. It demonstrates the generic application of the methodology to a system. The critical flows between components are captured and quantified, with upper bounds shown. In this system, all limits regard the upper range of variables, but there is no impediment to using the method to denote lower bounds, or ranges, as well. The heat gun in question is assumed to be a basic model in the 1 kW range with an on/off toggle switch, designed to produce a flow of hot air at 550°C based on the assumption of a 20°C operating environment; the calculated air flow values are based on those figures. The model was developed as follows:

5.1.1 DEFINING THE CONTROL VOLUME

The physical extent of the heat gun was considered to define the control volume. Specifically, any energy or signal passing into or out of the device, whether electrical energy from the power grid, human energy from the operator, or heat from the nozzle, was viewed as crossing the control volume.

5.1.2 DEFINING COMPONENTS

The primary subsystems, and hence component blocks in the model, along with their compatibility excess requirements, were determined to be:

- Case
    - Airflow (F-M-G)
    - Human energy (F-E-H)
- Cord
    - Electrical energy (F-E-E)
- Controller
    - Electrical energy (F-E-E)
    - Signal (F-S-C)
    - Volume (S-G-3)
- Fan
    - Load (S-S-L)
    - Reaction torque (S-S-T)
    - Electrical energy (F-E-E)
    - Airflow (F-M-G)
    - Volume (S-G-3)
- Heating coils
    - Load (S-S-L)
    - Electrical energy (F-E-E)
    - Airflow (F-M-G)
    - Volume (S-G-3)
- Nozzle
    - Load (S-S-L)
    - Thermal energy (F-E-T)

This description of the system was arrived at by considering the components which would be evident if the device were disassembled, i.e. the self-contained subsystems which exhibit modular behavior.

5.1.3 DEFINING EXCESS TYPES

Consideration of the purposes of each component resulted in the following breakdown of functional excess types:

- Case
    - Load (S-S-L)
    - Torque (S-S-T)
    - Airflow (F-M-G)
    - Volume (S-G-3)
- Cord
    - Electrical energy (F-E-E)
- Controller
    - Electrical energy (F-E-E)
- Fan
    - Airflow (F-M-G)
- Heating coils
    - Thermal energy (F-E-T)
- Nozzle
    - Thermal energy (F-E-T)

Designer discretion, coupled with a consideration of the core function of the components, was required when determining how to define the functional excesses present, as well as choose the most meaningful associated units. For example, the nozzle could technically provide an impediment to the airflow. However, it is unlikely that a change to another component would so significantly increase the required airflow as to necessitate a nozzle redesign; the airflow required to maintain the design temperature would scale linearly with current, meaning that at most a 50% increase in airflow is possible based on the environment limitations. More likely is that the desired output temperature could be increased (meaning airflow would remain the same or decrease) above a temperature that the nozzle could tolerate. Therefore, thermal energy with units of degrees Kelvin was chosen to describe the nozzle's functional excess. Alternatively, the unit of Joule could have been used without sacrificing accuracy, but the informational content would have had less value to a designer.

5.1.4 DEFINING AND QUANTIFYING FLOWS

The process of mapping excess flows between components requires knowledge of the system architecture and matching the functional excess outputs to the correct compatibility excess inputs. This is also the step in which the Environment block was defined. For this system, the required environmental flows were human energy (F-E-H) to support the heat gun, control signal provided by a human hand (F-S-C) to actuate the control switch, electrical energy (F-E-E) from the power grid (assumed to be 110V), and airflow (F-M-G). The flow limitations, where known or applicable, were ascribed to the flow arrows. In this case, it was assumed that (i) a human could reasonably exert 50N of force on the heat gun, (ii) the electrical supply was governed by a common 15A circuit breaker, and (iii) airflow from the environment was unlimited for the magnitude of airflow required by a heat gun.

Next, the excess blocks and flows were quantified to demonstrate the method's incorporation of numerical information. As the heat gun in question is to represent a general, commercially-available model, characteristics such as the amperage draw and output temperature were assumed to hold their typical values; most commercially available consumer grade heat guns output air at roughly 500°C and draw between 1 and 1.2 kW of power. This information was used to calculate thermodynamic properties of intermediate flows such as the mass of hair which could be heated.

As a last step, any outbound flows crossing the Control Volume were identified. The only such flow in this case was a flow of thermal energy passing from the nozzle to the environment.

5.2 HEAT GUN EXPANDED MODEL

The second model illustrates the envisioned usage of the method when a particular type of future evolution is identified as likely. Figure 10 is the same model, but with further decomposition. The decomposition is of the Case and Controller subsystems, based on the assumption that a possible evolution to the system is upgrading the heat gun controller from the binary on/off capability of the original model to one capable of adjusting the fan speed and power output in real time to maintain constant temperature.

Therefore, the Body is decomposed into Barrel and Grip, while the Controller is decomposed into Switch and Control Circuit. In the case of Barrel/Grip, no new types of excess are assigned; the decomposition is intended solely to increase the resolution of where volumetric excess exists within the body. However, in the case of Switch/Control Circuit, a new form of excess is documented. It is a flow of Binary Channel control signal with zero available excess from the Switch to the Control Circuit, and represents the control signal passed from the Switch to the Control Circuit within the Controller subsystem.

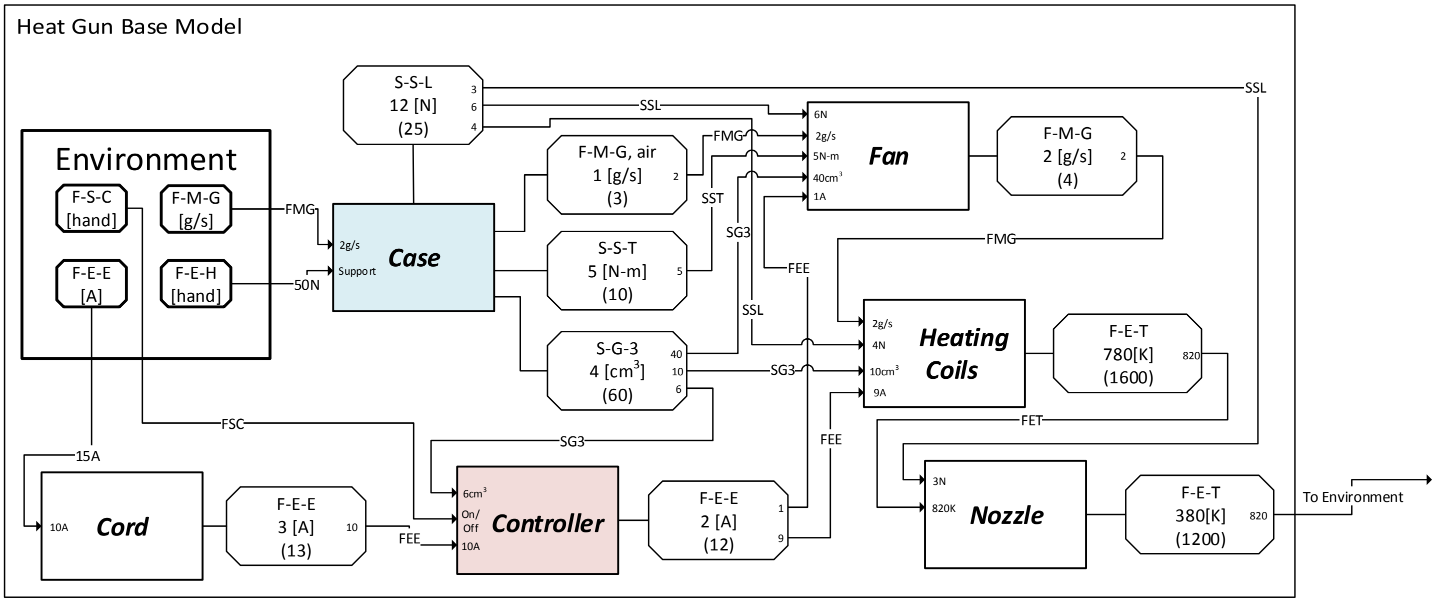


Figure 9: Base Heat Gun Excess Model

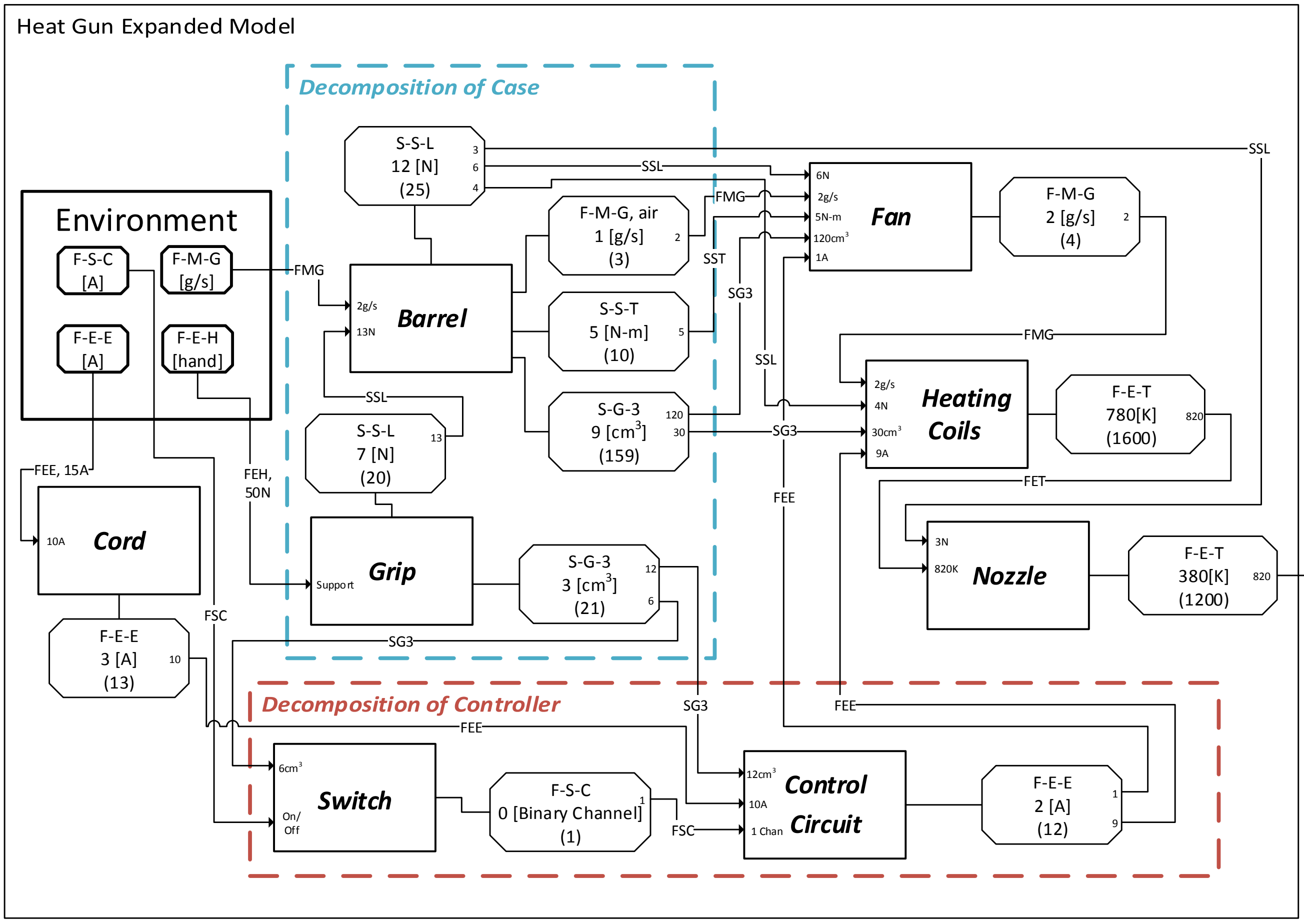


Figure 10: Expanded Heat Gun Model

5.3 DISCUSSION ON MODELS

The model developed demonstrates use of the Taylor Series Expansion analogy principle described in Section 3.3. For example, Load excess flows are described for the motor and heating coils, but not for the controller. This is because the weight of the controller is known to be relatively insignificant, and therefore unnecessary to model since it would increase complexity but not add any useful information. Similarly, while it is known that there are screws holding components of the system in place, each possessing ability to carry excess load, they are not modeled, nor are the excesses present in the holes with which the screws interface. Modeling such parameters would not be invalid, but they would constitute superfluous excess in terms of the flows which are critical to describe the system. While this avoidance of unnecessary complexity may not seem important for this system, consider the method applied to a highly complex system with many inter-component interactions; such reductions in model complexity will aid in the designer's navigation of the model.

Figure 10 demonstrates the ability of the method to accommodate locally increased resolution without affecting the remainder of the system map. As noted previously, this expansion was based on the idea that a possible future evolution of the heat gun is an upgrade of the controller. As a result, the mapping resolution was increased for the body, which contains the controller, and for the controller itself.

Evolution scenario 1: Controller upgrade

The model presented in Figure 10 can be used to explore the ability of a system to receive an enhanced controller. Specifically, the enhanced controller is defined as one which allows the user to specify a desired output temperature, receives sensor input from a thermocouple in the nozzle, and adjusts in real time the power given to the heating coils in order to maintain the user-specified temperature.

First, the existing controller must be examined to determine if it may be evolved to meet the upgraded specifications. Since the switch has zero excess in its control signal flow output, it would have to be replaced entirely to meet the requirement that the user select a temperature. Likewise, since the control circuit is shown to receive only binary signals, and further has no regulatory ability documented, it would also have to be replaced entirely. The model therefore shows that, if a new switch and control circuit are to be placed in the grip, they may occupy a maximum of 21 cubic centimeters.

It is also shown that, in order to maintain the functional excess flows already documented, the new control circuit must be capable of passing at least 10 amperes of current. It is not viewed as necessary to make provision for a thermocouple wire within the model, as it would require negligible volume and be an integral part of the proposed new controller subsystem in terms of its signal flow.

Evolution scenario 2: Increase output temperature

Another example of the value provided by the model in Figure 10 is the ability to consider an evolution that increases the output temperature of the heat gun. It can be seen that, if an output temperature of 1000°C (1273K) were desired, the heating coils would accommodate the change, but the nozzle would have to be replaced. Also, the calculations which informed the electrical flows and airflow through the system would have to be revisited by the designer as well; an increase in temperature to 1000°C would represent a roughly 50% increase in the heat addition to the air. This implies that the mass air flow could either decrease by roughly one third, or that the amperage consumption could increase by roughly one half, or some intermediate shift could be effected in both properties. For any considered modification to those flows, the available excesses would have to be queried to ensure feasibility.

6 CONCLUSIONS AND FUTURE WORK

This work defines a framework with which to map excess in an engineered system via a synthesis of two methodologies from the literature: HD-DSM's and functional diagrams. The resulting method uses block diagrams coupled with quantified flows, and defines two component-level forms of excess: functional and compatibility. Functional excess occurs in parameters in which the component is capable of supplying excess, and compatibility excess occurs in the parameters which the component requires externally. Components are identified for inclusion in the mapping based firstly on their relevance to primary system functions and secondly on particular anticipated relevance to future system evolution, if applicable. Inter-component excess flows are labeled according to the working excess basis, which identifies excesses based on their Class, Category, and Type. Overall, the excess basis represents an extension to the extant functional modeling flow set in order to encompass all possible types of excess interactions within a system. It accomplishes this by adding a Storage class, which addresses that all flows may be stored in a system and that geometric and structural excess considerations may also be treated as a form of storage. This basis may be further extended in future if designers desire finer description. Based on the type of excess represented in a flow, an 'excess block' is attached to the component block and four pieces of primary information are recorded: the three-character identification of excess, its quantity, its units, and the nominal total capacity of the component.

With this approach, system information made available within the detailed design phase can be selectively transferred to a conceptual-phase design in terms of future evolvability. In general, the maps generated by this method offer insight into the governing flow relationships between the system components, thereby allowing designers to determine if a system is likely capable of a particular evolution. Importantly, this method makes such determinations more readily possible than would be possible if designers were required to consult the detailed system models for every posited evolution. This application of excess maps is the immediate use delivered by this paper, but is not the ultimate goal of the method. When coupled with future work of determining gains per unit excess, it is expected for excess maps to become the basis of predicting system evolvability.

The next direction of intended research is the development of metrics to analytically describe the excess present in a system. It is expected that continued development of excess modeling will result in the ultimate ability to determine the gains per unit excess required by the prior work of [5]. Another potentially attractive direction for application of the method to systems of significant complexity is the development of a software-based GUI which would allow for multiple-level representations of a system, i.e. where a primary block representing a subsystem may be opened to reveal the constituent components with specifically mapped excess relationships.

ACKNOWLEDGMENTS

The authors would like to recognize the National Science Foundation Grant CMMI-1301247 for funding this research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] United States Department of Defense. "F-35 Lighting II Program," http://www.jsf.mil.

[2] Newport News Shipbuilding. "Gerald R. Ford (CVN 78) – The Ford-class," http://www.thefordclass.com.

[3] Boeing. "Boeing: 787," http://www.boeing.com/boeing/commercial/787family/.

[4] Tackett, M. W. P., 2013. "A mathematical model for quantifying system evolvability using excess and modularity". Master's thesis, Brigham Young University, May. http://hdl.lib.byu.edu/1877/etd6180.

[5] Tackett, M. W. P., 2014. "A Model for Quantifying System Evolvability Based on Excess and Capacity," *Journal of Mechanical Design, accepted.*

[6] Greenert, J. W., 2012. "Payloads over platforms: Charting a new course". *U.S. Naval Institute Proceedings*, **138**(7): pp. 16–23.

[7] Beesemyer, J. C., Julcoly, D. O., Ross, A. M., Rhodes, D. H., 2011. “Developing Methods to Design for Evolvability: Research Approach and Preliminary Design Principles,” *9th Conference on Systems Engineering Research*, Los Angeles, CA.

[8] Keese, D., Tilstra, A. H., Seepersad, C. C., Wood, K. L., 2007. “Empirically-Derived Principles for Designing Products with Flexibility for Future Evolution,” *ASME Design Engineering Technical Conferences*, Las Vegas, NV, DETC2007-35695.

[9] Howard, W., 2002. Future of the aircraft carrier. Tech. rep., Defense Science Board Task Force, Accession Number ADA- 408129.

[10] Giffin, M., de Weck, O., Bounova, G., Keller, R., Eckert, C., & Clarkson, P. J., 2009. “Change propagation analysis in complex technical systems.” *Journal of Mechanical Design*. **131:** 081001.

[11] Bloebaum, C. L., McGowan, A. R., Lightfoot, M. C., Collopy, P. D., and Hazelrigg, G., 2012. “NSF/NASA workshop on the design of large-scale complex engineered systems – from research to product realization,” In Proceedings of the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, IN, AIAA-2012-5572.

[12] Braha, D. and Y. Bar-Yam. The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. Management Science. Vol. 53 (7). pp. 1127-1145. 2007.

[13] Eckert, C., Clarkson, P. J., Zanker, W., 2004. “Change and customization in complex engineering domains,” Research in Engineering Design. **15**(1): pp. 1-21.

[14] Clarkson, P. J., Simons, C., Eckert, C., 2004. “Predicting Change Propagation in Complex Design,” *Journal of Mechanical Engineering*. **126**(5): pp. 788-797

[15] Browning, T. R., Co, L. M. A., and Worth, F., 2001. “Applying the design structure matrix to system decomposition and integration problems: A review and new directions.” *IEEE Transactions on Engineering Management*. **48**(3): pp. 14.

[16] Smith, R.P., Eppinger, S.D. Identifying controlling features of engineering design iteration. Management Science, 43 (3), pp. 276-293. 1997.

[17] Yassine, A. and Braha, D. Complex Concurrent Engineering and the Design Structure Matrix Approach. Concurrent Engineering: Research and Applications. Vol. 11 (3). 165-177. 2003.

[18] Tilstra, A. H., Seepersad, C. C., and Wood, K. L., 2009. “Analysis of product flexibility for future evolution based on design guidelines and a high-definition design structure matrix”. In Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, CA, DETC2009-87118.

[19] Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L., 2002. “A functional basis for engineering design: Reconciling and evolving previous efforts,” Research in Engineering Design. **13**(2): pp. 65-82.

[20] Stewart, J., 2004. *Calculus: Concepts and Contexts*. Belmont, CA: Thomson Brooks/Cole