Computer Support for Preliminary Concept Completion & Evaluation/Analysis of Design Concepts

by

Maryam Khorshidi

A Thesis Presented in Partial Fulfillment

of the Requirements for the Degree

Master of Science

Approved November 2013 by the

Graduate Supervisory Committee:

Jami Shah, Chair

Teresa Wu

Esma Gel

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

Creative design lies at the intersection of novelty and technical feasibility. These objectives can be achieved through cycles of divergence (idea generation) and convergence (idea evaluation) in conceptual design. The focus of this thesis is on the latter aspect. The evaluation may involve any aspect of technical feasibility and may be desired at component, sub-system or full system level. Two issues that are considered in this work are:

1. Information about design ideas is incomplete, informal and sketchy
2. Designers often work at multiple levels; different aspects or subsystems may be at different levels of abstraction

Thus, high fidelity analysis and simulation tools are not appropriate for this purpose. This thesis looks at the requirements for a simulation tool and how it could facilitate concept evaluation. The specific tasks reported in this thesis are:

1. The typical types of information available after an ideation session
2. The typical types of technical evaluations done in early stages
3. How to conduct low fidelity design evaluation given a well-defined feasibility question

A computational tool for supporting idea evaluation was designed and implemented. It was assumed that the results of the ideation session are represented as a morphological chart and each entry is expressed as some combination of a sketch, text and references to physical effects and machine components. Approximately 110 physical effects were identified and represented in terms of algebraic equations, physical variables and a

textual description. A common ontology of physical variables was created so that physical effects could be networked together when variables are shared. This allows users

to synthesize complex behaviors from simple ones, without assuming any solution sequence. A library of 16 machine elements was also created and users were given instructions about incorporating them.

To support quick analysis, differential equations are transformed to algebraic equations by replacing differential terms with steady state differences), only steady state behavior is considered and interval arithmetic was used for modeling. The tool implementation is done by MATLAB; and a number of case studies are also done to show how the tool works.

textual description. A common ontology of physical variables was created so that physical effects could be networked together when variables are shared. This allows users to synthesize complex behaviors from simple ones, without assuming any solution sequence. A library of 15 machine elements was also created and users were given instructions about incorporating them.

To support quick analysis, differential equations are transformed to algebraic equations by replacing differential terms with steady state differences), only steady state behavior is considered and interval arithmetic was used for modeling. The tool implementation is done by MATLAB; and a number of case studies are also done to show how the tool works.

ACKNOWLEDGMENTS

I would like to sincerely thank Dr. Jami Shah for being my advisor during my Master's degree. It was a great experience working in his prestigious research group and I really appreciate his continued encouragement to explore new paths and his invaluable insights in research.

I would also like to thank Dr. Esma Gel and Dr. Teresa Wu for being a part of the advisory committee. Their valuable comments and feedback have improved this research work greatly.

I would also like to thank my colleagues at the Design Automation Lab, Sumit Narsale, Manikandan Mohan, Ying Chen, Prabath Vemulapalli, SamirSavaliya, Prashant Mohan, Payam Haghighi, and Mahmoud Dinar; if it was not for their support and help, this work has never been completed.

Finally, I would like to thank the National Science Foundation for providing support for this research work (National Science Foundation Grant #1045644, 12010162).

TABLE OF CONTENTS

Page

CHAPTER

LIST OF FIGURES

CHAPTER 1: PROBLEM STATEMENT

Problem Statement

Not all design concepts developed during ideation are technically or economically feasible to be pursued during later stages of design; therefore there is a need to evaluate concept feasibility and proceed only with the ones that are feasible.

Previously, a tool for helping designers to generate concepts was designed and implemented at www.ideationstste.com. This study aims to complement that tool and allow designers to work with little information available to roughly simulate and evaluate feasibility. To that purpose, a software platform for generating quick and simplified models has been developed. The designers would be provided with a function and behavior representation of their model in a graphical form and a module for quantifying and simulating the quantified model to see whether it satisfies the design objectives and/ or constraints or not.

The main challenge of this work is to present a physical ontology to formulate a concept design and enable low-fidelity analysis of it with little available information while technical analysis requires detailed information.

Scope

The effort presented in this study focuses on developing a computer tool for modeling and simulation of mechanical power transmission systems. It will not model systems that are electrically/electronically controlled.

The modeling environment proposed in this work gives the user access to many common artifacts (Commercial Off-The Shelf or COTS) that are typically used in mechanical systems and are available in the market, Physics principles that are generally used for modeling mechanical system behavior are also embedded in the tool.

Types of Evaluation

This tool is capable of answering a physically described feasibility question using concept models that are formulated using PVs (physical variables). If the feasibility question is described without physical variables or the concept model is not associated with PEs (physical effects) and COTS models, evaluation is not achievable (physical ontology is required).

The tool provides a platform for assessment of technical feasibility (vs. economical feasibility, etc.) through mathematical formulation and simulation of physical variables.

System Input and Output

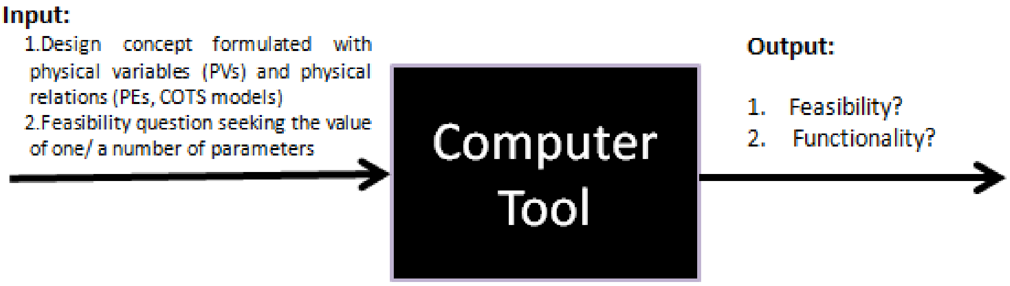


Figure 1 - Problem Input and Output

This tool is used in conjunction with the ideation tool available at www.ideationspace.com[1]. Using that tool for generating and developing concepts, the

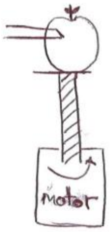
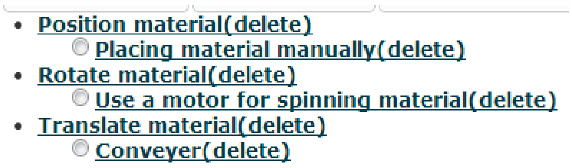
---

[1] More detailed description of the ideation tool is provided in next chapter

designer should come up with a structure of desired behaviors that serve the design goal in total.

A concept needs to be articulated with enough information to be evaluated; i.e. the designer needs to develop his concept to a stage in which the behavior described by the concept is describable using physical variables. In the ideation space, a concept is described in a morph chart; in order to be transferrable to this tool though, the morph chart needs to address the design solution clearly either with a number of components or with some physical relations.

For instance, the solution proposed on the left describes the main functions required for an *industrial apple peeler*. To answer the question whether or not the apple peeler could peel 100 apples per minute, the designer needs to complete his initial concept (generated using the ideation space tool) by associating it with two already established components: A conveyer and a motor (on the right):

Solution developed initially (coming from the ideationspace's FunctionCAD): Concept sketch:  Concept function structure:	Solution developed to a physically realizable state (Morph chart created in ideationspace):  • Position material(delete) ○ Placing material manually(delete) • Rotate material(delete) ○ Use a motor for spinning material(delete) • Translate material(delete) ○ Conveyer(delete)

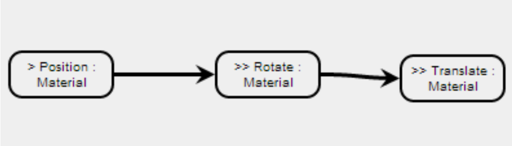
 **(Not ready for evaluation)**	**(Ready for evaluation)**

Figure 2 - Valid and Invalid Problem Inputs

At this state, the designer is able to use the evaluation tool to create a model, simulate it and answer the feasibility question. In summary, for a given feasibility question, the tool provides the designer with a platform to create a physical model consisting of physical variables (PVs) using the COTS (component models) or PEs (Physical effects).

A detailed overview of this process is given in the following chapters; next section discusses the basics of evaluation using this method:

Synopsis of the Work

In this work, formulating the right question and modeling the concept to address that question is facilitated by providing a platform for the designer to generate a simplified model which is capable of addressing the feasibility question. To satisfy that goal, the tool has modules that enable:

1. Creation of an ontology of physical variables and physical principles
2. Formal encoding of physical effects (mechanical Physics principles)
3. Formal encoding of component models
4. Abstraction of mathematical relations

5. Formulation of feasibility questions by the designers using the physical variable ontology and predefined physical relations, COTS math models or user defined models

Therefore, the software provides a pre-developed library of simplified model fragments representing common physical behaviors and component models (Approximately 110 physics principle and 16 components are included in a built-in library). The built-in library provides information on the application of each physical behavior/component model, and more importantly, a mathematical network of physical variables and relations that simulates mechanical behavior.

In order to create the model to address the feasibility question, relevant model fragments are added and linked together by the user; as this is done, the tool generates a parametric network of physical variables and relations. For ease of understanding and interaction with the model, the tool also generates a graphical representation of the physical model developed. The physical ontology is later quantified, checked for solvability and solved to see whether the feasibility criterion being evaluated is satisfied or not. The chart presented below shows the steps of this process:

Figure 3 - Steps to answering a feasibility question

CHAPTER 2: STUDY BACKGROUND

Chapter Overview

This chapter reviews the conceptual design phase focusing on concept generation and concept evaluation (the two major steps of this phase of design). In the first section of this chapter, methods used for concept generation are briefly discussed here and the ASU's ideation space tool is reviewed.

In the second section of this chapter, concept evaluation and its role in early design are discussed.

Overview of the Conceptual Design

Engineering design is defined as a technical activity that involves understanding of the requirements for a need and the creation of plans to satisfy that need -Jami Shah's MAE 540 Lecture Notes-. It involves generation of alternative solutions (concepts), engineering and economic analysis, decision making, experimentation, verification and detailing/documentation of product plans. It is usually conducted by a team involving engineers and industrial designers from various disciplines and stylists with both experiential and analytical knowledge [1]. As an early phase of design, conceptual design is responsible for generating design solutions and eliciting promising ones for later stages of product development. During this phase, designers determine the principle solution by abstracting the essential problems, establishing function structures, searching for suitable working principles and then combining those principles into a working structure. This phase of design results in the specification of a principle solution generally known as an idea or concept. Concepts transform abstract and qualitative ideas to an organized qualitative or quantitative formulation of the original idea[2].

Stages of Conceptual Design

In traditional categorization of conceptual design stages, five main phases are identified for it [1]:

1. Functional decomposition
2. Sub solution generation
3. Solution combinations
4. Solution evaluation
5. Documentation

More modern classifications however, classify phases of conceptual design as concept generation, testing, and refining/ re-generation.

During ideation, designers generate as many solutions (concepts) as possible; the quality of the solutions generated is not judged during this phase though. Encouraging creativity and providing the opportunities for the designers to explore the design space as much as possible are the main goals of this stage (the divergent phase of conceptual design).

However, proceeding to later stages of design with the large pool of solutions developed during ideation is not possible. Concept evaluation and preliminary analysis is required to link the ideation to embodiment design [1].

---

[2] This is a prescriptive view and is not always practiced. Sometimes design practices are much more chaotic and ad hoc.

The rest of this chapter reviews concept generation (ideation) and concept evaluation in two sections:

Section I - Ideation

The purpose of this design phase is to propose a solution to address a need; to that end, the designers might need to break the design problem into multiple smaller and perhaps easier problems earlier during problem decomposition. To that end, designers usually approach the design problem by decomposing the problem to a number of functions and creating a functional model (other methods are modeling requirements/ constraints/ objectives/etc.). This approach is generally considered to be the main framework used for design activities; it converts customer requirements into engineering approaches and produces hierarchical models of function, process and environment [2].

Once the problem is formulated using any desired technique, designers would need to generate sub-solutions for each sub-problem. To help them generate concepts, numerous intuitive and experiential methods have been proposed. These methods are categorized into (1) intuitive methods, (2) experiential methods, and (3) combined methods.

Following, a brief description of each is presented.

(1) Intuitive Methods

Intuitive techniques are chance-based techniques that depend on the knowledge of the designer and do not depend on any catalogs or physical principles. Free-form thinking, provocative stimuli, and problem reframing are common instances of such techniques Other approaches include: Method 635[3], Gallery method [4], C-Sketch[5], Brain storming [6].

In all these methods, a stimulus is usually provided by graphical objects like pictures, videos, and animations; online resources such as Imagenet [7] provide a large database of such stimuli. For reframing the problem and exploring similar alternatives also, Wordnet [8] provides a useful resource.

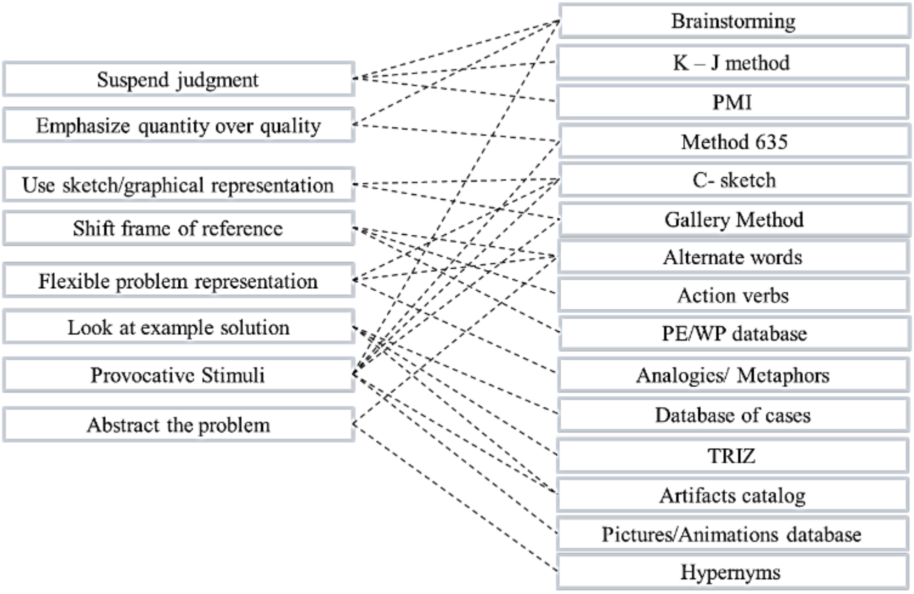


Figure 4 - Intuitive Ideation Methods [9]

(2) Experiential Methods (Logical Methods)

Unlike intuitive methods, experiential methods (logical methods) offer a systematic approach to design by relying on the past solutions. Logical ideation methods depend on physical principles, catalogs and databases; some examples of such methods include physical effects based catalog [10], working principles catalogs [11], Bio-Mimetics resources (like Ask-Nature [12]), and design repository [13].

(3) Combined Methods

Some ideation techniques rely on both intuitive and logical approaches. These methods are known as mixed methods. Examples of such methods include TRIZ [14], Morphological charts [15].

Since engineering design requires both creativity and functional quality, it is argued that for an effective ideation process, both experiential methods and intuitive methods must be incorporated [16]. To that end, previously a computer tool was developed at ASU (Ideation Space) that integrates many of the mentioned techniques. Below a short description of this tool is provided

ASU's Ideation Tool - Ideation Space

Prior to this work, a computer tool was developed at Arizona State University for helping the designers with concept generation (ideation). This tool creates a platform for the designer to do:

1. Function modeling with FunctionCAD
2. Concept generation for each function
3. Combination of concepts with morphological charts
4. Documentation of the results using graphical and textual documentation

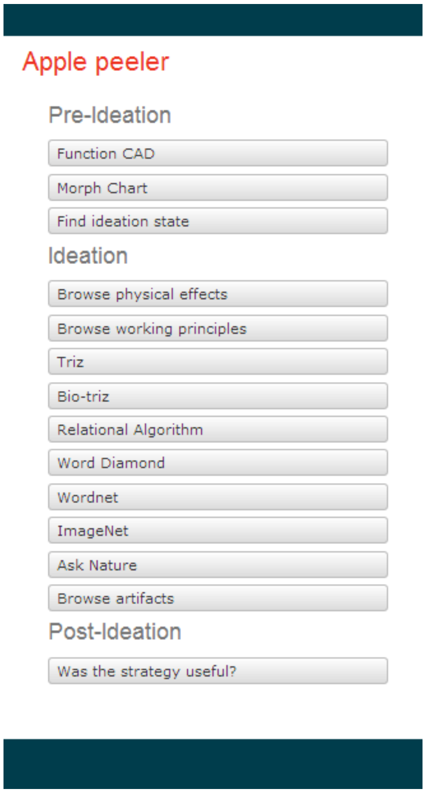


Figure 5 - Ideation Space's User Interface

FunctionCAD is a pre-ideation module that allows designers to create a function structure of the overall problem and formulate the concept solution as a structure of functions.

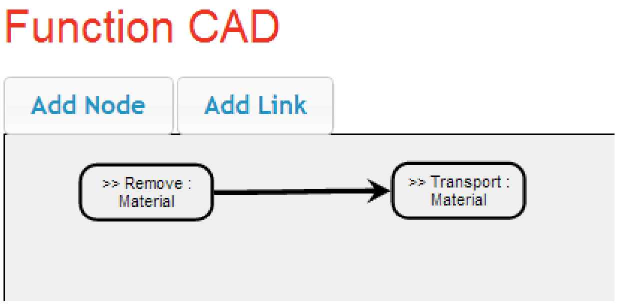


Figure 6 - FunctionCAD of the Ideation Space

Once the function structure is built, the designer should proceed to the ideation modules. Modules in this tool are a combination of the logical and intuitive methods that altogether provide the user the chance to explore the design space based on his ideation state and propose solutions (or sub-solutions) regardless of the quality of the solution. In the following figure, a list of the available ideation techniques is displayed:

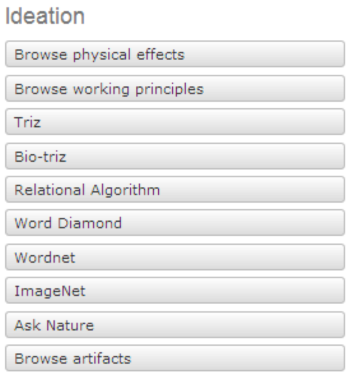
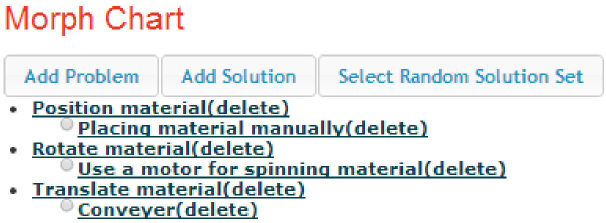


Figure 7 - Ideation Modules of the Ideation Space

At the end of ideation, a number of concepts are developed for each individual sub-problem. Effective combination of them for gaining proper functionality is another major phase in conceptual design. Being very critical in design of complex systems, finding compatible sub-solutions and evaluating feasible ones in a large network of sub-solutions is a major challenge for engineering teams. To that end, the Ideation Space tool, provides access to a Morph Chart [15] in order to help the designers organize and sort the sub-solutions, add descriptions or sketches to each and combine them for higher quality or more novel solutions.

A morph chart would be an output of this tool (also an input to the tool developed in this study).Below, a screen shot of a morph chart is provided.



- Position material(delete)
  - Placing material manually(delete)
- Rotate material(delete)
  - Use a motor for spinning material(delete)
- Translate material(delete)
  - Conveyer(delete)

Figure 8 - A sample morph chart developed by the ideation space

More details about the Ideation Space tool could be found in [9, 16].

*

A successful design may be constrained by decisions made and solution approaches generated during conceptual design [1]; it is very difficult to make major architectural changes in later stages during embodiment or detailed design. If there exist shortcomings in conceptual design, the tendency is to patch and refine which may result in a clumsy design.

Early concepts are usually presented in the form of a working structure as a combination of sketches and textual description. Such a working structure is not assessable until it is transformed into a more concrete representation. This concretization often involves selecting preliminary materials, producing a rough dimensional layout, and considering technological possibilities. Only then, it is possible to assess the essential aspects of a solution principle and review whether the objectives and constraints are satisfied or not [1].The phase for assessing the concept, is known as concept evaluation and its purpose is to see whether the concept could function as desired or not and should therefore be followed in later stages of design. Such decisions typically affect the rest of the design and development process: While successive evaluation guides the course of design activity, incorrect or inaccurate evaluations could delay the design process or impose extra work. Therefore, this stage of design is of critical importance in the success of a design and development process.

Section II - Concept Evaluation and Feasibility Analysis

Since it is not technically/economically feasible to pursue all concepts generated during concept generation, concepts are evaluated based on design objectives such as performance, cost, size, etc. and some are selected through preliminary analysis of initial ideas. Design concepts that do not satisfy the demands of the requirements list are then eliminated; the rest would be judged by the methodical application of specific criteria. During this phase, the chief criteria are of a technical nature, though rough economic criteria also play a part in decision making [17].

Concepts that are to be evaluated should be detailed enough to accommodate evaluation since pre-mature evaluation leads to pre-mature decision making which could further lead to elimination of a potentially good concepts or failing to filter unworthy ones. Thus as critical as evaluation is to valid decision making, sufficient detailing of a concept is to making a correct judgment. In other words, designers should proceed to evaluation if and only if, the concept is developed with sufficient details (this task is done as a part of the ideation phase). Once all design concepts are evaluated and some eliminated, one/some of the concepts are selected to proceed to the embodiment design for further detailing. It may be that several concepts look equally promising, and that a final decision can only be reached on a more concrete level later during the embodiment or detail design.

Concepts could be evaluated on a relative or absolute scale. In absolute concept evaluation, design criteria (customer requirements, design objectives, etc.) are used as a measure for evaluating how good or bad the behavior (performance) of a concept is (Examples of absolute techniques include the use-value analysis, cost- benefit analysis, and the weighted objective trees). However, in relative concept evaluation, concepts are compared with each other to determine which one performs better (Examples of such techniques include the Dominance matrix, the AHP method, and the Pugh method).

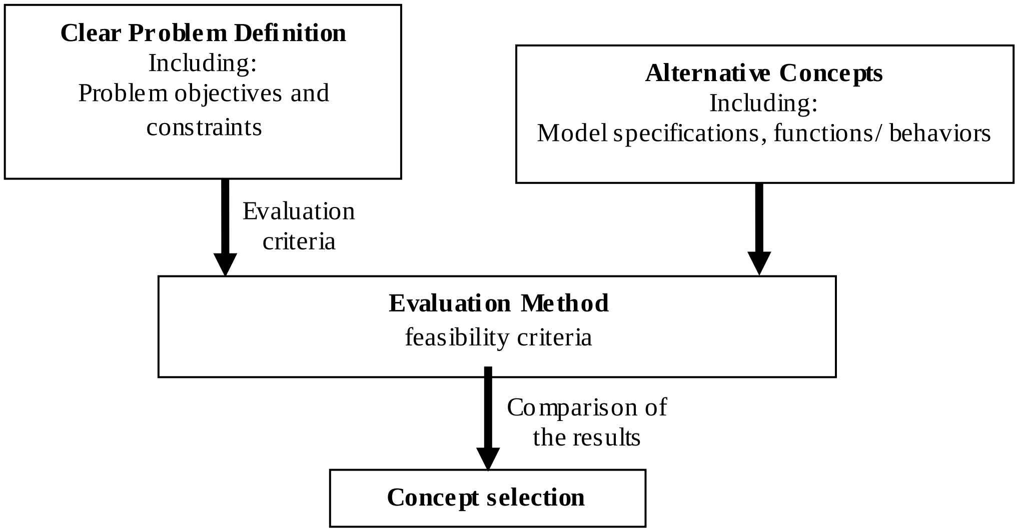


Figure 9 - Concept Evaluation/Selection Procedure

*

Regardless of the technique used, meeting customer/ product requirements, taking wise and well-planned steps forward, avoiding hassles later during the detail design, reducing the number of alternatives and eliciting only useful ones are some of the unique advantages that are gained by successful concept evaluation.

Next chapter reviews the literature in this area and focuses on the techniques and methods proposed for low- fidelity analysis and system modeling and evaluation. Later, the fourth chapter introduces the tool developed within this research work and reviews the basics of the techniques used for that purpose.

CHAPTER 3: LITERATURE REVIEW

Chapter Overview

Unlike detail and embodiment design, very little information is available during conceptual design. Due to the abstract and qualitative information available, system modeling and analysis in this stage is very restricted; in this stage, only rough models could be built and very low-fidelity analysis could be conducted.

This Chapter reviews the techniques and tools built for system modeling and simulation. The first section of this chapter reviews the techniques proposed for modeling and analyzing systems with little available information (qualitative analysis techniques). At the end of this section, these techniques are compared.

The second section of this chapter reviews the academic and commercial tools developed for that purpose. At the end of this chapter, tools developed in academia and industry are compared with each other and their suitability for evaluation of design concepts are discussed.

Section 1 - Techniques for Dealing with Qualitative Information

People draw useful conclusions about the physical world without differential equations. In daily life, humans figure out what is happening around them and how they can affect it, working with far less data (usually imprecise data), than would be required to use traditional, purely quantitative methods. Therefore methods to reason with abstract and qualitative data (such as qualitative Physics [18, 19]) have been introduced for creating representations for continuous aspects of the world, such as space, time, and quantity which support reasoning with very little information.

Main characteristics (both strengths and weaknesses) of modeling/analyzing qualities using this method include:

1. The capability to draw important conclusions about a broad pattern of physical behaviors with surprisingly little information.
2. Generating probable outcomes of a situation using qualitative and under-defined models (Results might identify a number of possible states -not a unique one-)
3. (Like all models) Capability to resolve ambiguities with higher resolution information if needed[3].

Application

In the context of engineering applications, two distinct areas of application are discernible for such techniques:

1. **Design**

   Many of the costly mistakes in design occur during the conceptual design phase where the overall goals, constraints, and functions of the artifacts are established and constructible artifacts/systems are planned for [17]. Since this stage of design deals with qualitative data, application of qualitative reasoning techniques that could handle partial information is very critical to timely and efficient design.

---

[3] Resolution is known to be one of the two main characteristics of qualitative reasoning. Resolution concerns the level of information detail in a representation. It is considered of high importance since one main goal of qualitative reasoning is to understand how little information suffices to draw useful conclusions or alternatively determine critical aspects/ important questions to ask.

2. **Monitoring and Diagnosis**

Like design, monitoring a system requires summarizing behavior at a level of description that is useful for taking action. Diagnosis on the other hand, aims to isolate problem, demanding a rough enough model for fault detection (examples could be found in [20, 21]).

Providing the opportunity to be faster, cheaper, and more efficient, both of the mentioned applications benefit much from qualitative techniques. Following techniques for reasoning with qualitative data is discussed.

Qualitative Reasoning Techniques

To address the above needs, a number of qualitative reasoning techniques are identified in the literature; some of the most applicable methods in areas of science and engineering include [22]:

- Parametric Model Formulation
- Causal Reasoning
- Comparative Analysis[4]

Method -1: Parametric Model Formulation

Design could be expressed in terms of some key parameters which describe performance. Using those parameters, a model of the design could be constructed and used for reasoning purposes. Parametric models could be generated with little information; unlike

---

[4] These methods could be used individually or in combination with each other depending on how much information is available and how detailed an outcome is expected to be gained from analysis. In this study, the parametric model formulation is used (more details given in chapter 5).

quantitative models, they are easily comprehensible and modifiable as well. For instance, the parametric model of an analogue circuit consists of voltage, resistance, and current -while the quantitative model consists of numeric values of each of those parameters-.

In fact, like any model, parametric models are built to simulate and predict behavior, once the model is built, it is assessed against available information to check its validity/accuracy. In case of incorrect/undesirably inaccurate results/ predictions, it is refined for a better prediction capability usually by increasing the resolution of the model (level of detail in the model).

For such an iterative process, parametric models provide the opportunity to easily change a portion of the model, without having to manually modify a complex and inter-related network of relationships. Thus by addressing the abstract level of the solution-space and the resilience to be modified rapidly, low-detail parametric models act as an important reasoning technique for rather complex problems.

Conceptual models built with this technique could be used at qualitative and quantitative levels; they are detailable (scalable; which in this context means that are transferrable to advanced stages of design).

Parametric Modeling Ontology

System modeling using parametric formulation is inspired by the work of Henry Paynter [32] as Bond graphs. Bond graphs are extremely helpful in design of system configurations during early design. They could be at qualitative and quantitative levels; therefore models generated with them are transferrable to later stages of design. Bond graphs were originally introduced for modeling power transmitted between different systems; they act as graphical representation of physical systems which are similar to the better known block diagram and signal-flow graph. Using the physical variables, they represent "effort" and "flow" transmitted between system sections.

In order to ensure compatibility between model components, ports ontology is used. In this ontology, each model fragments has a number of ports and linking fragments is possible if and only if the ports of both match [22].

More on the ontology and how it is implemented in this study is discussed in the next chapter.

Method - 2: Causal Reasoning

This type of reasoning constitutes much of the evaluation and decision making process that is typically done by human beings. The vast use of qualitative deductive reasoning in all stages of design is an instance of using causal reasoning in engineering applications.

In fact, causal reasoning is the ability to identify relationships between the cause and the effect. In this context, causal reasoning explains an aspect of a situation in terms of others, in such a way that the aspect being explained can be changed if so is desired [22].

The techniques used for causal reasoning all share a common structure of:

1. Identifying the factors within a state.
2. Identifying how the properties of a state contribute to a/some transition(s) to another state.

Unlike the parametric models which are generally quantifiable, models made with this reasoning technique are not quantifiable. Therefore, models made with this technique are not detailable; i.e., in the context of design, conceptual-level models are not transferrable to embodiment and detail design. This type of reasoning also requires very clear and specific declaration of rule sets.

Method - 3: Comparative Analysis

Humans commonly incorporate this technique to reason by using basic assumptions, heuristics, experiential or higher-resolution information [22]. Comparative analysis answers a specific kind of "what if" questions, namely the changes that result from changing the value of a parameter in a situation. For instance, the effect of increasing the velocity of a vehicle on the time it takes for it to stop could be surveyed through comparison.

This method benefits from the comparison between states, behavior, etc. and acts as a prediction mechanism. In order to use this technique, one needs to have sufficient information on the outcome of at least one case and use that to conduct comparison with another case [22]. An instance of this technique includes incorporation of *exaggeration* for predicting worst case scenarios with purely qualitative information [25] (For such problems, one needs to know the outcome of the extreme case.).

As stated before, models built with this technique require quantitative information and it is not possible to model qualitative data with this model; therefore, it is not a suitable reasoning technique for conceptual design.

*

Overall, it could be stated that causal reasoning requires a set of known rules based on which one could conclude whether an effect is obtainable from a certain cause or not. Comparative analysis however requires a set of cases with known outcomes between which a comparison is made and a conclusion is derived [22]. Comparison typically requires numerical values; yet causal chains are usually formed by a set of qualitative rules. Parametric models are defined at qualitative level; such model are quantifiable and could be used at quantitative level as well.

Section 2 - Academic and Commercially Developed Tools for System Modeling and Analysis

While design analysis during embodiment and detailed design requires extensive quantitative information and produces accurate results, conceptual analysis is restricted to little and rather qualitative information and produces approximate results. Therefore techniques used for each is very different from the other. In this section, only techniques and tools that are used for system design with little information are reviewed.

Following, some system modeling and analysis tools and their main features are briefly discussed; later in this thesis, the tool developed by this work is compared to them and the characteristics of each are discussed.

1- Academic Tools

Much research has been done on aiding with and automating design analysis [26- 30]; some leading to establishing techniques for system representation and some for system analysis.

An instance of a design/analysis tool is the work of Huang and Mak who introduced a concept generation and evaluation tool in 1999 [32] in an HTML- based environment by incorporating morphological chart analysis technique. This tool provides the opportunity for clear definition of functional requirements using FAST diagrams [33], generating sub-solutions using morphological charts, and evaluating them by assessing each generated concept with respect to a design objective or criteria. The technique used for assessing concepts is referred to as morphological chart analysis; i.e. scoring (ranking) of each concepts vs. the design criteria.

Among the approaches for solving conceptual design problems with computational methods (computer tools), are two agent-based systems presented in the A-Design research [34] and the catalog design method used in [35]. In both of these approaches, input-output characteristics of components are incorporated to analyze a system-level design by using components as building blocks and by integrating them according to design requirements.

Other methods to computational design analysis include representations that manage and manipulate functional descriptions which are later converted into configurations of components. [36] used a method based on energy flows as a foundation for their Scheme builder tool to automatically explore alternative conceptual schemes and appropriate allocation of function between electromechanical components.

In 2005 Sridharan and Campbell used function component matrices to encode physical compatibilities between components. In this method, the goal was to generate a formal set of rules that will describe function structures for a range of products, based on a common basis, which will help in the designers in generation of function structures (a function generation "grammar" that aims to help create a viable function structure based on current products). Matrix algebra was employed on these two matrices to construct a final product matrix describing the overall solution space [37]. The rule set of this that is based on 20+ consumer products such as juicer, drill, cooker, etc. The tool developed is tested in a study in [37] in which the function structures developed with and without the tool are compared; the results of the study shows a much better outcome in developing function structures in case of using the tool.

In terms of implementation, this work resulted in a large and unorganized set of java files. Thus, the rules were later rewritten in a new graphical environment known as GraphSynth which allows one to graphically create the rules and manage the resulting data as a series of portable xml files. The tool uses a function structure grammar, configuration design grammar, and a component selection algorithm to find the optimal choice of components [38]. This tool emphasizes the interconnectivity and qualitative representation of system network more than parametric definition, quantification and detailing; it functions at a qualitative level suggesting components that could perform a specific function and does not have a function structure evaluation module. The tool benefits from a database of 300 commercial artifacts and 213 general rules that contain many heuristics stored as grammar rules with which the reasoning mechanism of the tool functions. However, compared to humans who collectively know more, each likely containing many caveats, exceptions, and useful minutiae, in some ways, the tool might generate excessive infeasible solutions or hinder novel solutions.

The design operators from the TEAM model developed in 1991, modified for practical use in the language for design procedures specification provided by the ASU Design Machine [39], are another example of such system with five generic types of "step"s in the language as: function step, lookup step, input step, calculate step, and rule step. In this system, one or more procedures could be associated with a given part class (e.g., gear) and design procedures may calculate some or all parameters of the part. The language provides all the necessary facilities to specify major design activities which can be performed automatically. This method lacks means for (and is not intended to) specifying design activities not performed by the computerized system (e.g., interaction between designers) and there is also no mechanism designed for recording rationale for each action.

Using a similar function-based approach, Bryant et al. [40] developed another concept generation technique that utilizes a repository of existing design knowledge and a set of matrix-manipulation algorithms. The Function Design Framework (FDF) in 2008 and the FunctionCAD in 2009 developed by Nagel et al. [41, 42] are two other instances of frameworks that allow representation and modification of design concepts. FDF integrates functional modeling based on the Functional Basis [43] with process modeling [44, 45] to provide a unified modeling architecture where function provides depth to system modeling and process provides breadth. FunctionCAD, the second generation of FDF, is however a modular, open source application (written in C++) designed to create integrated, hierarchical models consisting of environments, processes and functions. It creates integrated functional and process models that are generated through three phases to represent the environments where the product would be used, the processes capturing the jobs, tasks that a product does and the functionality that the product must contain.

A different approach was taken by Goel and Bhatta in 2004 which resulted in development of a system called IDeAL [46]. IDeAL evaluates the candidate design by qualitative simulation of the model it generates, where the qualitative simulation is done simply by tracing the causal behaviors and propagating the effects of the initial conditions and constraints imposed by transitions. IDeAL uses a theory of analogy-based design called model-based analogy (MBA) that transfers design patterns from source cases to target problems. In particular, for the domain of physical devices, it identifies a class of design patterns, called generic teleological mechanisms (or GTMs), that specify generic functional relations and abstract causal structure of a class of devices. [46] IDeAL has been developed as a proof of concept; it uses causal reasoning for modeling systems and according to its developers, the underlying algorithm for it requires addressing generality and scalability before it could be put into further practice. It should also be noted that the method does not support quantification (none has been reported about such a capability).

In 2013, Sen et al. [48] developed a graph-based function-structure tool called ConMod which creates visual rendering of function models. It allows an interactive construction of design concepts using a GUI that is intended to replace pencil and paper as the modeling medium. [48].Unlike commercially available tools such as Microsoft Visio and academically developed tools such as FunctionCAD [42], ConMod uses a basic intelligent reasoning approach (causal reasoning as discussed earlier in this chapter), i.e., **v**ocabulary, grammar, and algorithms - to perform basic reasoning and compatibility check [48].

At the end of this chapter, these tools as well as the commercial tools are compared with each other and their specific applications are discussed.

*

2- Commercial Tools

In this section, 5 different modeling tools are examined and their suitability for preliminary analysis is discussed. Unlike commercial tools which are mostly rule-based, all these tools use mainly parametric modeling for simulating system behavior. The tools discussed in this chapter include: Modelica, Phoenix Integration, I-CAD, Knowledge Fusion by NX Siemens, and Ingenious.

Modelica

Modelica is originally a system modeling tool for configuration modeling and analysis. Geometrical design and/or optimization cannot be carried out with most of its versions. It is famous for its rich library of cases which makes modeling easy and intuitive. As a programming language, earlier versions of Modelica based tools require coding but more recently developed ones are more user-friendly with an intuitive GUI. It is worthy to note that the coding capability of Modelica is available in all commercial/free versions of it for customized use.

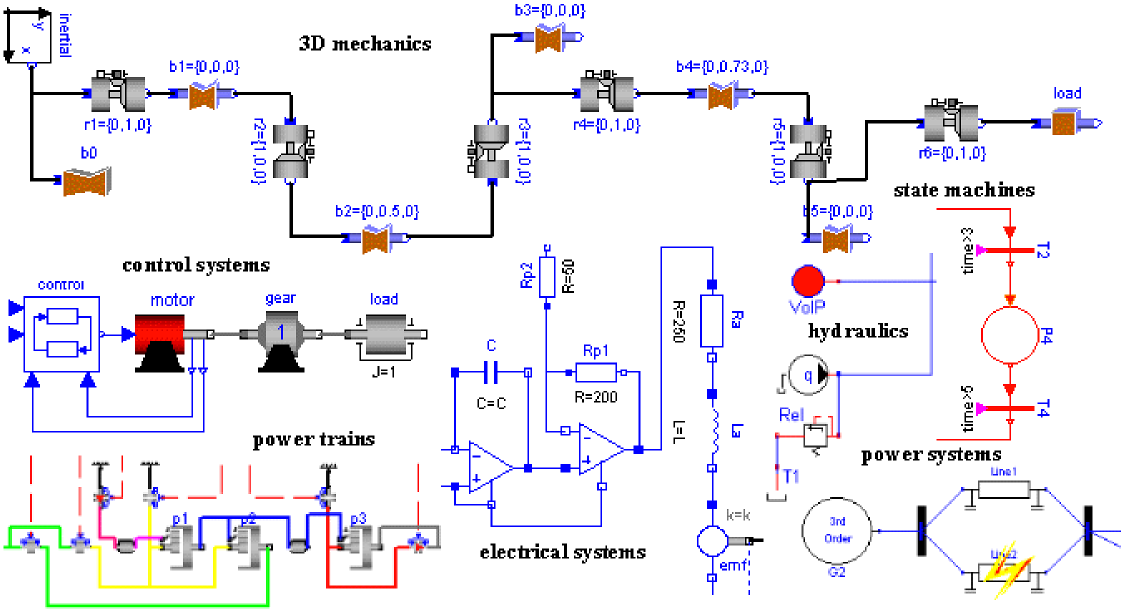


Figure 10 - Modelica Interface

Model Center

Model Center by Phoenix Integration is another system configuration modeling tool which enables creation and simulation of a configurational representation (model) of a system, linking them and creating a hub for different contributors of a project to work together. It allows trade studies and optimization using its built-in features. Unique features of this software package are its rich library and the capability to work in a network. Like Modelica, Model Center is adaptable to the MATLAB/ Simulink package.

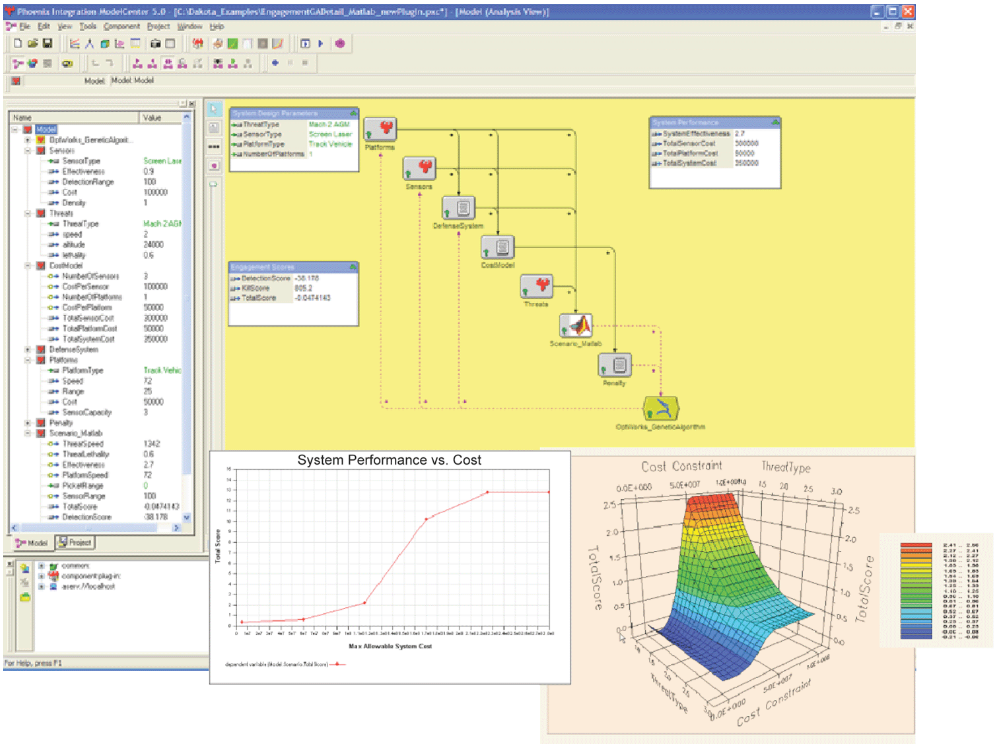


Figure 11 - Model Center's Interface

I-CAD

I-CAD was a popular configuration modeling and analysis tool which is no more available as a separate tool. It is now embedded in the CATIA package. Reviews on KBE tools do not usually recommend CATIA's package mentioning other more developed packages with superior capabilities such as built-in libraries.

NX- Knowledge Fusion

NX-Knowledge Fusion is very robust in designing and modifying parametric CAD models; it works well with Excel and transfers and updates data (as of values and rules) automatically back and forth. In Contrast to its unique features in parametric CAD, Knowledge Fusion is not as useful in configuration modeling; it requires external tools such as MATLAB/ Simulink to perform such modeling/ analysis.

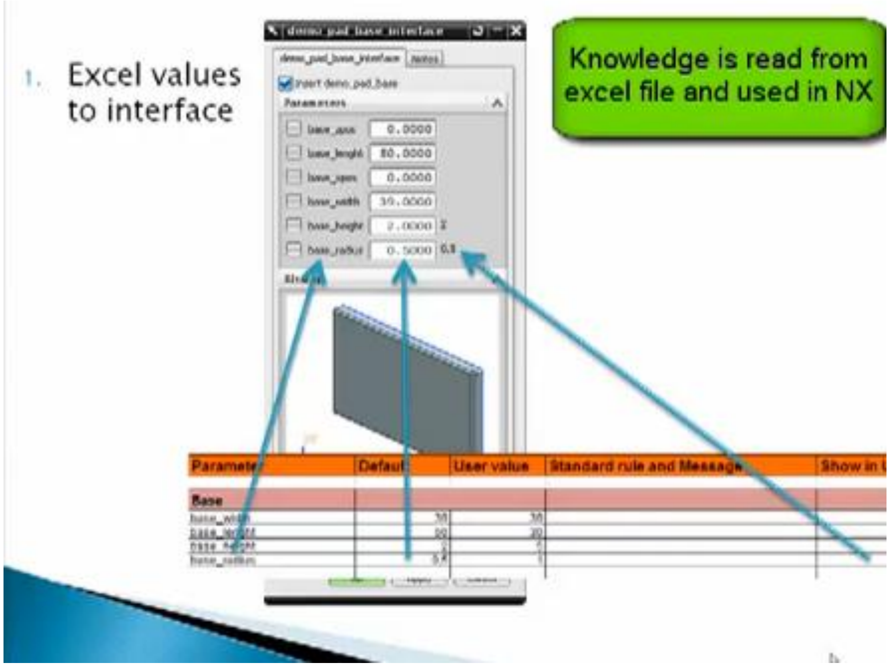
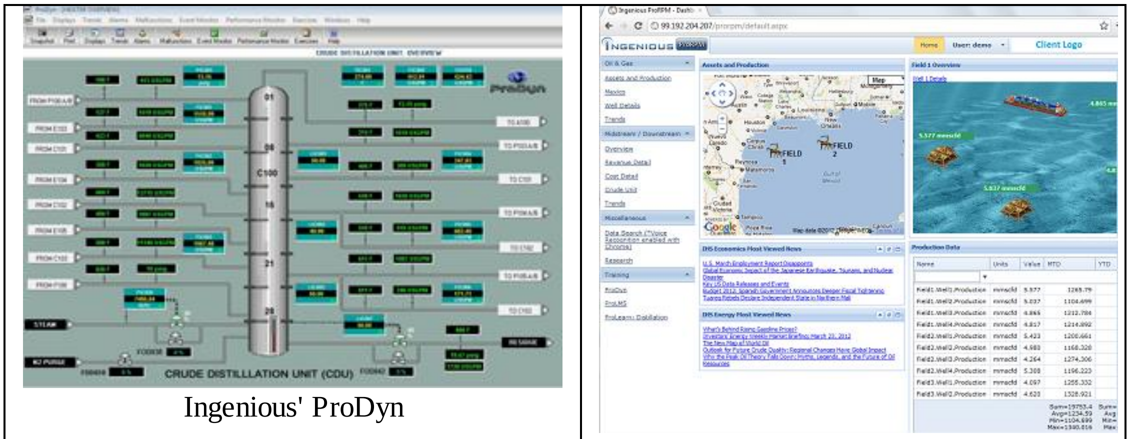


Figure 12 - Knowledge Fusion's Interfce

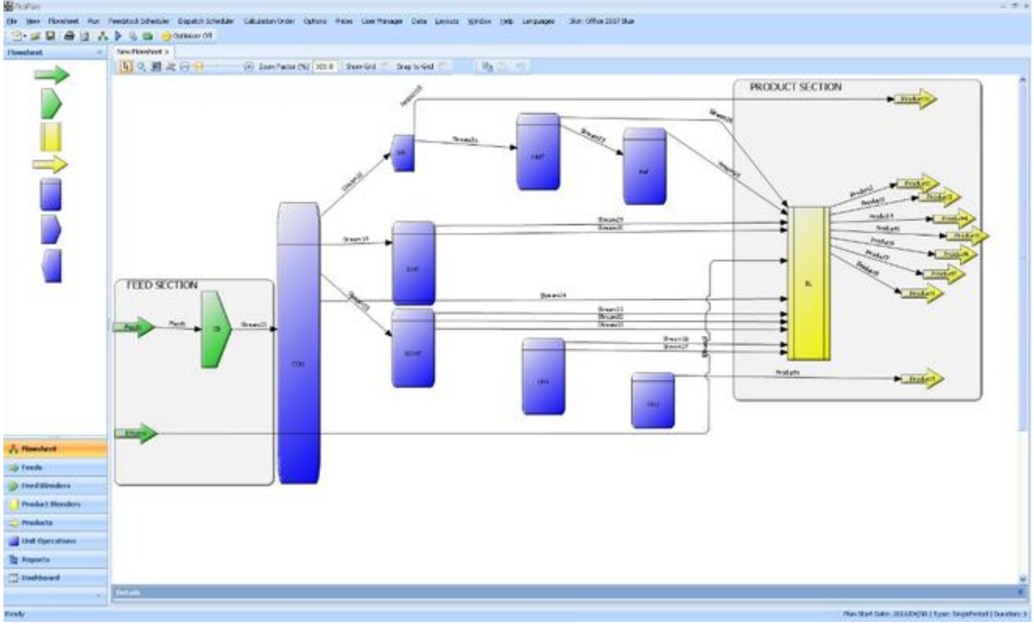
Ingenious

Ingenious is another package which is mainly designed and used for process planning and optimization. It deals very little with product modeling and analysis (vs. process modeling). As a process monitoring tool however, the tool offers unique features of network operation and dynamic process tele-monitoring.



Ingenious' ProDyn

Ingenious' ProRPM



Ingenious' ProPlan

Figure 13 - Ingenious' Interface

Wolfram's System Modeler

Wolfram's System Modeler allows efficient multi-domain modeling of mechanical, nautical, aerospace, biological systems. It allows simulation and visualization of system behavior. Its 3D capabilities are however much less than unified modeling and simulation packages like NX. Not all the simulation capabilities of Mathematica package is built in the System Modeler and for in-depth analysis of the model, users need to use Mathematica itself. Overall, compared to other available tools, it seems that the System Modeler is still in early developmental stages and yet not much functional.

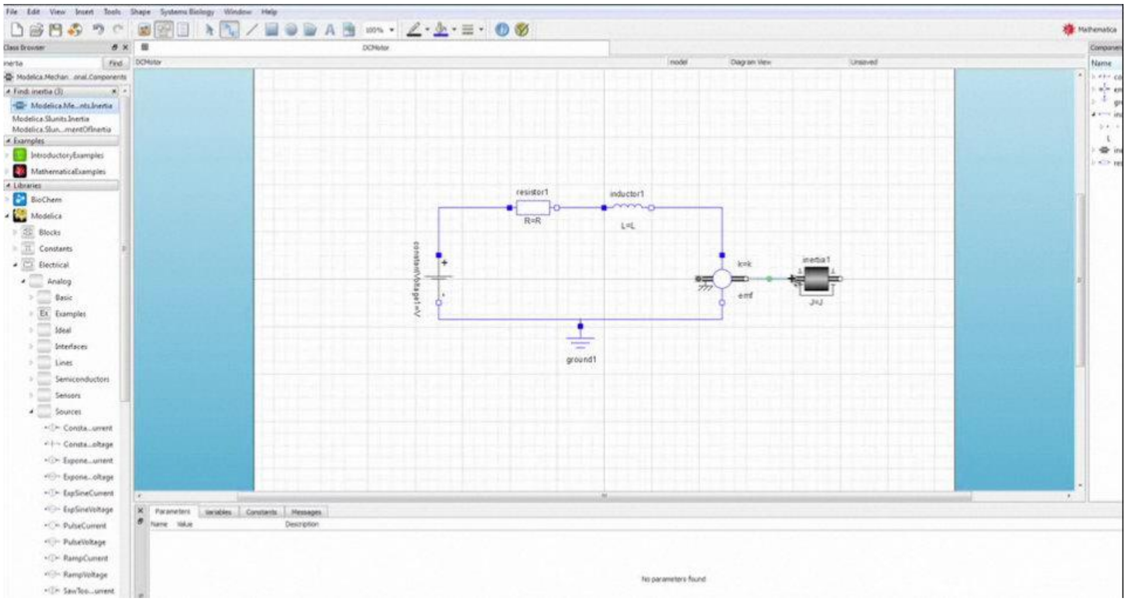


Figure 14 - Wolfram's Interface

*

Comparison of the mentioned tools

In general, commercial tools are mostly suitable for embodiment and detail design; they require extensive details and produce accurate results. Their formulation is time-dependent and are ususally connected to major CAD tools such as NX Knowledge Fusion. On the other hand academic tools , are more suitable for early design; some of these tools allow analysis through parametric modeling (such as ASU's Design Machine); some just provide a platform for system representation (such as FunctionCAD); some provide extensive information on artifacts and their functionality without platform for system modeling and analysis (such as OSU's Design Repository), and some are rule-based systems that examine connectivity and ensure compatibility between model fragments (such as ConMOD). Some of these tools are capable of qualititive modeling (such as IDeAL), while some use a quantifiable platform for modeling (such as Design Machine). Below These tools are compared:

	Representation	Qualitative	Parametric	Quantitative	Specific Values
FunctionCAD	*				
GraphSynth	*	*			
ConMOD	*	*			
IDeAL	*	*			
Design Machine				*	
Commercial Tools (eg. NX Knowledge Fusion, Dymola, Model Center)	*		*	*	*

CHAPTER 4: STUDY METHOD- PART I: SYSTEM DESIGN

Chapter Overview

This chapter discusses the method used for model development and analysis. Modeling ontology is discussed in detail in this chapter. This chapter consists of three sections: Required input information, modeling ontology, model development, and model solution are discussed in detail in section I of this chapter.

Section II discusses the equation abstraction and how the time dependent mathematical formulations are converted to simplified equations (Building the knowledge bases from these simplified equations is discussed in the next chapter.)

*

Within this study, a tool is built for concept modeling and simulation which is capable of answering a feasibility question. Due to its scalability and extendibility characteristics, parametric modeling ontology is used for system modeling.

Below the details of the ontology are discussed:

Section I- Modeling Ontology

Overview of the Method

In this study, concepts are represented by physical variables (PVs); and the way they achieve the desired functionality (behavior) are described by physical relations (Physical Effects, and Component Off The Shelf models). For instance, in order to address whether a concept vehicle reaches maximum velocity of x m/s (as a dynamic behavior), the concept is described with physical parameters such as velocity, and acceleration and its behavior by the dynamic relations in this case, Newton's law of motion.

Such a mathematically realizable and physically explainable representation (called a behavior model) is capable of reflecting the behavior and therefore addressing the feasibility question if enhanced with sufficient input.

Appropriate Modeling Ontology

As stated before, design is a recursive process with several re-design-evaluation phases and a good technique for modeling design is one that is expandable (also known as detailable) and scalable (compositionality[5]).

Among the qualitative techniques introduced in the previous chapter, the parametric modeling is capable of generating scalable and extendable models. It can facilitate product and process ontology which are the main ontologies used for system modeling. Therefore, this modeling technique is incorporated for this study[6].

---

[5] By definition compositionality refers to the ability to combine representations for different aspects of a phenomenon or system to create a representation of the phenomenon or system as a whole.

[6] Ontology is central to qualitative modeling since one of its main goals is to formalize the art of building models of physical systems. The most commonly used in system modeling are:

1. the device ontology [18]
2. the process ontology [49]

The device ontology is inspired by the network theory and system dynamics. Like those formalisms, it construes physical systems as networks of devices whose interactions occur solely through a fixed set of ports. Unlike those formalisms, it provides the ability to write and reason automatically with device models whose governing equations can change over time. The process ontology however suits the practice of Thermodynamics and Chemical Engineering. It construes physical systems as consisting of entities whose changes are caused by physical processes. Process ontology therefore postulate a separate ontological

This modeling ontology uses physical variables (PVs), physics principles (PEs) and component models (COTS). Models created with this technique are known as behavior model. Following this modeling ontology is described in detail:

Required Input Information

Two major inputs are required for the tool:

1. A clearly defined **feasibility question**
2. A clearly described **concept**

1. Feasibility Question

During the conceptual design one intends to answer the critical question of whether or not a concept works as desired. This question should consider several aspects such as structural stability, kinematic, thermal, etc. behavior for reliable outcome. Being unspecific and general, as expected for concept design stage, the designer is required to indicate "how" s/he wants the feasibility to be evaluated. This is achieved by providing criteria and defining feasibility questions that could be presented to the tool such as "Would this concept satisfy this criterion - the criterion being represented by a/ a number of PV(s) range or value-?".

2. Concept

---

category for causal mechanisms, unlike device ontologies, where causality arises solely from the interaction of the parts.

Being appropriate for a particular domain, both device ontology and process ontology are used for modeling concepts. Depending on the context, each provides useful information about model properties. For instance, device modeling of an electronic circuit vs. process model of a chemical plant both reveal useful information about their performance.

Like the question asked above, the concept should be realizable; from a Physics standpoint, that means that its function/behavior should be describable by physical parameters. For instance, a concept satisfying typical functionality of "moving" is non-realizable unless it is described by the physical parameter of "*position*" changing from *state1* to *state2*.

While it is acceptable that a concept describes behavior without much detail and at a high and abstract level, it should be able to clearly express "how" the feasibility criterion being investigated is achieved. In the case of the example provided above, answering whether or not the *displacement (movement)* exceeds a certain value is impossible by addressing only *state1* (*state2* is also required). Therefore, for answering a feasibility question, concepts coming from the ideation should be complete or well-developed.

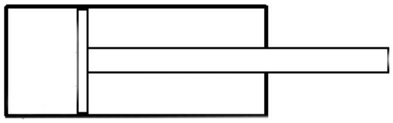
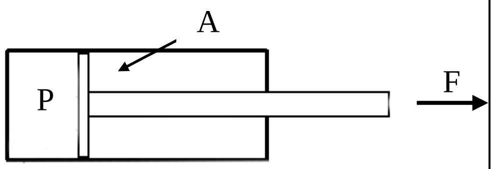
Sketch - Undeveloped Concept: "mechanism to generates force"	Sketch associated with a Physics principle - Well-developed Concept:
	 Associated with: Pascal's law of pressure F = P*A

Figure 15 - Improper and Proper Input to the Tool

If a concept is incomplete, it is not assessable and the designer needs to go back to ideation and complete it. However, even for a completed concept, the designer might not be able to answer the feasibility question asked. The feasibility question asked should be in line with the physical attributes that the concept is described with.

Therefore as stated before, in order to use this evaluation technique, the designer needs to have:

1. A clear feasibility question seeking a physics variable
2. A well-developed concept describing concept behavior with PVs, PEs, and COTS models

Modeling Ontology - Behavior Models

Within the scope of this study, models are developed to address one/ a number of feasibility questions. As an abstraction of the real world, a model represents a simplified version of the actual situation by incorporating:

1. Model elements
2. Relationships between the model elements

In this approach, model elements are represented by physical parameters. The relationship between them is also represented by mathematically formulated physical relations that describe basic function of a system.

For the instance presented above, the concept model would consist of physical parameters such as position, velocity, and acceleration as model elements, and the kinematic relations (physics principles such as Newton's law of motion) as the method for relating model elements to each other.

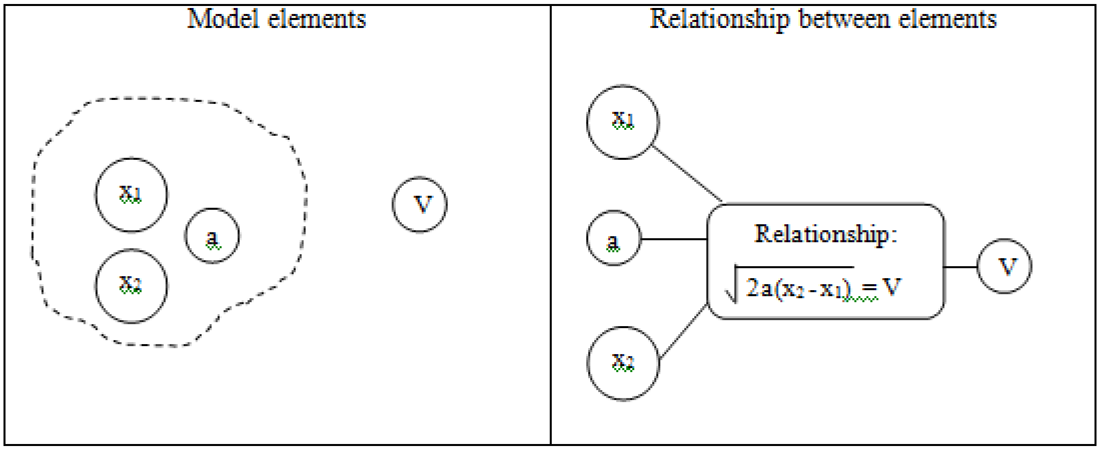


Figure 16 - Model Parameters and Relations

Since the model might involve multiple aspects and could not be fully represented by only one physical relation, a system of physical relations should be incorporated to represent it. Therefore, parameters are connected to each other at two levels:

1. parameters that are connected to each other through a single physical relationship

2. parameters that are commonly shared between more than one physical relation

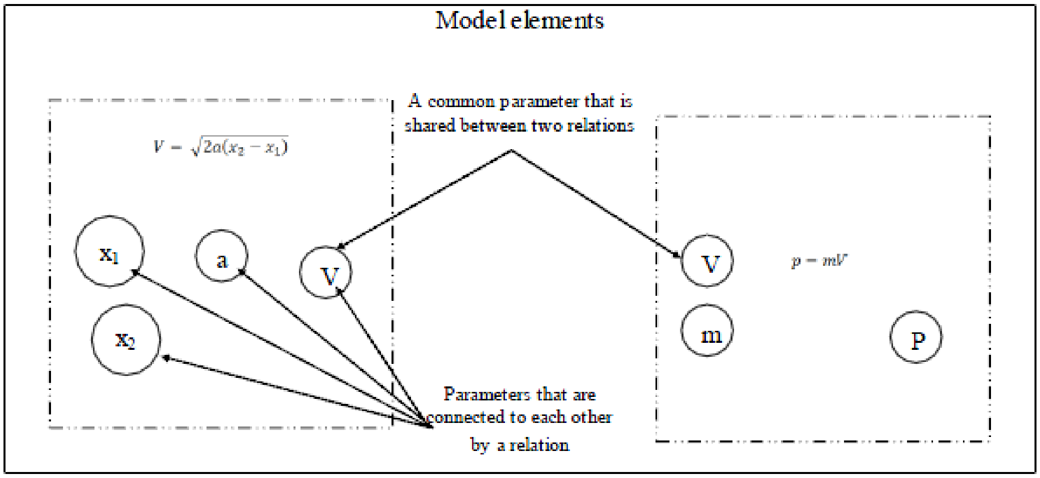


Figure 17 - Two types of parameters used in the model

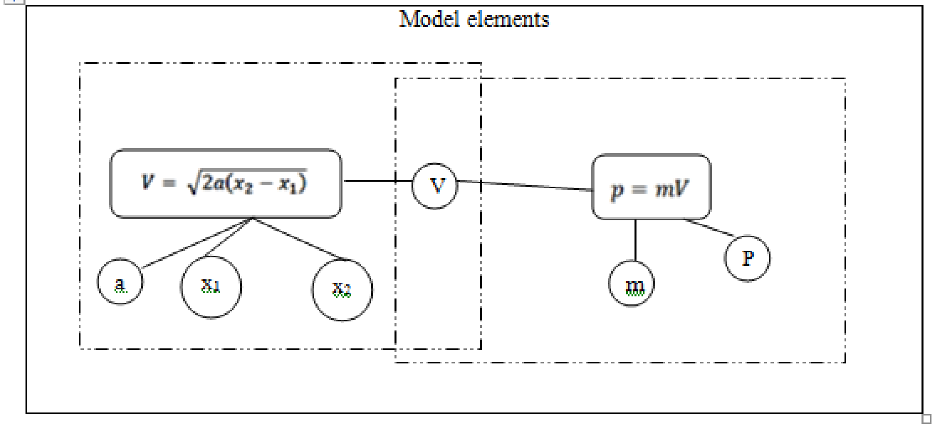


Figure 18 - Bi-partite graph of the concept model shown in the above figure

Model Formulation

This formulation is reflected in model development by using bi-partite graphs. These graphs represent two different types of connections: in this modeling approach, they are: (1) the relationship between vertexes that belong to one physical relationship, and (2) the connection between parameters that are common between two or more physical relations. From the perspective of mathematical formulation, with a single relation describing behavior, the model consists of *n* different variables (each representing a physical parameter) that are related to each other (Type 1 connection from above). Parameters that belong to two/more physical relations (Type 2 connection from above) also indicate the same variable that belongs to two/more of the mathematically formulated equations. Therefore the mathematical formulation of the example presented above would be:

$$\begin{cases} V - \sqrt{2a(x_2 - x_1)} = 0 \\ p - mV = 0 \end{cases} \rightarrow \begin{cases} x_1 - \sqrt{2\, x_2(x_3 - x_4) = 0} \\ x_5 - x_6.x_2 = 0 \end{cases}$$

In general, the mathematical model constructed from the system of physical relations would be formulated as:

$$\begin{cases} f_1(x_1, x_2, \dots, x_m) = 0 \\ f_2(x_1, x_2, \dots, x_m) >< 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_m) = 0 \end{cases}$$

Model Compatibility

It is important to note that shared parameters between different equations should be of the same type. In system modeling perspective, this is equivalent to having model fragments compatible for being linked to each other. As an instance, the *displacement* output of an actuator cannot be the *torque* input of a turbine shaft; in this modeling ontology, this translates to the *displacement* being a different PV than *torque* and therefore referred to as two different PVs.

In this system, compatibility between model fragments is addressed by the ports [24]. In ports ontology, only certain ports could be connected to each other. Here ports are represented by PVs and only similar PVs could be connected to each other (in terms of mathematical formulation, PVs shared between PEs and COTS).

Model Status and Model Solution

The formulation presented above serves as a parametric model for the problem with which the status of the model feasibility could be evaluated if the model is solved.

Assuming that none of the equations are redundant, in order for the system of equations presented above to be solvable, the number of model variables should be greater than the number of available equations in the formulation ($n \geq m$) -This is the necessary condition, not the sufficient condition-. Therefore, in case of $n < m$, the model is over constrained and there would be no unique answer to the formulated problem model. In this case, model formulation should be modified and redefined with less physical relations describing the relationships between the model parameters. In reality however, this situation would happen very rarely and is probably caused by bad formulation.

If $n = m$, the formulation is solvable and has a unique answer. However, in case of $n > m$ (under constrained model), some of the variables should be replaced by known quantitative values in order for the system of equations to be solvable. Variable quantification is done by the designer and would be based on the problem specifications and feasibility question asked. More on this topic would be discussed in the next chapter, in section 2.2.

Section II - Equation Abstraction

The systems of equations that govern certain behavior (in electrical circuits, chemical kinetics, etc.) contain a combination of differential equations and algebraic equations. The differential equations are responsible for the dynamical evolution of the system, while the algebraic equations serve to constrain the solutions to certain manifolds. In addition to complicated and time-taking solution processes required, such equations consist of many derivative terms which would require much further quantification of derivative terms or boundary conditions. Information required however is not available and it would therefore be desirable to modify the formulation such that it requires much less input information.

If a behavior is described by

$$\begin{cases} \dfrac{dx}{dt} - x(t) = 0 \\ x(t) + \dfrac{dx}{dt} + C = 0 \end{cases}$$

Such an equation is replaced by

$$\begin{cases} \dfrac{x(t2) - x(t1)}{t2 - t1} - x(t) = 0 \\ x(t) + \dfrac{x(t2) - x(t1)}{t2 - t1} + C = 0 \end{cases}$$

As seen, the above formulation still requires so many input variables. In order to simplify the equation, based on the desired output variable, the above equations are modified to:

Assuming the desired output of x(t2)

$$\begin{cases} \dfrac{x(t2) - x(t1)}{t2 - t1} - x(t2) = 0 \\ x(t2) + \dfrac{x(t2) - x(t1)}{t2 - t1} + C = 0 \end{cases}$$

And then re-formulated as:

$$\begin{cases} x(t2)\left(\dfrac{1}{t2 - t1} - 1\right) - (\dfrac{1}{t2 - t1})x(t1) = 0 \\ x(t2)\left(1 + \dfrac{1}{t2 - t1}\right) - (\dfrac{1}{t2 - t1})x(t1) + C = 0 \end{cases}$$

Now this system of equations could be replaced by one equation of:

$$x(t2)\left(\frac{1}{t2 - t1} - 1\right) - [x(t2)\left(1 + \frac{1}{t2 - t1}\right) + C] = 0$$

Now this equation could be solved and the desired output could be calculated using simple algebraic operations.

*

This chapter reviewed the basics of the approach used in this study:

- Modeling ontology is explained to address how a feasibility question and a concept are transferred to an analyzable model;
- model generation and solution are explained in a step by step approach to address the technical approach taken in this study;
- model simplification through equation abstraction is explained. The purpose of this step was originally to help with model generation and analysis with little available information.

Next chapter reviews the implementation of the mentioned techniques; later in this thesis, chapter 7 briefly discusses how the tool helps with addressing some of the technical issues encountered during design.

CHAPTER 5: STUDY METHOD - IMPLEMENTATION

Chapter Overview

This chapter discusses the tool implementation and the main methods used for modeling and simulating concepts. The contents of this chapter are presented in two sections:

The first part of this chapter reviews the tool implementation focusing on 1.database design (section 1), 2.model representation (both the graphical representation and the mathematical formulation) (section 2), and 3.model simulation and solution (section 3).

The second part of this chapter reviews the GUI of the tool and briefly discusses the flow of information in the tool.

Part 1. Tool Implementation

The tool is developed using MATLAB 2011 and is implemented using modular programming technique. It has been tried to design a user-friendly GUI with intuitive features that require a short learning period for designers.

Knowledge base of the tool consists of the physical variables and equations abstracted (PVs and PEs). PEs and COTS models could either represent change of state within a process or the relationship between PVs within a state. These formulations are presented in the tool as libraries of PEs and COTS models that represent relationships between different PVs (A comprehensive list of those data bases are presented in the Appendix A-C.).

Variable types used in the tool include the mechanical/ physical parameters that are commonly used for describing power transmission systems (e.g. torque, speed, etc.). The data bases are developed using the SI system units. For the current implementation, unit conversion is not possible. As described in the previous section, all embedded equations are steady state type; and no differential terms are present in the formulation.

The search technique used for finding queries in the data base is based on finding relevant PEs and COTS through PVs that are being modeled.

The data bases are developed as MATLAB .MAT tables utilizing easy retrieval of data from the data bases and fast modification of them. Three major data bases were created for this purpose:

1. A database of physical variables that consists of different types of physical parameters and their states. In this database, parameters with similar units compose the same category of data in the data table. For instance, Pressure, pressure_1, pressure_2, Surface pressure are all of same nature (could represent the same physical variable) and therefore belong to one data unit[7].
2. A comprehensive list of Physics principles (PEs) with a brief description of each principle and the equation corresponding to it[8].
3. A comprehensive data set of component sets with a brief description of their application[9], pictures from their common types, read more links, links to commercial catalogues of each, and their mathematical equations.
4. A data set of the problem model being developed that gets modified as changes are made to the model.

---

[7] This database is presented in Appendix A.

[8] This database is presented in appendix B.

[9] This data base is presented in appendix C.

Section I - Data Base Design:

Following the organization of data in each data base is presented:

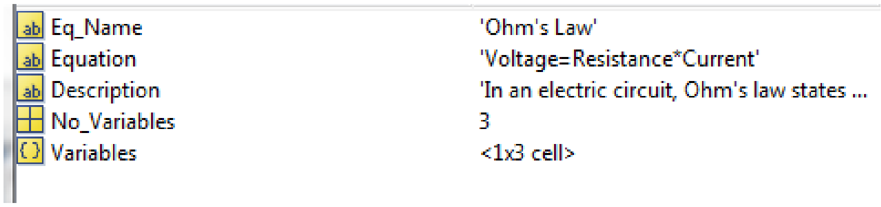
1. Physic Variables

Main variable	State variable (s)

2. Physic Principles

Area (Thermal& Fluids, Mechanical, Electrical)-->	Index
	Equation Name
	Equation
	Number of Variables
	Variables
	Description

A sample query of this database is shown below; the query displayed is on the Ohm's law:



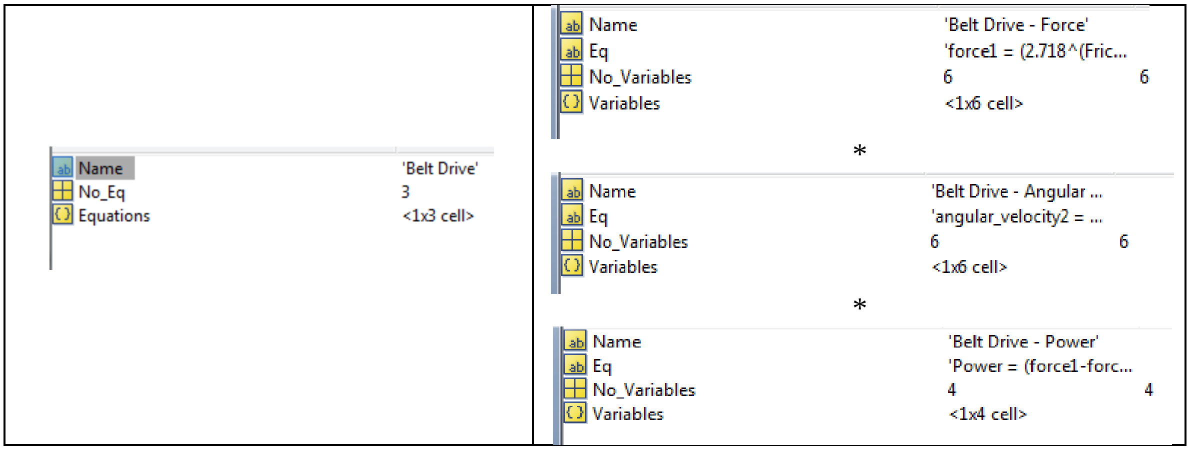
Eq_Name	'Ohm's Law'
Equation	'Voltage=Resistance*Current'
Description	'In an electric circuit, Ohm's law states ...
No_Variables	3
Variables	<1x3 cell>

Figure 19 - A sample query of the PE's Database

3. Component Sets

Index	Name	Number of Equations	Equations ->
			Name Equation Number of Variables Variables

A sample query of this database is shown below. The mathematical model of the belt drive is displayed below and the three equations describing it are displayed:



Name	'Belt Drive'
No_Eq	3
Equations	<1x3 cell>

Name	'Belt Drive - Force'	
Eq	'force1 = (2.718^(Fric...	
No_Variables	6	6
Variables	<1x6 cell>	

*

Name	'Belt Drive - Angular ...	
Eq	'angular_velocity2 = ...	
No_Variables	6	6
Variables	<1x6 cell>	

*

Name	'Belt Drive - Power'	
Eq	'Power = (force1-forc...	
No_Variables	4	4
Variables	<1x4 cell>	

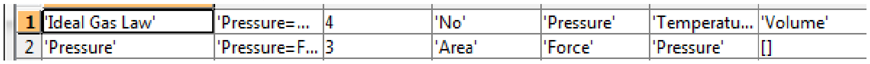
Figure 20 - A sample query of the COTS' Database

4. Problem Model (The Behavior Model)

Equation Index	Equation Name	Number of variables	Variables	Equation output	Value for the rest of variables	Equation output value

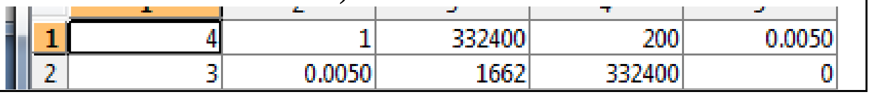
A sample query of this database for a simple model is shown below.

Qualities stored in the database:



1	'Ideal Gas Law'	'Pressure=...	4	'No'	'Pressure'	'Temperatu...	'Volume'
2	'Pressure'	'Pressure=F...	3	'Area'	'Force'	'Pressure'	[]

Quantified values of the above parameters (either gotten from the user or

	calculated) stored in the database: 

1	4	1	332400	200	0.0050
2	3	0.0050	1662	332400	0

*

Section II - Model Representation:

A Parametric Behavior Model is a representation of a problem formulated using PVs, PEs, and COTS behavior models. In this representation, the system behavior is expressed in terms of mathematical relations between PVs. The ontology of modeling using this technique is discussed in the previous chapter. This section reviews creating the model representation of the network of physical relations describing the behavior from two perspectives:

1. Graphical representation
2. Mathematical formulation

Following the model representation and formulation are discussed through an example:

2. 1. Creating Parametric Behavior Models

In general, the steps required for this process include:

Step 1- Associating the model fragment with a PE

Step 2- Adding model fragments to the initial model fragment

Step 3- Connecting model fragments to each other

Step 4 - Adding constraints

Following the procedure is explained in detail. First, the genera procedure is discussed; later the tool's graphical outputs are displayed.

2.1.1. Graphical Representation - General Procedure:

As the first step towards answering a well-defined feasibility question, the designer needs to develop a consistent parametric behavior model that includes all the parameters affecting the objective variable (the variable whose calculated value determines whether the feasibility question is satisfied or not).

As an instance, in order to know how much force does a hydraulic press generates (feasibility question), the designer needs to create a model that consists of at least *Force* as the parameter. Since force is the objective variable and should be calculated, he needs to associate it with a PE or COTS model. Therefore, based on the available information, he selects the Pascal pressure law which has three parameters: Pressure, area of the press, and force. Therefore, the PVs of the press would include:

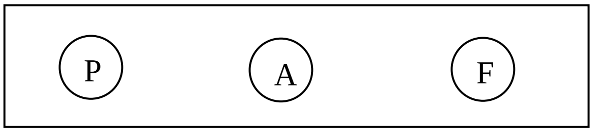


Figure 21 - Parameters of a Sub-System

It is noteworthy that the PVs could be associated with either a pre-defined relation (association with artifacts/Physics principles COTS/ PE) or by a manually input a relationship that the user inputs (Such a relationship needs to follow the PV ontology of the tool to be consistent with the rest of the model).

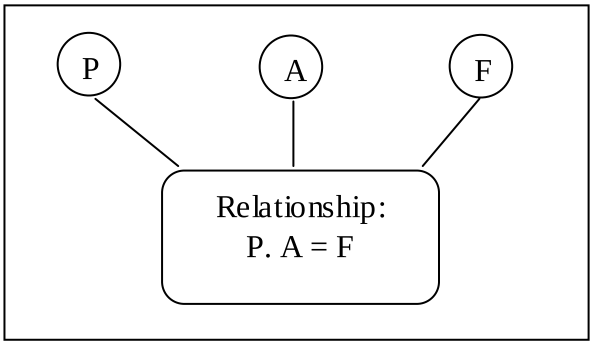


Figure 22 - A complete Sub-System with Relations and Parameters

The above box represents a model fragment of a behavior that has three parameters and a relationship which determines how the parameters are related.

At this point, the designer could add other behaviors/functions and connect them through shared parameters; i.e., this model could be added to another model and a larger problem is simulated. For instance, in order to incorporate the above hydraulic press in a press machine, the user needs to calculate how much does a steel bar deflect after being pressed by the hydraulic press; he would have to update the initial behavior model to:

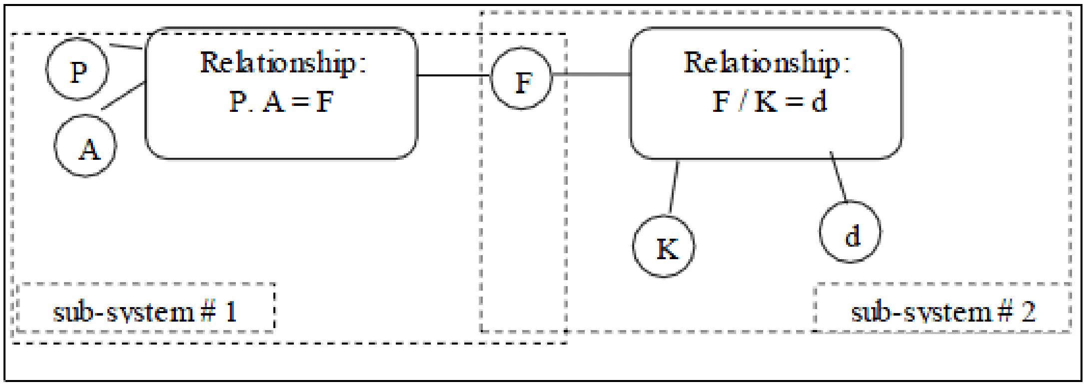


Figure 23 - Parametric Behavior Model of Two Connected Sub-Systems

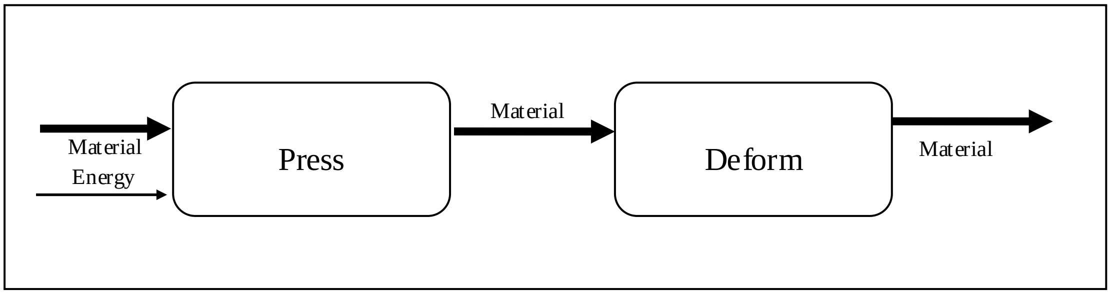


Figure 24 - Equivalent function structure of the hydraulic press model for answering the question on whether or not the press is capable of deflecting a bar to a desired amount

Now the designer can add constraints to the model. For instance, for the above problem, the designer might decide to add a constraint on controlling the numerical value of the cross sectional area keeping it above 0 in order to avoid geometric conflicts. Therefore, the behavior model would look like:

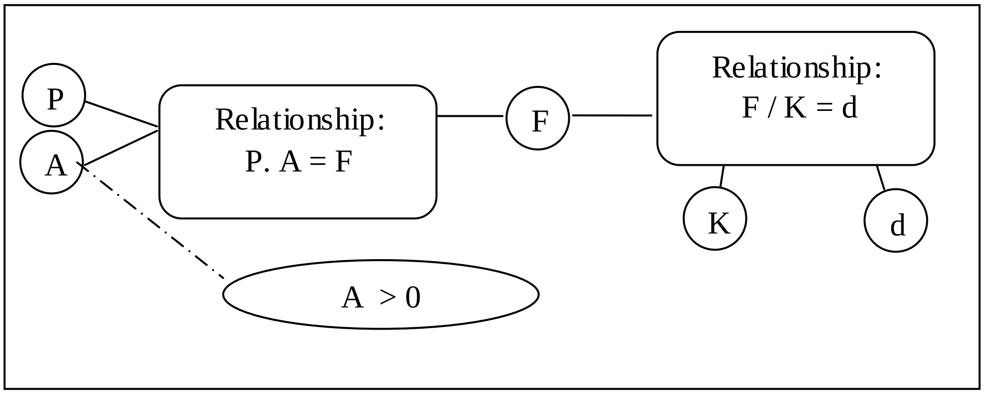


Figure 25 - A Complete Behavior Model with Parameters, Relations, and Constraints

2.1.2. Model Representation - The Tool's Output:

In this section, the model discussed above is generated using the tool:

Step 1- Associating the initial model fragment with a PE:

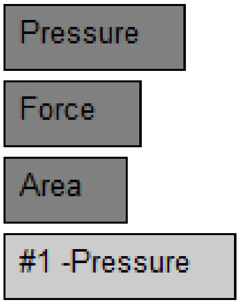


Figure 26 - Model Fragment Representation by the Tool's GUI

Step 2- Adding model fragments to the above model fragment:

Figure 27 -Model Fragments Inserted into One Model

Step 3- Connecting model fragments to each other:

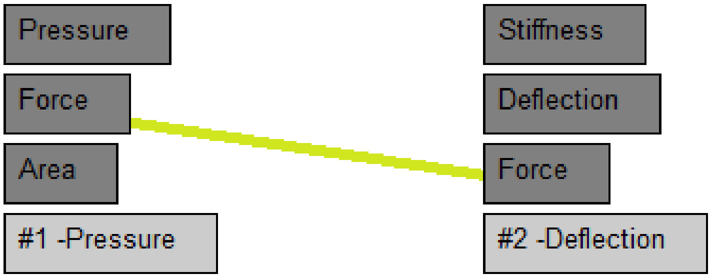
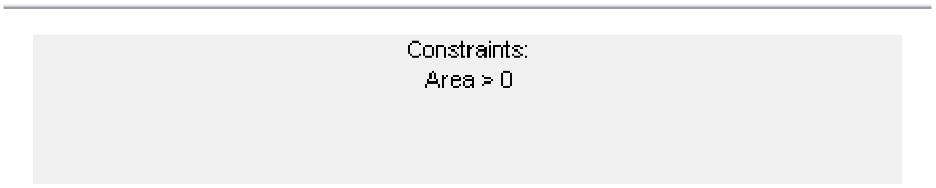


Figure 28 - Model fragments connected to each other (New model generated)

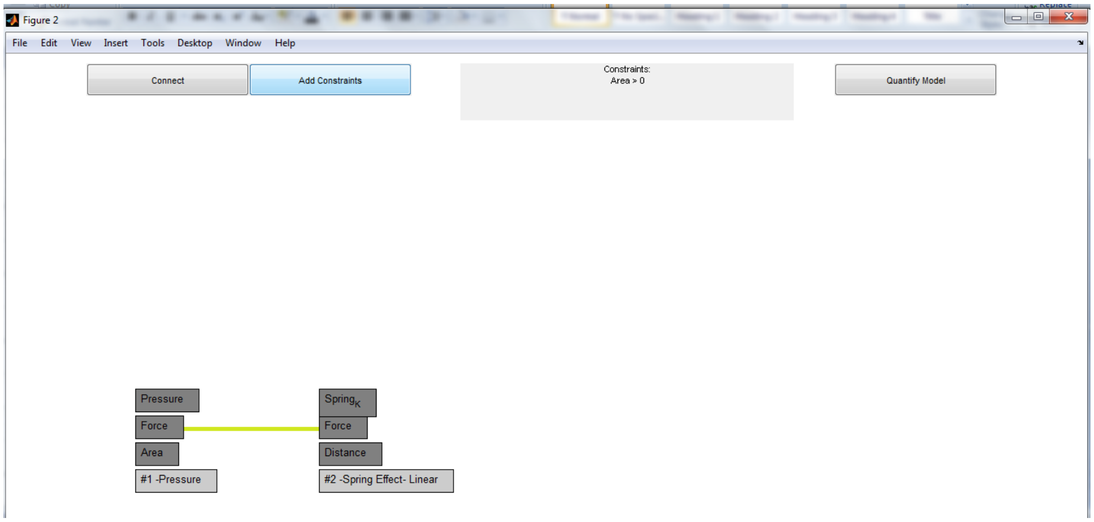
Step 4- Adding constraints:



Constraints:
Area > 0

Figure 29 - Constraints added to the model

Overall graphical representation would look like:



Constraints:
Area > 0

Figure 30 - Total model

2.2.1. Mathematical Formulation - General Procedure:

Step 1- Associating the model fragment with a PE

For the example shown above, the mathematical formulation would consist of one equation initially,

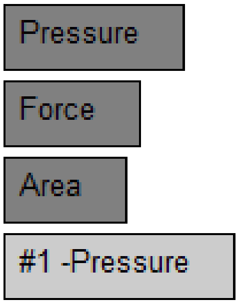
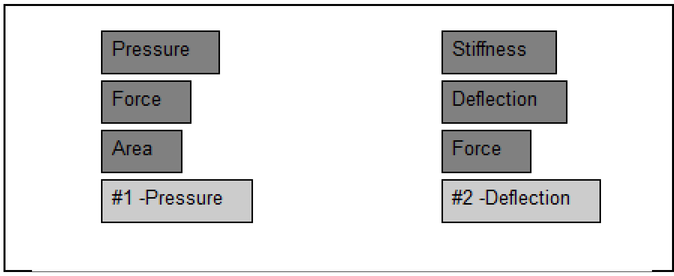
	$Force - Pressure * Area = 0$ $\rightarrow$ $x(1) - x(2) * x(3) = 0$

Figure 31 - Mathematical formulation - Step # 1

Step 2- Adding model fragments to the initial model fragment

As model is being completed more fragments are being added to it; here two physical relations are used to represent the model:



$$\begin{cases} Force - Pressure * Area = 0 \\ Force - Stiffness * Deflection = 0 \end{cases}$$

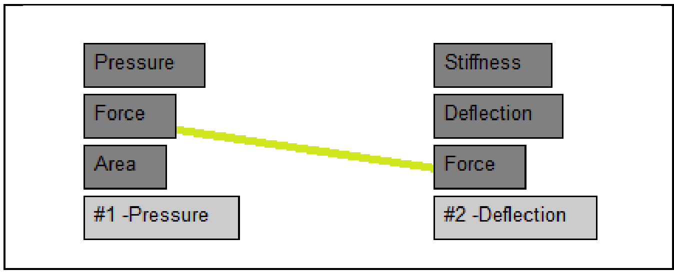
$$\rightarrow$$

$$\begin{cases} x(1) - x(2) * x(3) = 0 \\ x(4) - x(5) * x(6) = 0 \end{cases}$$

Figure 32 -Mathematical formulation - Step # 2

Step 3- Connecting model fragments to each other

As model fragments are linked together, some variables in the mathematical formulation get eliminated (shared variables):



$$\begin{cases} Force - Pressure * Area = 0 \\ Force - Stiffness * Deflection = 0 \end{cases}$$

$$\rightarrow [x(1) = x(4)]$$

$$\begin{cases} x(1) - x(2) * x(3) = 0 \\ x(1) - x(5) * x(6) = 0 \end{cases}$$

Figure 33 -Mathematical formulation - Step # 3

Section III. Model Solution

Once the problem model is developed as above, it should be quantified and solved. The solution steps are as following:

Step 1 - Specifying the measures of goodness

Step 2 - Quantifying the model

Step 3 - Checking Solvability

Step 4 - Solving the model

Following, all four steps are discussed in detail; an example is also provided to show how these steps are performed:

Step 1: Specifying the Measures of Goodness (Determining Model Inputs and Outputs)

Mathematically speaking, this step translates to re-ordering the mathematical formulation so that the feasibility criteria is expressed as a function of the some/ all of the other parameters. In other words, at this point, the developed model is a network of PVs that are related to each other by some physical relations. In order to solve it, the mathematical network of equations should be re-organized as a number of known variables (problem inputs) and a number of unknown variables (problem outputs).

The designation of model input and outputs is based on the feasibility question asked and the available input information of the design problem.

For the example problem mentioned above, the feasibility questions were:

1. How much force would be generated by the press?
2. How much deflection would the press cause in the bar?

Therefore, the current state of formulation would be converted to

$$\begin{cases} Force - Pressure * Area = 0 \\ Force - Stiffness * Deflection = 0 \end{cases} \quad or \quad \begin{cases} x(1) - x(2) * x(3) = 0 \\ x(1) - x(5) * x(6) = 0 \end{cases}$$

$$\rightarrow \begin{cases} Force = Pressure * Area \\ Deflection = \dfrac{Force}{Stiffness} \end{cases} \quad or \quad \begin{cases} x(1) = f\big(x(2), x(3)\big) \\ x(6) = f\big(\, x(1), x(5)\big) \end{cases}$$

Step 2: Quantifying the Model

Once the model is reorganized to be solved for the feasibility question asked, the user should quantify the known variables based on design specifications or simply estimations that he could provide.

For the example provided earlier, the user needs to quantify 3 parameters of pressure, area, and stiffness to be able to solve the system of equations:

Model inputs	***Model outputs***
Area, Pressure, Stiffness [x(2), x(3), x(5)]	Force, Deflection [x(1), x(6)]

Within this step, it is ensured that the shared variables are not designated as unknown variables and are transferred in the model as the model is being solved.

Step 3: Checking Solvability of the Model

At this stage the model should be checked for solvability to ensure that the system of the equations is solvable with the current state of the known and unknown variables. If all the required inputs are provided, then the model is ready to be solved.

In order to be solvable, the model should be appropriately constrained. In most cases, in order for the model to be solvable, the number of model variables (n) should not exceed or be less than the number of available relations between the model variables - relations here consist of the PE, COTS (m) and shared parameter between the model fragments (p)-[10]. Therefore, the status of solvability is determined by:

---

[10] The purpose of checking the solvability of the model is to quickly check the correct connectivity of the parameters and equations defining the model and it is in fact just a rough check of the system of equations created by the user. It cannot ensure obtaining a unique result.

In order to ensure solvability and derivation of a unique result, the designer needs to solve the model and then manually select from a number of solution cases in case of multiple solutions or re-quantify model

$$\begin{cases} n < m + p \rightarrow \ Model\ is\ overconstrained\ \ \rightarrow\ not\ solvable \\ n = m + p\ \ \rightarrow Model\ is\ appropriately\ constrained\ \ \rightarrow solvable \\ n > m + p\ \ \rightarrow Model\ is\ underconstrained\ \ \rightarrow not\ solvable \end{cases}$$

Step 4: Solving the Model

If the mathematical formulation is solvable, the quantified equations are solved for the feasibility objective parameter and the results are presented to the user.

Since the mathematical formulation consists only of linear equations, the solver used for solving the equations is MATLAB's Solve command which uses MATLAB's the symbolic toolbox to solve the system of equations.

Part 2. The Tool's User Interface

As discussed in detail in part 1 of this chapter, the tool has two main sub-modules for:

1. Creating the behavior models - The Model Developer
2. Analyzing the behavior models - The Model Analyzer

Following, the user interfaces of these modules are discussed in detail:

Section I - The Model Developer

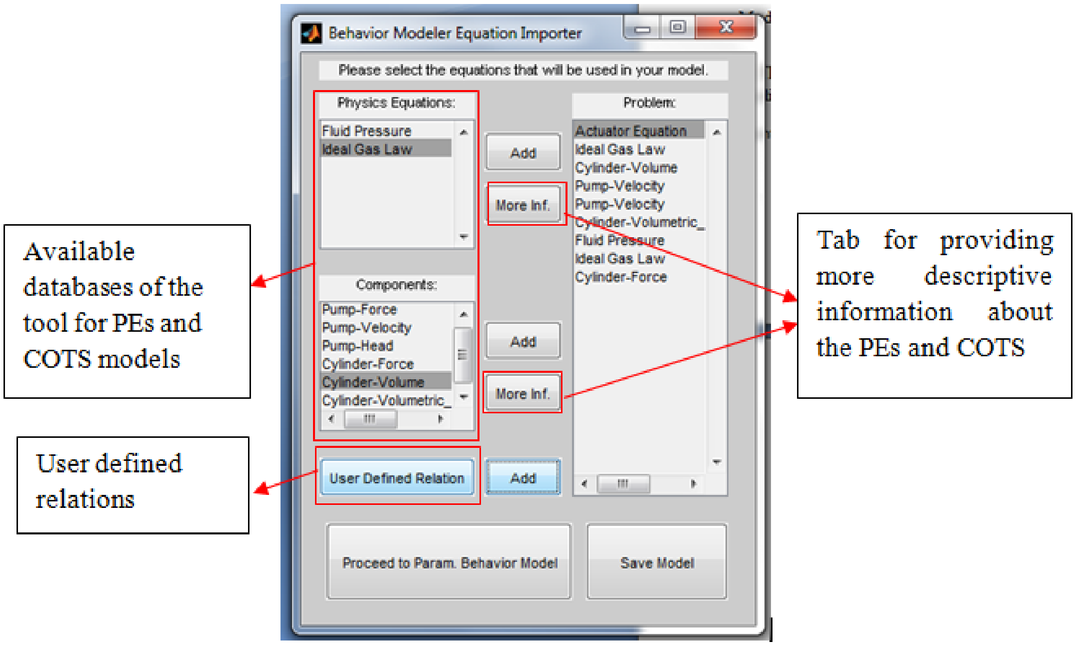
This module supports association of the concepts with PEs, COTS models and user defined relations. It also generates graphical representation of the models generated.

The main window of the tool is displayed below; this window allows: Association of the model with PEs, COTS models (two list boxes shown in the left are connected to the tool libraries on PEs and COTS models), and user defined relations. The user can add any of these relations to his problem model.

---

input parameters and re-solve in case of zero solutions. It is worthy to note that such cases are very rare in actual design practice.

All of the relations that are imported to the model are added to the left list box (displaying all the relations that together generate the problem model).



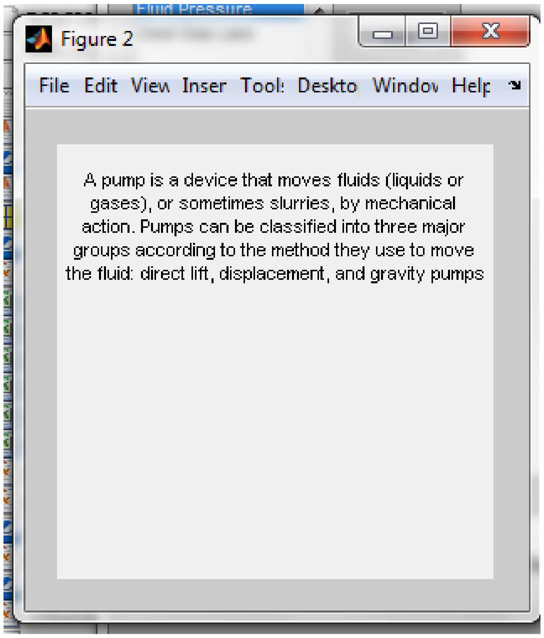
Available databases of the tool for PEs and COTS models

User defined relations

Tab for providing more descriptive information about the PEs and COTS

Figure 34 - Parametric Behavior Modeler/ Analyzer Main Screen

This module helps designers in selecting the right Physics principle/component set by providing general information about them:



A pump is a device that moves fluids (liquids or gases), or sometimes slurries, by mechanical action. Pumps can be classified into three major groups according to the method they use to move the fluid: direct lift, displacement, and gravity pumps

Figure 35 - More Information about the Component Sets/ Physics Principles

Once all of the equations that describe the model are imported to the problem model, the user could proceed to creating the graphical representation of the problem model. This module would allow displaying PVs and the links between them.

In this representation, PVs are displayed (categorized with the equations that describe the relationship between them). Figure below displays a representation:

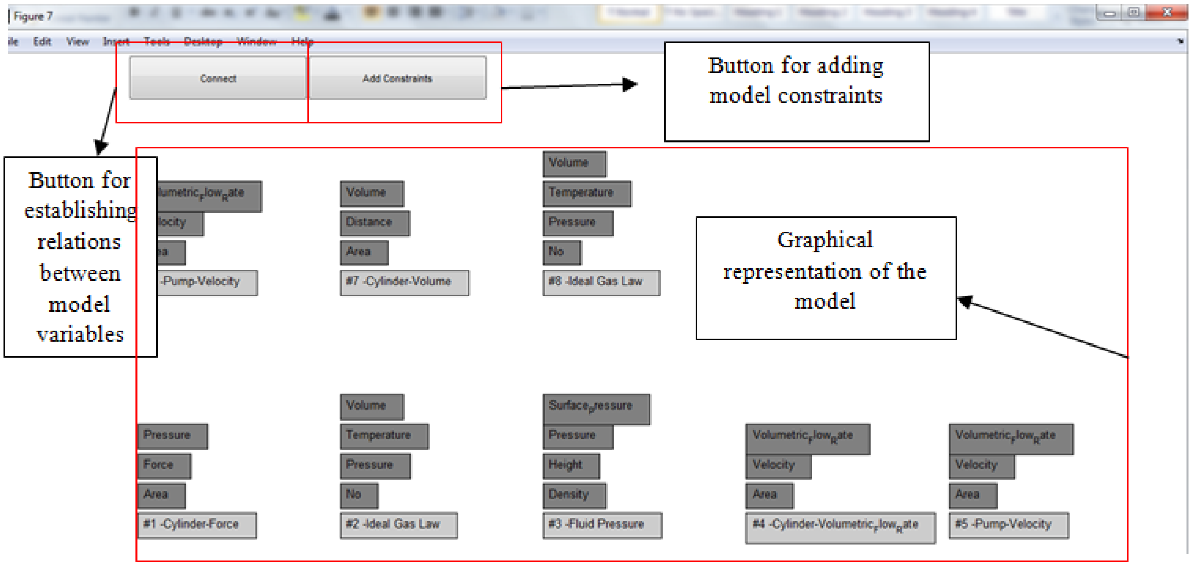


Figure 36 - Parametric Behavior Model's Graphical Representation

At this point, the user could establish the relations between the variables, specifying shared parameters. The "connect" button on the top of the screen could be used to establish the "shared status" between the variables. Constraints that apply to the model could also be added to the model using the "Add Constraint" button. Figure below shows the interface for establishing the connections between the PVs:

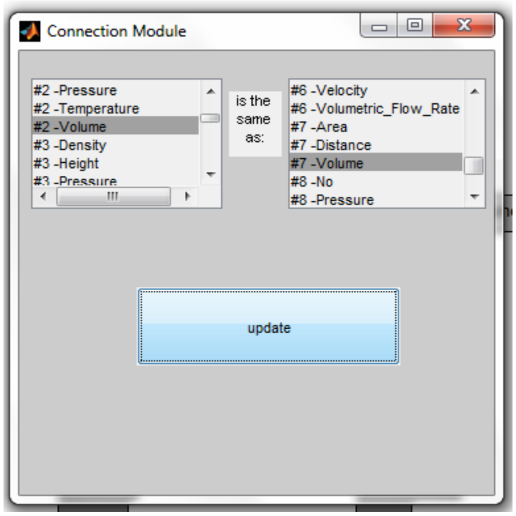


Figure 37 - The Connect Module for Linking Shared Parameters

As stated before, connections could be established if and only if the variables are of the same type (compatible). Once approved of compatibility, the connections between the variables are represented by lines between the PVs. Figure below displays the connections between the PVs by lines:

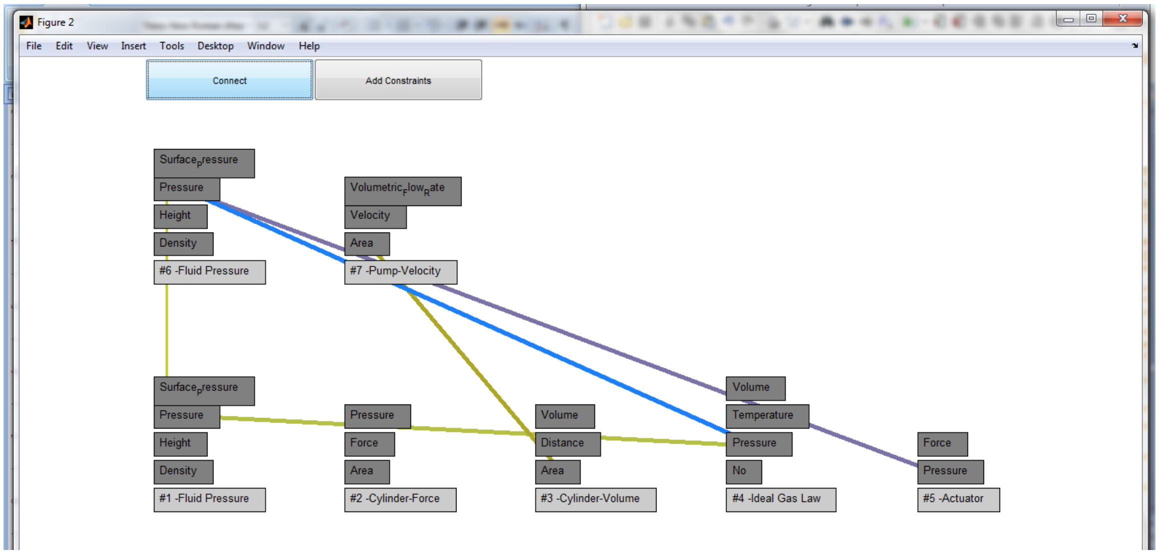
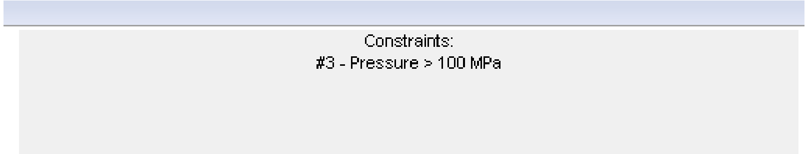


Figure 38 -Behavior Model's Shared Parameters Connected by Lines



Constraints:
#3 - Pressure > 100 MPa

Figure 39 - Constraints' Representation in the Parametric Behavior Modeler's GUI

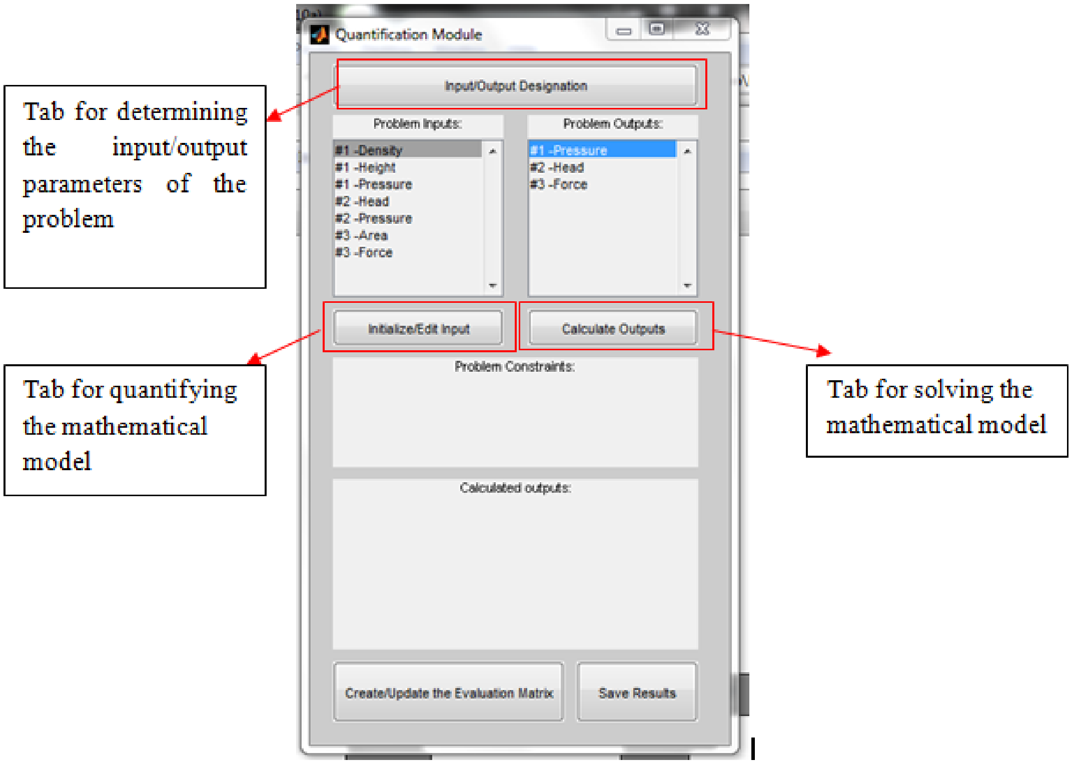
Section II - The Model Analyzer

Once the model is fully developed, the designer could proceed to analysis. In order to do so, he still needs to:

1. Define the purpose of the model by determining the measure(s) of goodness
2. Complete his model

In order to do the first task, he needs to determine the PVs' that are affect the feasibility criteria (measures of goodness). Therefore, for each equation describing the behavior, he needs to determine the known and unknown variables. Figure 40 shows the interface that performs this task; the tab "input/output designation" connects this module to the interface that performs this task .

After determining the measure of goodness, the user would still need to complete his model. He needs to quantify the PVs that are specified as problem variables (inputs). The tab "Initialize inputs" shown in Figure 40 directs the user to the module that performs this task.



Tab for determining the input/output parameters of the problem

Tab for quantifying the mathematical model

Tab for solving the mathematical model

Figure 40 - The Model Quantification Module

At this stage, the model should be examined for solvability. If adequately defined (done via the input/output designation) and initialized (done via the initialize input), the user could solve the problem and view the results. In the figure displayed below, the tab for checking solvability is highlighted:

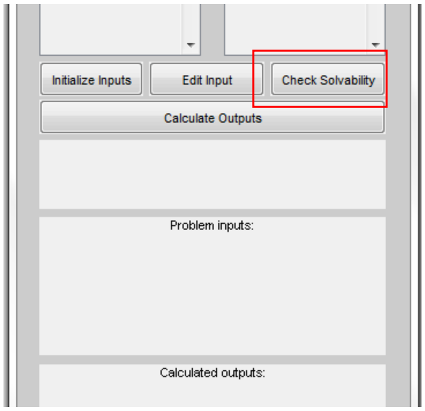
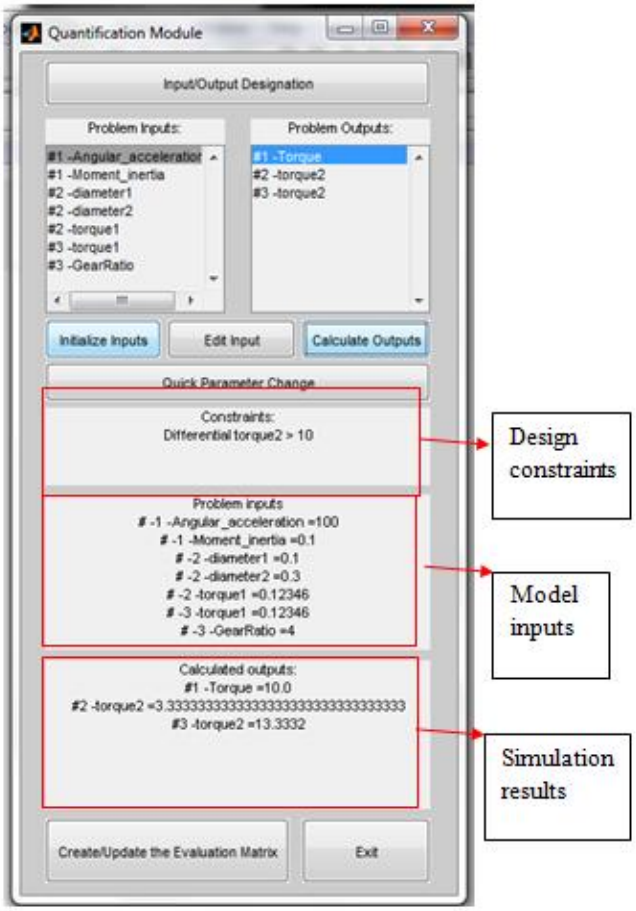


Figure 41 - The tab for checking solvability

Of course, after checking the constraints, he might decide to modify some of the inputs, re-examine solvability, and re-solve the model. Once solution is completed, he could request mapping of the output/ any other design criteria to the evaluation matrix as a design criteria and evaluate the developed concept's performance.

A solved model would be reflected in the GUI as:



Problem Inputs:
#1 -Angular_acceleratior
#1 -Moment_inertia
#2 -diameter1
#2 -diameter2
#2 -torque1
#3 -torque1
#3 -GearRatio

Problem Outputs:
#1 -Torque
#2 -torque2
#3 -torque2

```
Constraints:
Differential torque2 > 10
```

```
Problem inputs
# -1 -Angular_acceleration =100
# -1 -Moment_inertia =0.1
# -2 -diameter1 =0.1
# -2 -diameter2 =0.3
# -2 -torque1 =0.12346
# -3 -torque1 =0.12346
# -3 -GearRatio =4
```

```
Calculated outputs:
#1 -Torque =10.0
#2 -torque2 =3.33333333333333333333333333333333
#3 -torque2 =13.3332
```

Figure 42 - Results of the Simulation Displayed

CHAPTER 6: CASE STUDIES

Using the tool for real-world design problems

A number of sample design problems are solved by the tool and the results are reported here to further display the advantage of using the tool in early design. The two first case studies of this chapter are the D#1 and D#2 projects of the Advanced Product Design and Development course lectured by Dr. Jami Shah at Arizona State University. The third one is on answering a system configuration design question which seeks the best power train configuration for a given vehicle.

Case Study - 1:

For this study, the class project (D1) of the MAE 540 was used. The project required developing an autonomous vehicle that can patrol a figure '8' path around two boxes of maximum size 1'x1'x1' cubes along with other various design parameters.

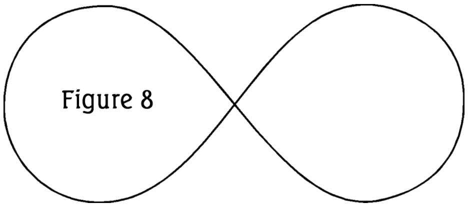


Figure 43 - Path to be taken

The feasibility of the design proposed by one of the students is surveyed by the tool proposed within this study. The procedure for this study and its results are reported below. The steps taken for this study are:

1. Acquiring the concept proposed
2. Reviewing the feasibility questions asked and converting them to PV- specific questions

3. Creating models from the design to answer the feasibility questions
4. Defining the measures of goodness
5. Quantifying the model
6. Checking solvability of the model
7. Solving the model and reviewing the results
8. Making changes and resolving (if desired)

Step # 1: Acquiring the design proposed

The figure displayed below shows the working principle of the concept vehicle:

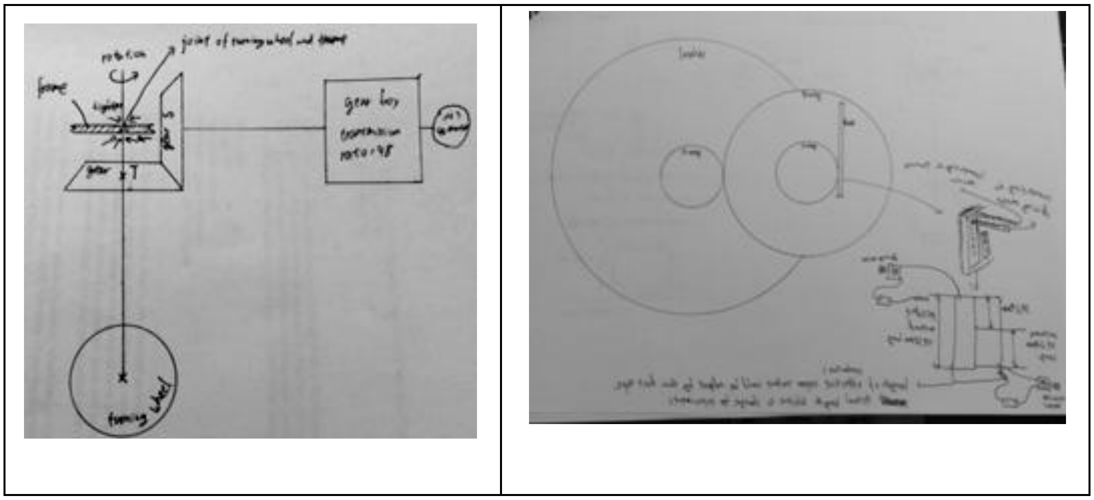


Figure 44 - Concept power train

The design consists of a motor, three gear sets, and wheels (one wheel is steered). It is supposed that the vehicle circles 18" circles around the boxes. In order to turn, at certain times, the turning wheel would rotate 33.75° to satisfy the requirements.

Assumptions made:

Length of the vehicle: 12", Total weight: 500 gr, Efficiency of the power train: 80%.

The motor selected is 273-047 which is a 9V motor with a maximum current of 115 mA and maximum power output of 1.035W. The car is equipped with 2 9V batteries of 1604A type which generate 565mAh in total. The wheel diameter is chosen as 6".

Step #2: Reviewing the feasibility questions

For this problem, checking feasibility would consist of checking whether the vehicle would be capable of moving and turning as desired. Therefore, the feasibility questions are:

1. Would the selected motor satisfy the power requirements for this vehicle?
2. Would the proposed steering be tight/loose enough to facilitate smooth turning of the vehicle?
3. Would the batteries supplied for the vehicle satisfy the power requirement?

The first question asked seeks whether or not the '*power*' generated by the engine would be sufficient for running the vehicle. The second question also seeks the value of the '*torque*' required for fastening the joints that allow steering. The third question seeks whether the capacity of the batteries (energy production capacity of the battery) would be sufficient for the test.

The procedure and the results for modeling and simulating each of the questions are discussed separately below:

Question -1: Would the power generated by this engine be sufficient for running the vehicle?

Step #3: Creating models for the feasibility questions asked

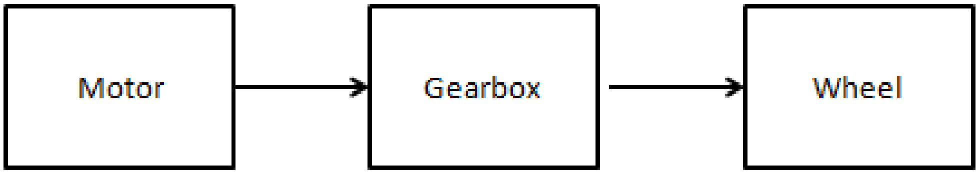


Figure 45 - The schematic of the power train

Assumptions made for this problem are:

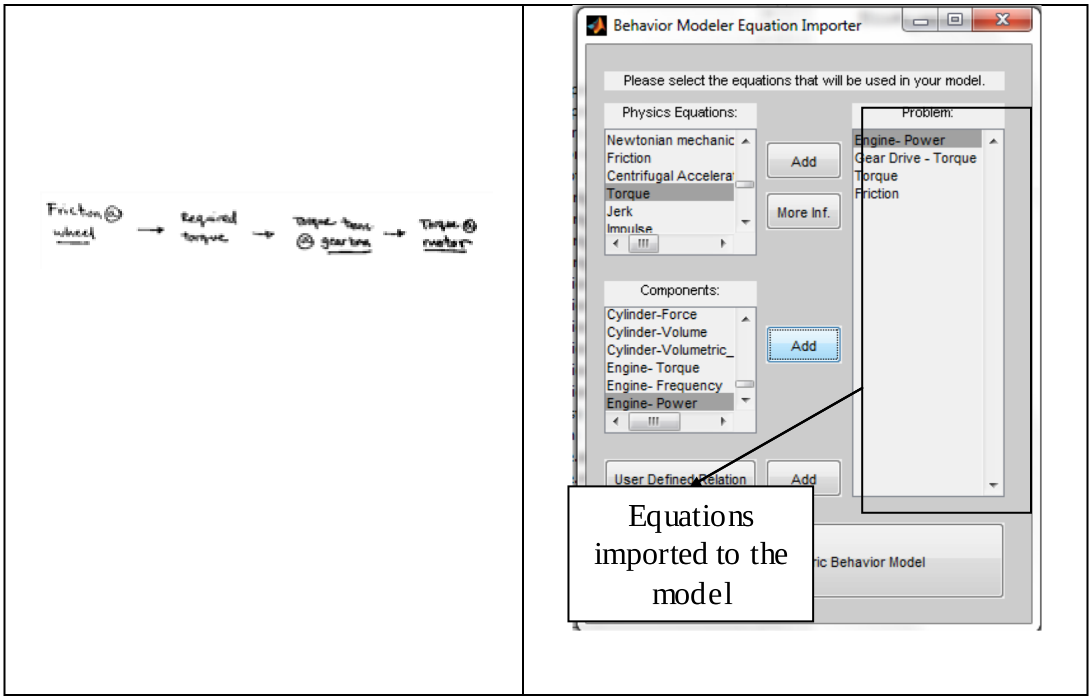
Constant rolling resistance of 0.08

Vehicle speed equal to 0.6 m/s

Overall transmission ration: 72

The procedure for creating the problem model is displayed in the following:

1. Importing relevant equations



Please select the equations that will be used in your model.

Equations imported to the model

Figure 46 - Developing the model- step#1

2. Creating the graphical representation of the model

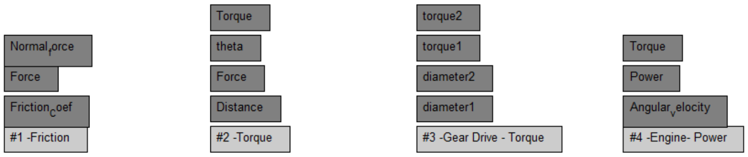


Figure 47 - Graphical representation of the model - PVs are still not connected to each other

3. Connecting the model fragments through ports (PVs)

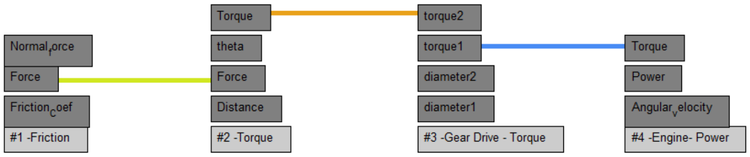
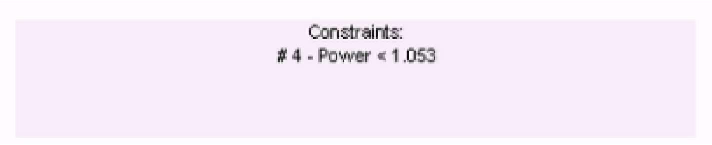


Figure 48 - Graphical representation of the concept model - PVs are connected

4. Adding the problem constraints

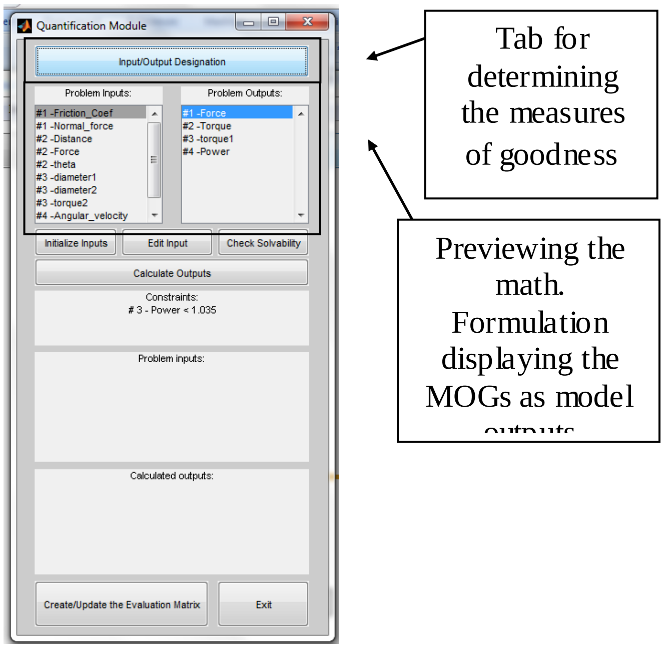


Constraints:
4 - Power < 1.053

Figure 49 - Constraints of the problem

Step #4: Determining the measures of goodness

In this step, the user determines the measures of goodness in his model. Based on the feasibility question asked and the available inputs, the measures of goodness are defined as following.



Problem Inputs:
#1 -Friction_Coef
#1 -Normal_force
#2 -Distance
#2 -Force
#2 -theta
#3 -diameter1
#3 -diameter2
#3 -torque2
#4 -Angular_velocity

Problem Outputs:
#1 -Force
#2 -Torque
#3 -torque1
#4 -Power

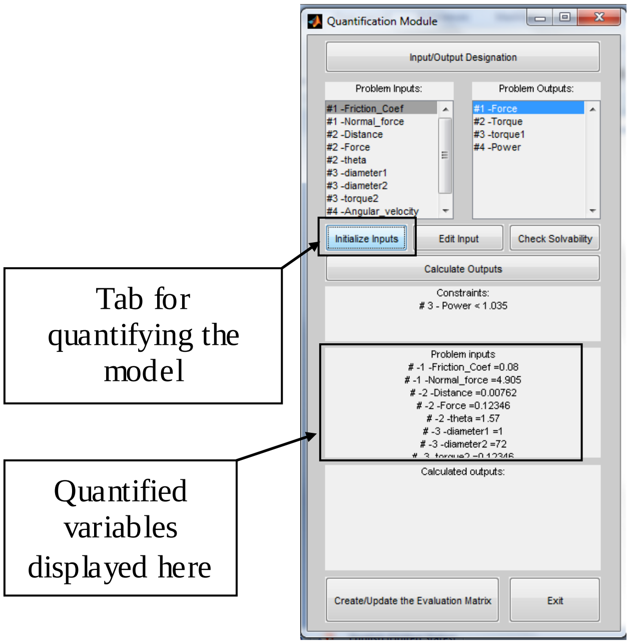
Constraints:
3 - Power < 1.035

Tab for determining the measures of goodness

Previewing the math. Formulation displaying the MOGs as model outputs

Figure 50 - Determining the measures of goodness and them being reflected in the tool GUI

Step #5: Quantifying the model



Problem Inputs:
#1 -Friction_Coef
#1 -Normal_force
#2 -Distance
#2 -Force
#2 -theta
#3 -diameter1
#3 -diameter2
#3 -torque2
#4 -Angular_velocity

Problem Outputs:
#1 -Force
#2 -Torque
#3 -torque1
#4 -Power

Constraints:
3 - Power < 1.035

Problem inputs
-1 -Friction_Coef =0.08
-1 -Normal_force =4.905
-2 -Distance =0.00762
-2 -Force =0.12346
-2 -theta =1.57
-3 -diameter1 =1
-3 -diameter2 =72

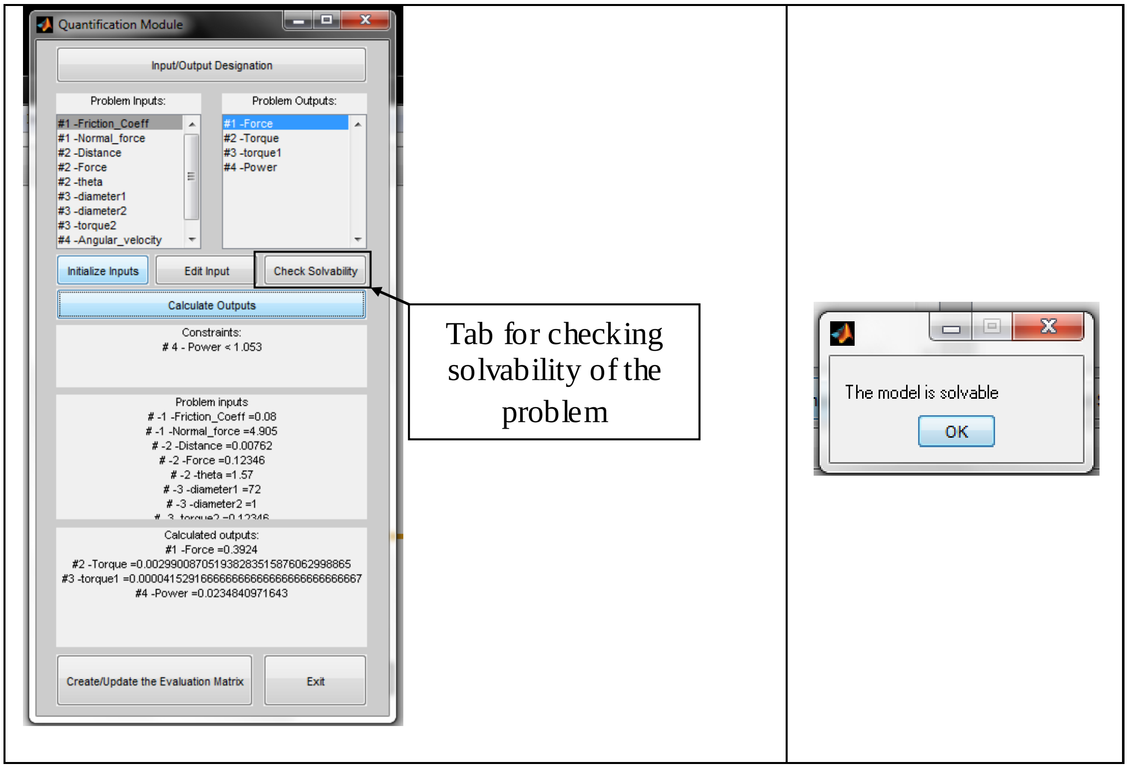
Tab for quantifying the model

Quantified variables displayed here

Figure 51 - Quantifying the model

Step #6: Checking solvability of the model

After initializing the quantities of the problem, it could be checked whether the formulated mathematical model is solvable or not. If solvable (as the message produced by the tool shows), the model could be solved:



```
#1 -Friction_Coeff
#1 -Normal_force
#2 -Distance
#2 -Force
#2 -theta
#3 -diameter1
#3 -diameter2
#3 -torque2
#4 -Angular_velocity

#1 -Force
#2 -Torque
#3 -torque1
#4 -Power

Constraints:
# 4 - Power < 1.053

Problem inputs
# -1 -Friction_Coeff =0.08
# -1 -Normal_force =4.905
# -2 -Distance =0.00762
# -2 -Force =0.12346
# -2 -theta =1.57
# -3 -diameter1 =72
# -3 -diameter2 =1
# -3 -torque2 =0.12346

Calculated outputs:
#1 -Force =0.3924
#2 -Torque =0.002990087051938283515876062998865
#3 -torque1 =0.00004152916666666666666666666667
#4 -Power =0.0234840971643
```

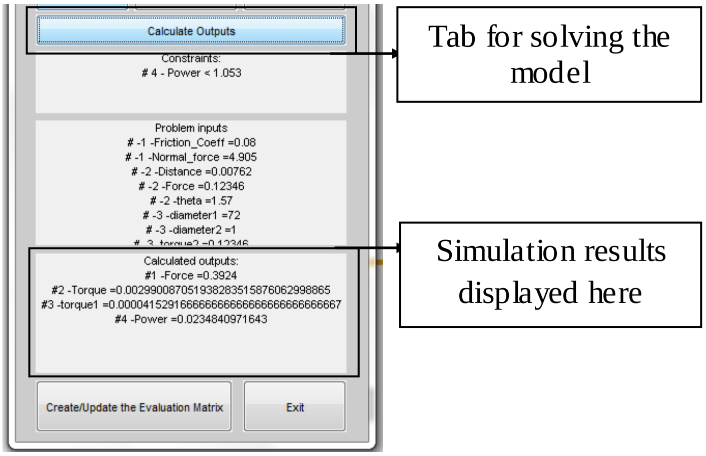
Tab for checking solvability of the problem

The model is solvable

Figure 52 - Checking solvability of the model

Step #7: Solving the model and reviewing the results

Once it is determined that the model is solved and PVs are calculated:



```
Constraints:
# 4 - Power < 1.053

Problem inputs
# -1 -Friction_Coeff =0.08
# -1 -Normal_force =4.905
# -2 -Distance =0.00762
# -2 -Force =0.12346
# -2 -theta =1.57
# -3 -diameter1 =72
# -3 -diameter2 =1
# -3 -torque2 =0.12346

Calculated outputs:
#1 -Force =0.3924
#2 -Torque =0.002990087051938283515876062998865
#3 -torque1 =0.0000415291666666666666666666666667
#4 -Power =0.0234840971643
```

Figure 53 - Simulation results

According to the simulation results, the power output of the engine is 0.023W which is smaller than the maximum output power of the motor (1.053W). Therefore, it could e concluded that the motor is capable of moving the vehicle.

Question - 2: Would the torque required for the steering joints allow smooth steering?

Step #3: Creating models for the feasibility questions asked

For this feasibility question, the user needs to consider only a portion of the vehicle that performs steering, just the second motor and the gear box used for steering. Therefore the model only needs to review the power generation of the motor and the resistance of the gear set that is used for steering:

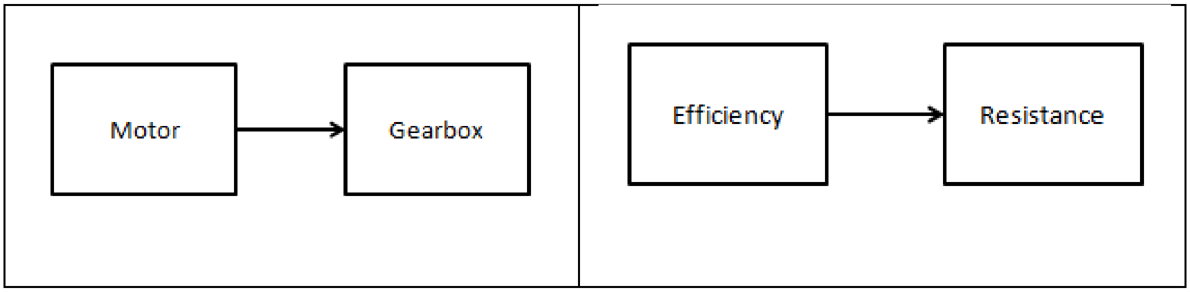


Figure 54 - The components used in the steering system and the problem model created for it

Step # 4: Creating models with the tool for the feasibility questions asked

For this model, the efficiency of the second motor and the resistivity of the gear set are modeled. For simulating the resistivity, the designer has used a self-defined physical relation (Torque = Power* Gear Ratio/ Angular velocity):

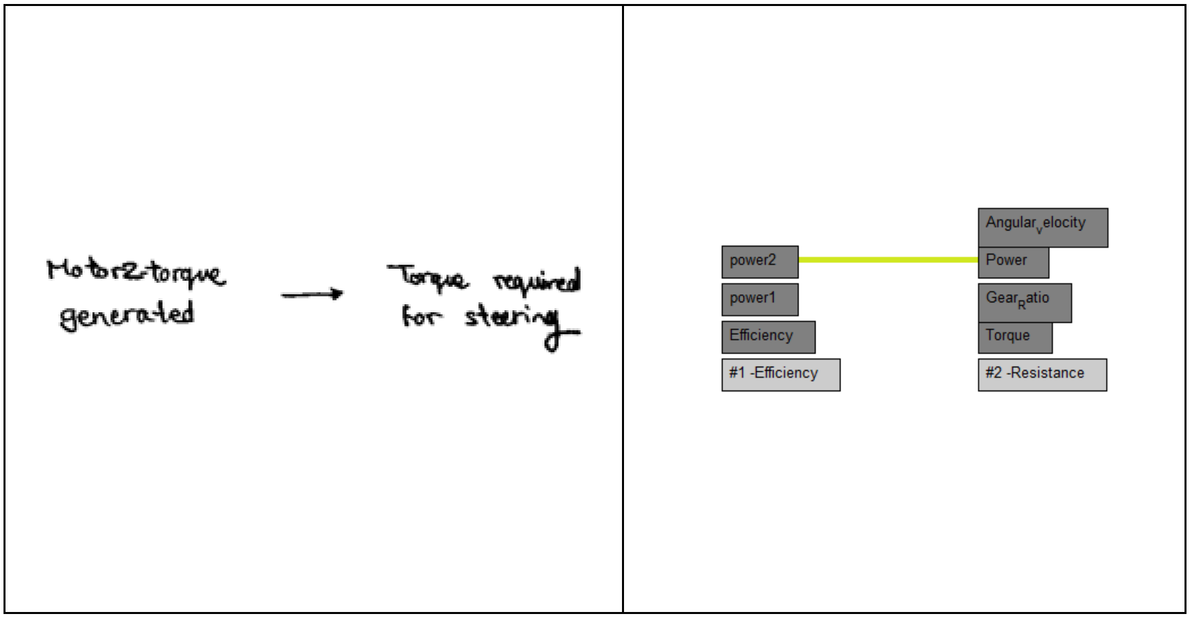
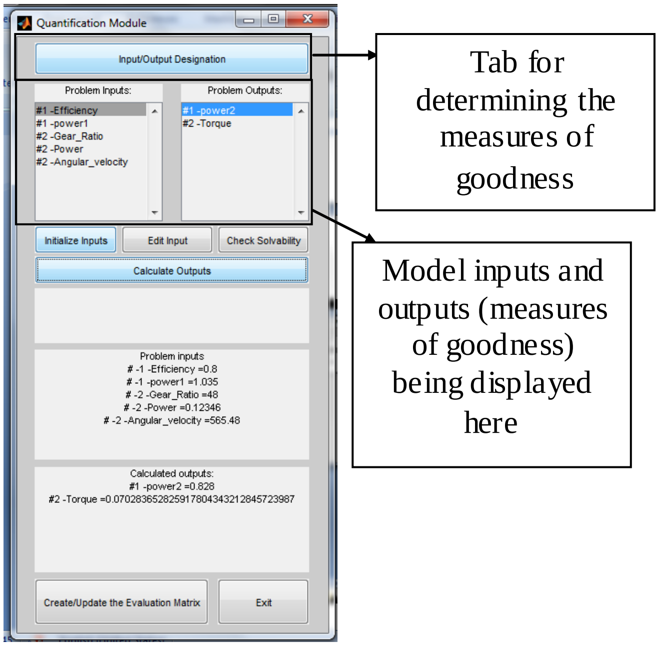


Figure 55 - The problem model created for the steering system

Step # 5: Defining the measures of goodness

In this problem, the goal is to find the torque required for fastening the joint in the steering system; therefore, the measure of goodness is the output torque.



Problem Inputs:
#1 -Efficiency
#1 -power1
#2 -Gear_Ratio
#2 -Power
#2 -Angular_velocity

Problem Outputs:
#1 -power2
#2 -Torque

Problem inputs
-1 -Efficiency =0.8
-1 -power1 =1.035
-2 -Gear_Ratio =48
-2 -Power =0.12346
-2 -Angular_velocity =565.48

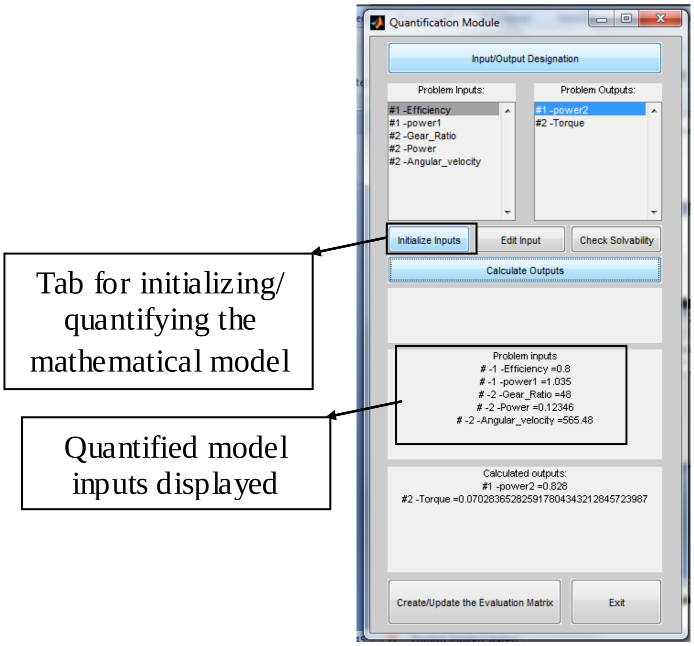
Calculated outputs:
#1 -power2 =0.828
#2 -Torque =0.070283652825917804343212845723987

Tab for determining the measures of goodness

Model inputs and outputs (measures of goodness) being displayed here

Figure 56 - Designating the measures of goodness for the simulation

Step # 6: Quantifying the model



Problem Inputs:
#1 -Efficiency
#1 -power1
#2 -Gear_Ratio
#2 -Power
#2 -Angular_velocity

Problem Outputs:
#1 -power2
#2 -Torque

Problem inputs
-1 -Efficiency =0.8
-1 -power1 =1.035
-2 -Gear_Ratio =48
-2 -Power =0.12346
-2 -Angular_velocity =565.48

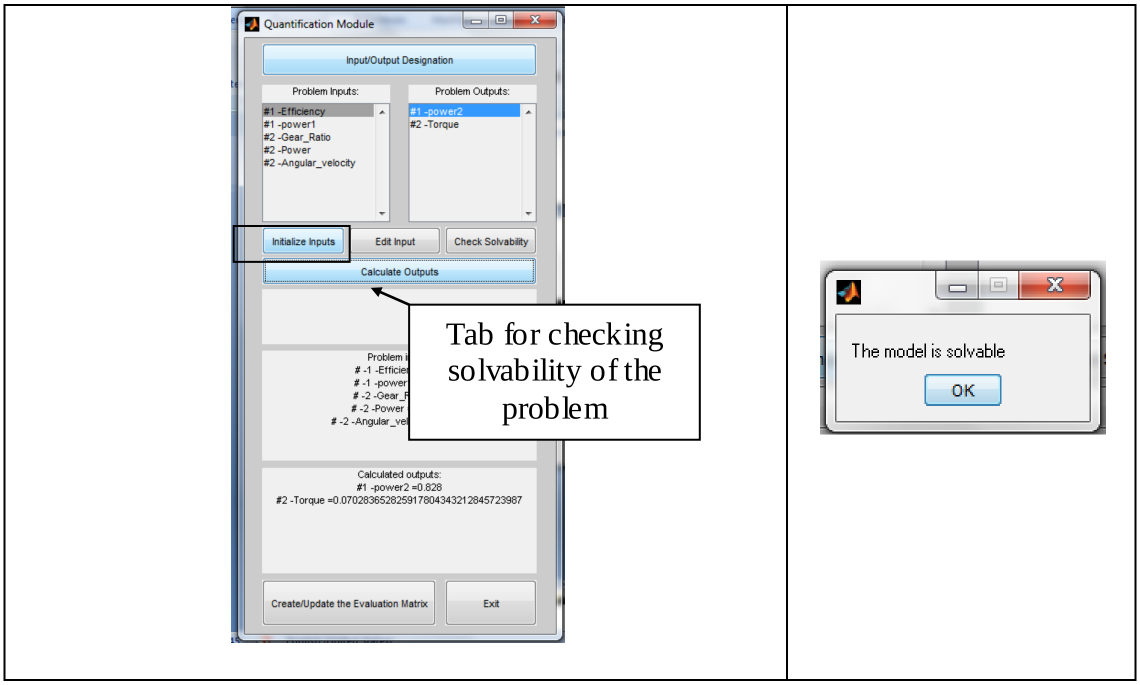
Calculated outputs:
#1 -power2 =0.828
#2 -Torque =0.070283652825917804343212845723987

Tab for initializing/ quantifying the mathematical model

Quantified model inputs displayed

Figure 57 - Initializing the mathematical model (quantifying the model)

Step # 7: Checking solvability of the model



#1 -Efficiency
#1 -power1
#2 -Gear_Ratio
#2 -Power
#2 -Angular_velocity

#1 -power2
#2 -Torque

Problem i
-1 -Efficier
-1 -power
-2 -Gear_F
-2 -Power
-2 -Angular_vel

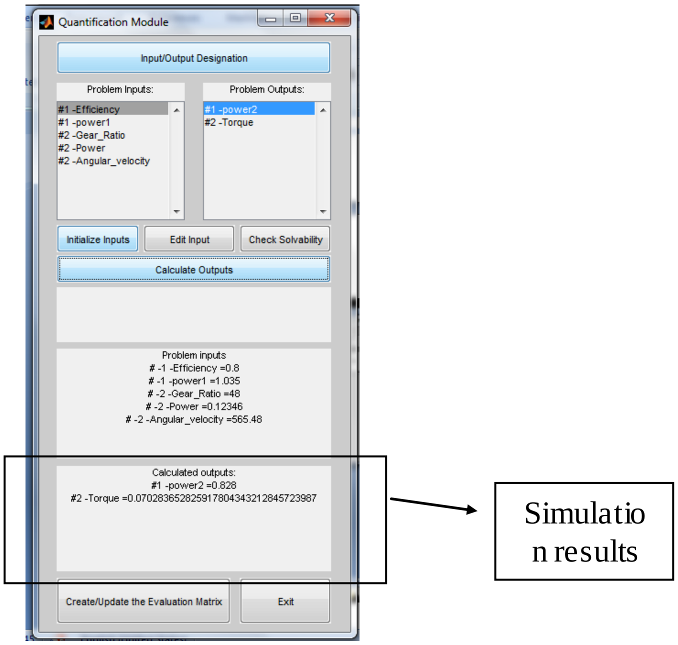
Calculated outputs:
#1 -power2 =0.828
#2 -Torque =0.070283652825917804343212845723987

Tab for checking solvability of the problem

The model is solvable

Figure 58 - Checking model solvability

Step# 8: Solving the model and reviewing the results



#1 -Efficiency
#1 -power1
#2 -Gear_Ratio
#2 -Power
#2 -Angular_velocity

#1 -power2
#2 -Torque

Problem inputs
-1 -Efficiency =0.8
-1 -power1 =1.035
-2 -Gear_Ratio =48
-2 -Power =0.12346
-2 -Angular_velocity =565.48

Calculated outputs:
#1 -power2 =0.828
#2 -Torque =0.070283652825917804343212845723987

Simulation results

Figure 59 - Simulation Results

According to the simulation results, the required torque for smooth steering is 0.07 Nm which is within the acceptable range.

Question - 2: Would the torque battery provide sufficient energy for the vehicle to run for a reasonable amount of time?

Step #3: Creating models for the feasibility questions asked

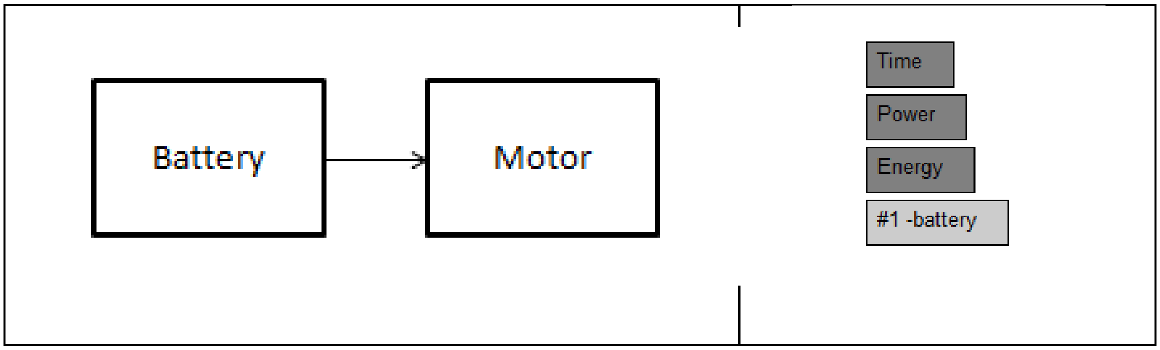


Figure 60 - Concept model for answering the third feasibility question

Step # 4: Determining the MOGs

The measure of goodness is the time for the vehicle to travel with the supplied power generation unit.

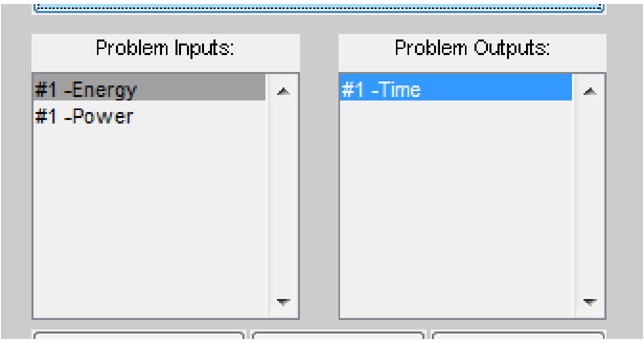
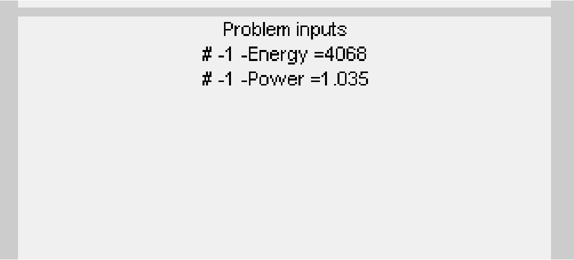


Figure 61 - Time as the measure of goodness for this model

Step #5: Quantifying the model

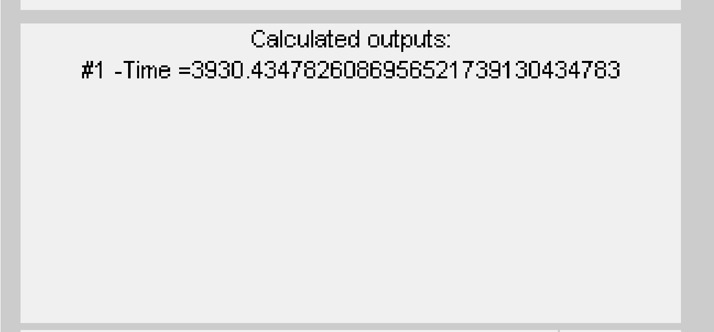
The value of the motor power and the energy output of the battery are inserted to the model.



```
Problem inputs
# -1 -Energy =4068
# -1 -Power =1.035
```

Figure 62 - Model inputs initialized

Step # 7: Checking solvability of the model and solving it results



```
Calculated outputs:
#1 -Time =3930.4347826086956521739130434783
```

Figure 63 - The simulation result

The time calculated for the batteries to run the engine is equal to 3930 seconds which is equal to 65 minutes approximately. The duration of batteries running the vehicle is equal to an hour which satisfies the design requirements.

Conclusion

For this project, there were two main issues that were surveyed:

1. To check the selected motor and see whether or not it could generate the required power for moving the vehicle and

2. To check the torque required for smooth steering

3. To check the energy requirement of the engine to run during the test

Based on the simulation results, the power train is capable of supporting the desired motion (required power less than the motor output). Also, vehicle could steer smoothly if the fastening torque for the steering joint is 0.07 Nm. Therefore, the concept vehicle passes the first phase of evaluation and more detailed design could be performed. The power requirement of the engine is also satisfied by the energy generated by the batteries. Therefore, considering the requirements for the test, the concept vehicle seems feasible.

*

Case Study - 2:

For this study, the class project (D2) of the MAE 540 was used. The project goal was to design and fabricate a vehicle that goes around a field and collects waste. In this study, the feasibility of the design concept proposed by one of the students is surveyed.

The procedure for this study and its results are reported below. The steps taken for this study are:

1. Acquiring the concept proposed
2. Reviewing the feasibility questions asked and converting them to PV- specific questions
3. Creating models from the design to answer the feasibility questions
4. Defining the measures of goodness
5. Quantifying the model
6. Checking solvability of the model
7. Solving the model and reviewing the results
8. Making changes and resolving (if desired)

Step # 1: Acquiring the design proposed

The design proposed by the student consists of a set of motor- gearbox and a wheel.

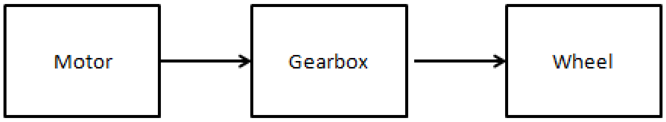


Figure 64 - The design proposed

The design specifications and assumptions made are:

Total weight = 450 gr (load included)

Load = 200 gr

Gradability = >= 35° (with full load)

Speed (flat surface) = 0.15 m/s with 70% load

Speed (Sloped surface) = 0.04 m/s with 100% load

Wheel diameter = 58 mm

Gear ratio = 196.7:1

Rolling resistance = 0.01

Gearbox efficiency = 80%

Motor output: 87.65/ efficiency = 0.577 gr.cm

Rotation speed of the vehicle: 7000 rpm

Length of the field (flat surface) = 3048 mm

Step # 2: Reviewing the feasibility questions asked

In order to check whether the vehicle is capable of collecting the waste in the allotted time, there are two main feasibility criteria to check:

1. Would the vehicle satisfy the time requirement of the contest (flat surface)?
2. Would the vehicle climb up the ramp in a reasonable amount of time?

3. Would the batteries generate enough energy for the vehicle to run during the test?

In order to survey the first question, the *'time'* required for traveling the contest field is examined. To survey the second question, the *'velocity'* generated by the power train is evaluated and compared with the requirement of the test.

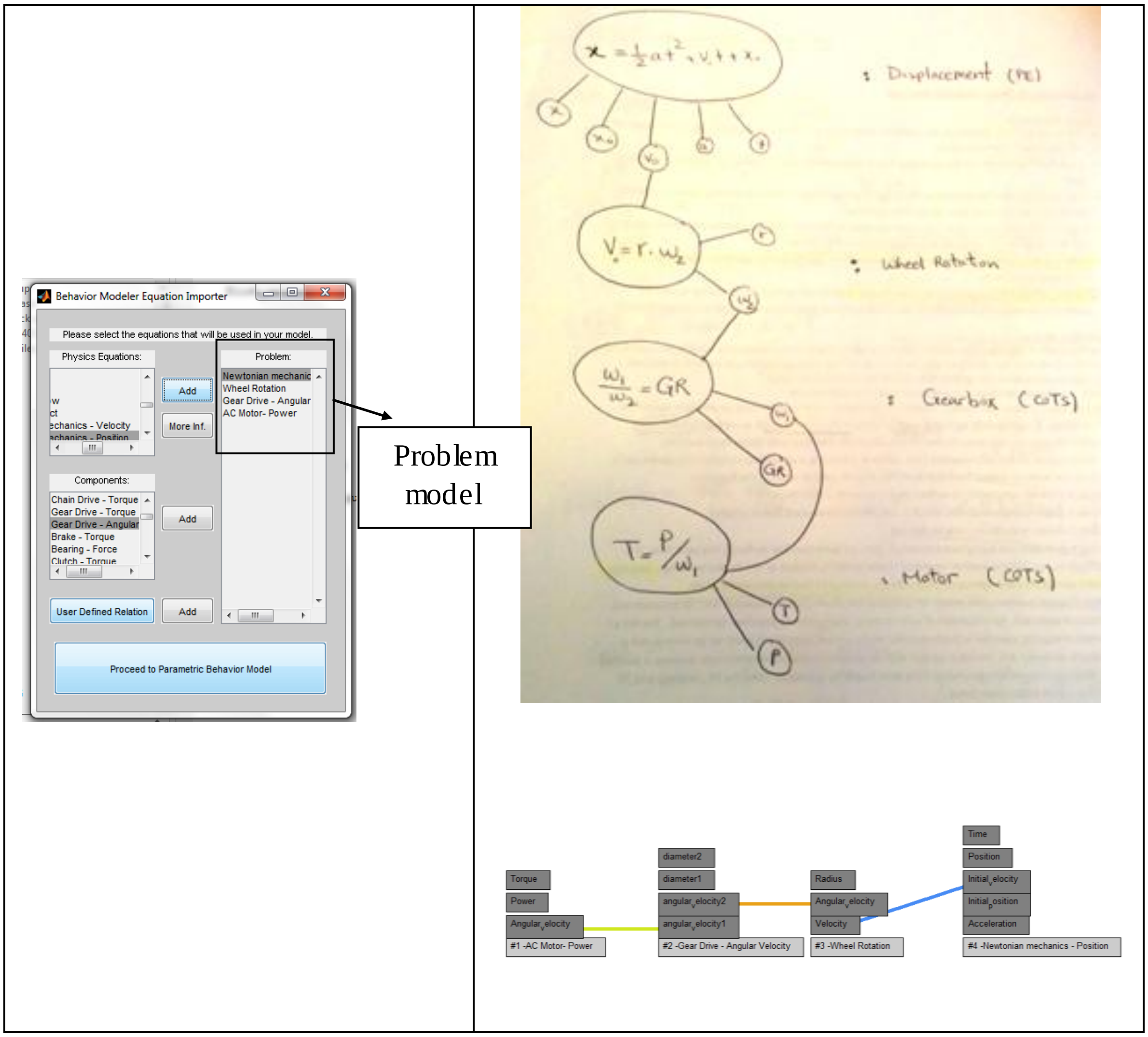
Following, each feasibility criterion is examined individually:

Question # 1 - How fast would the vehicle move?

Step - 3: Creating model from the concept to answer the feasibility questions

Since the requirement for this contest is to travel the contest area as fast as possible, the PV examined for this question is *'time'*.

To that end, a model is created to simulate the motor output to check whether or not the power train could produce the required velocity to move the vehicle within the allotted time. The model includes modeling the torque required at the wheels to overcome the friction, the torque transmission at the gear box and the torque generated at the motor. Below the model is created using a PE (Newtonian equation for calculating position), a user defined relationship for calculating toque from friction force, and two COTS models for gear drive and motor.



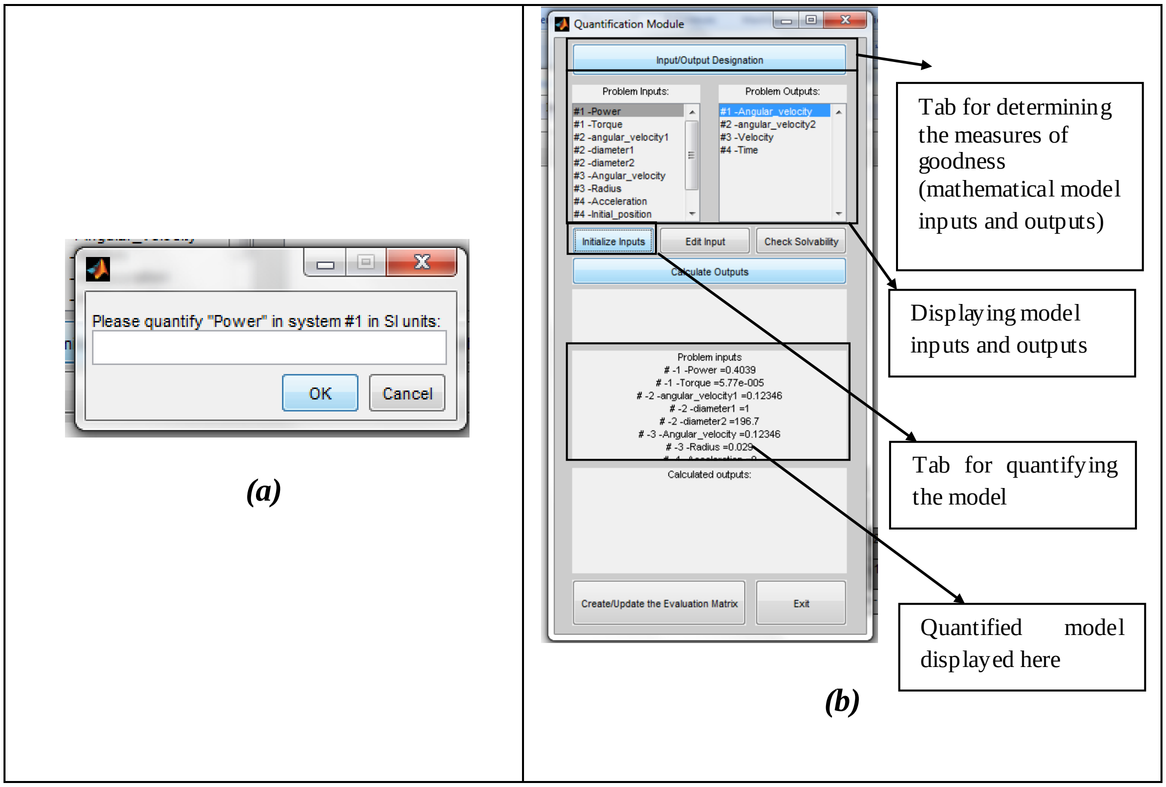
Please select the equations that will be used in your model.

Figure 65 - Model created for this feasibility question

Step -4: Defining the measures of goodness

As stated before, the measure of goodness for this model would be the time taken by the vehicle to travel the field.

Step -5: Quantifying the model



Please quantify "Power" in system #1 in SI units:

(a)

Problem Inputs:

- #1 -Power
- #1 -Torque
- #2 -angular_velocity1
- #2 -diameter1
- #2 -diameter2
- #3 -Angular_velocity
- #3 -Radius
- #4 -Acceleration
- #4 -Initial_position

Problem Outputs:

- #1 -Angular_velocity
- #2 -angular_velocity2
- #3 -Velocity
- #4 -Time

```
Problem inputs
# -1 -Power =0.4039
# -1 -Torque =5.77e-005
# -2 -angular_velocity1 =0.12346
# -2 -diameter1 =1
# -2 -diameter2 =196.7
# -3 -Angular_velocity =0.12346
# -3 -Radius =0.029
```

Tab for determining the measures of goodness (mathematical model inputs and outputs)

Displaying model inputs and outputs

Tab for quantifying the model

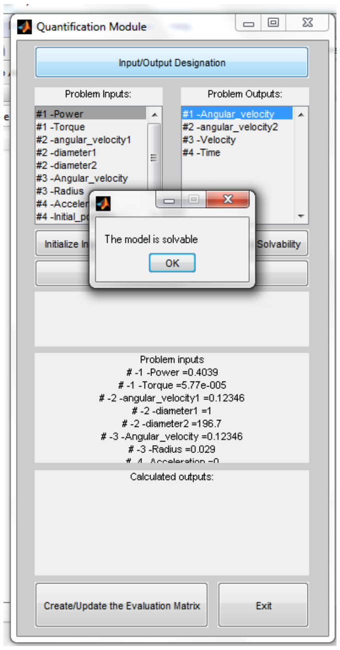
Quantified model displayed here

(b)

Figure 66 - Quantifying the problem model (mathematical inputs) (a: the tool module for quantification, b: quantified variables displayed)

Step -6: Checking solvability of the model

The purpose of this step is to check whether the mathematical model developed is solvable or not. If not, the quantification step or the model development should be reviewed for incorrect input.



```
#1 -Power
#1 -Torque
#2 -angular_velocity1
#2 -diameter1
#2 -diameter2
#3 -Angular_velocity
#3 -Radius
#4 -Acceler
#4 -Initial_p
```

```
#1 -Angular_velocity
#2 -angular_velocity2
#3 -Velocity
#4 -Time
```

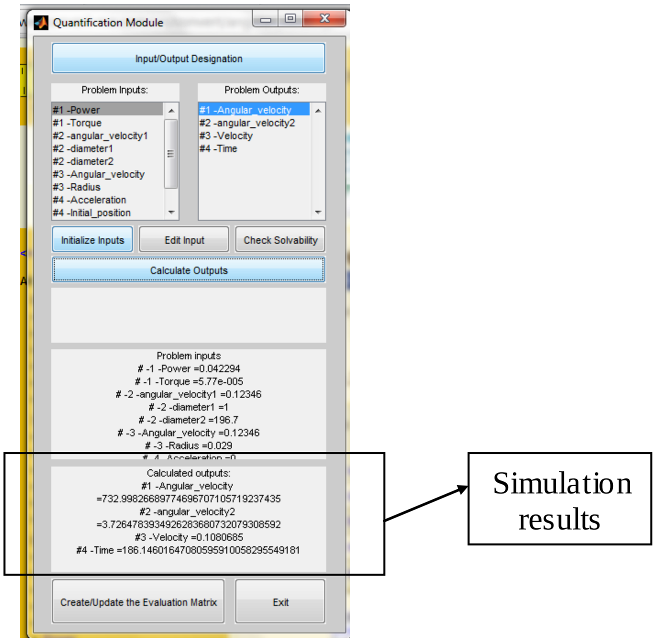
The model is solvable

```
Problem inputs
# -1 -Power =0.4039
# -1 -Torque =5.77e-005
# -2 -angular_velocity1 =0.12346
# -2 -diameter1 =1
# -2 -diameter2 =196.7
# -3 -Angular_velocity =0.12346
# -3 -Radius =0.029
# -4 -Acceleration =0
```

```
Calculated outputs:
```

Figure 67 - Checking the solvability of the model

Step -7: Solving the model and reviewing the results



```
#1 -Power
#1 -Torque
#2 -angular_velocity1
#2 -diameter1
#2 -diameter2
#3 -Angular_velocity
#3 -Radius
#4 -Acceleration
#4 -Initial_position
```

```
#1 -Angular_velocity
#2 -angular_velocity2
#3 -Velocity
#4 -Time
```

```
Problem inputs
# -1 -Power =0.042294
# -1 -Torque =5.77e-005
# -2 -angular_velocity1 =0.12346
# -2 -diameter1 =1
# -2 -diameter2 =196.7
# -3 -Angular_velocity =0.12346
# -3 -Radius =0.029
# -4 -Acceleration =0
```

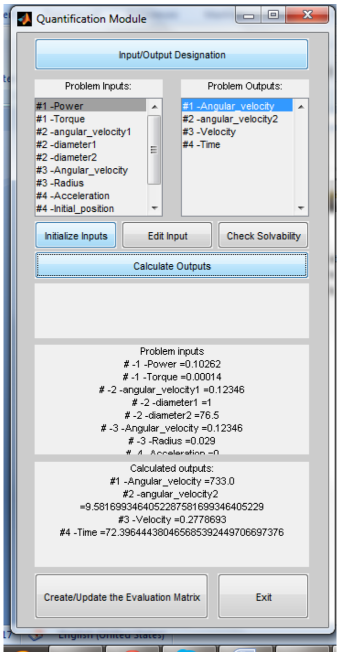
```
Calculated outputs:
#1 -Angular_velocity
=732.99826689774696707105719237435
#2 -angular_velocity2
=3.7264783934926283680732079308592
#3 -Velocity =0.1080685
#4 -Time =186.14601647080595910058295549181
```

Figure 68 - Simulation Results

After solving the model, the results indicate that it takes 186.1 secs for the vehicle to travel 1.5 rounds of the contest perimeter. Since this is a long period of time, some variables are changed and the model is re-solved.

Step -8: Making changes and resolving (if desired)

The transmission ratio is changed to 76.5:1 (instead of the 196.7:1) and the model is re-solved to see how the performance of the vehicle is improved. Below, the new simulation results are displayed:



```
Problem inputs
# -1 -Power =0.10262
# -1 -Torque =0.00014
# -2 -angular_velocity1 =0.12346
# -2 -diameter1 =1
# -2 -diameter2 =76.5
# -3 -Angular_velocity =0.12346
# -3 -Radius =0.029
# -4 -Acceleration =0
```

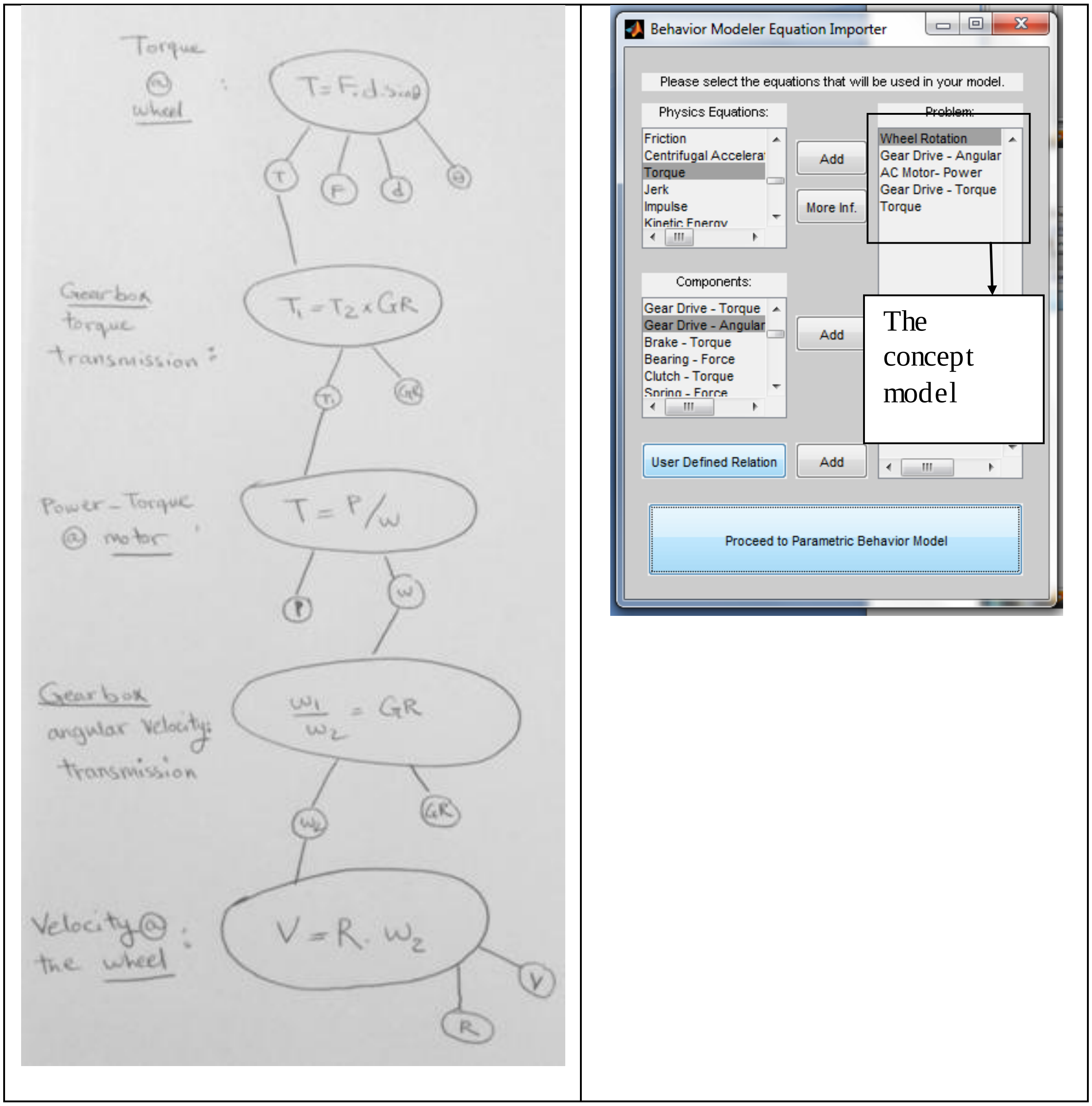
```
Calculated outputs:
#1 -Angular_velocity =733.0
#2 -angular_velocity2
=9.5816993464052287581699346405229
#3 -Velocity =0.2778693
#4 -Time =72.396444380465685392449706697376
```

Figure 69 - Results of modifying the model and simulating it

With the new gear ratio used for the transmission, the time for traveling the contest field has decreased to 72seconds (performance improved by approximately 60%).

Question 2- Would the vehicle climb up the ramp in a reasonable time?

In order to answer this feasibility question, the velocity generated by the motor when climbing the ramp should be compared with the torque generated by the power transmission system. Therefore the model developed consists of:



Please select the equations that will be used in your model.

The concept model

Figure 70 - The concept model which consists of PEs, COTS, and user-defined relations

The graphical representation of the model developed by the tool is displayed below:

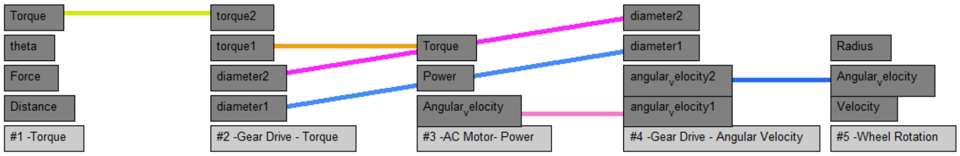
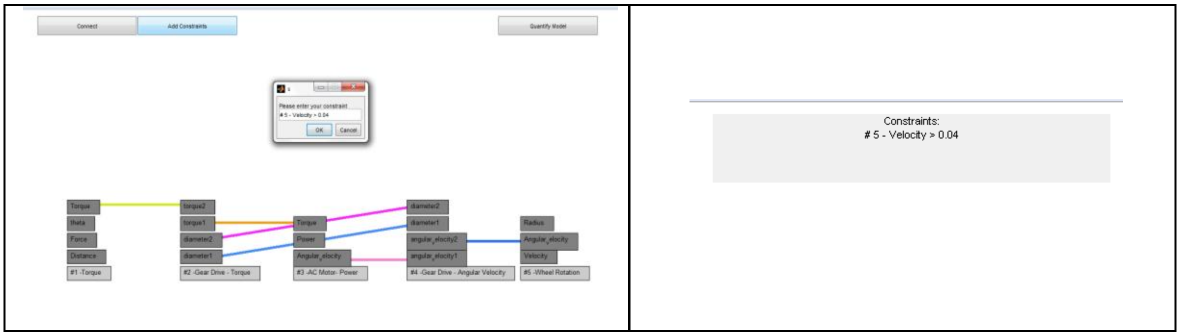


Figure 71 - Concept model developed by the tool

At this point, constraints that are applied to the model are added:

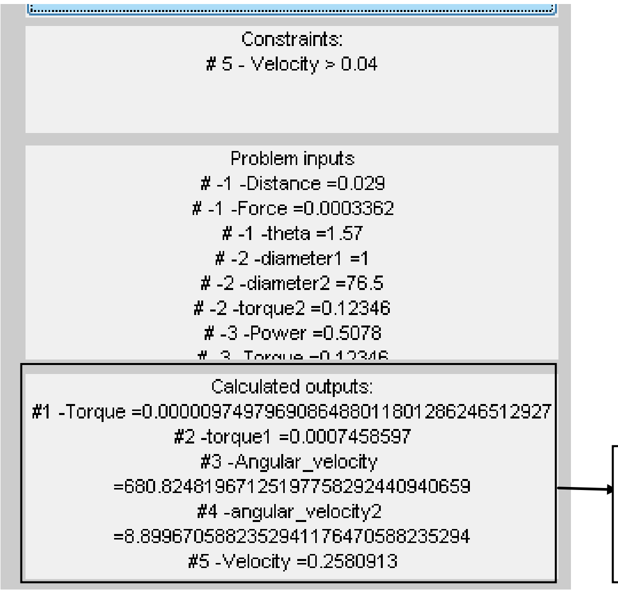


Please enter your constraint
#5 - Velocity > 0.04

Constraints:
5 - Velocity > 0.04

Figure 72 - Model constraint added

After the measures of goodness are determined, the model variables are quantified. Afterwards, the model is solved and the results indicate a velocity of 0.25 m/s which indicates that the performance of the vehicle is acceptable.



Constraints:
5 - Velocity > 0.04

Problem inputs
-1 -Distance =0.029
-1 -Force =0.0003362
-1 -theta =1.57
-2 -diameter1 =1
-2 -diameter2 =76.5
-2 -torque2 =0.12346
-3 -Power =0.5078
-3 -Torque =0.12346

Calculated outputs:
#1 -Torque =0.0000097497969086488011801286246512927
#2 -torque1 =0.0007458597
#3 -Angular_velocity
=680.82481967125197758292440940659
#4 -angular_velocity2
=8.8996705882352941176470588235294
#5 -Velocity =0.2580913

Measures of goodness calculated

Figure 73 - Simulation Results

Question - 3: Would the battery provide sufficient energy for the vehicle to run during the test?

Step #3: Creating models for the feasibility questions asked

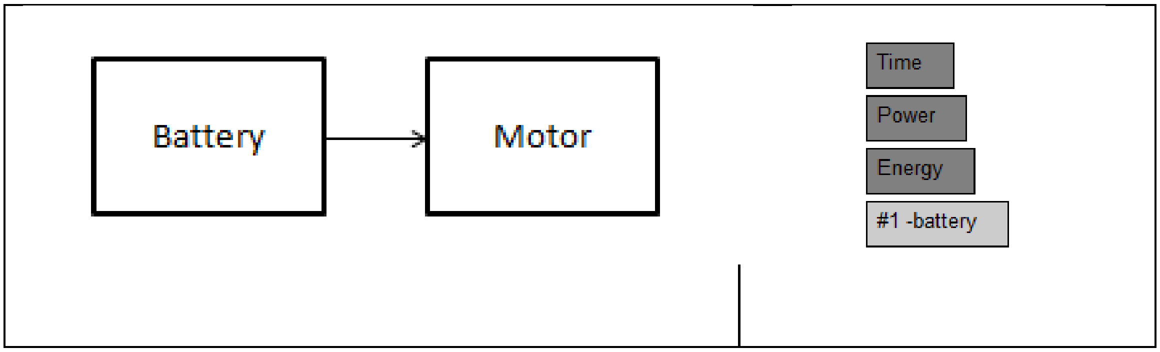
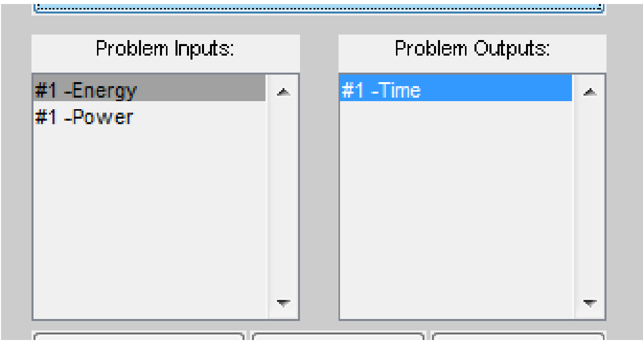


Figure 74 - Concept model for answering the third feasibility question

Step # 4: Determining the MOGs

The measure of goodness is the time for the vehicle to travel with the supplied power generation unit.



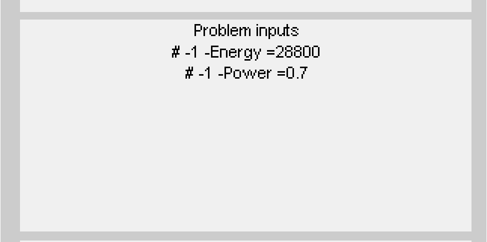
Problem Inputs:
#1 -Energy
#1 -Power

Problem Outputs:
#1 -Time

Figure 75 - Time as the measure of goodness for this model

Step #5: Quantifying the model

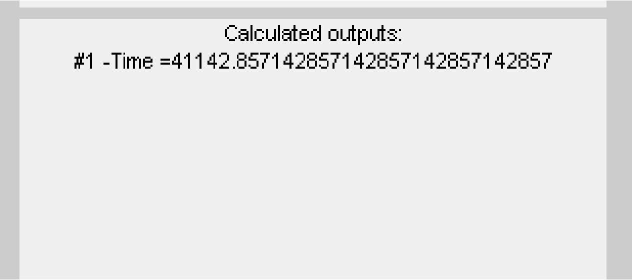
The value of the motor power and the energy output of the battery are inserted to the model.



Problem inputs
-1 -Energy =28800
-1 -Power =0.7

Figure 76 - Model inputs initialized

Step # 7: Checking solvability of the model and solving it results



Calculated outputs:
#1 -Time =41142.857142857142857142857142857

Figure 77 - The simulation result

The time calculated for the batteries to run the engine is equal to 41000 seconds which is equal to 11 hours approximately. The duration of batteries running the vehicle is sufficient for the test requirements.

*

Conclusion

Based on the simulation results for both of the feasibility questions, the configuration of motor-gearbox-wheel would work well if the gearbox gear ratio is 76.5:1 instead of 196.7:1. The motor limitations will be served and the speed of the vehicle would be in an acceptable range for the test. Also, the battery would run for the test period.

*

Case Study - 3:

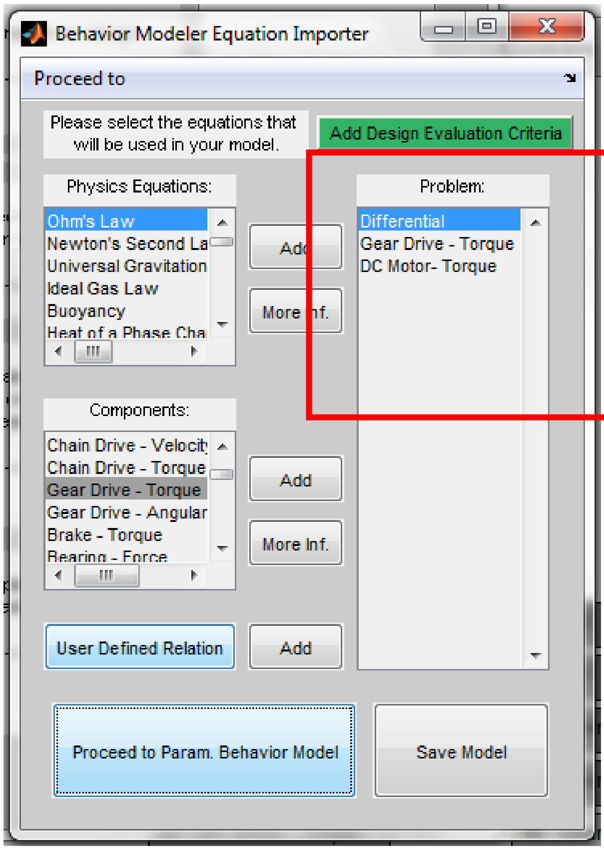
Problem statement:

> Determine which configuration is better for a vehicle power train. The power train should consist of an electric motor, a gear box or a chain drive and a differential. The electric motor can generate angular acceleration of 50- 150 rad/sec^2. The motor has a moment of inertia of approximately 0.1 kg.m^2. The gear ratio for the differential is 4. The output torque from the differential should be greater than 10.

Problem solution using the tool:

Section 1- Design with a gear box:

The user imports the relevant relations for the motor and gear box from the tool's library and adds a user defined relation for the differential.



Problem model having three sub-solutions for motor, gear box and the differential

Figure 78 - Related model relations are imported from the tool library or inputted by the user

The Graphical representation of the model for the system then provides a better understanding of the model.

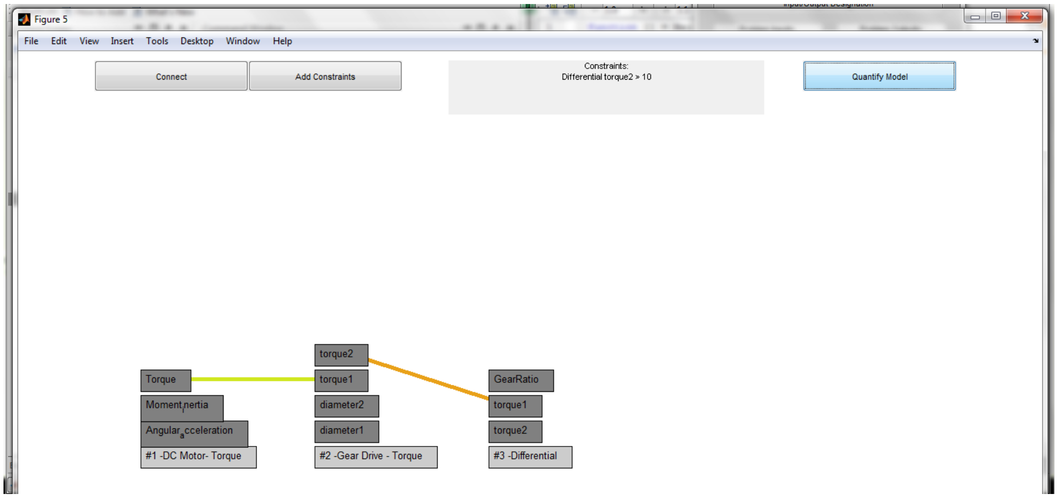
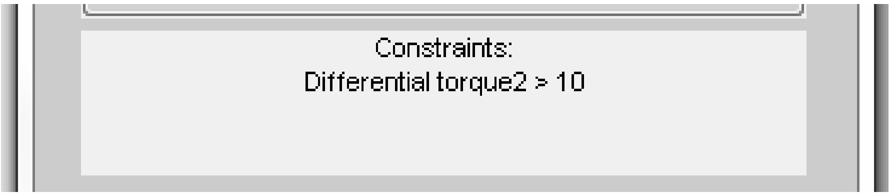


Figure 79 - Model's Graphical Representation and the sub-systems' connection

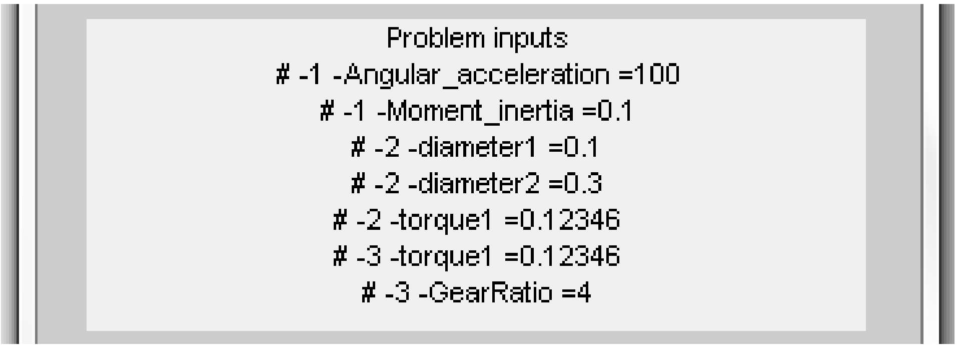
Model constraint is reflected into the model as:



Constraints:
Differential torque2 > 10

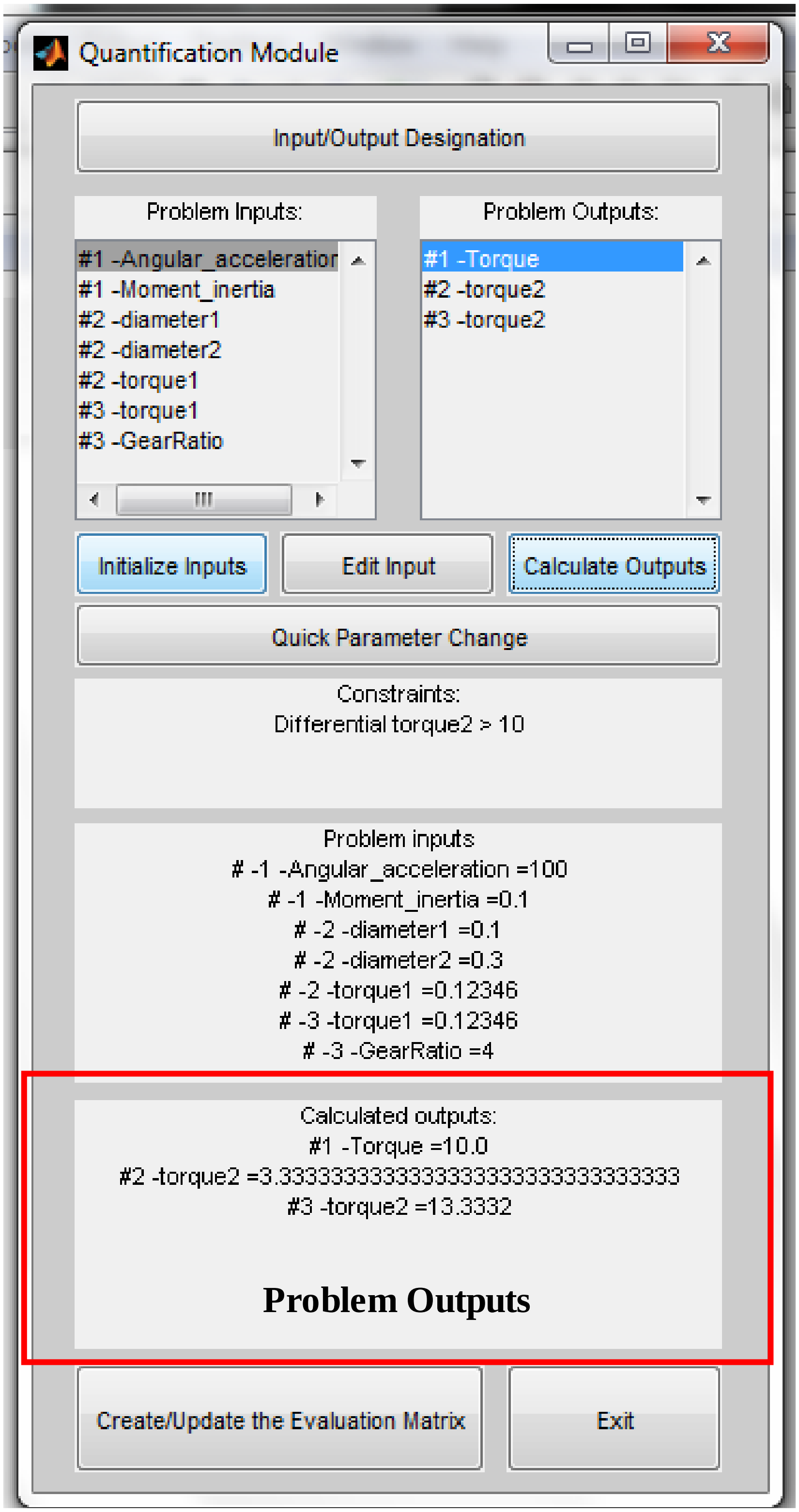
Figure 80 - Model Constraints

From the given design specifications, after defining the model input and outputs, user initializes the model and solves it:



Problem inputs
-1 -Angular_acceleration =100
-1 -Moment_inertia =0.1
-2 -diameter1 =0.1
-2 -diameter2 =0.3
-2 -torque1 =0.12346
-3 -torque1 =0.12346
-3 -GearRatio =4

Figure 81 - Model Inputs



Constraints:
Differential torque2 > 10

Problem inputs
\# -1 -Angular_acceleration =100
\# -1 -Moment_inertia =0.1
\# -2 -diameter1 =0.1
\# -2 -diameter2 =0.3
\# -2 -torque1 =0.12346
\# -3 -torque1 =0.12346
\# -3 -GearRatio =4

Calculated outputs:
#1 -Torque =10.0
#2 -torque2 =3.33333333333333333333333333333333
#3 -torque2 =13.3332

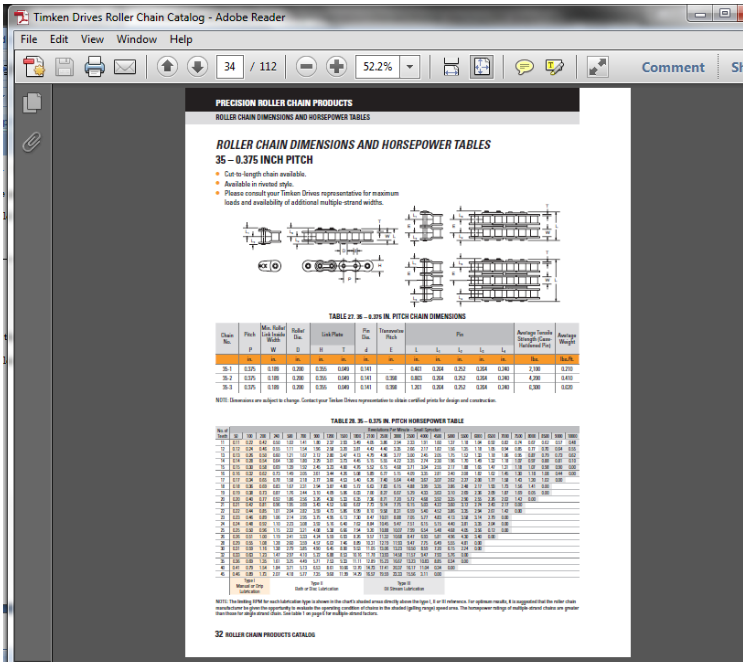
Figure 82 - Solution Results

As seen, the results match the problem constraints and there is no need for re-iteration unless the user wants to enhance the results (e.g. increase output torque, etc.). This configuration produces final torque of 13.3 N.m.

Section 2 - Design with a chain drive:

This problem is the same as the previous problem, yet instead of using a gear drive, the designer is asked to use a chain drive.

The designer imports the relevant relations for the motor and the chain drive from the tool's library and adds a user defined relation for the differential. Also, from the commercial catalogue on the chain drives, he adds another user-input relation for the chain drive:



PRECISION ROLLER CHAIN PRODUCTS

ROLLER CHAIN DIMENSIONS AND HORSEPOWER TABLES

ROLLER CHAIN DIMENSIONS AND HORSEPOWER TABLES

35 – 0.375 INCH PITCH

- Cut-to-length chain available.
- Available in riveted style.
- Please consult your Timken Drives representative for maximum loads and availability of additional multiple-strand widths.

TABLE 27. 35 – 0.375 IN. PITCH CHAIN DIMENSIONS

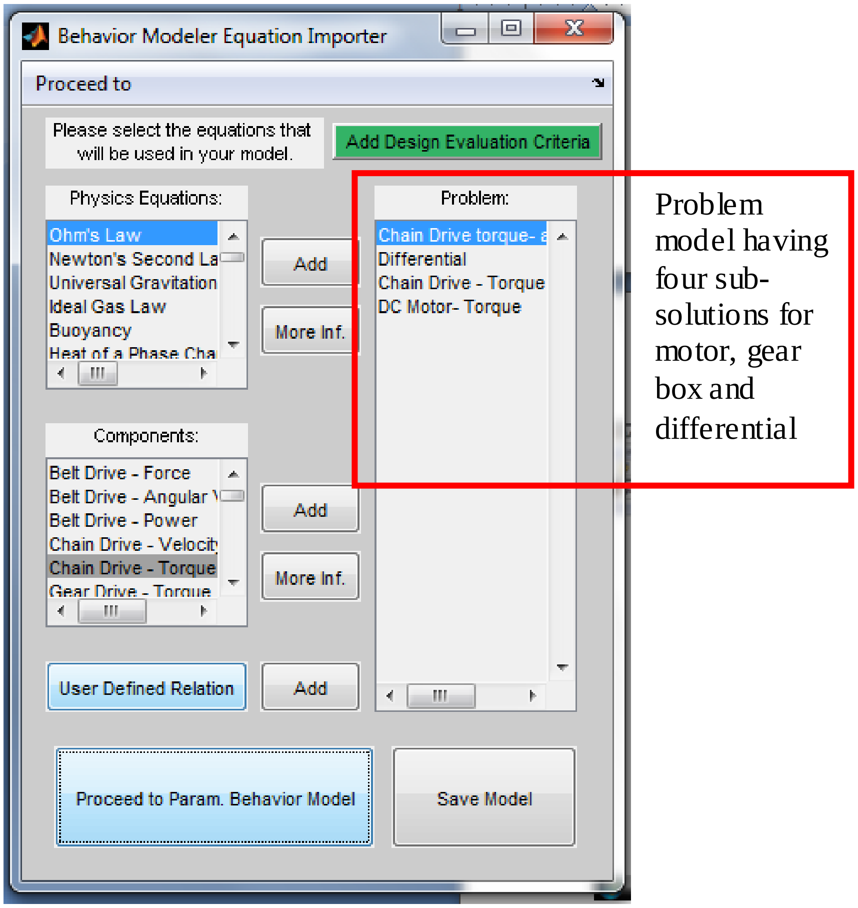
Chain No.	Pitch	Min. Roller Link Inside Width	Roller Dia.	Link Plate		Pin Dia.	Transverse Pitch	Pin					Average Tensile Strength (Case-Hardened Pin)	Average Weight
	P	W	D	H	T	d	E	L	L_1	L_2	L_3	L_4		
	in.	in.	in.	in.	in.	in.	in.	in.	in.	in.	in.	in.	lbs.	lbs./ft.
35-1	0.375	0.189	0.200	0.355	0.049	0.141	–	0.461	0.264	0.252	0.264	0.240	2,100	0.210
35-2	0.375	0.189	0.200	0.355	0.049	0.141	0.398	0.863	0.264	0.252	0.264	0.240	4,200	0.410
35-3	0.375	0.189	0.200	0.355	0.049	0.141	0.398	1.261	0.264	0.252	0.264	0.240	6,300	0.620

NOTE: Dimensions are subject to change. Contact your Timken Drives representative to obtain certified prints for design and construction.

TABLE 28. 35 – 0.375 IN. PITCH HORSEPOWER TABLE

32 ROLLER CHAIN PRODUCTS CATALOG

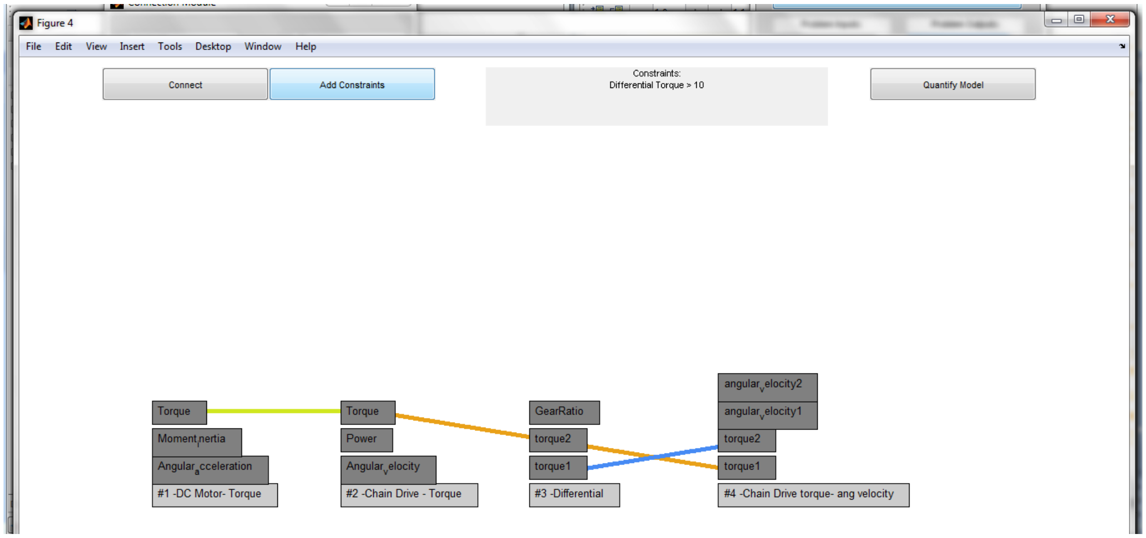
Figure 83 - Commercial catalogue used for modeling and initializing the chain drive model



Problem model having four sub-solutions for motor, gear box and differential

Figure 84 - Related model relations are imported from the tool library or inputted by the user

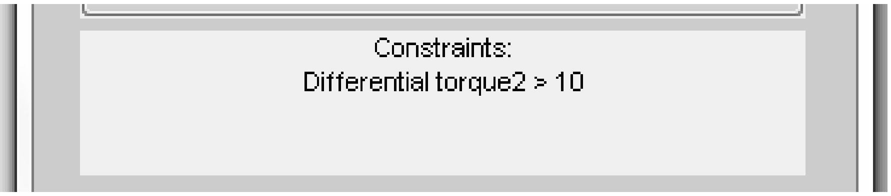
The Graphical representation of the model for the system then provides a better understanding of the model.



Constraints:
Differential Torque > 10

Figure 85 - Model's Graphical Representation and the sub-systems' connection

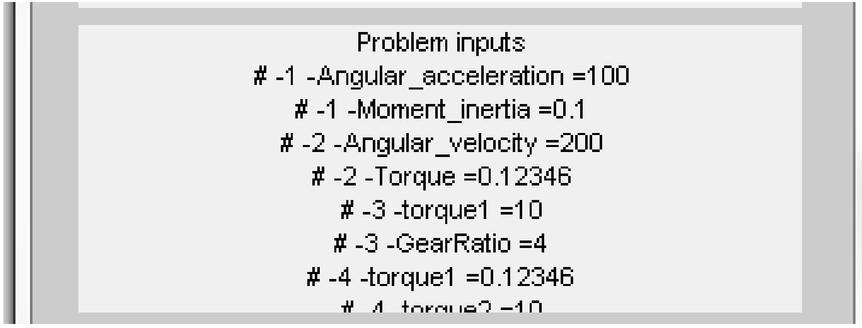
Model constraint is reflected into the model as:



Constraints:
Differential torque2 > 10

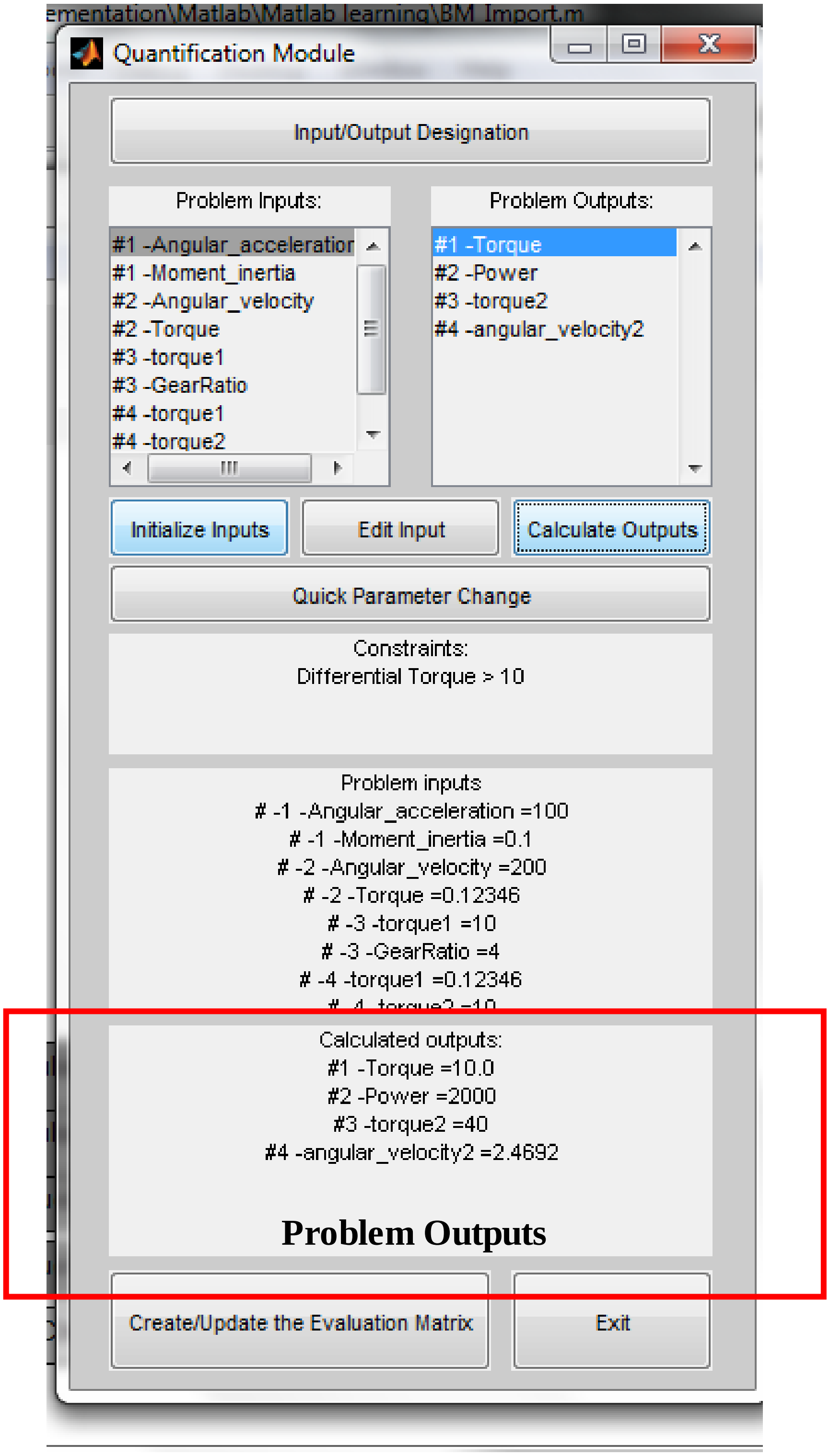
Figure 86 - Model Constraints

After defining the model input and outputs, user initializes the model and solves it:



Problem inputs
-1 -Angular_acceleration =100
-1 -Moment_inertia =0.1
-2 -Angular_velocity =200
-2 -Torque =0.12346
-3 -torque1 =10
-3 -GearRatio =4
-4 -torque1 =0.12346
-4 -torque2 =10

Figure 87 - Model Inputs



```
#1 -Angular_acceleratior
#1 -Moment_inertia
#2 -Angular_velocity
#2 -Torque
#3 -torque1
#3 -GearRatio
#4 -torque1
#4 -torque2
```

```
#1 -Torque
#2 -Power
#3 -torque2
#4 -angular_velocity2
```

Constraints:
Differential Torque > 10

Problem inputs
-1 -Angular_acceleration =100
-1 -Moment_inertia =0.1
-2 -Angular_velocity =200
-2 -Torque =0.12346
-3 -torque1 =10
-3 -GearRatio =4
-4 -torque1 =0.12346
-4 -torque2 =10

Calculated outputs:
#1 -Torque =10.0
#2 -Power =2000
#3 -torque2 =40
#4 -angular_velocity2 =2.4692

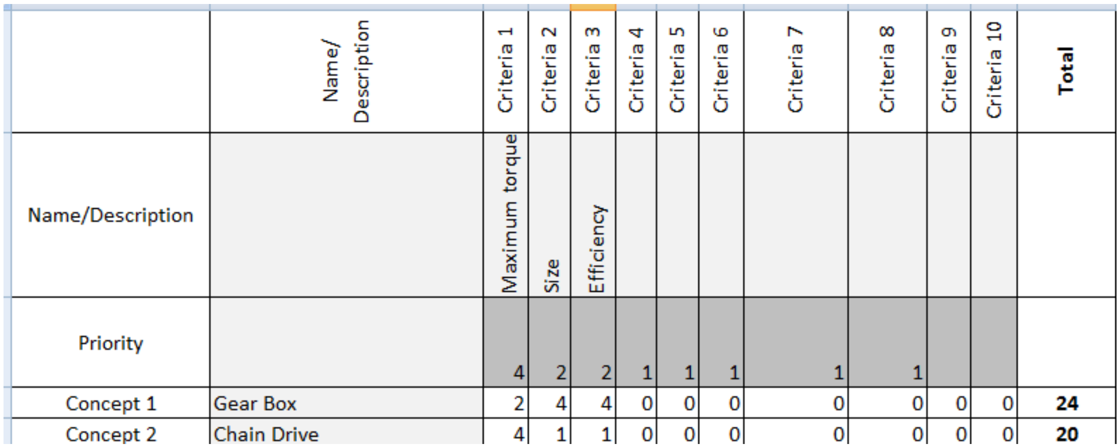
Problem Outputs

Figure 88 - Solution Results

As seen, the results match the problem constraints and there is no need for re-iteration unless the user wants to enhance the results (e.g. increase output torque, etc.).

This model required a more-detailed modeling of the power train, yet gave more in-depth results about the power required for the motor. With this configuration, the user gets 40N.m. torque output from the differential.

At this point, the user proceeds to the concept selection module to compare and rank the two designs using the evaluation/comparison module. Knowing that maximum torque is an important characteristic of the design, he uses it as a criterion and scores the two designs:



	Name/ Description	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	Criteria 6	Criteria 7	Criteria 8	Criteria 9	Criteria 10	**Total**
Name/Description		Maximum torque	Size	Efficiency								
Priority		4	2	2	1	1	1	1	1			
Concept 1	Gear Box	2	4	4	0	0	0	0	0	0	0	**24**
Concept 2	Chain Drive	4	1	1	0	0	0	0	0	0	0	**20**

Figure 89 - Comparison of the two designs

Conclusion:

Having considered both designs, the ranking suggests that the gear drive is a better configuration, even though it produces smaller output torque.

CHAPTER 7: CONTRIBUTIONS OF THIS WORK AND CONCLUSIONS

Contributions of This Work

The main advantages of this of this work could be discussed from two perspectives:

1. Functionality (application)
2. Technology

Looking at its application in early design, one could notice the tool providing:

- A simplified framework for quick formulation and simulation of concepts which suits the little information available during conceptual design,
- the capability to model concepts using PVs, PEs, and COTS in an organized structure allowing adding, editing PEs, COTS models, and user defined relations while ensuring modularity of the model and compatibility between linked model fragments,
- the capability to formulate models in a parametric framework as well as the capability to quantify and simulate the parametric formulation,
- the capability to create models that are detailable and therefore transferable to more advanced design stages such as embodiment and detail design.

Therefore, this platform assists the designers to create a rough model for answering the feasibility problem at hand and provides a chance to simulate the model and roughly estimate the behavior. This ultimately leads to making a rationalized decision during conceptual design.

From the technological perspective, this tool provides a platform for structuring design concepts using a physical ontology and smoothly transitioning it to more advanced stages. The modular modeling technique used in the tool enables adding details to the model; while it is completely up to the designer to decide how detailed he wants the concept model to be. Features for modifying the model configuration as desired by the user, and is an instance of the tool's technological organization.

Also, the tool provides the benefit of modeling not only function structures, but a broader variety of behaviors (performance), through modeling PEs and COTS behavior models (a more flexible method of tackling design problems (compared to function modeling)). The graphical model representation of the tool also makes it easy for the user to interact with the model and gain more insight into his design and the relationship between the variables.

Unlike many agent-based systems are based on traditional models and theories of designing that assume the world as being fixed, this tool allows modification and addition of model fragments (as essential for every design process).

Comparison with Similar Tools from Industry and Academia

This tool is suitable for rough analysis required for conceptual design when little is known about the final design. However, most commercial tools are designed for more advanced stages of design and require extensive information about the design (e.g. NX Knowledge fusion requires geometrical information which is usually undetermined during conceptual design.). The simulation done with such tools is computationally expensive and requires advanced solvers; while this tool uses algebraic equations that model steady state behavior and therefore simulations done with it are computationally cheap (e.g. Modelica libraries model COTS using differential equations and numerical solvers are required to solve them.).

Many academic tools do not perform any analysis (eg. FunctionCAD) and are solely for representation purposes (OSU's Design repository is suitable for associating design with COTS but does not create any mathematical formulation and does not facilitate analysis.). Other tools such as CONMOD generate qualitative models but do not facilitate model quantification. IDeAL models are not transferrable to more advanced stages of design.

Following various aspects of the tool developed in this study vs. the tools developed at industry and academia are compared:

	Representation	Qualitative	Parametric	Quantitative	Specific Values
FunctionCAD	*				
GraphSynth	*	*			
ConMOD	*	*			
IDeAL	*	*			
Design Machine				*	
Commercial Tools (eg. NX Knowledge Fusion, Dymola, Model Center)	*		*	*	*
Current Work	*	*	*	*	

Conclusions and Future Work

This research work was an early effort to develop a tool for formulation and rough evaluation of design concepts. Within this work, the ontology for developing concept models was reviewed and a tool was developed for modeling the mechanical power transmission systems. While the tool is incapable of accurate simulation, it allows quick formulation, modification, and simulation of design concepts to address a feasibility question. This tool works with physical variables, physical principles and component models to create a mathematical network of physical equations. The mathematical formulation is capable of being quantified and solved to simulate behavior and answer the feasibility question asked. The tool generates a graphical representation of the concept model for easier interaction with the model. It is also used for some case studies and the results of the studies are discussed.

Since the tool is unable to deal with incomplete concepts/ideas, one future work could be adding such a capability to it. Working with ranges of variables instead of specific values could be another aspect of improving the current tool.

*

REFERENCES

[1] G. Pahl, W. Beitz, J. Feldhusen, K.H. Grote, (2007), *Engineering Design: A Systematic Approach*, 3rd. ed. Springer-Verlag, 2007.

[2] R. Nagel, R. Perry, R. B. Stone, (2009), FunctionCAD: A functional modeling application based on function design framework, IDETC/CIE 2009, San Diego, California USA.

[3] G. Rohrbach, (1969), "Kreativ Nach Regeln - Method 635, Eine New Technik Zim Lösen von Problemen", Absatzwirtschaft 12.

[4] H. Hellfritz, (1978), "Innovation via Gallerie Methode", Königstein/Ts: Eigenverlag.

[5] J. J. Shah, N. Vargas-Hernandez, J. D. Summers, S. Kulkarni, (2001), "Evaluation of Collaborative Sketching as an Idea Generation Technique for Engineering Design", Journal of Creative Behavior, Vol. 35, No. 3, pp.1-31.

[6] A. F. Osborn, (1953), *Applied imagination; principles and procedures of creative problem-solving*, Scribner.

[7] Library of image net: http://www.image-net.org/

[8] Library of word net: http://wordnetweb.princeton.edu/perl/webwn

[9] M. Mohan, J. Shsh, S. Narsale, M. Khorshidi, (2012) "Capturing Ideation Paths for Discovery of Design Exploration Strategies in Conceptual Engineering Design", Design Computing and Cognition DCC'12.

[10] K. Roth, (2000), *Konstruieren, Mit Konstruktionskatalogen:Band 1: Konstruktionslehre*, Springer, Berlin.

[11] VDI-Fachbereich Produktentwicklung und Mechatronik, Konstruktionsmethodik/ 1 VDI-Handbuch Konstruktion. 2222,1, Methodisches Entwickeln von Losungsprinzipien, 1997.

[12] AskNature Bio-Mimicry Design Portal: http://www.asknature.org/

[13] R. B. Stone, Design Engineering Lab. Repository, Oregon State University: http://designengineeringlab.org/

[14] G. Altshuller, (2001), "40 Principles: TRIZ Keys to Technical Innovation", Tech Innov Center.

[15] P. Zwicky, (1969), "Discovery, Invention, Research through Morphological Analysis", McMillan, NY.

[16] M. Mohan, Y. Chen, J. J. Shah, (2011), Towards a Framework for Holistic Ideation in Conceptual Design, DETC 2011/ CIE.

[17] K. Ulrich, S. Eppinger, (2011), *Product Design and Development,* 5th Edition, McGraw Hill/ Irwin

[18] J. De Kleer, J. Seely Brown, (1984), "A Qualitative Physics Based on Confluences", Artificial Intelligence 24,pp. 7-83.

[19] J. De Kleer, (1977), "Multiple Representations of Knowledge in a Mechanics Problem Solver", Proceedings of IJCAI-77, pp.299-304.

[20] Y. Shimomura, S. Tanigawa, Y. Umeda, T. Tomiyama, (1995), "Development of Self-

Maintenance Photocopiers", Proceedings of IAAI-95. 171-180.

[21] P. Dague, O. Raiman, P. Deves, (1987), "Troubleshooting: When Modeling Is the Trouble",

Proceedings of AAAI-87, pp. 590-595.

[22] K. D. Forbus, (1996), Qualitative Reasoning, Chapter for the CRC Handbook of Computer Science, IL, USA.

[23] H. Paynter, (1961), Analysis and Design of Engineering Systems, The M.I.T. Press. ISBN 0-262-16004-8.

[24] V. Liang, C. Paredis, (2004), "A Port Ontology for Conceptual Design of Systems", Journal of Computing and Information Science in Engineering 4, pp. 206- 217, DOI: DOI: 10.1115/1.1778191.

[25] D. Weld, (1990), Theories of Comparative Analys.i sCambridge, Mass.: MIT Press.

[26] A. K. Goel, and S. R. Bhatta, (2004), "Use of Design Patterns in Analogy-Based Design," *Advanced Engineering Informatics* 18, pp. 85-94.

[27] J. S. Gero, U. Kannengiesser, (2002), "The Situated Function-Behavior-Structure Framework", Artificial Intelligence in Design'02, pp. 89-104.

[28] P. Sridharan, and M. Campbell, (2004), "A Grammar for Function Structures", ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

[29] J. S. Gero, (1996), "Creativity, Emergence and Evolution in Design", Knowledge-Based Systems 9 pp. 435-448.

[30] M. Suwa, J. Gero, T. Purcell, (1999), "Unexpected Discoveries: How Designers Discover Hidden Features in Sketches", *in J.* Gero, B. Tversky, *Visual and Spatial Reasoning in Design*, Key Centre of Design Computing and Cognition, University of Sydney, pp. 145-162.

[31] J. Gero, (1990), "Design Prototypes: A Knowledge Representation Schema for Design", AI Magazine, Vol. 11, No.

[32] G. Huang, K. Mak, (1991), "Web-based Morphological Charts for Concept Design in Collaborative Product Development", Journal of Intelligent Manufacturing , Vol 10, pp. 267-278.

[33] C. W. Bytheway, (1971), "The creative aspects of FAST diagramming", Proceedings of the SAVE conference.

[34] M. Campbell, J. Cagan, K. Kotovsky, (2000), "Agent-based synthesis of electromechanical design configurations" Journal of Mechanical Design 122.

[35] A. Chakrabarti, T. P. Bligh, (1996), "An approach to functional synthesis of mechanical design Concepts: Theory, applications, and emerging research issues", Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, Vol. 10, pp. 313-331.

[36] R.H. Bracewell, J.E.E. Sharpe, (1996), "Functional descriptions used in computer support for qualitative scheme generation - "Schemebuilder"", Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, Vol. 10 pp. 333-345.

[37] P. Sridharan, M. Campbell, (2005), "A Study on the Grammatical Construction of Function Structures", *Artificial Intelligence for Engineering Design, Analysis & Manufacturing*, **19**(3) pp. 139-160.

[38] T. Kurtoglu, A. Swantner, M. I. Campbell, (2010), "Automating the conceptual design process: “From black box to component selection”", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 24, Special Issue 01, pp. 49-62.

[39] J. Shah, S. Sen, S. Ghosh, "An Intelligent CAD Environment for routine mechanical design", ASME Computers in Engineering Conf, Santa Clara, August 1991.

[40] C. Bryant, R. Stone, J. Greer, D. McAdams, T. Kurtoglu, M. Campbell, (2007), "Function-Based Component Ontology for Systems Design", International Conference on Engineering Design.

[41] R. L. Nagel, R. B. Stone, R. S. Hutcheson, D. A. McAdams, J. A. Donndelinger, (2008), "Function Design Framework (FDF): Integrated Process and Function Modeling for Complex Systems", Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE).

[42] R. L. Nagel, K. L. Perry, R. B. Stone, D. A. McAdams, (2009), "FinctionCAD: A Functional Modeling Application Based on the Function Design Framework", Proceeding of IDETC/CIE.

[43] Data Processing Techniques: Flowcharting Techniques, C20-8152-1, IBM: International Business Machine Corporation, (1970).

[44] Flowchart Symbols and their Usage in Information Processing, American National Standard, A. X3.6-1970.

[45] G. Darnton, M. Darnton, (1997), Business Process Analysis, London, International Thomson Business Press.

[46]G. Fischer, K. Nakakoji, J. Ostwald, G. Stahl and T. Sumner, (1993-a). 'Embedding computer-based critics in the contexts of design", In: S. Ashlund, K. Mullet, A, Henderson, E. Hollnagel and T. White (eds.), Human Factors in Computing (Interact '93 and CHI '93), pp. 157-164.

[47] G. Fischer, K. Nakakoji, J. Ostwald, G. Stahl and T Sumner, (1993-b), "Embedding critics in design environments", The Knowledge Engineering Review 8(4), pp. 285-307.

[48] C. Sen, (2011), "A Formal Representation of Mechanical Functions to Support Physics-Based Computational Reasoning in Early Mechanical Design", PhD Dissertation, Clemson University.

[49] K. Forbus, (1984), "Qualitative process theory", Artificial Intelligence 24, pp. 85-168, p. 333-345.

APPENDIX A

PHYSICAL VARIABLES

Parameter	Process Parameters
Voltage	
Resistance	Resistance1, Resistance2 (86,87)
Current	
Mass	Mass1 Mass2 (3,11,12,13)
Acceleration	Acceleration1, Acceleration2 (63)
Distance	Distance_Load, Distance_Effort(21)
Pressure	Pressure3, Pressure4 (47) Pressure1, Pressure2 (51, 57)
No	
Temperature	Temperature 2, Temperature1 (19,40, 41, 51, 95, 96), Environment_Temperature (42) Temperature3, Temperature4 (46, 47)
Volume	Volume1, Volume2 (99)
Density	
Heat	
Latent_Heat	
Adh_Coeff	
Area	Area1, Area2 (94)
Angular_Acceleration	
Torque	
Moment_Inertia	
Bending_Stress	
Force	Force_Effort, Force_Load(21) Force1, Force2 (102)
Centripetal_Acceleration	
Velocity	Velocity12,Velocity22(10,11,13), Velocity 21,Velocity11(12,13) Velocity1, Velocity2(22, 57, 94)
Coriolis_Acceleration	
Angular_Velocity	
Radial_Velocity	
Friction_Coeff	
Normal_Force	
Stress	
Thickness	
Shear_Stress	
Diameter	
Width	
Strain	

Coeff_Expansion	
Impact	
Time	
Momentum	Momentum1, Momentum2 (64)
Fatigue_Stress	
Length	Length2, Length1 (27, 95, 108)
Frequency	
Poisson_Nu	
Elastic_Modulus	
Spring_K	
Weight	
Surface_Tension	
Torsion	
Polar_Moment_Inertia	
Shear_Modulus	
Theta	
Potential_Energy	
Mechanical_Advantage	
Coeff_Wear	
Probability	
Yield_Stress	
Average_Wear_Depth	
Deflection	
Yield_Strength	
Stress_Intensity_Factor	
Plastic_Zone_Size	
Fracture_Constant	
Half_Crack_Length	
Heat_Flux	
Thermal_Conductivity	
Power	
View_factor12	
Energy_Flow	
Coeff_Heat_Transfer	
Emmision_Current_Density	
Material_Correction_Factor	
Work	
Coeff_Partition	
Concentration_Solute	Concentration_Solute1, Concentration_Solute2 (44)
Dissociation_Constant	
Equilibrium_Concentration	Equilibrium_Concentration_A, Equilibrium_Concentration_B, Equilibrium_Concentration_AB (45)
Fuel_Air_Ratio	

Fuel_Heating_Value	
Specific_Heat	
Gas_Mean_Free_Path	
Temperature_Gradient	
Viscosity	
Height	Height1, Height2 (57)
Vaporization_Enthalpy	
Adsorped_Quantity	
Adsorbent_Adsorbate_k	
Adsorbent_Adsorbate_n	
Rate_Constant	
Saturation _Concentration_Solute	
Empirical_Exponent	
Root_Mean_Square_Velocity	
Mass_Flow	
Permeability	
Delta_Pressure	
Volumetric_Flow_Rate	
Initial_Velocity	
Initial_Position	
Position	
Maximum_Friction_Force	
Centrifugal_Acceleration	
Jerk	
Impulse	
Kinetic_Energy	
Potential_Energy	
Period	
EMF	
Charge	Charge1, Charge2 (74)
Electric_Field	
Average_Electric_Field	
Capacitance	Capacitance1, Capacitance2 (84, 85)
Resistivity	
Total_Capacitance	
Total_Resistance	
Magnetic_Field	
Magnetic_Constant	
Magnetic_Flux	
Surface_Pressure	
Average_Kinetic_Energy	
Internal_Energy	
Efficiency	
Enthropy_Change	

APPENDIX B

PHYSICS PRINCIPLES DATABASE

ID	Area	Eq_Name	Symbolic_Eq	No_Variables	Parameters	Descrption
1	Electrical	Ohm's Law	Voltage=Resistance*Current	3	V: Voltage, R: Resistance, I: Current	In an electric circuit, Ohm's law states that the current through a conductor between two points is directly proportional to the potential difference across the two points.
2	Mechanical	Newton's Second Law of Motion	Force=Mass*Acceleration	3	F: Force, m: Mass, a: Acceleration	The acceleration of a body is directly proportional to, and in the same direction as, the net force acting on the body, and inversely proportional to its mass.
3	Mechanical	Universal Gravitation	Force=6.67*(10^-11)*Mass1*Mass2/(Distance*Distance)	4	F: Force, m1: Mass #1, m2: Mass #2, r: Distance	**Newton's law of universal gravitation** states that every point mass in the universe attracts every other point mass with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them.
4	Thermal and Fluids	Ideal Gas Law	Pressure=No*8.31*Temperature/Volume	4	P: Pressure, n: # of moles, T: Temperature, Vol: Volume	The **ideal gas law** is the equation of state of a hypothetical ideal gas. It is a good approximation to the behaviour of many gases under many conditions, although it has several limitations. It was first stated by Émile Clapeyron in 1834 as a combination of Boyle's law and Charles's law.
5	Thermal and Fluids	Buoyancy	Force=Density*Volume*9.81	3	F: Force, Density: Density of the fluid, Vol: Volume of	**buoyancy** is an upward force exerted by a fluid that opposes the weight of an immersed object.

					the fluid displaced	
6	Thermal and Fluids	Heat of a Phase Change	Heat=Mass*Latent_Heat	3	Q: Heat, m: Mass, L: Latent heat of phase change	Heat of phase change is the heat that is transferred during phase change, e.g. from liquid to solid, etc.
7	Mechanical	Adhesion	Force= -1*Adh_Coeff*Area/(24*3.14*(Distance*Distance*Distance))	4	F: Force, A: Cross Section, adh: Specific adhesive coefficient, r: Separaration distance	Adhesive materials fill the voids or pores of the surfaces and hold surfaces together by interlocking. Sewing forms a large scale mechanical bond, velcro forms one on a medium scale, and some textile adhesives form one at a small scale
8	Mechanical	Angular acceleration	Angular_Acceleration = Torque/Moment_Inertia	3	Alpha: Angular acceleration, I: Moment of inertia, To: Torque	Rate of change of angular velocity wrt time. Also, for constant torque exerted by a body, there will be a constant angular acceleration
9	Mechanical	Bend	Bending_Stress= Torque*Distance/Moment_Inertia	4	Sigma_b: Bending stress, M: Moment about neutral axis, y: Prependicular distance from neutral axis, Ix: Second moment of inertia about the neutral axis X	In engineering mechanics, bending (also known as flexure) characterizes the behavior of a slender structural element subjected to an external load applied perpendicularly to a longitudinal axis of the element
10	Mechanical	Centripetal Acceleration	Centripetal_Acceleration= Velocity*Velocity/Distance	3	Acen: Centripetal acceleration, v: Velocity, r: Radius	Acceleration due to motion in a curved path

11	Mechanical	Collision - Inelastic	Velocity12 =Velocity22	2	v12: Velocity of the 1st object after collision, v22: Velocity of the 2nd object after collision	Isolated event in which two or more bodies exert relatively strong forces on each other for a relatively short time.
12	Mechanical	Collision - Elastic	(Mass1*Velocity11*Velocity11)+(Mass2*Velocty21*Velocity21)=(Mass1*Velocity12*Velocity12)+(Mass2*Velocity22*Velocity22)	6	v12: Velocity of the 1st object after collision, v22: Velocity of the 2nd object after collision, v11: Velocity of the 1st object before collision, v21: Velocity of the 2nd object before collision, m1: Mass of the first object, m2: Mass of the second object	Isolated event in which two or more bodies exert relatively strong forces on each other for a relatively short time.
13	Mechanical	Collision - Elastic/Inelastic	(Mass1*Velocity11)+(Mass2*Velocity21) = (Mass1*Velocity12)+(Mass2*Velocity22)	6	v12: Velocity of the 1st object after collision, v22: Velocity of the 2nd object after	Isolated event in which two or more bodies exert relatively strong forces on each other for a relatively short time.

					collision, v11: Velocity of the 1st object before collision, v21: Velocity of the 2nd object before collision, m1: Mass of the first object, m2: Mass of the second object	
14	Mechanical	Coriolis acceleration	Coriolis_Acceleration=-2*(Angular_Velocty+Radial_Velocity)	3	Ac: Coriolis acceleration, Rv: Radial velocity, w: Angular velocity	An acceleration which, when added to the acceleration of an object relative to a rotating co-ordinate system and to its centripetal acceleration, gives the acceleration of the object relative to a fixed co-ordinate system
15	Mechanical	Dynamic/Kinetic friction	Force=Friction_Coeff*Normal_Force	3	Ff: Frictional force, mu: Coefficient of kinematic friction, Fn: Normal force	Friction between two solid objects that are moving relative to each other
16	Mechanical	Form closure - Compressive Stress	Stress=Force/(Thickness*Distance*No)	5	Sigma_c: Compressive stress, Fs: Shear load, t: Plate thickness, Dr: Rivet diameter, Nr: Number of load carrying rivets	Join or fasten (plates of metal or other material) - ex. Riveting

17	Mechanical	Form Closure-Shear Stress	Shear_Stress=4*Force/(3.14*Diameter*Diameter*No)	4	Ts: Shear stress, Fs: Shear load, t: Plate thickness, Dr: Rivet diameter, Nr: Number of load carrying rivets	Join or fasten (plates of metal or other material) - ex. Riveting
18	Mechanical	Form Closure - Tensile Stress	Stress=Force/(Width-No*Diameter)*Thickness	6	Sigma_t: Tensile stress, Fs: Shear load, t: Plate thickness, Dh: Hole diameter, Nr: Number of load carrying rivets, b: Gross plate width	Join or fasten (plates of metal or other material) - ex. Riveting
19	Mechanical	Heat strain	Strain=Coeff_Expansion*(Temperature2-Temperature1)	4	et: Thermal strain, alpha: Coefficient of expansion, T2: Temperature at state 2, T1: Temperature at state 1	A solid body expands as the temperature increases and contracts as the temperature decreases.this causes the thermal strain.
20	Mechanical	Impact	Force=Mass*Velocity/Time	4	F: Force, m: Mass, v: Velocity, t: Time	Impact is a high force or shock applied over a short time period
21	Mechanical	Lever effect	Force_Effort=Force_Load*Distance_Load/Distance_Effort	4	Fe: Effort load, Fl: Load force, dl: Lever	It is a rigid object used with an appropriate fulcrum/pivot point to either multiply the mechanical force that can

					distance from load, de: Lever distance from effect	be applied to an object or resistance force, or multiply the distance and speed at which the opposite end of the rigid object travels.
22	Mechanical	Linear acceleration	Acceleration=(Velocity2-Velocity1)/Time	4	a: Aceleration, v2: Velocity at state 2, v1: Velocity at state 1, t: Time	Change of Linear velocity with respect to time
23	Mechanical	Linear momentum	Momentum= Mass*Velocity	3	p: Momentum, m: Mass, v: Linear velocity	Product of mass and velocity of an object
24	Mechanical	Material joining - Butt Welding	Stress=Fatigue Stress*Force/(Thickness*Distance)	5	Sigma_w: Weld stress, Kf: Fatigue stress, P: Tensile load, t: Plate thickness, Lw: Length of the weld	Process of joining materials by using coalescence. Ex. Welding
25	Mechanical	Material joining - Fillet Welding	Shear_Stress=Force/(0.707*Width*Length)	4	Tou_w: Weld shear stress, P: Tensile load, s: Weld triangle width, Lw: Length of the weld	Process of joining materials by using coalescence. Ex. Welding
26	Mechanical	Mechancial Resonance	Frequency = (1/6.28)*(9.81/Length)^0.5	3	f: Natural frequency, g: Gravity (9.81), L: Length	It is the tendency of a mechanical system to absorb more energy when the frequency of its oscillations matches the system's natural frequency of vibration (its resonance frequency or resonant frequency) than it does at other frequencies.

27	Mechanical	Poisson effect	Poisson_Nu=Elastic_Modulus*Area*(Length2-Length1)/(Length1*Force)	6	L2: Length at state 2, L1: Length at state 1, F:Force, A: Cross section, E: Elasticity modulus	When a material is compressed in one direction, it usually tends to expand in the other two directions perpendicular to the direction of compression. This phenomenon is called the Poisson effect.
28	Mechanical	Spring Effect-Linear	Force=Spring_K*Distance	3	F: Force, k: Spring constant, x: Displacement	Elastic object to store mechanical energy is a spring. When the force deflecting the spring is in direct proportion to the distance it travels, it is called linear spring
29	Mechanical	Weight	Weight = Mass* 9.81	2	W: Weight, M: Mass	Weight of an object with the Earth's gravitational acceleration.
30	Mechanical	Surface tension	Surface_Tension = Diameter*Distance*Distance*Density/4	5	r: Radius, ro: Density, Gamma: Surface tension, g: Gravity, h: Height	Property of the surface of the liquid that allows it to resist an external force
31	Mechanical	Torsion - Shear Stress	Torsion= Polar_Moment_Inertia*Shear_Stress*2/Diameter	4	T: Torque, tou_max: Maximum shear stress, R: Radius, J: Polar moment of inertia	Torsion is the twisting of an object due to an applied torque
32	Mechanical	Torsion - Theta	Torsion=Polar_Moment_Inertia*Shear_Modulus*theta/Length	5	T: Torque, G: Shear modulus, theta: torsion angle, J: Polar moment of inertia, L: Length	Torsion is the twisting of an object due to an applied torque
33	Mechanical	Torsional Spring	Potentail_Energy=0.5*Spring_K*(theta*theta)	3	PE: Potential energy, k: Spring constant, Theta:	Spring that stores mechanical energy when twisted

					Torsion	
34	Mechanical	Wedge Effect-Mechanical Advantage	Mechanical_Advantage=Width/Length	3	Ma: Mechanical advantage, S: Width of the wedge, H: Height	A wedge is a triangular shaped tool, a compound and portable inclined plane, and one of the six classical simple machines
35	Mechanical	Wear - Coefficient of Adhesion	Coeff_Wear=Probability/(9*Yield_Stress)	3	Kadh: Wear coefficient, k: Probability of formation of transferred segment, Sigma_yp: Yield strength of the material	Erosion or sideways displacement of material by the action on another surface
36	Mechanical	Wear - Average Wear Depth	Average_Wear_Depth=Coeff_Wear*Pressure*Distance	4	Dadh: Average wear depth, Kadh: Wear coefficient, pm: Nominal contact pressure, ls: Sliding distance	Erosion or sideways displacement of material by the action on another surface
37	Mechanical	Impact	Deflection=Force*Length/(Area*Elastic_Modulus (1 + (2*Distance*Elastic_Modulus*Area/(Force*Length)^0.5))	6	ymax: Maximum end deflection, Wi: Impact load, l: Length of the bar, A: Cross section, E: Young's modulus, h: Height from which	High force or shock applied over a short period of time

					impact occurs	
38	Mechanical	Fracture - Yield Strength	Yield_Strength=Stress_Intensity_Factor/(6.28*Plastic_Zone_Size)	3	Sigma_yp: Yield strength, K: Stress intensity factor, rp: Plastic zone size	Local separation of an object into two or more pieces under the action of stress
39	Mechanical	Fracture - Stress Intensity	Stress_Intensity_Factor=Fracture_Constant*Stress*(3.14*Half_Crack_Length)^0.5	4	K: Stress intensity factor, Cf: Fracture constant, Sigma: Stress, a: Half crack length	Local separation of an object into two or more pieces under the action of stress
40	Themal and Fluids	Thermal conduction	Heat_Flux=-1* Thermal_Condutivity*(Temperature2-Temperature1)	4	q: Heat flux, k: Material conductivity, T2: Temperature at state 2, T1: Temperature at state 1	In heat transfer, conduction (or heat conduction) is the transfer of thermal between regions of matter due to a temperature gradient.
41	Themal and Fluids	Thermal radiation	Power=5.670373×(10^−8)*Area*View_Factor12*(Temperature1^4 - Temperature2^4)	5	P: Radiated power, sigma: Stephan-Boltzman constant, A: Surface area, F12: View factor frm surface 1 to surface 2, T1: Temperature at state 1, T2: Temperature at state 2	Thermal radiation is electromagnetic generated by the thermal motion of charged in matter.

42	Themal and Fluids	Thermal convection	Energy_Flow=Coeff_Heat_Transfer*Area*(Temperature-Environment_Temperature)	5	P: Thermal energy flow, h: heat transfer coefficient, A: Surface area, Tobj: Temperature of the object, Tenv: Temperature of the environment	Convection is the movement of molecules within fluids (i.e. liquids, gases). It cannot take place in solids, since neither bulk current flows nor significant diffusion can take place in solids.
43	Themal and Fluids	Thermionic emision	Emission_Current_Density=1.20173*10^6*Material_Correction_Factor*(Temperature^2)*2.71828^(-1*Work/(1.3806488 × 10^-23 *Temperature))	6	J: Emission current density, A0: 1.20173e6 Am-2K-2, lambdar: material-specific correction factor, T: Temperature of the metal, W: Work of the metal, k: Boltzman constant	Thermionic emission is the heat-induced flow of charge carriers from a surface or over a potential-energy barrier.
44	Themal and Fluids	Absorption	Coeff_Partition=Concentration_Solute1/Concentration_Solute2	3	x: Concentrations of solute, Kn: Partition coefficient	Absorption, in chemistry, is a physical or chemical phenomenon or process in which atoms, molecules enter some bulk phase - gas, liquid, or solid material.
45	Themal and Fluids	Thermal dissociation	Dissociation_Constant=Equilibrium_Concentration_A*Equilibrium_Concentration_B/Equilibrium_Concentration_AB	3	A,B: Equilibrium concentrations, Kd: Dissociation constant	Reversible breakdown of a chemical compound into simpler substances by heating it (see dissociation)

46	Themal and Fluids	Combustion - Otto Cycle - temperature	Temperature4=Temperature3+Fuel_Air_Ratio*Fuel_Heating_Value/Specific_Heat	5	T: Temperature, f: fuel-air ratio, Q: Fuel heating value, Cv: Specific heat	the sequence of exothermic chemical reactions between a fuel and an oxidant accompanied by the production of heat and conversion of chemical species.
47	Themal and Fluids	Combustion - Otto Cycle - temperature	Pressure4=Pressure3*(Temperature4/Temperature3)	4	T: Temperature, P: Pressure	the sequence of exothermic chemical reactions between a fuel and an oxidant accompanied by the production of heat and conversion of chemical species.
48	Themal and Fluids	Thermo-diffusion - Force	Force=-1*Pressure*Gas_Mean_Free_Path*(Diameter^2)*(Temperature_Gradient))/Temperature	6	Fth: Thermophoretic force, p: Gas pressure, lambda: Gas mean free path, dp: Particle diameter, nablaT: Temperature gradient, T: Absolute temperature of the particle	A phenomenon observed when a mixture of two or more types of motile particles (particles able to move) is subjected to the force of a temperature gradient and the different types of particles respond to it differently.
49	Themal and Fluids	Thermo-diffusion - Velocity	Velocity=-0.55*(Viscosity)*(Temperature_Gradient))/(Density*Temperature)	5	Vth: Thermophoretic velocity, eta: Gas viscosity, ro_g: Gas density, T: Absolute temperature of the particle, nablaT: Temperature gradient	A phenomenon observed when a mixture of two or more types of motile particles (particles able to move) is subjected to the force of a temperature gradient and the different types of particles respond to it differently.

50	Themal and Fluids	Capillary effect	Height= 4*(Surface_Tension)*cos(theta)/(Density*9.81*Diameter)	5	h: Height of meniscus, gamma: liquid-air surface tension, theta: contact, ro: density of liquid, g: local gravitation (9.81), r: Radius of the tube	Capillary action, or capillarity, is the ability of to flow against gravity where liquid spontaneously rises in a narrow space such as a thin tube, or in porous materials such as paper or in some non-porous materials such as liquefied carbon fiber.
51	Themal and Fluids	Evaporation	ln(Pressure2/Pressure1)=(-1*Vaporization_Enthalpy/R)*(1/Temperature2-1/Temperature1)	5	P1, P2: Vapor pressure, T1, T2: Temperatures, delHvap: Enthalpy of vaporization, R: Universal constant	Evaporation is a type of vaporization of a liquid that occurs only on the surface of a liquid.
52	Themal and Fluids	Adsorption	Adsorped_Quantity/Mass=Adsorbent_Adsorbate_k*Pressure^(1/adsorbent_adsorbate_n)	5	x: Quantity adsorped, m: Mass of the adsorbent, P: Pressure of the adsorbate, k, n: Empirical constants	Adsorption is the adhesion of atoms, ions, biomolecules or molecules of gas, liquid, or dissolved solids to a surface.
53	Themal and Fluids	Crystallization	No=Rate_Constant*(Concentration_Solute-Saturation_Concentration_Solute)^Empirical_Exponent	5	B: Number of nuclei formed per unit volume peer unit time, Kn: Rate constant, c: Instantane	Crystallization is the (natural or artificial) process of formation of solid crystals precipitating from a solution, melt or more rarely deposited directly from gas.

					ous solute concentration, cstar: Solute concentration at saturation, n: Empirical exponent	
54	Themal and Fluids	Effusion	Temperature=(1/(3*5.670373*(10^-8)))*Mass*Root_Mean_Square_Velocity^2	4	vrms: Root mean square of molecular speed, m: Molecular weight, Kb: Boltzman constant, T: Temperature	Effusion is the process in which individual molecules flow through a hole without collisions between molecules.
55	Themal and Fluids	Permeation	Mass_Flow=Permeability*Area*Delta_Pressure/Thickness	5	qm: Mass flow, Perm: Specific material permeability, A: Surface area, delp:Pressure difference, t: Material thickness	The penetration of a permeate (such as a liquid, gas, or vapor) through a solid, and is related to a material's intrinsic permeability
56	Themal and Fluids	Volumetric flow	Volumetric_Flow_Rate=Area*Velocity*cos(theta)	4	Q: Volumetric flow rate, A: Area of the surface, C: Fluid velocity, theta: Angle from prependicular to the given surface	the volume of fluid which passes through a given surface per unit time.

57	Themal and Fluids	Bernoulli effect	(Velocity2^2)/2 + 9.81*Height2 + (Pressure2/Density) = (Velocity1^2)/2 + 9.81*Height1 + (Pressure1/Density)	6	v: Fluid flow speed, g: Acceleration due to gravity, z: Elevation, p: Pressure, ro: Density	**Bernoulli's principle** states that for an inviscid flow, an increase in the speed of the fluid occurs simultaneously with a decrease in pressure or a decrease in the fluid's potential energy.
58	Mechanical	Newtonian mechanics - Velocity	Velocity=Initial_Velocity +Acceleration*Time	4	V:Velocity, v0: Initial velocity, a: Acceleration, t:Time	**velocity** is the rate of change of the position of an object, equivalent to a specification of its speed and direction of motion.
59	Mechanical	Newtonian mechanics - Position	Position=Initial_Position+ Initial_Velocity*Time+ 0.5*Acceleration*Time*Time	5	x: Position, x0: Initial position, v0: Initial velocity, t: Time, a: Acceleration	an object's **location** at any particular time.
60	Mechanical	Friction	Maximum_Friction_Force = Friction_Coeff*Normal_Force	3	Ff_Max: Maximum friction force, mu: Coefficient of friction, N: Normal force	the force resisting the relative motion of solid surfaces, fluid layers, and material elements sliding against each other.
61	Mechanical	Centrifugal Acceleration	Centrifugal_Acceleration =Velocity*Velocity*2/Diameter	3	a_c: Centrifugal acceleration, v: Velocity, r: Radius	The acceleration that is caused by the force that draws a rotating body away from the center of rotation.
62	Mechanical	Torque	Torque= Distance*Force*sin(theta) /2	4	to: Torque, r: Radius, F: Force, theta: Angle between force and radius	**moment** or **moment of force** (see the terminology below), is the tendency of a force to rotate an object about an axis, fulcrum, or pivot.

63	Mechanical	Jerk	Jerk = (Acceleration2-Acceleration1)/Time	3	J:Jerk, a: Acceleration, t: Time	**jerk**, also known as **jolt**, **surge**, or **lurch**, is the rate of change of acceleration; that is, the derivative of acceleration with respect to time, the second derivative of velocity, or the third derivative of position.
64	Mechanical	Impulse	Impulse= Momentum2-Momentum1	3	J: Impulse, p2: Momentum at time 2, p1: Momentum at time 1	**impulse** (noted as **I** or **J**) is defined as the integral of a force with respect to time, which gives you the change in the momentum of the body being acted on by the force.
65	Mechanical	Kinetic Energy	Kinetic_Energy=0.5*Mass*Velocity*Velocity	3	K: Kinetic energy, m: Mass, v: Velocity	the **kinetic energy** of an object is the energy which it possesses due to its motion.
66	Mechanical	Potential Energy	Potential_Energy = Mass*9.81*Height	4	U: Potential energy, m: Mass, g: gravity, h: Height	**potential energy** is the energy of an object or a system due to the position of the body or the arrangement of the particles of the system.
67	Mechanical	Work	Work = Force*Distance*cos(theta)	4	W: Work, F: Force, r: Distance, theta: Angle between force and distance	A force is said to do work when it acts on a body so that there is a displacement of the point of application in the direction of the force.
68	Mechanical	Power	Power = Work/Time	3	P: Power, W: Work, t: Time	power is the rate at which energy is transferred, used, or transformed.
69	Mechanical	Mechanical Power	Power = Force*Velocity*cos(theta)	4	P: Power, F: Force, v: Velocity, theta: Angle between force and velocity	power is the rate at which energy is transferred, used, or transformed.

70	Mechanical	Spring - Potential Energy	Potential_Energy= 0.5*Spring_K*Distance*Distance	3	U: Potential energy, k: Spring constant, x: Position	Potential energy stored as a result of deformation of an elastic object, such as the stretching of a spring. It is equal to the work done to stretch the spring, which depends upon the spring constant k as well as the distance stretched.
71	Mechanical	Period - Spring	Period=2*3.14*(Mass/Spring_K)^0.5	3	Ts: Period, m: Mass, k: Spring constant	The time it takes for spring to pass one complete cycle.
72	Mechanical	Period-Pendulum	Period=2*3.14*(Length/9.81)^0.5	3	Tp: Period, l: Length of the pendulum, g: gravity constant	The time it takes for pendulum to pass one complete cycle.
73	Mechanical	Period-Frequency	Period=1/Frequency	2	T: Period, f: Frequency	Frequency is the number of occurrences of a repeating event per unit time.
74	Electricity and Magnetism	Coulomb's Law	Force = 1/(4*3.14*EMF)*Charge1*Charge2/(Distance*Distance)	5	F: Force, ep0: EMF, q1, q2: Point charges, r: Distance	Coulomb's inverse-square law is a law of physics describing the electrostatic interaction between electrically charged particles.
75	Electricity and Magnetism	Electric Field	Electric_Field=Force/Charge	3	E: Electric field, F: Force, q: Point charge	An **electric field** is generated by electrically charged particles and time-varying magnetic fields. The electric field describes the electric force experienced by a motionless electrically charged test particle at any point in space relative to the source(s) of the field.
76	Electricity and Magnetism	Electric Potential Energy	Potential_Energy=Charge*Voltage	3	U: Potential energy, q: Point charge, V: Potential difference	**Electric potential energy**, or **electrostatic potential energy**, is a potential energy (measured in joules) that results from conservative Coulomb forces and is associated with the configuration of a particular set of point charges within a defined system.
77	Electricity and Magnet	Average Electric Field	Average_Electric_Field= -1*Voltage/Distance	3	Eavg: Average electric	Average electric field is defined as the electric potential difference divided

	ism				field, V: Potential difference, d: Distance	by the distance
78	Electricity and Magnetism	Capacitance	Capacitance=Charge/Voltage	3	C: Capacitance, Q: Charge, V: Electric potential	Capacitance is the ability of a body to store an electrical charge.
79	Electricity and Magnetism	Capacitance	Capacitance=EMF*Area/Distance	4	C: Capacitance, ep0: EMF, A: Area, d: Distance	Capacitance is the ability of a body to store an electrical charge.
80	Electricity and Magnetism	Potential Energy - Capcitor	Potential_Energy=0.5*Capacitance*Velocity*Velocity	3	U: Potential energy, C: Capacitance, V: Potential difference	The potential energy that is stored in a capacitance due to potential difference
81	Electricity and Magnetism	Current	Current=Charge/Time	3	I: Current, Q: Charge, t: Time	Electric current is a flow of electric charge
82	Electricity and Magnetism	Resistance	Resistance=Resistivity*Current/Area	4	R: Resistance, ro: Resistivity, I: Current, A: Area	the opposition to the passage of an electric current through that conductor; the inverse quantity is electrical conductance, the ease at which an electric current passes.
83	Electricity and Magnetism	Electrical Power	Power=Current*Voltage	3	P: Power, I: Current, V: Potential difference	**Electric power** is the rate at which electric energy is transferred by an electric circuit.
84	Electricity and Magnetism	Capacitors in parallel	Total_Capacitance=Capacitance1+Capacitance2	3	Ct: Total capacitance, C1, C2: Individual capacitance	The total capacitance when two capacitators are placed in parallel to each other.
85	Electricity and Magnetism	Capacitors in series	Total_Capacitance=(Capacitance1*Capacitance2)/(Capacitance1+Capacitance2)	3	Ct: Total capacitance, C1, C2: Individual capacitance	The total capacitance when two capacitators are placed in series.

86	Electricity and Magnetism	Resistances in series	Total_Resistance=Resistance1+Resistance2	3	Rt: Total resistance, R1, R2: Individual resistance	The total resistance when two capacitators are placed in series.
87	Electricity and Magnetism	Resistances in parallel	Total_Resistance=(Resistance1*Resistance2)/(Resistance1+Resistance2)	3	Rt: Total resistance, R1, R2: Individual resistance	The total resistance when two capacitators are placed in parallel to each other.
88	Electricity and Magnetism	Lorentz' Force Law	Force=Charge*Velocity*Magnetic_Field*sin(theta)	5	F: Force, q: Point charge, v: Velocity, B: Magnetic field, theta: Angle	the force on a point charge due to electromagnetic fields. If a particle of charge *q* moves with velocity **v** in the presence of an electric field **E** and a magnetic field **B**, then it will experience this force.
89	Electricity and Magnetism	Ampere's Law	Force=Magnetic_Field*Current*Length*sin(theta)	5	F: Force, B: Magnetic field, I: Current, l: Length, theta: Angle	This law relates the integrated magnetic field around a closed loop to the electric current passing through the loop.
90	Electricity and Magnetism	Magnetic Field	Magnetic_Field=Magnetic_Constant*Current/(2*3.14*Distance)	4	B: Magnetic field, mu0: Magnetic Constant, I: Current, r: Distance	A **magnetic field** is a mathematical description of the magnetic influence of electric currents and magnetic materials.
91	Electricity and Magnetism	Magnetic Flux	Magnetic_Flux= Magnetic_Field*Area*cos(theta)	4	Phi: Magnetic flux, B: Magnetic field, A: Area, theta: Angle	the magnetic flux through a surface is the component of the magnetic B field passing through that surface.
92	Electricity and Magnetism	EMF	EMF= Magnetic_Field*Current*Velocity	4	Epsilon: EMF, B: Magnetic field, l: Length, v: Velocity	EMF refers to voltage generated by a battery or by the magnetic force according to Faraday's Law, which states that a time varying magnetic field induces an electric current.
93	Themal and Fluids	Fluid Pressure	Pressure=Surface_Pressure+Density*9.81*Height	4	P: Pressure, p0: Pressure	**Fluid pressure** is the pressure at some point within a fluid, such as water or air

					on free surface, ro: Density, h: Height	
94	Themal and Fluids	Pipe Velocity	Velocity2=Area1*Velocity1/Area2	4	v2: Velocity at state 2, v1: Velocity at state 1, A1: Area at state 1, A2: Area at state 2	The velocity of fluid in a pipe with two different diameters.
95	Themal and Fluids	Extension/ Compression due to Temperature Change	Length2= Length1+Coeff_Expansion*Length1*(Temperature2-Temperature1)	5	L2: Length at state 2, L1: Length at state 1, T2, T1: Temperature at each state, alpha: Coefficient of linear expansion	**Thermal expansion/contraction** is the tendency of matter to change in volume in response to a change in temperature
96	Themal and Fluids	Heat	Heat=Mass*Specific_Heat*(Temperature2-Temperature1)	5	Q: Heat, m: Mass, C: Specific heat, T: Temperature	**heat** is energy transferred between a closed system and its surroundings by mechanisms other than work
97	Themal and Fluids	Pressure	Pressure=Force/Area	3	P: Pressure, F: Force, A: Area	the ratio of force to the area over which that force is distributed.
98	Themal and Fluids	Average Molecular Kinetic Energy	Average_Kinetic_Energy = 1.5*5.670373*(10^-8)*Temperature	3	Kavg: Average molecular kinetic energy, Kb: Boltzman constant, T: Temperature	**The average kinetic energy the molecules have**

99	Themal and Fluids	Work Done on a System	Work=-1*Pressure*(Volume2-Volume1)	4	W: Work, P: Pressure, V2, V1: Volumes	work performed by a system is the energy transferred by the system to another that is accounted for by changes in the external generalized mechanical constraints on the system.
100	Themal and Fluids	Internal Energy	Internal_Energy=Heat+Work	3	U: Internal energy, Q: Heat transferred to a system, W: Work	the **internal energy** is the total energy contained by a thermodynamic system
101	Themal and Fluids	Thermal Efficiency	Efficiency=Work/Heat	3	e: Efficiency, W: Work, Q: Heat transferred to a system	An indication of how well an energy conversion or transfer process is accomplished.
102	Mechanical	Mechanical Advantage	Mechanical_Advantage=Force2/Force1	3	MA: Mechanical advantage, Fo: Force Out, Fin: Force in	**Mechanical advantage** is a measure of the force amplification achieved by using a tool, mechanical device or machine system.
103	Mechanical	Centripetal Force	Force=2*Mass*Velocity*Velocity/Diameter	4	F: Force, m: Mass, v: Velocity, r: Radius	The force that makes a body follow a curved path: its direction is alwaysorthogonal to the velocity of the body, toward the fixed point of the instantaneous center of curvature of the path.
104	Mechanical	Mechanical torque-Angular Acceleration	Torque=Moment_Inertia*Angular_Acceleration	3	to: Torque, I: Moment of interia, alpha: Angular acceleration	Torque caused by angular velocity.
105	Mechanical	Rotational Kinetic Energy	Kinetic_Energy=0.5*Moment_Inertia*Angular_Velocity*Angular_Velocity	3	KE: Rotational kinetic energy, I: Moment of inertia, w: Angular	The kinetic energy of a rotating object

					velocity	
106	Thermal and Fluids	Entropy Change at Constant Temperature	Entropy_Change= Heat/Temperature	3	Delta_S: Entropy change, Q: Heat transferred to a system, T: Temperature	The change in entropy (Entropy is a measure of the number of specific ways in which a system may be arranged, often taken to be a measure of disorder.)
107	Mechanical	Stress	Stress= Force/Area	3	Sigma: Stress, F= Force, A: Area	The term stress (s) is used to express the loading in terms of force applied to a certain cross-sectional area of an object.
108	Mechanical	Strain	Strain= (Length2-Length1)/Length1	3	Strain: Mechanical strain, L2: length at state 2, L1: Length at state 1	the mathematical expression of the shape changes resulting from mechanical stresses
109	Mechanical	Stress-Strain	Stress=Elastic_Modulus* Strain	3		The term stress (s) is used to express the loading in terms of force applied to a certain cross-sectional area of an object.

APPENDIX C

COMPONENTS' BEHAVIOR MODEL

Mechanical Function		Component	Equation ID	Equation
Reduce Speed	Speed Reduction Elements	Belt Drvies	1	Force1 = (2.718^(Friction_Coeff*Angle))*(Force2-(Mass*Velocity*Velocity))+(Mass*Velocity*Velocity)
			2	Angular_Velocity_Large_Pulley = Angular_Velocity_Small_Pulley*((Diameter_Small_Pulley+Thickness)/(Diameter_Large_pulley+Thickness)*(1-Belt_Slip)
			3	Power = (Force1-Force2)*Velocity
		Chain Drives	4	Velocity = Pitch*No*Angular_Velocity
			5	Torque = P/Angular_Velocity
		Gear Drives	6	Torque2 = Torque1*Diameter1/Diameter2
			7	Angular_Velocity2 = Angular_Velocity1*Diameter1/Diameter2
Dissipate Energy	Energy Dissipation Elements	Brakes	8	Torque = (No*(Diameter_in+Diameter_Out)/4)*Friction_Coeff*Normal_Force
		Bearings	9	Force = Friction_Coeff* Normal_Force
Transfer Energy	Intermittant Energy Transfer	Clutches	10	Torque = (No*(Diameter_in+Diameter_Out)/4)*Friction_Coeff*Normal_Force
Store Energy	Energy Storage Elements	Springs	11	Force = Spring_K*Distance
		Damper	12	Force = Damper_Constant*Velocity
		Flywheels	13	Energy_Max = 0.5* Mass* Stress/Density
Convert Energy	Prime Movers	DC Motors	14	Torque = Moment_Inertia* Angular_Acceleration
			15	Torque = Voltage*Torque_Constant/Resistance-(Friction_Constant+(Back_Electro_Magnetic_Force_constant*Torque_Constant/Resistance))*Angular_Velocity
			16	Efficiency = Power/(Voltage*Current)

			17	Voltage = (Resistance/Torque_Constant)*Torque + (Resistance*Friction_Constant/Torque_Constant)*Angular_Velocity+Back_Electro_Magnetic_Force*Angular_Velocity
		AC Motors	18	Torque = Moment_Inertia* Angular_Acceleration
			19	Frequency = No* Velocity/120
			20	Current = Power/ (Voltage*Power_Factor)
			21	Power = Torque * Anguular_Velocity
Hydraulics	Fluid Transfer	Duct	22	Flow_Resistance = (Pressure2-Pressure1)* Volumetric_Flow_Rate
			23	Flow_Conductance = 1/Flow_Resistance
			24	Fluid_Capacitance = Area/ (Density*9.81)
		Flow Transformer	25	Volumetric_Flow_Rate2 = Volumetric_Flow_rate1*Pressure1/Pressure2
		Pump	26	Force1 = Area*Pressure2
			27	Velocity1 = Area* Volumetric_Flow_Rate2
			28	Head = 0.102*Pressure/Specific_Gravity
		Cylinder	29	Force = Pressure *Area
			30	Volume = Area* Distance
			31	Volumetric_Flow_Rate = Velocity * Area