Milton Bouchard;Michael Dina;Onoriode Onokpise;Jackson Raines;Zachary Shapiro

  2525 Pottsdamer St. Tallahassee, FL. 32310

Team 505: Model Based System Quadruped

1/10/2023



# Abstract

The abstract is a concise statement of the significant contents of your project. The abstract should be one paragraph of between 150 and 500 words. The abstract is not indents.

*Keywords*: list 3 to 5 keywords that describe your project.

# Disclaimer

Your sponsor may require a disclaimer on the report. Especially if it is a government sponsored project or confidential project. If a disclaimer is not required delete this section.

# Acknowledgement

These remarks thanks those that helped you complete your senior design project. Especially those who have sponsored the project, provided mentorship advice, and materials. 4

* Paragraph 1 thank sponsor!
* Paragraph 2 thank advisors.
* Paragraph 3 thank those that provided you materials and resources.
* Paragraph 4 thank anyone else who helped you.

# Table of Contents

[Abstract ii](#_Toc118396677)

[Disclaimer iii](#_Toc118396678)

[Acknowledgement iv](#_Toc118396679)

[Table of Contents v](#_Toc118396680)

[List of Tables ix](#_Toc118396681)

[List of Figures x](#_Toc118396682)

[Notation xiii](#_Toc118396683)

[Chapter One: EML 4551C 1](#_Toc118396684)

[1.1 Project Scope 1](#_Toc118396685)

[1.1.1 Project Description 1](#_Toc118396686)

[1.1.2 Key Goals 1](#_Toc118396687)

[1.1.3 Markets 1](#_Toc118396688)

[1.1.4 Assumptions 2](#_Toc118396689)

[1.1.5 Stakeholders 2](#_Toc118396690)

[1.2 Customer Needs 3](#_Toc118396691)

[1.3 Functional Decomposition 7](#_Toc118396692)

[1.3.1 Introduction 7](#_Toc118396693)

[1.3.2 Data Generation 7](#_Toc118396694)

[1.3.3 Discussion 9](#_Toc118396695)

[1.3.4 Functional Relationships 10](#_Toc118396696)

[1.4 Targets and Metrics 12](#_Toc118396697)

[1.4.1 Critical Targets/Metrics 12](#_Toc118396698)

[1.4.2 Targets/Metrics Derivation 13](#_Toc118396699)

[1.4.3 Method of Validation 14](#_Toc118396700)

[1.5 Concept Generation 17](#_Toc118396701)

[1.5.1 Concept Generation Tools 17](#_Toc118396702)

[1.5.2 Concept 1: MATLAB GUI textbox focused input 17](#_Toc118396703)

[1.5.3 Concept 2: Simscape Model 18](#_Toc118396704)

[1.5.4 Concept 3: MATLAB GUI with information dashboard 19](#_Toc118396705)

[1.5.5 Concept 4: Racing car game selection 20](#_Toc118396706)

[1.5.6 Concept 5: MATLAB command line function 21](#_Toc118396707)

[1.5.7 Concept 6: MATLAB variables to Simulink model 22](#_Toc118396708)

[1.5.8 Concept 7: MATLAB GUI with dropdowns 23](#_Toc118396709)

[1.5.9 Concept 8: System Composer with GUI and Simulink 23](#_Toc118396710)

[1.6 Concept Selection 25](#_Toc118396711)

[1.6.1 House of Quality (HoQ) 25](#_Toc118396714)

[1.6.2 Pugh Chart 27](#_Toc118396715)

[1.6.3 Analytical Hierarchy Process (AHP) 29](#_Toc118396716)

[1.6.4 Final Selection 30](#_Toc118396717)

[1.7 Spring Project Plan 31](#_Toc118396718)

[Chapter Two: EML 4552C 32](#_Toc118396719)

[2.1 Spring Plan 32](#_Toc118396720)

[2.1.1 Project Plan. 32](#_Toc118396723)

[2.1.2 Build Plan. 32](#_Toc118396724)

[Appendices 33](#_Toc118396725)

[Appendix A: Code of Conduct 34](#_Toc118396726)

[Appendix B: Functional Decomposition 41](#_Toc118396727)

[Appendix C: Target Catalog 42](#_Toc118396728)

[Appendix D: Concept Generation 45](#_Toc118396729)

[Appendix E: Concept Selection 50](#_Toc118396730)

[Appendix A: APA Headings (delete) 62](#_Toc118396731)

[Heading 1 is Centered, Boldface, Uppercase and Lowercase Heading 62](#_Toc118396732)

[Heading 2 is Flush Left, Boldface, Uppercase and Lowercase Heading 62](#_Toc118396733)

[2.1.3 Heading 3 is indented, boldface lowercase paragraph heading ending with a period. 62](#_Toc118396734)

[Appendix B Figures and Tables (delete) 63](#_Toc118396735)

[Flush Left, Boldface, Uppercase and Lowercase 64](#_Toc118396736)

[References 65](#_Toc118396737)

# List of Tables

[Table 1 - Stakeholders 2](#_Toc118396832)

[Table 2 - Customer Needs Questions, Responses, and Interpretations 4](#_Toc118396833)

[Table 3 - Functional Cross Reference 10](#_Toc118396834)

[Table 4 – Critical Targets and Metrics 12](#_Toc118396835)

[Table 5 – Customer Needs used in BPwC 25](#_Toc118396836)

[Table 6 – Selection criteria used in Pugh Charts 27](#_Toc118396837)

[Table 7 - Full Target Catalog 42](#_Toc118396838)

[Table 8 The Word Table and the Table Number are Normal Font and Flush Left. The Caption is Flush Left, Italicized, Uppercase and Lowercase 64](#_Toc118396839)

# List of Figures

[Figure 1 - Functional Hierarchy Chart 8](#_Toc118396789)

[Figure 2 - Text-based Inputs for a GUI 18](#_Toc118396790)

[Figure 3 - Current Simscape model used in CISCOR for their ET-Quad robot 19](#_Toc118396791)

[Figure 4 – A information dashboard for the weather in Tallahassee, Florida 20](#_Toc118396792)

[Figure 5 - An example of kart selection in Mario Kart 8 21](#_Toc118396793)

[Figure 6 – A MATLAB function that accepts parameters from the user and outputs critical targets 22](#_Toc118396794)

[Figure 7 – MATLAB variables tied to blocks in a Simulink model 23](#_Toc118396795)

[Figure 8 – An example System Composer architecture with attached requirements and Simulink model 24](#_Toc118396796)

[Figure 9 - Initial Pugh Chart 28](#_Toc118396797)

[Figure 10 – Final Pugh Chart 29](#_Toc118396798)

[Figure 11 – Preliminary example of final concept: System Composer with a MATLAB GUI and Simulink models 31](#_Toc118396799)

[Figure 12 – Binary Pairwise Comparison table from Excel. 50](#_Toc118396800)

[Figure 13 – House of Quality table from Excel. 50](#_Toc118396801)

[Figure 14 – First Pugh Chart. 51](#_Toc118396802)

[Figure 15 – Second Pugh Chart. 51](#_Toc118396803)

[Figure 16 – Final Pugh Chart. 51](#_Toc118396804)

[Figure 17 – AHP for determining the importance of each selection criterion. Green boxed indicate where user input is necessary. 52](#_Toc118396805)

[Figure 18 – The Normalized Criteria Comparison Matrix, which will be used to determine criteria weights. 52](#_Toc118396806)

[Figure 19 – Criteria weights and their Consistency Vectors. Lambda is the average of the Consistency Vectors. 52](#_Toc118396807)

[Figure 20 – Our Consistency Index and Ratio. CR should be less than 1 to prevent bias. 53](#_Toc118396808)

[Figure 21 – Comparison of concepts for the Accepts old parameters and performance data criterion. 53](#_Toc118396809)

[Figure 22 – Normalized comparison of concepts for the Accepts old parameters and performance data criterion. 54](#_Toc118396810)

[Figure 23 – Criteria weights and Consistency Vectors for Accepts old parameters and performance data criterion. 54](#_Toc118396811)

[Figure 24 – Consistency check for the Accepts old parameters and performance data criterion. 54](#_Toc118396812)

[Figure 25 - Comparison of concepts for the Accepts General Robot Characteristics criterion. 55](#_Toc118396813)

[Figure 26 – Normalized comparison of concepts for the Accepts General Robot Characteristics criterion. 55](#_Toc118396814)

[Figure 27 – Criteria weights and Consistency Vectors for the Accepts General Robot Characteristics criterion. 55](#_Toc118396815)

[Figure 28 – Consistency check for the Accepts General Robot Characteristics criterion 56](#_Toc118396816)

[Figure 29 - Comparison of concepts in the Accepts Performance Specifications criterion. 56](#_Toc118396817)

[Figure 30 – Normalized comparison of concepts for the Accepts Performance Specifications criterion. 56](#_Toc118396818)

[Figure 31 – Criteria weights and Consistency Vectors for Accepts Performance Specifications criterion. 57](#_Toc118396819)

[Figure 32 – Consistency check for Accepts Performance Specifications criterion. 57](#_Toc118396820)

[Figure 33 – Comparison of concepts for Calculate critical targets based on user inputs criterion. 57](#_Toc118396821)

[Figure 34 – Normalized comparison of concepts for Calculate critical targets based on user inputs criterion. 58](#_Toc118396822)

[Figure 35 – Criteria weights and Consistency Vectors for Calculate critical targets based on user inputs criterion. 58](#_Toc118396823)

[Figure 36 – Consistency check for Calculate critical targets based on user inputs criterion. 58](#_Toc118396824)

[Figure 37 – Comparison of concepts for Generates simple block model criterion. 59](#_Toc118396825)

[Figure 38 – Normalized comparison of concepts for Generates simple block model criterion. 59](#_Toc118396826)

[Figure 39 – Criteria weights and Consistency Vectors for Generates simple block model criterion. 60](#_Toc118396827)

[Figure 40 – Consistency check for Generates simple block model criterion. 60](#_Toc118396828)

[Figure 41 – Final rating matrix for our three final concepts in each of our selection criteria. 61](#_Toc118396829)

[Figure 42 – Overall ratings of our final concepts. System Composer GUI is highlighted because it was the idea that was selected 61](#_Toc118396830)

[Figure 43. Flush left, normal font settings, sentence case, and ends with a period. 63](#_Toc118396831)

# Notation

|  |  |
| --- | --- |
| CISCOR | Center for Intelligent Systems, Control and Robotics |
| GUI | Graphical User Interface |
| UI | User Interface |
|  |  |

# Chapter One: EML 4551C

## Project Scope

### Project Description

### The objective of this project is to develop a software tool that expedites the design and construction of quadrupedal robots. The tool will use the knowledge gained from robots previously built at the Center for Intelligent Systems, Control, and Robotics (CISCOR) to create a database that updates and improves the models used by the software tool. This ensures that the theoretical values are more realistic. Key Goals

The primary goal of this project is to develop a tool that will assist users in the development of new quadrupedal robots. The tool will be able to return critical parameter values such as mass, motor size, and motor speed based on user inputs such as desired gaits and requirements. These outputs will be calculated quickly, reducing the amount of time spent in the development phase of each robot. In addition to being a development tool, the final product will also act as a storage of knowledge for robotic development. Initially, the tool will pull from historical data of robots developed by CISCOR, and, as more developments are made using the tool, new data will continue to be added to its database.

### Markets

There are a variety of groups that would have interest in our final product. The primary market for this tool will be CISCOR, as they approached Team 505 asking for a tool to speed up the design and production of their robot. The tool will be developed with its employees, students, and faculty in mind as the main users. This means our tool must be useable for people with varying levels of experience, such as undergraduate students and professors. CISCOR primarily focuses on legged robotics, so our tool will be designed to optimize legged robots.

Secondary markets are other organizations or groups that would be interested in or benefit from the final tool. The secondary markets for this tool include private robotics companies, other educational institutions around the country, and individual robotics hobbyists. Our initial tool will focus on quadrupedal robots, but there are many different types of robots, including bipedal and wheeled robots; therefore, this tool can later be expanded beyond our initial goals to become useful to the robotics field in general.

### Assumptions

To ensure we can finish this project by April 2023, we must constrain our scope. If any software is used, it must be in the MathWorks software suite because CISCOR has readily available access to MathWorks. CISCOR primarily focuses on quadrupedal robots, so the data we have access to will be from these types of robots; therefore, the focus of the tool will be on quadrupedal robots, as this will be most beneficial to our client. Lithium polymer (lipo) batteries are the primary battery chemistry utilized in CISCOR, and as a result our tool will utilize a model of lipo batteries for analysis.

### Stakeholders

Stakeholders are people who have interest, control, or investment in the project. The left column of Table 1 shows who these people are, and the column headings across the top show how they affect the project.

Table 1 - Stakeholders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Investor** | **Decision Maker** | **Advisors** | **Receivers** |
| **Sponsor**  CISCOR, Dr. Clark, Dr. McConomy | **X** | **X** | **X** | **X** |
| **Manager**  Dr. McConomy |  | **X** | **X** |  |
| **Experts**  Dr. Clark, Dr. Hubicki, Dr. Hollis, Dr. McConomy |  |  | **X** |  |
| **Operators**  Graduate researchers, undergraduate students |  |  |  | **X** |
| **General Readers**  Other educational institutions, robotics companies, robotics hobbyists |  |  |  | **X** |

*Note:* Some of the stakeholders can impact the project in more than one way. For example, managers can be advisors and decision-makers.

## Customer Needs

After our initial meeting established the scope of our project, our team met again with our sponsors, Dr. Jonathan Clark and Dr. Shayne McConomy. Dr. Clark is the director of CISCOR and served as our source of information for the specific robotics requirements, and Dr. McConomy answered our questions about systems engineering and provided insight about similar tools already used in the automotive industry. We were able to meet with Dr. McConomy in person and directly ask our questions; we met with Dr. Clark via Zoom to ask similar questions. Our most important questions were related to what inputs they wanted to give the tool and what outputs they expected to get back. Table 2 below shows the questions we asked our sponsors, their answers, and our interpretations.

Table 2 - Customer Needs Questions, Responses, and Interpretations

|  |  |  |
| --- | --- | --- |
| **Questions** | **Responses** | **Interpretations** |
| What kinds of inputs do you expect to provide to the tool? | “End goal variables like size, speed, and types of gaits. Also, the general class of robots. Think like cars – sedan, SUV, etc.” – Dr. McConomy | The tool accepts the general classification of the robot from the user. |
| “Performance specifications of the robot. Things like payload, how long it’s expected to last, the speed, and so on.” – Dr. Clark | The tool accepts end goal performance specifications as inputs from the user. |
| What would you like the tool to determine for you and output? | “How mass is allocated to different systems, power requirements, essentially a target catalog for a robot.” – Dr. McConomy | The tool returns the critical targets. |
| “I want to know how we should build the robot with respect to old robots. What motors do we need, what battery size, how large does it need to be. The compliance in the legs and their link lengths.” – Dr. Clark | The tool uses information from previous CISCOR robots to return the critical targets. |
| Do you want a visual display returned along with the parameter outputs? | “I would like to see a spider plot illustrating the comparison of different designs to specific parameters. Also, an information dashboard.” – Dr. McConomy | The tool returns a visual overview of information. |
| “It’d be great to have a default image of a quadruped in the tool’s GUI that gets updated in real time as you adjust sliders that affect target specifications. Just a simply MATLAB model made of blocks” - Dr. Clark | The tool provides a simple parametric model that updates with changing inputs. |
| Given unlimited time and resources, what would be the ideal end product of this project? | “It would be nice if this then populated a Bill of Materials that was used to create a parametric CAD model.” – Dr. McConomy | The tool returns a Bill of Materials that is used to generate a parametric CAD model. |
| “Using a simple physics engine or Simscape model to see how changing input parameters would affect the speed and motor torques would be really nice.” – Dr. Clark | The tool utilizes a physics engine to visualize how input parameters affect performance characteristics. |

From the interpreted customer needs, the needs that were determined to be most important included the tool’s ability to accept performance specification from the user, and then use data from previous CISCOR robots to return critical parameter targets for new robots. These were determined to be the most important due to their constant emphasis from our project sponsor when gathering information. The last row of Table 2 is determined to be more of a want rather than a need for the project, as these were emphasized as desirable from the sponsor rather than something that is necessary for the project’s success.

## Functional Decomposition

### Introduction

In functional decomposition, the actions and outcomes of the project developed in customer needs are split up into broad systems. Within these broad systems, there are more specific functions arranged and set into each of their appropriate categories. The four broad systems of this project’s functional decomposition consisted of Inputs, Outputs, Modeling, and Simulation. To properly organize our functions by system, a hierarchy chart was created, shown below in Figure 1. To compare each function and determine which are the most important, a cross reference chart was also made. The importance of each function was established by how many systems a particular function could satisfy.

### Data Generation

The data generation of the functions originated in the project scope and was supplemented by the customer needs. The functions we determined describe what the product needs to do, but they must conform to the constraints described in our scope and customer needs. The customer needs depended upon the primary market, CISCOR, and were described in terms of robotics specifications and systems engineering approaches for analysis. The responses from the customers were then interpreted and those interpretations acted as a foundation for our functions. Figure 1 shows the functional hierarchy chart. The systems and functions in blue are necessary based on our customer needs, while those in orange are additional features that our sponsor would like to have but are not critical to the success of the project.

A close-up of a calculator

Description automatically generated with medium confidence

Figure 1 - Functional Hierarchy Chart

### Discussion

The primary goal of this project is to develop a tool that aids the user in the design of quadrupedal robots. From our customer needs, we determined that it was critical to: store and update a database of previous robot specifications; accept user inputs in the form of general robot classifications and performance specifications; and use relevant models to calculate and produce critical targets. These high-level functions each fall under one of our systems.

The first system of the hierarchy chart selected was Inputs. This is due to the goal of having the tool accept several classifications of the quadruped along with the final performance specifications. Key parameters and the associated performance characteristics from past quadrupeds can also be entered and stored into the system to aid in calculations. The user input entered into the tool will serve as the basis for the development of later systems shown in the hierarchy table such as the Output, Modeling, and Simulation.

The Output system is composed of producing and displaying information back to the user. A catalog of critical targets and graphics present information to the user that was calculated based on the user’s inputs. Besides displaying the critical targets to the user, the tool must also save the targets for the user to reference later. Other functions within the system include: Generates Bill of Materials, Populates parametric CAD model, Generates simple block model, and Generates dynamic model.

The Modeling system focuses on generating physical models of the quadruped as well as the equations for determining the critical targets. This system includes four functions: Calculates critical targets based on user inputs, Populates parametric CAD model, Generates simple block model, and Generates dynamic model; of these four, only the calculation of the critical targets is essential to the project’s success.

The Simulation system focuses on using a physics engine to provide the user with a dynamic visual of the proposed quadruped. This system was determined to be supplementary, rather than necessary. It consists of the Populate parametric CAD model, Generate simple block model, and Generate dynamic model functions.

### Functional Relationships

Table 3 - Functional Cross Reference

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Inputs** | **Outputs** | **Modeling** | **Simulation** |
| Accepts old parameters and performance data | X |  |  |  |
| Updates model with new information | X |  |  |  |
| Stores old parameters and performance data | X |  |  |  |
| Accepts General Robot Characteristics | X |  |  |  |
| Accepts Performance Specifications | X |  |  |  |
| Produces and stores critical targets catalog |  | X |  |  |
| Produces graphic for comparisons |  | X |  |  |
| Displays an information dashboard |  | X |  |  |
| Calculate critical targets based on user inputs |  | X | X |  |
| Generates a Bill of Materials |  | X |  |  |
| Populates parametric CAD model |  | X | X |  |
| Generates simple block model |  | X | X |  |
| Generates dynamic model |  | X | X | X |

The cross-reference table above provides insight to the priority level of each system. In order from most important to least important, the systems rank as follows: Outputs, Inputs, Modeling, and then Simulation. This is indicated by both the number of functions that each system requires to achieve the final goal of the project, as well as the needs versus wants that were interpreted from the conversation with the project sponsors. The orange functions were determined to be wants and were not critical for the project being deemed successful. These functions will be treated as bonuses to the project and will be focused on after all other major functions have been solidified. As stated above, the Output and Input systems were determined to be the most important. The tool will not be able to assist its users in robot design if it is not able to take in old information and performance data and return new critical targets based on user goals, which is the focus of the project.

When looking at the functions that were determined to be true needs of the project, only the function of Calculates critical targets based on user inputs was determined to span across multiple systems. CISCOR has published many papers which propose equations that model different types of robots, so our team included Calculates user targets in the Modeling system. These models are additionally used for determining and outputting the critical targets, which is why we also included it in the Output system. It should be noted that while the other critical functions do not belong to more than one system, they will interact with each other to complete the goal of the project. For example, any critical parameter that will be calculated will be based on the user inputs and the robot database, but they are not all contained in the same system.

Beyond the critical needs, all the additional wants had cross-system interaction besides the Bill of Materials function. This was because, at their core, all these additional wants were based on different models or simulations that would provide different forms of predictive representation. These models and simulations would provide no use to the users if they were not outputted effectively, therefore creating the interaction between the Output, Modeling, and Simulation systems.

## Targets and Metrics

After determining the functions for this model-based system tool, each function must be assigned a target and metric. The target given to each function represents a quantified goal that the tool must attain to complete the desired function. The metric is a specific parameter, aligning with the target value, for how the function will be validated or completed. The most challenging part of this process was determining useful metrics for software, as our team is more accustomed to mechanical devices. The fully developed target catalog is in Appendix C: Target Catalog. Of our targets, the designated critical targets and metrics are essential for our project’s success.

### Critical Targets/Metrics

Our critical targets and metrics are shown below in Table 4.

Table 4 – Critical Targets and Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| *System* | *Function* | *Target* | *Metric* |
| Inputs (User Defined) | Accepts general robot characteristics | 1 | Binary |
| Inputs (User Defined) | Accepts performance specifications | 1 | Binary |
| Outputs | Produces and stores critical targets catalog | 1 | Binary |
| Outputs/Modeling | Calculate critical targets based on user input | 1 | Binary |
| Additional | Time to order | 15 minutes | Time |
| Additional | Force required at the foot | ±10% | Margin of Error |
| Additional | Torque required at joint | ±10% | Margin of Error |

### Targets/Metrics Derivation

Because this project’s goal is to develop a software tool, many of the targets are focused on whether our tool fulfills the given function or does not. For these functions, our targets are binary, and receive a 1 if that function is met, and a 0 if the function is not met. Another significant concern for our tool is the usability and heuristics of our tool. Some of these are not explicitly stated as functions but are none the less important for our project’s completion. These are the time to order, the force required at the foot, and the torque required at the joint.

Our tool must accept performance specifications and general characteristics of the robot, and then calculate critical targets and store them in a critical targets catalog for our project to be functional. As a result, they are given a binary target of 1.

The time to order of components specified by our tool is another critical target of the tool. The motivation for the project is the reduction of time in the early stages of robot development so our tool must reduce the time to determine critical targets, or else the tool is not useful. As a result, we determined that the user must be able to order components based on information from the tool in 15 minutes. This means that it should take 15 minutes from the time the user fully enters the required inputs to when everything is output, including simulations, if any, and the target catalog. This was determined by conversations with graduate students in the CISCOR lab about how long it takes to develop accurate models and their current simulations to run.

There are also critical targets that our tool must calculate accurately for our user for our tool to be successful. These are the force required at the foot to propel the robot forward, and the torque at the hip joint. We define a target of a ±10% margin of error for these critical targets. This margin of error was chosen based upon a 90% confidence level. When choosing the target for each margin of error, a 95% confidence was deemed too strict for this metric since a physical robot will not be developed and tested.

### Method of Validation

Our team plans to look at one function at a time before adding additional functions, ensuring that additional functions do not disrupt the performance of the tool. For example, our team can start with one of the functions in the Input system, such as Accepts General Robot Characteristics. A simple function or script will be created that prompts the user to enter values or parameters for the robot characteristics, and our team will see if the code successfully accepts the input, meaning it does not give an error, and saves it to a variable. Similar steps can be used to test if the rest of our functions are performed correctly. With each success, we will add another function to the code and repeat the process.

We also plan on our tool being visual in nature which making a graphical user interface (GUI) or showing plots. We will test these capabilities after ensuring the text-based user interface (UI) functions as intended. Our team will repeat the process outlined in the paragraph above: we will take the working code and add visuals to each function that requires it, such as dropdowns for the inputs or outputting a simple block model. If the visual produced matches the numerical data and works with the code, we will add an additional visual, one at a time.

The torque at a leg joint is driven by the required force at the foot to propel the robot in a certain mode, such as swimming. Ideally, our team would provide specifications to our tool, and we would use the results to build a robot and check if the output parameters are correct; unfortunately, we do not have the time nor resources to do so. We are particularly concerned with this target because motors are normally specified by their stall torque and no-load speed, which defines a torque-speed curve. A specific curve can be achieved by a single motor or motor and gearbox combination. Gearboxes have efficiencies that are less than 100%, so there are losses in torques. As such, our team will model the torque requirements assuming a worst-case efficiency of 80% (Max Power, 2017). We will also determine the motor torque from the foot forces to start but check our results by also calculating the forces acting on the foot after we have determined a motor’s torque-speed curve. If the original foot force matches the force back calculated from the motor torque, then the model has accurately calculated the torque requirements.

Many of our metrics are defined as binary, as our project is heavily based on software. Our tool either accepts an input and produces an output or it doesn’t; however, what defines a 0 and 1 can be unclear, so our team plans on using a 5-point Likert scale to clearly define what is a 0 or 1. We will take a 4 or higher on the Likert scale as being a binary 1, while anything between 1 and 3 is a binary 0. Measuring the success of our outputs is partially subjective, so each team member plans on individually using the tool and expressing their satisfaction with the outputs based on the previously described scale. Additionally, we will give the tool to our sponsors and their graduate students to gather the perspectives of our end users to ensure our opinion matches their expectations.

A crucial part of the usefulness of our tool is the time required for the tool to run through its processes. For our functions that generate models, we will use MATLAB’s built-in “Run and Time” function to determine how long they take to complete. When it comes to the total time-to-order, we will additionally validate this by having multiple users go through a typical use scenario of the software, from input to the creation of a target catalog, and use a stopwatch to time the users.

We intend for our tool to be run with the foundation provided by MATLAB already in mind, so the computer system requirements metrics will be validated by comparing against MATLAB’s internal required specifications. Our simulation functions may require more capability from the user’s computer, so our targets for system requirements have been adjusted accordingly. We can also use CISCOR computers to check if they are capable of running our tool.

## Concept Generation

With an understanding of the targets we are trying to achieve and the constraints of our project, our team developed one hundred potential concepts for our tool. The complete list is shown in Appendix D: Concept Generation, but we discuss a few below.

### Concept Generation Tools

Our team found it difficult to generate one hundred potential concepts because we are not used to thinking about software design, but we used tools to make the process easier. These included brainstorming, forced analogy, and anti-problem. For our brainstorming, members of the team shared all ideas that might accomplish all our functions detailed in Functional Decomposition and reserved any judgement until after all our ideas were generated. Team members built off each other ideas by modifying or rearranging elements of a specific idea, and this allowed us to generate a larger variety of ideas. We also thought of different and seemingly unrelated items and listed their attributes. We could then force an analogy between our concepts and the attributes of different items. Anti-problem encourages the team to think of solutions that solve problems that are the opposite of our problem; for example, how can our team make it more difficult to design robots? Our team used the attributes of these opposite solutions to generate more concepts that deal with our problem. This tool was the least used.

### Concept 1: MATLAB GUI textbox focused input

One of the medium fidelity concepts is an app developed in MATLAB that allows the user to type in the desired value for each parameter for their robot design. Figure 2 shows what this idea might look like and outlines the textboxes in red. The figure shows an example of what this concept might look like and include. For each parameter, there will be an empty text box for the user to type a word or number defining the performance characteristics they want their robot to obtain. These inputs will be used by a function our term creates to determine the output parameter, such as torque and speed requirements. The GUI also allows us to visually display the outputs to the user.

Graphical user interface, application, PowerPoint

Description automatically generated

Figure 2 - Text-based Inputs for a GUI

### Concept 2: Simscape Model

Another medium fidelity solution was building a model using MathWorks’s Simscape. Simscape is a multi-body engine that allows the user to create dynamic models, and Figure 3 shows an example of a model currently used in CISCOR. Because we will have an extensive database of robot builds, including the specific components and performance, Simscape could potentially prove to be a very effective tool. When discussing potential software with sponsors, it was expressed that Simscape works well when you know the items you are building rather than the equations to describe the system. Generating new robots may be difficult when only using historical data, and we would also potentially limit ourselves to components that have been used in the past but working through those challenges could potentially lead to rewarding solutions.

Diagram

Description automatically generated

Figure 3 - Current Simscape model used in CISCOR for their ET-Quad robot

### Concept 3: MATLAB GUI with information dashboard

The third medium fidelity concept also uses a user interface made in MATALB to present information to the user. The user will input values describing the wanted performance characteristics to the interface and MATLAB will display an information dashboard presenting all important parameters needed to accurately develop the robot the user desires. This information dashboard will present values like the motor specs, link length, etc. It is like Concept 1, but it presents the output targets to the user in a different manner. Figure 4 shows an example dashboard for the weather in Tallahassee, FL. Our information does not need to be quite as complex, but the concept is similar.

Graphical user interface, application, website

Description automatically generated

Figure 4 – A information dashboard for the weather in Tallahassee, Florida

### Concept 4: Racing car game selection

A fourth medium fidelity potential solution was a program that acts similarly to a vehicle selection tool from a video game such as Mario Kart. Figure 3 shows this selection screen from Mario Kart 8 (Mario Kart 8, 2012). The yellow box with arrows shows the user selecting between a variety of available karts and modifications. Our tool would ideally take in the desired performance parameters from the user, then allow the user to select multiple different options for general robot characteristics such as number of legs and leg type. Additionally, a visual comparison tool in the form of a spider plot would be provided for the user showing how each potential design compares to others in a variety of performance areas. If our model is efficient enough, we could also simulate what the potential design would look like as the user selects different options.

Graphical user interface

Description automatically generated

Figure 5 - An example of kart selection in Mario Kart 8

### Concept 5: MATLAB command line function

The last medium fidelity concept is to have the user call a function in the command line of the MATLAB software and input the wanted performance characteristic values as function arguments. These characteristics will be sent to the specified function, and MATLAB will output the parameter values for the provided constraints. MATLAB also has plotting capability, and these can be leveraged to show the user the outputs in a visual genre. Figure 4 shows an example of what this might look like in MATLAB.

Graphical user interface, text, application, email

Description automatically generated

Figure 6 – A MATLAB function that accepts parameters from the user and outputs critical targets

### Concept 6: MATLAB variables to Simulink model

Our first high fidelity idea is to create a MATLAB script that divides the necessary input parameters into different sections. The user can then modify the workspace variables by editing their values in the script and then rerunning the script. These variables can be tied to values in the Simulink model, so the Simulink values are modified to reflect the user’s inputs to the MATLAB script. The Simulink model is then run using the user’s inputs, such as payload and run time, to determine possible values for critical targets like battery size and link lengths. Simulink can generate plots, and we can also have a supplementary function or script to provide the user with the critical targets in a visual format, such as spider plots. Other functions can also save the critical targets to a file for the user. Figure 7 shows this collaboration between MATLAB and Simulink.

Graphical user interface, application

Description automatically generated

Figure 7 – MATLAB variables tied to blocks in a Simulink model

### Concept 7: MATLAB GUI with dropdowns

Another high-fidelity concept was a MATLAB GUI with dropdown menus to assist the user with inputting their desired design characteristics. We believe that having dropdown menus with clear and predefined options for the user will result in a better user experience than requiring users to type into text prompts. These drop-down menus will be included for input parameters such as desired run time and payload capabilities. For outputs, the GUI can calculate them using a function that accepts the user’s inputs and contains the necessary models. The outputs can also be shown visually as spider plots or information dashboards.

### Concept 8: System Composer with GUI and Simulink

One of our high-fidelity ideas involves using a MATLAB GUI that provides values to a System Composer architecture and models the robot in Simulink. System Composer, MATLAB, and Simulink are all software packages in the MathWorks suite, so these are programs CISCOR has access to. System Composer allows the user to define the architecture of a system using functions and targets. The user can provide performance characteristics, such as run time, and general characteristics, such as quadruped, to the prompts present on the GUI. The performance characteristics can populate specific targets for different functions in System Composer, and switch statements in System Composer and Simulink can use the general characteristics to determine what models to use. This allows us to attach specific Simulink models to different System Composer functions, so the Simulink results can be compared against the targets to see if we meet our constraints. Figure 8 shows an example architecture with a motor, battery, and leg lengths, and shows requirements, outlined in blue, and a Simulink model, outlined in green. The GUI also allows us to visually present information to the user, which is something our sponsor emphasized.

Diagram

Description automatically generated

Figure 8 – An example System Composer architecture with attached requirements and Simulink model

## 1.6 Concept Selection

When beginning the process of selecting the best of our remaining concepts, several concept selection tools were used to assist in narrowing down our list and removing bias from the process. These tools included Binary Pairwise Comparisons (BPwC), House of Quality (HoQ), Pugh Charts, and Analytical Hierarchy Processes (AHP). The complete selection process and outcomes can be found in Appendix E: Concept Selection.



### House of Quality (HoQ)

When beginning the process of selecting the best of choice of our remaining concepts, we first had to determine which interpreted customer needs would have the highest priority in our design choices. To do this, we established a binary pairwise comparison table. The complete list of customer needs is in Table 2, but Table 5 below shows the summary of customer needs we used for our BPwC. The customer need in a row was compared to the customer need in the column, giving a 1 for more important and a 0 for less important, and a hyphen was placed in the main diagonal. Using the results from this table, an importance weight factor was established and later used in our House of Quality table. From our pairwise comparison table, it was determined that the highest priority for our tool was the reduction of development time, and the lowest priority was the dynamic model. The entire binary pairwise comparison table can be found in Appendix E: Concept Selection.

Table 5 – Customer Needs used in BPwC

|  |
| --- |
| Input General Class |
| Input Performance Specs |
| Output Critical Targets |
| Based on History |
| Visuals for Data |
| Simple Block Model |
| Parametric CAD Model |
| Bill of Materials |
| Dynamic Model |
| Development Time |
| User Interface |

Utilizing the importance weight factors determined above, the customer requirements and engineering characteristics were compared to determine how each individual characteristic would contribute to the customer requirements and rank them against each other. Our functions served as our engineering characteristics. We ranked them by giving each characteristic a score of 0, 1, 3, or 9. A 0 meant that the characteristic did not contribute to the requirement while 9 meant it significantly contributed. This score was then multiplied by the weight factor calculated in during the BPwC and then summed vertically within the characteristic. These sums were also summed to create a total raw score, which is used to rank them. The relative weight of each engineering characteristic was determined by dividing the characteristic’s raw score by the total raw score. The HoQ allowed us to translate the ideas and needs of our customer into engineering characteristics that we have control over in the design process. This ensures that the concepts our team feel are superior are also satisfy our customer needs. From our House of Quality table, the most important characteristics were identified to be the tools’ ability to accept performance specifications from the user and for it to be able to calculate critical targets based on user inputs. The least important characteristics of the tool included its ability to continuously update the model with new information and its ability to store old parameters and performance data. These are reasonable results as our sponsor has expressed that they are more concerned with the components necessary to build a new robot than creating a database of old robots. The entire HoQ can be found in Appendix E: Concept Selection.

Some of our criteria from the HoQ continued with us to our Pugh Charts. Our group determined the cutoff by calculating the average criteria weights and selected the ones that were greater than the average. We also considered calculating the maximum difference between the weight when they were sorted in ascending order, but this resulted in only one criterion continuing. The selection criteria we used are listed in Table 6.

Table 6 – Selection criteria used in Pugh Charts

|  |
| --- |
| Accepts old parameters and performance data |
| Accepts general robot characteristics |
| Accepts performance specifications |
| Produces and stores critical targets |
| Calculate critical targets based on user inputs |
| Generates simple block model |

### Pugh Chart

The next step in the design selection process was utilizing Pugh Charts to compare potential design choices directly. Pugh Charts were used to measure our high and medium fidelity concepts and compare them to a datum. Within the charts, the designs are compared to the datum using symbols: + meaning better than the datum, - meaning worse than the datum and S meaning satisfactory compared to the datum. As the process continues, ideas that are found to be much worse than the datum are removed, a new neutral datum is selected, and the process continues.

A picture containing application

Description automatically generated

Figure 9 - Initial Pugh Chart

In the first iteration of our Pugh charts, our initial datum was SolidWorks shown in the figure above. This datum was chosen because this is an existing software CISCOR uses to aid in robot design. We removed the Simscape Model and MATLAB Command Line concepts after this iteration because they scored the lowest and replaced SolidWorks with MATLAB to Simulink. This concept was chosen because it had five pluses and zero minuses, but there four other concepts with the same scoring, any of which would have served as a good datum. After two more iterations with MATLAB to Simulink being our datum, we chose our three concepts: MATLAB to Simulink, MATLAB GUI Info Dashboard, and System Composer GUI. Figure 10 below displays the final Pugh Chart. MATLAB to Simulink was chosen over MATLAB GUI Dropdowns because the former was better at accepting performance specifications, which was one of the most important functions to our sponsor. Our three final concepts were then used in an Analytical Hierarchy Process. The complete set of Pugh Charts are in Appendix E: Concept Selection.

Table

Description automatically generated

Figure 10 – Final Pugh Chart

### Analytical Hierarchy Process (AHP)

The Analytical Hierarchy Process (AHP) was used to establish weights for our selection criteria and determine if our concept judgement was biased. The complete charts are in Appendix E: Concept Selection, but their results are summarized here. The first chart developed in the process tested criteria against other criteria, using a ranking of 1, 3, 5, 7, or 9 to determine how much better the horizontal criteria was compared to the vertical criteria. If the vertical criteria were determined to be better, the inverse of the value would be applied (Example: 0.33 vs 3). 1 meant that the criteria demonstrated equal importance while a 9 meant that one criterion was significantly more dominant than the other.

Following the completion of the initial table, the values were then normalized in a different comparison matrix. The normalization was done by taking each cell and dividing it by the sum of its column vertically. Criteria weights were then established by averaging each row horizontally, giving a value for how impactful each criterion would be in the final design selection. Once criteria weights had been established, the Weighted Sum Vector (Ws) and Consistency Vector (Cons) were calculated using matrix operations and vector division. Using those values, a consistency ratio was calculated. The Consistency Ratio is ideally under 0.10, which we were able to achieve with a value of 0.9.

### Final Selection

To determine the best concept for our problem, the same process above was completed for each established criteria and tested the three remaining concepts against each other. The desired consistency ratios are once again ideally below 0.10, with a lower value meaning better. This process was completed for all 6 of the criteria being analyzed for this project, and each of those tables and charts can be found in the appendix of this document. Our highest consistency ratio returned from this process was only 0.06, indicating very little bias was involved in the selection process.

Finally, a Final Rating Matrix was calculated to show the performance of each concept in each criterion. These values are then used to calculate how each concept performed overall by multiplying them by the criteria weights that were established earlier in the AHP. To perform the matrix operation, the Final Rating Matrix was first transposed, and then multiplied by the criteria weight matrix, resulting in three final values. The larger of these values indicates a design that is more applicable for the task at hand and, in turn, our best and final design.

After the completion of all Analytical hierarchy processes and Alternative Comparison methods, it was determined that the potential solution of using a Systems Composer GUI was the best design for the task at hand; however, it should be noted that its alternative value of 0.42 was not much larger than the MATLAB to Simulink 0.39, so additional justification would still be required to move forward with this solution. System Composer allows the user to create requirements and link them to functions and models. We decided that this ability to check if our outputs met the user defined constraints makes it truly the better solution over the Simulink option. This is not an explicit criterion but is implicit to our tool’s ability to accept constraints from the user in the form of performance specifications. Figure 11 again shows the System Composer concept that was first presented in Concept Generation.

Diagram

Description automatically generated

Figure 11 – Preliminary example of final concept: System Composer with a MATLAB GUI and Simulink models

## 1.7 Spring Project Plan

# Chapter Two: EML 4552C

## 2.1 Spring Plan



### Project Plan.

### Build Plan.

# Appendices

# Appendix A: Code of Conduct

**Mission Statement**

Our mission is to engage in engineering endeavors and technology to propel ourselves into the next generation of engineering design. Team 505 will go above and beyond the call of duty to produce high-quality work while maintaining a productive, friendly, and professional atmosphere.

**Outside Obligations**

Each team member has specified external obligations which they may need to attend to, as listed below. These obligations are subject to change.

**Milton B.:** No outside obligations related to the university other than classes.

**Michael D:** TA and undergraduate researcher for a combined 20 hours a week. Office hours are TBD but will be chosen so that they do not interfere. A member of the CSU on campus, which meets Thursdays from 7:30 PM – 9:00 PM.

**Onoriode O.:** Is a field technician on call that may need to head to work on short notice, often outside of Tallahassee. Does not work on Sundays but can be flexible if needed.

**Jackson R.:** Works as a service technician at CCS within the FAMU-FSU College of Engineering on Mondays, WednesdaysFridays. Is flexible on those days for standup meetings but generally unavailable for full-length meetings unless an exception is made.

**Zachary S.:** Works as a part-time engineering intern remotely. Time obligation is at most ten hours a week, without a set schedule concerning workdays each week.

**Team Roles**

**Milton B.: Modeling Engineer**

The Modeling Engineer is responsible for the modeling and testing of the Simulink model architecture. They must ensure that all information from the user is modeled and developed properly to link to the System Composer architecture.

**Michael D.: Systems Engineer**

The Systems Engineer is responsible for the design and coding of the System Composer model architecture. They must ensure that all Simulink models are properly attached to different System Composer Components and that all ports and interfaces are properly defined so that data is shared across components.

**Onoriode O.: User Interface Engineer**

The User Interface Engineer is responsible for the design of user interfaces for the software to be developed. They will analyze the functionality the software should have as well as define and design its navigation model. They will design style, content, and graphics which connect a user to the software.

**Jackson R.: Testing Engineer**

The Testing Engineers will be responsible for validating that designs will function as expected. This involves validation of the outputs of our tool and testing physical components of relevant robot platforms and compiling relevant data.

**Zachary S.: Testing Engineer**

The Testing Engineers will be responsible for validating that designs will function as expected. This involves validation of the outputs of our tool as testing physical prototypes and compiling relevant data.

As the project scope and our understanding of the project expands, Team 505 expects to discuss new roles or expand existing ones so that every aspect of the project is thoroughly fulfilled. New responsibilities will be assigned to each team member on a case-by-case basis at full team meetings.

**Communication**

The primary means of communication within the group will be through Microsoft Teams, which will also facilitate file sharing and the transfer of important documents. Text messaging will be used as a secondary means of communication for instantaneous updates or during Teams service outages. Members are expected to reply within 24 hours of a message; reactions, such as liking a message, are considered a response, but, if the author wants elaboration, they can solicit a more detailed response. If a message or notification requires a more urgent response, a member can mention a channel in Teams to notify everyone.

Currently, only one channel, the General channel, exists in our team. As more details about the project become available, Team 505 understands that more channels and/or tabs may be created to help organize and focus conversations relating to different aspects of the project.

**Professional Communication**

External communications with entities such as the project manager (Dr. McConomy) and sponsor (Dr. McConomy and Dr. Clark) will be handled primarily through email. The designated point of contact for email communications is Zachary S, who can be emailed at zbs19@fsu.edu.

**Dress Code**

The following sections specify the dress code policies for different circumstances that require the student team to be present.

**Presentation Dress Code**

For presentations, all team members are expected to wear professional business attire. Color and style schemes will be arranged to guarantee a professional look between group members. E.g.: Long sleeve button shirt, a tie, slacks, and dress shoes.

**Sponsor Meeting Dress Code**

For meetings with sponsors and advisors, all team members are expected to wear polo style collared shirts and khaki pants with closed toed shoes.

**Senior Design Day Dress Code**

For Senior Design Day and future team photos, all team members are expected to wear professional business attire including a jacket. E.g.: Jacket, long sleeve button down shirt, a tie, slacks, and dress shoes.

**Attendance Policy**

Team meetings will occur daily, with three out of five days consisting of 15-minute standing meetings; two days are reserved for a 1-hour long meeting. The standing meetings will occur virtually with the schedule being as follows: Mondays, 4:45 PM to 5 PM; Wednesdays, 9:30 AM to 9:45 AM; and Fridays, 2:30 PM to 2:45 PM. Due to outside obligations, the team understands that Michael Dina cannot attend the Tuesday and Thursday meetings. Standing meetings will consist of members updating the rest of the team on any progress made, any issues that the member needs help with, and goals for that day.

The longer meeting will occur in person on Tuesdays and Thursdays from 11 AM to 12:30 PM; a location will be determined beforehand. Members will update each other on progress made since the previous week and outline goals for the coming week. This time is also used for the group to work together on any tasks that would require all members to be present. Team 505 has agreed that no weekend meetings will occur without prior notice. If the team is to meet during the weekend, a decision must be made by Thursday at the latest. If a team member wants to schedule an emergency meeting outside of regularly scheduled hours, 24-hour notice, if possible, is required.

All members are expected to attend meetings, except for the circumstances mentioned above. If a member expects to be late, they should try to notify the rest of the team as soon as possible, but the remaining members will be understanding if extreme circumstances arise. If a member persistently misses meetings without warning, the team will sit down at a Wednesday meeting to discuss the issue and a resolution.

**Actions Before Involving Dr. McConomy**

Any grievances between group members will be brought up at the daily standup meeting. If the problem persists, using a three strike system, a sit-down discussion will be held during the weekly full-length meeting. If the issue requires input from the entire group, a vote will be held to determine what should be done to remedy the situation. If the issue is not able to be resolved in this manner, then we consult Dr. McConomy for his advice on an equitable solution.

**Project Sponsor Meeting Structure**

Each meeting with project sponsors will begin with a brief outline of the planned procedure for that meeting. Then these topics will be discussed by the group member most familiar with the topic, or whose team role best aligns with the nature of the topic. After all, topics have been discussed, there will be a round table section of the meeting for general or last-minute questions, as well as any relevant housekeeping information.

**Code of Conduct Amendment Process**

An amendment to the code of conduct can be proposed by any team member. For the proposed amendment to be ratified, a team meeting with a formal vote will be conducted. For the amendment to pass, 4 of the 5 members must approve. If one member feels that they are being personally targeted by a proposed amendment, they have the right to raise a complaint with the rest of the team, allowing them to voice their discontent. Team 505 will then follow the guidelines listed in the **Actions Before Involving Dr. McConomy** section. If the team member still feels like they are being treated unfairly, they can request input from Dr. McConomy on the fairness of the amendment being proposed. After the amendment passes, it will be added to the document accompanied by the date of its approving vote, and all members of the group must re-sign the edited code of contact.

**Statement of Understanding / Group Signatures**

By agreeing to and signing the presented code of conduct, each member of the team acknowledges they have read, understand, and agree to the code of conduct terms presented above. Each student also agrees that they have agreed to this document to give their maximum effort to follow the agreement and agreed to it of their own will, free of any other influences.

**Milton Bouchard**

Signed: Date:

**Michael Dina**

Signed: Date:

**Onoriode Onokpise**

Signed: Date:

**Jackson Raines**

Signed: Date:

**Zachary Shapiro**

Signed: Date:

# **Appendix B: Functional Decomposition**

A close-up of a calculator

Description automatically generated with medium confidence

# Appendix C: Target Catalog

Table 7 - Full Target Catalog

*Note*. Targets in blue are required for the project, while those in orange are optional additions. Bolded targets are critical targets.

|  |  |  |  |
| --- | --- | --- | --- |
| *System* | *Function* | *Target* | *Metric* |
| **Inputs (User Defined)** | **Accepts general robot characteristics** | **1** | **Binary** |
| **Inputs (User Defined)** | **Accepts performance specifications** | **1** | **Binary** |
| Inputs (Database) | Accept old parameters and performance | 1 | Binary |
| Inputs (Database) | Updates model with new information | 1 | Binary |
| Inputs (Database) | Stores old parameters and performance | 1 | Binary |
| **Outputs** | **Produces and stores critical targets catalog** | **1** | **Binary** |
| Outputs | Produces graphic for comparisons | 9th grade | Reading level comprehension |
| Outputs | Displays an information dashboard | 9th grade | Reading level comprehension |
| **Outputs/Modeling** | **Calculate critical targets based on user input** | **1** | **Binary** |
| **Additional** | **Force required at foot** | **±10%** | **Margin of Error** |
| **Additional** | **Torque required at joint** | **±10%** | **Margin of Error** |
| Additional | Battery Capacity | ±10% | Margin of Error |
| **Additional** | **Time to order** | **15 minutes** | **Time** |
| Additional | Useability (Total Errors Made) | 20 | Number of Errors |
| Additional | Useability (Errors per Interaction) | 2 | Number of Errors |
| Additional | System Requirements | 8GB+ | RAM |
| Additional | System Requirements | Intel i3, Ryzen R3 or greater | Processor Model |
| Additional | System Requirements | 2 GB+ | Storage |
| Outputs | Generates a Bill of Materials (BoM) | 1 | Binary |
| Outputs/Modeling | Populates parametric CAD model | 1 | Binary |
| Outputs/Modeling | Generates simple block model | 1 minutes | Time |
| Outputs/Modeling/Simulation | Generates dynamic model | 8 minutes | Time |
| Outputs/Modeling/Simulation | Generates dynamic model | 2 | Degrees of Freedom |
| Outputs/Modeling/Simulation | Generates dynamic model | 1 | Binary |

# Appendix D: Concept Generation

**Brainstorming**

1. Look up tables
2. Encyclopedia
3. Call center
4. Hand drawn drafting
5. 3D Robot model that can be modified in real-time
6. MATLAB GUI with information dashboard
7. MATLAB GUI with a spider plot
8. MATLAB GUI using a Simulink model
9. MATLAB script calling Simulink model
10. MATLAB script calling Simscape model
11. MATLAB GUI using a function
12. MATLAB GUI drag and drop points for speed and torque
13. Simulink model
14. Simscape model
15. Command line function
16. Command line MATLAB script
17. Command line Python script
18. Command line tool which prompts for each parameter one at a time
19. Command line tool which allows for menu-based input
20. User’s manual
21. Python GUI
22. MATLAB web app
23. MATLAB desktop app
24. MATLAB phone app
25. MATLAB GUI with slider focused input
26. MATLAB GUI with drop down focused input
27. MATLAB GUI with text box focused input
28. Tool which uses simple algebraic relations for calculations
29. Tool which uses first order calculus
30. Tool which uses complex higher order calculus (Jacobian)
31. Library stored in GUI
32. Database stored in an internet cloud
33. User uploads their database
34. Peer reviewed results
35. System Composer with Simulink model
36. System Composer with MATLAB function
37. System Composer with MATLAB script
38. System Composer with MATLAB GUI
39. Video game that walks the user through selection of parameters
40. Direct a movie walking user through parameter analyzation process
41. Escape room-esque game that allows the user to choose parameters by walking through door and monitors their choices
42. Magazine that walks the user through the equations
43. YouTube video walking user through equation concepts
44. CD program with MATLAB app installed on it
45. CD that has the database on it and prompts the user for inputs from the terminal
46. Phone app that has tabs for the different user inputs
47. Smart watch app to help calculate user inputs
48. MATLAB app that has tabs for different user inputs
49. Pray for an idea
50. Dream about robot parameter values
51. In-person lecture to determine the necessary specs
52. Kahoot game with the user with questions made by roboticists based on old robot parameters
53. Ask a magic 8-ball for robot specifications
54. Ask 100 roboticists about robot parameters for design
55. Place all the different robot parameters on a dart board, one at a time, and play darts to determine robot’s design
56. Coin toss tournament to determine robot specs
57. Offer Excel spreadsheet with previous robot data
58. Break down all previous robots and give to user to determine parameter values
59. Give user all previous robots to break down and determine parameter values
60. Step by step video tutorial on how to spec motors

**Forced Analogy**

1. Siri-like voice prompt input
2. Television Pay-per-view style access and input
3. Racing car game selection
4. Hologram modified by user
5. Linear regressions based on history
6. Evolutionary algorithm based physical model
7. Intuitive AI
8. Shopping cart of ideas
9. Robot Model Optimization tool
10. Hire full time robotics expert
11. Phone a robotics journal
12. Attend a robotics conference
13. Simulink model with multiple equations running at a time
14. Model recursively asks user if the outputs are satisfactory
15. Tool with a hardware lockout dongle to protect sensitive information
16. Model created based on computer hot keys
17. Tool which has previous robots available as editable templates
18. Unentered values are populated by default minimums
19. GUI that uses history as baseline for defaults
20. Function that uses incremental values for a specific parameter
21. Textbook
22. TI-84 graphing calculator program
23. TI-Inspire calculator program
24. Useful equations pamphlet
25. Online support chat window
26. Handheld dedicated robotic design device
27. Zoom video chat with a roboticist
28. Webex video chat with a roboticist
29. Twitter thread with Dr. Hubicki
30. Reddit AMA with a roboticist
31. Touch based UI with MATLAB script
32. Script that accepts input parameters from physical sensors
33. Mr. Potato Head-style modular robot with parameters editable in a mobile app
34. Tool that procedurally generates random robots for the user to select
35. Custom computers build idea – offer different historical options of motors and other components and let user choose

**Anti-Problem**

1. Interview Dr. Clark
2. Create a website with all the links to various papers published by CISCOR
3. Set up a parametric model to simulate in PTC Creo
4. Set up a parametric model to simulate in SolidWorks
5. Meeting framework that forces project members to meet and hand calculate specifications

# Appendix E: Concept Selection

Table

Description automatically generated

Figure 12 – Binary Pairwise Comparison table from Excel.

Application, table, Excel

Description automatically generated

Figure 13 – House of Quality table from Excel.

Table

Description automatically generated

Figure 14 – First Pugh Chart.

Table

Description automatically generated

Figure 15 – Second Pugh Chart.

Table

Description automatically generated

Figure 16 – Final Pugh Chart.

Graphical user interface, application, table, Excel

Description automatically generated

Figure 17 – AHP for determining the importance of each selection criterion. Green boxed indicate where user input is necessary.

Table

Description automatically generated

Figure 18 – The Normalized Criteria Comparison Matrix, which will be used to determine criteria weights.

A screenshot of a computer

Description automatically generated with low confidence

Figure 19 – Criteria weights and their Consistency Vectors. Lambda is the average of the Consistency Vectors.

Table

Description automatically generated

Figure 20 – Our Consistency Index and Ratio. CR should be less than 1 to prevent bias.

Table

Description automatically generated

Figure 21 – Comparison of concepts for the Accepts old parameters and performance data criterion.

Table

Description automatically generated

Figure 22 – Normalized comparison of concepts for the Accepts old parameters and performance data criterion.

Table

Description automatically generated

Figure 23 – Criteria weights and Consistency Vectors for Accepts old parameters and performance data criterion.

Table

Description automatically generated

Figure 24 – Consistency check for the Accepts old parameters and performance data criterion.

Table

Description automatically generated

Figure 25 - Comparison of concepts for the Accepts General Robot Characteristics criterion.

Table

Description automatically generated

Figure 26 – Normalized comparison of concepts for the Accepts General Robot Characteristics criterion.

Table

Description automatically generated

Figure 27 – Criteria weights and Consistency Vectors for the Accepts General Robot Characteristics criterion.

Table

Description automatically generated

Figure 28 – Consistency check for the Accepts General Robot Characteristics criterion

Table

Description automatically generated

Figure 29 - Comparison of concepts in the Accepts Performance Specifications criterion.

Table

Description automatically generated

Figure 30 – Normalized comparison of concepts for the Accepts Performance Specifications criterion.

Table

Description automatically generated

Figure 31 – Criteria weights and Consistency Vectors for Accepts Performance Specifications criterion.

Table

Description automatically generated

Figure 32 – Consistency check for Accepts Performance Specifications criterion.

Table

Description automatically generated

Figure 33 – Comparison of concepts for Calculate critical targets based on user inputs criterion.

Table

Description automatically generated

Figure 34 – Normalized comparison of concepts for Calculate critical targets based on user inputs criterion.

Table

Description automatically generated

Figure 35 – Criteria weights and Consistency Vectors for Calculate critical targets based on user inputs criterion.

Table

Description automatically generated

Figure 36 – Consistency check for Calculate critical targets based on user inputs criterion.

Table

Description automatically generated

Figure 37 – Comparison of concepts for Generates simple block model criterion.

Table

Description automatically generated

Figure 38 – Normalized comparison of concepts for Generates simple block model criterion.

Table

Description automatically generated

Figure 39 – Criteria weights and Consistency Vectors for Generates simple block model criterion.

Table

Description automatically generated

Figure 40 – Consistency check for Generates simple block model criterion.

Table

Description automatically generated

Figure 41 – Final rating matrix for our three final concepts in each of our selection criteria.

Table

Description automatically generated

Figure 42 – Overall ratings of our final concepts. System Composer GUI is highlighted because it was the idea that was selected

# Appendix A: APA Headings (delete)

# Heading 1 is Centered, Boldface, Uppercase and Lowercase Heading

## Heading 2 is Flush Left, Boldface, Uppercase and Lowercase Heading

### Heading 3 is indented, boldface lowercase paragraph heading ending with a period.

#### Heading 4 is indented, boldface, italicized, lowercase paragraph heading ending with a period.

##### Heading 5 is indented, italicized, lowercase paragraph heading ending with a period.

See publication manual of the American Psychological Association page 62

# Appendix B Figures and Tables (delete)

The text above the cation always introduces the reference material such as a figure or table. You should never show reference material then present the discussion. You can split the discussion around the reference material, but you should always introduce the reference material in your text first then show the information. If you look at the Figure 2 below the caption has a period after the figure number and is left justified whereas the figure itself is centered.



Figure 43. Flush left, normal font settings, sentence case, and ends with a period.

In addition, table captions are placed above the table and have a return after the table number. The second line of the caption provided the description. Note, there is a difference between a return and enter. A return is accomplished with the shortcut key shift + enter. Last, unlike the caption for a figure, a table caption does not end with a period, nor is there a period after the table number.

Table 8  
The Word Table and the Table Number are Normal Font and Flush Left. The Caption is Flush Left, Italicized, Uppercase and Lowercase

|  |  |
| --- | --- |
| Level of heading | Format |
| 1 | **Centered, Boldface, Uppercase and Lowercase Heading** |
| 2 | Flush Left, Boldface, Uppercase and Lowercase |
| 3 | Indented, boldface lowercase paragraph heading ending with a period |
| 4 | Indented, boldface, italicized, lowercase paragraph heading ending with a period. |
| 5 | Indented, italicized, lowercase paragraph heading ending with a period. |

# References

Mario Kart 8. (2012). Nintendo.

Max Power. (2017, November 21). *Know Which Gear Is More Efficient?* Retrieved from Max Power Gears: https://maxpowergears.com/know-gear-efficient/#:~:text=The%20efficiency%20of%20a%20hypoid,ratios%20up%20to%20200%3A1.&text=Helical%20gears%20can%20run%20with,ratios%20up%20to%2010%3A1