MobiCharged
Hazard Analysis

Team Super Charged (No.33)
Nashit Mohammad - mohamn31
Eric Nguyen - nguyee13
Samuel De Haan - dehaas1
Eamon Earl - earle2
Mustafa Choueib - choueibm

October 19th, 2022

Contents

List of Figures

List of Tables

1 Revision History

Table 1: **Revision History**

Author	Date	Version	Description
All	October 19, 2022	Rev 0	Created first draft of document

2 Introduction

2.1 Purpose

The purpose of this Hazard Analysis document is to examine the MobiCharged Project in its stage of development to outline all potential hazards. These hazards include, but are not limited to safety risks, areas of failure and security issues. Along with the highlighted areas of potential hazards, solutions to remove these issues (or mitigate these issues at best) will be outlined.

2.2 Background

Engineers are tasked with design in construction to exceed requirements without hindering safety. Safety is a topic that is never missed within the industry and is continuously being highlighted amongst designs; especially as Engineers are reminded of their moral obligations to society by their awarded rings upon graduation.

As a current process, the construction industry places sensors within concrete spaces to continuously test and/or monitor the integrity of buildings during as well as after construction. Ultimately however, these sensors run out of battery and are required to be re-charged. The industry still faces challenges when attempting to charge these sensors with the method of remote charging as the current products that satisfy remote charging abilities are yet to be optimized. There are a significant number of buildings being built in the GreaterToronto-Area, which is emphasized considering that 70% of cranes within Canada are in just the GTA alone. To place innovation in the sub-field of safety within the industry, it is indeed a requirement to modernize the ability of producing efficient remote charging systems and to have the design process optimized to provide the most effective results.

The system-solution for this will be the development of MobiCharged. This system is separated into two separate components - the software for users as well as the hardware / prototype. The software component of MobiCharged is a machine-learned system that will react to the input of users (in which the input will be the desired outputs / application requirements for the remote charging device) and provide the necessary results (these variables depending on the user inputs can be antenna types, layouts, wavelengths, phases, etc.) in order to satisfy the user's inputs such that they may proceed with producing the devices in a way that it is optimized. This software can be operated in any environment the user chooses such that it can be used in any computing system with sufficient speed, memory the required processors.

The hardware component of MobiCharged is a prototype to be developed for the purpose of demonstration as well as development for the software. This physical component will allow the system to be rooted to the core optimization problem in the real world, as it applies to real products. The physical system will allow placing absolute constraints and limitations into the software for optimal outputs in the software. In addition, this physical system can be implemented for an actual use-case in the field post-demonstrations.

2.3 Scope

The environments in which these physical systems operate are typically from roof-tops and/or high-altitude locations with spacial capabilities to place arrays of these systems. These systems react to user inputted (remotely) data such as the location of the device required to be charged, so that it may orient itself in a manner optimal for that application.

The purpose of the software system, MobiCharge, is a machine learning algorithm that will be used by Mobilite-Power, engineering consultant groups, general contractors and building maintenance teams to optimize the design process required to effectively and efficiently produce the most viable remote charging system. In doing so, this will negate the current process of manually conducting simulations (that requires lengthy computerized numerical calculations), ultimately minimizing cost, manual labor, and the time necessary to produce the required results.

This system will provide users with the optimal configuration of a remote charging device based on the desired output, encrypt data protecting users when producing design results and use data smoothing to ensure the accuracy of the system in a time efficient manner.

The purpose of the hardware system is to root our algorithms optimization in the real world environment. The production of a physical model will assist in the determination of the absolute boundaries that can be fed into the machine learning algorithm. Variable parameter ranges will be able to be derived from the physical model to determine the magnitude to which the boundaries can be pushed within the simulation. The physical system provides a secondary purpose in the form of data collection and verification. In order to increase the breadth of data that we can feed into the algorithm, we must determine the degree of computational error within the simulation results. A physical model will aid in determining this range and lead to further optimization through the machine learning algorithm.

2.4 Definitions & Assumptions

2.4.1 Definitions

Table 2: **Naming Conventions and Terminology**

Word	**Definition/Context**
System Hazard	A hazard associated with the system which typically exists regardless of the status of operation.
Accident	An unintended event which generally leads to a form of loss.
Risk	A probability of exposure to danger.
Phase-Shift System	A system designed to alter the waves distributed such that it moves the phase of a wave.
Antenna Array	A system of antenna designed to distribute waves in an organized layout.
ASCII Values	A standard data-encoding format for electronic communication between computers.
Output Limitation Timers	A software algorithm module designed to measure the time it takes for a process to complete and cancel the process if the time elapsed exceeds a programmed amount.
Functional Requirement	Requirements that describe what the product is supposed to do
Non-functional Requirement	Requirements that describe qualities that product will have
General Contractor	Third party companies that acquire services by Mobilite-Power
ECA	The Electrical Construction Association
Data Smoothing	The process of using old data as well as "future" data in order to predict designs.
ML	"Machine Learning" algorithm.

2.4.2 Assumptions

- There is an assumption that the developers will eventually have access to enough processing power to conduct large quantities of simulations.

- A large underlying assumption regarding this system is that the user does not intentionally attempt to enter inputs incorrectly, as well as provide positive feedback to the system when it is not correct.

3 Component Overview

3.1 Software System

3.1.1 Front End User Display

The Front End User Display component is the component in which the user is able to view. This area is where the user navigates through the software, log-in their accounts, enters inputs, requests data and/or verification, and receives outputs.

3.1.2 Back-End Calculations

The Back-End Calculations component of the software system is for the computerized calculations to occur based on the user's inputs. Note that this does not refer to simulations; this component is merely where the user's inputs are calculated to higher level variables such that the software may then be used to process to create outputs.

3.1.3 Machine-Learning Algorithm

The Machine-Learning component of this software system is where the system receives feedback either from the outputs themselves, or from the user in regards to desirable solutions. The more positive feedback it receives, the more of these inputs the system will retain. Similarly, the more negative feedback the system receives, the less of those specific inputs it will retain. As the system continues to learn, the concept of the system providing suggestions, limitations and of course the optimal solutions will become present.

3.1.4 Data Exporting

The Data Exporting component of the software system is the area of the software system where it exports the results. This not only refers to merely outputting the data to the front-end display component, but also refers to exporting into desirable file systems to be stored as well as encryption processes during transfer.

3.1.5 Server

The Server component will be used to maximize the training available to the machine learner, by having the simulations run on local machines, and passing the data via the online connections to an isolated machine and database, which will encapsulate the Machine-Learning Algorithm component. This will be in a future iteration of the design.

3.1.6 Simulation Integration Software

This component encompasses the pre-existing Matlab simulations with which we will integrate our machine learning algorithm, and the software required to integrate them. It will likely involve a database system as well as kernel modules for real-time polling of said database(s), with a dynamic scheduler.

3.2 Hardware Systems

3.2.1 Power Supply System

The purpose of the power supply system component is to provide usable power to subsequent systems.

3.2.2 Phase-Shift System

This component will work to provide the phase shift required for the antenna array system to properly create wave required interference. The purpose of this is to facilitate constructive interference at the desired location for charging.

3.2.3 Antenna-Array System

This component will contain multiple small arrays and will work in conjunction with the Power Supply system and Phase-Shift System.

3.2.4 System Enclosure

This component is present to enclose the system. Typically, the material of this enclosure is a form of wave-reflective metal. The purpose of the enclosure is such that the waves create a destructive interference in the direction that it is not desired to go towards, and creates an amplified constructive interference in the direction that it is desired in.

3.2.5 Hardware Display System

This component is for the user to understand when the device is operational, functional, etc. The current display system is under development, however, the use of LEDs will most likely be implemented.

3.2.6 Circuits Logic

This will consist of any circuitry required for proper control and use of the overall system.

4 Failure Modes Effects Analysis Table

Below is a failure modes and effects analysis (FMEA) for MobiCharged system.

Design Component	Ref.#	Failure Modes	Causes of Failure	Effects of Failure	Detection	Controls	Recommended Action
Front End User Display	4.1.1.0 (PAR1)	Incorrect Outputs	-Incorrect inputs from user -Incorrect input type -Unintended indexing in Database -Race Conditions	Users carry incorrect outputs that are later used to produce the remote charging devices. This results in a device produced that is not actually the one optimized for certain application	-Software can detect incorrect input types based on ASCII values -Other forms are not detectable	N/A	-Display confirmation screen containing inputs provided by the user -Display/Export the data of outputs along with the user's inputs at all times -Create an "Incorrect Input" pop-up display when the user enters an incorrect input type -Display examples of inputs for user -Display input limitations -Display a "Calculation Failed" screen if calculation fails (and have program execute fail-safe) -Ensure Race Conditions & Indexing errors do not occur by correctly writing program to avoid it (i.e. using "sleep()")
	4.1.1.1 (RFTR1)	Frozen Screen and/or Crash	-User inputs values exceeding calculation capabilities (eg;dividing by 0) -Removal of power to software system during process -Deadlock	-Reboot necessary -Loss of data	N/A	N/A	-Avoid incorrect programming that may cause deadlocks and ensure robustness in code -Encourage user to maintain power to system during operation -Limit users from inputting incorrect data -Provide users examples of data types and/or values that are acceptable

Design Component	Ref.#	Failure Modes	Causes of Failure	Effects of Failure	Detection	Controls	Recommended Action
Front End User Display	4.1.1.2 (ACR1)	Incorrect Login Information	User forgets password and/or username	Loss of previous data history for user	N/A	N/A	-Generate security questions during account creation, thus, if user forgets password they can reset it using security questions -Make software tied to online servers, allowing users to get a reset link their email address
Backend Calculations	4.1.2.0 (SR4)	Failed Calculations (undefined answers)	-User enters values that lead to undefined answers -User enters incorrect data types -User enters extreme data values	-Crash of calculations and no outputs -Potentially outputting incorrect data without warning, which could then be used to produce remote charging devices	N/A	N/A	-Limit the data types users can input based on ASCII values -Ensure correct and sufficient testing is implemented during development
Machine Learning Algorithm	4.1.3.0 (IR1)	Infinite loop	-Incorrect programming -Negligence of exiting loops	-Software crash -Computer crash -Reboot system -Loss of data	-Computer built-in exiting programs -Output limitation timers	N/A	-Ensure correct programming to avoid infinite loops -Enter states of polling to ensure processing does not exceed time limits -Create failure states within code
	4.1.3.1 (RFTR1)	Incorrect Data-Smoothing	-Ineffective algorithms implemented -Limit of data present	-Incorrect data output leading to non-optimized solutions and devices -Catastrophic errors may occur if positive feedback is provided to incorrect output	N/A	N/A	-Produce extensive research to implement the most effective data-smoothing algorithm -Increase data set overtime

Design Component	Ref.#	Failure Modes	Causes of Failure	Effects of Failure	Detection	Controls	Recommended Action
Machine Learning Algorithm	4.1.3.2 (SR2)	Positive feedback applied to incorrect results (mislabelled data)	-Incorrect algorithms implemented Feedback entered incorrectly, repeatedly	-Catastrophic as incorrect outputs will be produced every time -Incorrect data will be implemented when producing remote charging devices -System failure as a whole	Comparison through data	N/A	-Apply verification checks periodically to machine-learned algorithm to ensure it matches up correctly to existing solutions and data -Disallow users from directly inputting labelled data (can only be passed as the output to a simulation)
Data-Exporting	4.1.4.0 (SR3)	Unable to export	-Export file type not supported -Exporting process stopped due to higher priority preemption or power loss during process	-Exporting failed, data is not exported to user -Data is not saved	-Software check -Computer built in exit programs	N/A	-Provide user the requirements of installing the software to ensure the necessary support is present -Deny the installation of the software system if necessary support is not present -Create the programs modular and preemptable such that the process can continue after halt
	4.1.4.1 (SR4)	Incorrect data outputted	-Race conditions -Incorrect indexing through data	Incorrect solution provided	Visual check between correct data displayed to user and exported data	N/A	-Thorough programming to avoid race conditions -Apply verifications to ensure indexing is correct
	4.1.4.2 (SEC R1)	Vulnerable data	-Data leaks	Possibly critical client data available to malicious parties	Ethical hacking attempts to assess vulnerabilities	N/A	Encrypt outbound data on local machines before transmitting via the server (SR3)

Design Component	Ref.#	Failure Modes	Causes of Failure	Effects of Failure	Detection	Controls	Recommended Action
Server	4.1.5.0 (SR5)	Server inaccessible	-Server at capacity -server hardware malfunction / internet access restricted	-Inability to serve users -Loss of data	An inevitable hazard, must use recommended actions as fail-safes	N/A	-Timeout for idle clients on the server -Local backups of untransmitted data
Simulation Integration Software	4.1.6.0 (SR6)	Inaccurate results	-computational error	Our Machine-Learning Algorithm may only achieve a certain percent accuracy at best, even with infinite labelled input data	-Development of hardware, and comparison between Matlab simulation output and real-world testing	N/A	If error found to be large, alterations of the simulations would be in order to purify the data fed to our learner
	4.1.6.1 (SR1)	Data overflow	-Simulations produce outputs faster than can be processed by the ML algorithm, or considerably faster than the server polling speed	Simulation data is lost as the queue is at capacity	Have a flag for when overflow occurs	N/A	-Dynamic polling speeds, for increase in clients using the server in the future -Third party database monitoring software, for if simulation speeds greatly increase down the line

Design Component	Ref.#	Failure Modes	Causes of Failure	Effects of Failure	Detection	Controls	Recommended Action
Power Supply System	4.2.1.0 (RAR1)	No power supplied to subsequent systems	-Fault in power supply -Fault in power supply cables to downstream systems	-System crash -Loss of data	Current measurements taken at antenna-array system	N/A	-Disconnect power supply system from remaining systems -Conduct testing of power supply components to determine mode of failure
	4.2.1.1 (RAR1)	Voltage swell	Large change in loads seen by power system	System short circuited	Voltage measurements taken at antenna-array system	Protection devices (fuses) downstream from power supply	Conduct testing of power supply components to determine mode of failure
Phase-Shift System	4.2.2.0 (IR2)	Phase shifter component failure	Component break down	- System failure - Incorrect phase applied to system and unintended waves will be created - Device intended to be charged may not be charged	Measurement of induced radio waves	N/A	Testing of units prior to assembly
Antenna Array System	4.2.3.0 (IR2)	Antenna Array Component failure	-Component break down -Over current supplied	- System Failure - Waves will not be distributed - Devices will not be charged	Measurement of induced radio waves	Protection devices (fuses) to limit current to antenna within operating range	-Testing of units prior to assembly -Monitor power supplied to units

Design Component	Ref.#	Failure Modes	Causes of Failure	Effects of Failure	Detection	Controls	Recommended Action
System Enclosure	4.2.4.0 (RAR1)	Enclosure "leak"	Gap in wave reflective enclosure system	- Waves will leak through the gap and constructive interference will occur in unintended directions - The intended directions for the waves to be distributed will be minimized; device may not be charged due to length of waves not sent	- Measurement of induced radio waves -Visual inspection	N/A	-Inspection prior to use -Remove sensitive equipment from affected area
Hardware Display System	4.2.5.0 (RAR1)	False indication	Display stuck in "on" or "off" state	- Confusion amongst user - Incorrect usage may occur by user	Verification downstream to determine state of device	Wire indication in line with power supply to device	Disconnect the device from the power supply system until failure mode has been determined

5 Functional Architecture

As many constraints require feasible prototypes, the requirements are subject to change accordingly. Requirements indicated in bold writing are new additions.

5.1 Functional Requirements

5.1.1 Software System Functional Requirements

SR1. ML Model must optimize inputs faster than the existing process.
SR2. ML Model must be able to develop "new" simulations based on previous optimal models.
SR3. ML Model must be able to encrypt optimized data before exporting.

SR4. The software system must determine and output the optimized and correct solution.
SR5. ML Model must be able to process incoming simulation data from multiple source devices.
SR6. ML Model must be able to interpret data exported directly from Matlab simulations.

5.1.2 Hardware System Functional Requirements

HR1. The system must be able to wirelessly charge device through a reasonable amount of material in between.
HR2. The system must be able to charge a device within a reasonable amount of time.

5.2 Non-functional Requirements

5.2.1 Look and Feel Requirements

5.2.2 Appearance Requirements

APR1. The system will consist of a simple user interface.

5.2.3 Access Requirements

ACR1. Authorized users will have access to the system.

5.2.4 Integrity Requirements

IR1. The system must be able to store its current state locally in the event of a failure
IR2. The individual components of the physical system must be inspected and tested.

5.2.5 Style Requirements

N/A

5.2.6 Usability and Humanity Requirements

5.2.7 Ease of use Requirements

EUR1. The system shall be easy to use.
EUR2. The system shall be easy to install.

5.2.8 Learning Requirements

LR1. The system shall be understandable within an hour of use.

5.2.9 Understandability and Politeness Requirements

N/A

5.2.10 Performance Requirements

5.2.11 Speed and Latency Requirements

SLR1. The system must compute optimal configuration within 6 hours.

5.2.12 Safety Critical Requirements

N/A

5.2.13 Precision of Accuracy Requirements

PAR1. The system must have a relative accuracy of 5% compared to current Matlab simulation.

5.2.14 Reliability and Availability Requirements

RAR1. The system must be available at all times.

5.2.15 Robustness of Fault Tolerance Requirements

RFTR1. The system must be able to discard any corrupted data without adding it to the database.

5.2.16 Capacity Requirements

N/A

5.2.17 Operational and Environmental Requirements

5.2.18 Physical Environment

PER1. The hardware system must be able to withstand harsh weather.

5.2.19 Release Requirements

N/A

5.2.20 Maintainability and Support Requirements

5.2.21 Maintenance Requirements

N/A

5.2.22 Adaptability Requirements

ADAR1. The system must be functional on all operating systems.

5.2.23 Security Requirements

SECR1. The system must protect client data.

5.2.24 Access Requirements

N/A

5.2.25 Privacy Requirements

PRV1. The system must encrypt all exported data.

5.2.26 Legal Requirements

N/A

5.2.27 Health and Safety Requirements

N/A

6 Conclusion

Designing a software system is an intricate process, one that requires an inhuman-like insight into the very minute details of various sub-systems, independently nuanced and dependently coupled. For these reasons, they often contain far more mistakes and vulnerabilities than their proud creators suspect or even care to believe. This fact underlines the importance of acknowledging our faults and the likely faults of our current designs, which in turn allows us to not only protect against them but also further iterate on our pre-existing plans for development. By highlighting these hazards, we have been forced to further understand and define the constraints that are laid around our problem space, and how we might work to achieve all of them and the safest system possible. It is also important to notice the cyclical nature of data flow in our system, which can be seen in the various diagrams showing our system context in SRS Rev 0; thus our software system is especially vulnerable to the propagation of errors, and to the injection of poor data. In looking at the vulnerabilities we have been forced to understand the internal communications of all of our main components - the nature of their coupling as well as their own modular behaviour. The state of our problem definition, goals, and development plan are all better for it.

References

We will be referring to documentations provided by Mobilite-Power, however, as of now there are no references to mention.