Hazard Analysis
MTOBridge

Team 15, Alpha Software Solutions
Badawy, Adham
Yazdinia, Pedram
Jandric, David
Vakili, Farzad
Vezina, Victor
Chiu, Darren

Table 1: Revision History

Date	**Developer(s)**	**Change**
October 12 2022	Darren	Added System Boundaries & Components
October 19 2022	Adham	Added Adham/Victor/Farzads FMEA work into a latex table
October 19 2022	Pedram	Added Safety reqs and Roadmap
October 19 2022	Victor	Changes to FMEA table formatting

Contents

List of Tables

1 Introduction

This document is a hazard analysis of MTOBridge. A hazard is a potentially harmful event resulting from the conditions of the system and the environment.

2 Scope and Purpose of Hazard Analysis

This document describes the components of MTOBridge, the potential hazards associated with each, and any new functional requirements that can be derived from these hazards. This process is important to identify any potential issues with the system, and then design the system to eliminate the potential issues.

3 System Boundaries and Components

This hazard analysis addresses the system that consists of the following components:

1. UI Component, for providing a graphic display to the user and visualizing MATLAB results
2. Input Handler Component, for processing user inputs
3. MATLAB Interaction Component, for calling scripts and supplying specified arguments to them
4. MATLAB Engine Component, for performing bridge calculations
5. File Manager Component, for reading inputs from files and saving results in various formats

The system boundary includes these software components and any dependency files required for the application to operate. Although the MATLAB Engine Component is owned by the client and its exact contents verified independent of this project, this hazard analysis will address it due to being a crucial component of the system.

4 Critical Assumptions

We will not be making any critical assumptions about the system. As mentioned above, the MATLAB engine is a critical component of the system. We must address it with the hazard analysis and will not make any assumptions about its correctness and functionality.

5 Failure Mode and Effect Analysis

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI Displays truck config incorrectly	User confusion / Misleading interface	a. Incorrect processing of regular user Input b. Runtime error in truck display module c. Unexpected/boundary case user input	a. Thoroughly test the truck display module to avoid unexpected responses to input b. Ensure truck display module has proper error handling to avoid catastrophic failure if an error is encountered. c. Design modules with separation of concerns in mind to limit complexity and increase program robustness.	None	HA-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI does not update to match new truck config at all	User missing important information.	a. Failure to catch invalid user input. b. Runtime error in truck display module c. Unexpected/boundary case user input	a. Ensure truck display module has proper input bounds and safety nets to catch invalid user inputs, instead of just running with them. b. Same as HA-1b c. Same as HA-1c	None	HA-2
UI	UI Displays bridge config incorrectly.	User confusion / Misleading interface	a. Incorrect processing of regular user input b. Runtime error in truck display module c. Unexpected/boundary case user input	a. Same as HA-1a b. Same as HA-1b. c. Same as HA-1c.	None	HA-3

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI does not update to match new bridge config at all	User missing important information.	a. Failure to catch invalid user input. b. Runtime error in truck display module c. Unexpected/boundary case user input	a. Same as HA-2a. b. Same as HA-1b. c. Same as HA-1c.	None	HA-4
UI	UI attempts to display undesired calculation type	Display is worthless	a. Incorrect processing of regular user input. b. Misleading or incorrect display of user solver selection.	a. Thoroughly test solver configuration module to incorrect processing of user input b. Minimize complexity of input handler/solver selection display modules and the interaction between them to reduce chances of incorrect information passing and misleading or incorrect displays of input.	None	HA-5

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	Truck platoon trip display does not match bridge load display.	Display is impossible to parse	a. Incorrect calculation display logic. b. Unexpected bug or glitch is calculation display modules.	a. Thoroughly test calculation display module(s) to avoid unexpected behavior. b. Include checks to determine if the two displays align and catch/correct it if they don't instead of just displaying it anyways.	SR-1	HA-6
UI	Platoon trip and Bridge load synch check(s) provides false positives or negatives	Deny a fine display or let through an erroneous one.	a. Incorrect synch check logic. b. Unexpected bug or glitch in synch module.	a. Thoroughly test synch checks to avoid unexpected behavior. b. Simplify synch check logic as much as is possible while maintaining accuracy to limit chance of incorrect logic programming	SR-1	HA-7

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI incorrectly displays the concerned section	Display is worthless	a. Unexpected bug/glitch in concerned section display module. b. Misleading or incorrect display of user concerned section selection.	a. Thoroughly test concerned section display module to incorrect processing of user input b. Minimize complexity of input handler and concerned section display display modules and the interaction to reduce chances of incorrect information passing and misleading/incorrect displays of input.	None	HA-8
UI	UI incorrectly displays discretized bridge segments.	Display is worthless	a. Unexpected bug/glitch in concerned section display module. b. Misleading or incorrect display of user concerned section selection.	a. Same as HA-8a. b. Same as HA-8b.	None	HA-9

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI fails to display calculation results entirely.	Display is worthless.	a. Unexpected/boundary case user input. b. Runtime error in calculation display modules. c. Failure to catch invalid user input	a. Thoroughly test the calculation display modules to avoid unexpected responses to input b. Ensure calculation display modules have proper error handling to avoid catastrophic failure if an error is encountered, and design modules with separation of concerns in mind to limit error propagation and increase program robustness. c. Ensure calculation display modules have proper input bounds and safety nets to catch invalid user inputs, instead of just running with them.	None	HA-10

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI stops reacting to user inputs	User locked out from using UI, program is worthless	a. Runtime error in calculation display modules. b. Parallel computing issue such as deadlock that hangs the program.	a. Ensure all display modules have proper error handling to avoid catastrophic failure if an error is encountered, and design all modules with separation of concerns in mind to limit error propagation and increase program robustness. b. Implement proper thread safety measures to avoid deadlocks and other such issues.	SR-2	HA-11
UI	UI encounters parallel computing issue such as deadlock/race condition.	Incorrect results or unexpected program behavior	a. Multiple threads modifying the same values/waiting on each other.	a. Implement proper thready safety measures to avoid deadlocks and other such issues	SR-2	HA-12

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Matlab Interaction	Data received is incorrectly formatted.	Program cannot function	a. One-time error in cross-program communication caused by outside factors (OS, hardware, etc.) b. Bug or error in MATLAB engine	a. Try all calculations a second time when the first calculation fails b. The MATLAB engine will be tested thoroughly to try to reduce the amount of bugs it has. The program will always log in-depth error information and display an error message to the user telling them to contact the developers when there is an issue with the MATLAB component.	SR-3, SR-4	HB-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Matlab Interaction	Unable to call engine.	Program cannot function	a. One-time error in cross-program communication caused by outside factors (OS, hardware, etc.) b. Engine not installed / installed improperly	a. Same as HB-1a b. The program will always display a message telling the user that they must install the MATLAB engine with a reference to the installation section of the user manual when the MATLAB engine is not detected	SR-3, SR-4	HB-2
Matlab Engine	The engine crashes unexpectedly.	Program cannot function	a. One-time crash caused by outside factors (OS, hardware, etc.) b. Bug or error in MATLAB engine	a. Same as HB-1a b. Same as HB-1b	SR-3, SR-4	HC-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Matlab Engine	The engine calculations take more time than should be required (more than 1 second).	Program must wait for results	a. One-time error causing infinite looping caused by outside factors (OS, hardware, etc.) b. Bug or error in MATLAB engine	a. Same as HB-1a b. Same as HB-1b	SR-3, SR-4	HC-2
Matlab Engine	Data received from the engine is incorrect (as in physically impossible).	Program cannot function	a. One-time calculation error caused by outside factors (OS, hardware, etc.) b. Bug or error in MATLAB engine.	a. Same as HB-1a b. Same as HB-1b	SR-3, SR-4	HC-3
Input Handler	Handler passes inputs to other components that are too large or small.	Inaccurate analysis results	a. Accidental changes of input for example writing 100000 instead of 10000000	a. Validating numeric values are within an acceptable range	SR-5	HD-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Input Handler	Handler invariant to type changes.	System Crash	a. Accidental mix and match of inputs for example inputting structure material inside load section	a. Validating input type before passing it on.	SR-6	HD-2
Input Handler	Handler passes on incomplete set of inputs.	System Crash	a. Submitting before completing all input sections	a. Detecting if required inputs are missing from the model	None	HD-3
File Manager	File Manager loads corrupted configuration and saved files.	Inaccurate results or system crash	a. Process responsible for creating the file was interrupted. b. files edited manually by power users	a. Have metrics that indicates file creation was completed and if not it is communicated to the user when loading. b. Have metrics such as checksums to ensure the integrity of the files.	SR-7	HE-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
File Manager	File Manager partially saves file.	Data loss	a. Power outage b. System crash	a. Automatically save to a file whenever user changes the configuration or at reasonable time intervals b. Same as HE-2a	SR-8	HE-2

6 Safety and Security Requirements

Using the results of FMEA, we can derive the following safety and security requirements for our system in order to mitigate the identified hazards.

SR-1: The system must be able to synchronize the display of the truck platoon with the input bridge load through various logic checks and boundary tests. An alarm/error shall be produced if the display requirements across both modules are not met.

Rationale: While splitting the calculation display modules into two modules; platoon trip and bridge load display, can simplify each part, it can lead to mismatch in respective display.

Trace: HA-6, HA-7

SR-2: The system must have thread safety between the UI and other connected components.

Rationale: In order to avoid race conditions and deadlocks that could result in undesirable behavior, thread safety must be an integral part of the interaction between components.

Trace: HA-11, HA-12

SR-3: The system must produce a log of calls and functions with a detailed trace of function callbacks highlighting the code locations. The log will include timestamps along with software environment information such as input.

Rationale: The logs will be directly used in the debugging process which can be presented in different ways to the user. Such logs can prove useful in case of engine crashes, data loss and timeouts.

Trace: HB-1, HB-2, HC-1, HC-2, HC-3

SR-4: In case of an Matlab engine failure, the system must be able to use the recorded logs to provide a clear message to the user.

Rationale: Logs while carrying a lot of information can be too heavy to digest, we can help the user further with a bit of processing.

Trace: HB-1, HB-2, HC-1, HC-2, HC-3

SR-5: The system must validate numeric values are within an acceptable range before being passed on to other components.

Rationale: Accidental changes of input can cause massive shifts in the analysis and cause inaccurate results. (e.g. adding extra 0's)

Trace: HD-1

SR-6: The system must validate input type before being passed on to other components.

Rationale: This is to prevent accidental mix and match of inputs for example inputting structure material inside load section.

Trace: HD-2

SR-7: The system must include information that indicates the integrity of the file and whether the file creation is completed.

Rationale: This can help remove the risks associated with corrupt files during an import or export.

Trace: HE-1

SR-8: The system must automatically save to a file whenever user changes the configuration or at reasonable time intervals

Rationale: This is meant to mitigate risks associated with system crashes or power outages.

Trace: HE-2

7 Roadmap

The new requirements derived in the previous section can be prioritized based on time to develop, probability of happening and the severity with severity given the most weight. As such we believe requirements that can disrupt the user-flow must be integrated regardless of other factors. These requirements will be developed as part of the primary development phase while the rest are allocated as stretch requirements for later implementation.

7.1 Phase 1

Critical requirements include SR-1, SR-2, SR-3, SR-4, SR-5 and SR-6. The first requirement is meant to mitigate the risk of mismatch and miscalculations across modules which even though very unlikely can cause faulty results with a low chance of detection. SR-3 and SR-4 are meant to implement the logging system which will directly support debugging especially due to engine and pipeline errors. SR-5, SR-6 are meant to mitigate the risks associated with input and sophisticated time handling. While such requirements can quickly grow and take time to develop, they have high probability and severity due to continuous work models.

7.2 Phase 2

Phase two requirements include SR-7 and SR-8. SR-7 is another security requirement meant to validate imported files and their integrity through different methods. While a fairly severe requirement, we believe that it has a low probability and high time to develop. Similarly, SR-8 is another requirement which can protect the user against unforeseen environment changes such as BSOD's or power outages. Such requirements are of low priority and are scheduled as part of the stretch requirements for later development.