Reflection and Traceability Report on CVT Simulator

Team #17, Baja Dynamics
Grace McKenna
Travis Wing
Cameron Dunn
Kai Arseneau

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

The following table corresponds to SRS changes in response to feedback.

Feedback Item	**Source of Feedback**	**Change Made in Response**	**Commit Reference**
Functional Requirements do not seem to have measurable/fit criteria associated with them Issue #38.	Peer review	No change made, Functional requirements were deemed measurable.	N/A
Acknowledging possible floating point errors and specifying a tollerance range for error would be beneficial Issue #39.	Peer review	Added floating point accuracy consideration to NFR1.	b737c9d
Adding why specifc section were NA Issue #40.	Peer review	No change made, deemed not relevant.	N/A
Traceability matrix not clear Issue #41 and Issue #64.	Peer review and TA feedback	Traceability Matrices fixed to be more clear and complete.	f8dc185 d5080cc b7db68a
Reusability metric not practical Issue #42	Peer feedback	No change made as was deemed practical as previous CVT information was available to us.	N/A
Remove unused SRS folders Issue #55.	Supervisor feedback	Removed used SRS folders from the repository.	b6e45c2

Add References Issue #61.	TA feedback	Added references to SRS where needed.	1632c65
Discuss standards Issue #62.	TA feedback	Added standards section.	91148c8
Missing numerical rational as why certain numbers were chosen in requirements Issue #65.	TA feedback	Added rationale for the numbers selected.	b5b0137
Missing phase in plane of requirements Issue #66.	TA feedback	Added phase in plan for requirements.	43c023e
Add description of sheaves Issue #158.	Supervisor feedback	Added sheave description to improve user understanding.	e3953cc

The following table corresponds to Hazard Analysis changes in response to feedback.

Feedback Item	**Source of Feedback**	**Change Made in Response**	**Commit Reference**
The failure mode of ”insufficient frictional forces” has no requirement associated with it according to the FMEA table Issue #48.	Peer review	Addressed why one failure mode did not have requirement due to assumptions made.	430fb6d
Assumptions listed outside Assumptions section Issue #49.	Peer review	No change made as discussing some assumptions in the scope section was relevant.	N/A
Security requirements missing fit criteria Issue #50.	Peer review	No change made as security requirements were deemed practical and attainable.	N/A
Table is missing references for one failure mode Issue #52.	Peer review	No change made as this was addressed with Issue #48.	See 430fb6d
Table is missing failure modes and associated safety requirements for User Interface Component Issue #53.	Peer review	Added a section to address why there were no UI security requirements.	bce0b02

Should consider cancellation errors under numerical stability Issue #54.	Peer review	No change as not planned for this revision of our product.	N/A
Add traceability to security requirements in FMEA Issue #212.	TA feedback	Added traceability to FMEA table for Security Requirements.	430fb6d

1.2 Design and Design Documentation

The following table corresponds to the changes to the Module Guide(MG) and MIS documents.

Feedback Item	**Source of Feedback**	**Change Made in Response**	**Commit Reference**
MG - Section 6 not valueable to the reader Issue #100.	Peer review	No change made as section 6 was deemed valueable.	N/A
MG - Link back to SRS to reduce duplicate information and add traceability Issue #101.	Peer review	Link to SRS was provided.	c947278
MG - Anticipated changes could be more clear Issue #103.	Peer review	No change made as the anticipated change's section was deemed strait forward.	N/A
MG - Should the GUI module not make use of initialize module Issue #104.	Peer review	No change made as not applicable.	N/A
MG - Missing Add timeline and detail about priority of modules being implemented Issue #222.	TA feedback	Added details of priority of modules and breakdown of timeline.	a3867bb
MIS - Some exceptions were left blank. Issue #102.	Peer review	Blank exceptions were removed.	5c48147

The following table corresponds to the changes made in the design of the CVT Simulator's design.

Feedback Item	Source of Feedback	Change Made in Response	Commit Reference
Add units to the user inputs to improve user understanding. Issue #153.	Supervisor feedback	Units were added beside inputs.	54bafd2
Add ability to upload dxf files for ramp geometry. Issue #153.	Supervisor feedback	No changes as did not have time to implement.	N/A
Check for users entering negative inputs Issue #154.	Supervisor feedback	Users are not allowed to enter negative numbers.	54bafd2
Provide default values in the input fields Issue #156.	Supervisor feedback	Users are given default values allowing them to quickly tune CVT.	54bafd2
Provide user with loading information to improve user feedback Issue #204	Supervisor feedback	Added a loading bar to provide the user loading progress.	f9c304

1.3 VnV Plan and Report

The following table corresponds to changes regarding the VnV Plan.

Feedback Item	Source of Feedback	Change Made in Response	Commit Reference
Confusion with Validation report extra Issue #76.	Peer review	No change made, as one of our extras is a Validation report.	N/A
Repetitive information that can be listed in your development plan Issue #77.	Peer review	No change as was deemed not repetitive.	N/A
Opportunity for more automated tests Issue #78.	Peer review	No change as not planed for this revision of the product.	N/A
More granular items in the SRS verification plan checklist Issue #85.	Peer review	No change as not planed for this revision of the product.	N/A
The checklist format was not consistent. Issue #86.	Peer review	Checklist format was fixed so that the format of each checklist was consistent.	13e3596

Traceability labels could be shorter Issue #88.	Peer review	No change made, as test labels were deemed appropriate.	N/A
Add traceability to tests in the tables Issue #214.	TA feedback	Traceability added to the table.	c50551f
Link usability/understandability in the tests and add checklist where applicable Issue #215.	TA feedback	Added link to usability/understandability survey and added verifiability checklist.	ff2bb8c and b26ab16.

The following table corresponds to changes regarding the VnV Report.

Feedback Item	**Source of Feedback**	**Change Made in Response**	**Commit Reference**
Consolidating all your symbols, abbreviations and acronyms into one document Issue #201.	Peer review	No change made, as symbols are referenced in SRS already.	N/A
Adding expected result clarity explaining the results of your testing, Issue #202.	Peer review	No change made, as not enough time to implement.	N/A

2 Challenge Level and Extras

2.1 Challenge Level

The challenge level of this project is General.

2.2 Extras

2.2.1 Requirements Validation Report

Since our capstone is a simulation of a real-world system there is extra validation that has be done in addition to the VnV Report that was completed earlier this year. In the Requirements Validation Report we will analyze the simulation results vs actual data that was collected from the car using various data analysis techniques.

2.2.2 Usability Report

Users were given the application along with a usability survey that was created with the intention of getting the users to interact with all the core features. We will review the feedback collected from the users and organize the result into a Usability Report to analyze further.

3 Design Iteration (LO11 (PrototypeIterate))

From the initial prototype demo in November and the rev 0 demo in January our project has changed a lot. To more accurately capture the iterations that occurred in our project we will have to split this up into front end and back end.

3.1 Front End

The style and flow of the front end of our capstone remained the same throughout development. We decided to stick to the 3-page system and have users progress through them linearly. The areas that had lots of developmental iterations was the inputs page and results page, these both changed many times due to feedback received from stakeholders and team design decisions. These two pages each had there own specific question that impacted our design choices, on the input page it was "what should the user be able to do" and the results page was "what should the user be able to see".

The input page is where all of our input parameters to the system are stored and the user has the freedom to change any of these parameters to whatever they want. As well the user can upload a set of pre made parameters if they have them saved from an earlier run. Some key design iterations that occurred on this page included:

- **Figma Redesign**: The team decided a new design was needed after some feedback was received from test users about it being messy and unorganized.
- **Drop down options for ramps**: After discussing with our stakeholders on the Baja team about how best to handle the ramp inputs for the project it was concluded that instead of a completely free ramp designing option we will instead give them a list of ramps to chose from. The reasoning behind this decision was that a user will never able to input a ramp from scratch on the spot as the ramp design require a mathematical process to be created.
- **Saving/Uploading parameters**: During our rev 0 demo Dr Smith had some feedback about the user being able to save and upload parameters. We decided to implement that as apart of rev 1 as it allows users to essentially "keep there place" and pick up where they left off from last time.

The results page is where the user gets to view the simulation results and also where they can export parameters, graphs and data. This page is based around a playback component where the user can stop and start the simulation. Some key design iterations that occurred on this page included:

- **Adding the seekbar**: After presenting our rev 0 version of the project the team identified a missing aspect of our "playback" system was having a seek bar. It was put on the list of important features to add for the rev 1 demo as it ties together the whole idea of having playback control over the simulation, without it seemed we were just starting and stopping something with no identifiers of where we were in the simulation.
- **Displaying graphs** : After discussing the rev 0 demonstration of the project with our stakeholders on the Baja team and discussing what the most important information is from a run on the simulation. It was determined that instead of just having the option to export the data and analyze it elsewhere we should let them view important graphs while still on the application. The idea is to remove the wasted time of taking the data and creating the same graphs manually elsewhere. Therefore, for rev 1 we added the option to view graphs on the results page and it was an important part of our live demonstration.
- **Downloading parameters**: As mentioned earlier in the inputs page section Dr Smith provided us with the feedback of being able to download parameters. Originally, we had the download parameters option on the inputs page but after some usability testing it was determined that didn't make sense to the user. Therefore, we moved it to the results page because the feedback was that you wouldn't know if you wanted to download that set of parameters until you saw the results of the simulation.
- **Changing CVT material:** : In our rev 0 demo the CVT object we have on the results page was the default material in unity and it did not look good. After doing some UI review the team decided that we needed to dig more into unity to figure out how to change this material because originally we were not able to change it. It was then changed to the color it is now which fits much better with the rest of the application.

3.2 Back End

The backend saw several important design iterations driven by the need for more accurate results that better matched real data and improved overall code clarity. Some key design iterations that occurred on the backend include:

- **Updated Engine Dynamics**: Previously, the engine's angular velocity was simulated as a separate system with its own inertia, causing it to accelerate on its own due to missing forces. Recognizing discrepancies with real-world behavior and the benefits of the no-slip assumption, we unified the engine and car into one system—effectively connected like a timing belt rather than a rubber one—to improve accuracy.

- **Improved Ramp Representation**: Initially, the math for the circular portion of the ramp was faulty, leading to incorrect simulation outputs when compared with real data. We rewrote the equations and their derivatives to yield a far more accurate ramp representation which resulted in our results aligning much more closely with real data.
- **Modular Graphing System**: We separated out the graphing functionality to make it more modular. This change was driven by the decision to add a graphing feature on the front end for real-time data visualization. Now, the graphing module simply reads the output file and leverages established math modules, which simplifies the frontend code and enhances maintainability.
- **Added Progress Printing**: Given that the simulation frequently takes a while to run, progress printing was implemented to provide real-time feedback. This not only aids developers during debugging but also gives the frontend user clear indicators of how far along the simulation is, improving the overall user experience.
- **Decoupled FE output conversions**: Conversion calculations for output data have been moved from the frontend to the backend. This decoupling was done to improve the readability and maintainability of the frontend code, ensuring that the frontend focuses solely on user interaction while the backend handles all computational tasks.
- **Revisited and verified Math**: Various mathematical computations were re-evaluated because earlier inaccuracies resulted in outcomes that did not align with real data. This thorough review helped us identify and correct errors, leading to more reliable simulation results.
- **Incorperated frictional forces**: Frictional forces, once thought to have minimal impact, were added to introduce necessary dampening. Without these forces, the system exhibited oscillations that deviated from observed physical behavior, so their inclusion ensures that the simulation more accurately reflects real-world dynamics.

4 Design Decisions (LO12)

4.1 Limitations

The key limitation in this project was our understanding of the CVT system and how it functions. Even now with our simulation successfully producing results that accurately depict how the system should be functioning our key limitation is still our lack of knowledge of the CVT system. Essentially the

CVT system is a black box and there is so much math and physics that goes into it that fully understanding how each aspect works and how it impacts the overall performance is really challenging. Over the course of this capstone we have learned so much about the system and the "whys" of certain tuning parameters impact performance in a certain way. However, there is still so much more to learn about the system and that is why it is our biggest limitation.
Basically, as our project has progressed, we gained more and more knowledge about how the CVT functions and as a result the simulation became more accurate. We have been reducing the limitation over time, but it remains the biggest obstacle to achieving a "perfect" simulation. So, when talking about limitations I would label this as the most important/significant one when it comes to our capstone.

4.2 Assumptions

We made a lot of simplifying assumptions when doing this project to ensure we could capture the mathematics behind the CVT system. The two assumptions that affected our final design the most are:

- **No Belt Slippage**: As a result of assuming the belt does not slip the system overpredicts torque transfer and grip reliability.
- **No friction on noncritical components**: This resulted inaccuracies through various resistive/ dampening forces when shifting.

The other assumptions that were made very mode simple in concept, these two are the ones that have the most impacts on the actual results of our CVT simulation results as they fundamentally changed how we designed the backend architecture. However attempting to incorporate these assumptions from the beginning would have made the project significantly more complicated and we don't believe we would have achieved the same success if we had tried that.

5 Economic Considerations (LO23)

There is a market for our CVT Simulator, for instance the market would be the hundreds of Baja teams all over North America. Each team uses a CVT in their car and I would guess that they would be interested in having a way to simulate it for the same reasons why our team was interested in it. However, I don't think this project would succeed as a product that we would market and sell to other teams and the reason for that is the math. The math of the simulation is specifically based around how McMaster Bajas CVT is designed, and it was built off all the knowledge we have of how our CVT functions. If you were to give it to another team, they would have to change a bunch of math to make it align with there specific system before it started being of use to them. Therefore, I can see our capstone being more of an open-source project where other teams would have access to the codebase to be able to tailor the project to there needs. To go about attracting users to this project the approach I would first send it to the Baja SAE discord server which has thousands of members from across North America and let people know about our project. It would for sure get some people interested in looking at it. However, before that we would first have to rewrite the backend to abstract out the specific parts where the math is solely based on our CVT to make it easier for others to change it to there's. That would be a complicated undertaking but would have to been done if this was an actual goal for the project. Therefore, this capstone has the potential future to be an open source project where other Baja teams can come use the software to improve there teams CVT tuning but there would be a large barrier of entry because the domain and technical knowledge needed to interact with the codebase.

6 Reflection on Project Management (LO24)

6.1 How Does Your Project Management Compare to Your Development Plan

The team meeting plan in the development plan was an in person meeting every Monday (during our tutorial time) and then an online meeting every Friday. As the semester progressed, and we went got busy we stopped following this plan and began meeting on a need be basis.
Our communication plan was followed, and it became one of our strong points as a team, we communicated frequently and about everything. Our discord server has all our PR's, resources, notes and Todo's.
The team member roles were followed closely:

- Grace is the primary note taker at meetings and project manager
- Travis is the issue manager and was the front-end lead
- Kai is the primary reviewer for code and was the backend lead.
- Cam is the product owner and oversees meetings.

For different stretches of the project when we got busy the roles changed a bit but overall, this was the structure we followed.

The workflow plan was followed. For branches main was only updated when deliverables were due, develop was the primary branch we were interacting with and bigger features that took a while were kept on separate branches to be worked on. We added all of the various issues labels and used them throughout the project for better organization. We utilized CI/CD to run our test suites and to lint/check our backend code on each commit.

The technology was as expected:

- Backend: python
- Front-end: Unity/Csharp
- Github
- VSCode

6.2 What Went Well?

Processes:

- Git hub standards (PR's and reviewing) were all really well done
- Communication was the higlight of our team, utilzed our discord server to its fullest extent.
- Issue creation before development cycles, we issues out all the tickets needed for each cycle making it very easy to pick up work

Technologies:

- Python was a great choice for backend
- Testing, linting, formatting tools all are really easy to use in python

6.3 What Went Wrong?

Processes:

- Issue tracking during development was difficult, frequently had to go back and back update and link issues to PR's.
- Making issues when new issues/bugs came up, we defaulted to just fixing them as fast as possible.

Technologies:

- Using Unity for the front end was probably not a good choice in hindsight, the learning curve to use it effectively was much higher than we expected. Probably would have been better to make it a web application so that we would have been more familiar with the languages.
- Testing was difficult in unity because of the nature of our modules and how Unity works.

6.4 What Would you Do Differently Next Time?

- Do a lot more research into technologies before starting the project so we are sure we are using the right tech for the project.
- Having stricter GitHub regulations, reviewing, issue tracking, branch naming, etc.
- - Not having lapses in development due to getting busy with other courses, because we found that it really derails the project development.

7 Reflection on Capstone

7.1 Which Courses Were Relevant

Courses that were relevant to this capstone project include:

- SFWRENG 4X03 Scientific computation
- SFWRENG 3A04 Software Development
- ENG 3PX3, 2PX3, 1P13
- Calc 1, Calc 2
- Physics 1D03 and 1E03
- SFWRENG 3XB3

7.2 Knowledge/Skills Outside of Courses

Knowledge that had to be acquired outside of courses include:

- Unity and Csharp, we had to learn how to use Unity and Csharp to build the front end of our project.
- GitHub, we had to learn how to use GitHub effectively and how to use the issue tracking system as well as properly using CI/CD
- Python best practises, we were familiar with python but we had to learn how to use it in a more professional way.
- Physics and math of the CVT, we had to learn how a CVT works and how to simulate it.