Software Requirements Specification for CVT Simulator: subtitle describing software

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

October 8, 2024

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	**unit**	**SI**
m	length	metre
kg	mass	kilogram
s	time	second
°C	temperature	centigrade
J	energy	joule
W	power	watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)
N	force	newtons ($\mathrm{N} = \mathrm{kg\,m\,s^{-2}}$)
Nm	torque	newton metre ($\mathrm{Nm} = \mathrm{kg\,m^2\,s^{-2}}$)
rad	angle	radian

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	**unit**	**description**
A_C	$\mathrm{m^2}$	coil surface area
A_{in}	$\mathrm{m^2}$	surface area over which heat is transferred in

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model
RPM	Revolutions Per Minute
CVT	Continuous Variable Transmission
SEC	Secondary
PRIM	Primary

1.4 Mathematical Notation

N/A

[This SRS template is based on Smith and Lai (2005); Smith et al. (2007); Smith and Koothoor (2016). It will get you started. You should not modify the section headings, without first discussing the change with the course instructor. Modification means you are not following the template, which loses some of the advantage of a template, especially standardization. Although the bits shown below do not include type information, you may need to add this information for your problem. If you are unsure, please can ask the instructor. —TPLT]

[Feel free to change the appearance of the report by modifying the LaTeX commands. —TPLT]

[This template document assumes that a single program is being documented. If you are documenting a family of models, you should start with a commonality analysis. A separate template is provided for this. For program families you should look at Smith (2006); Smith et al. (2017). Single family member programs are often programs based on a single physical model. General purpose tools are usually documented as a family. Families of physical models also come up. —TPLT]

[The SRS is not generally written, or read, sequentially. The SRS is a reference document. It is generally read in an ad hoc order, as the need arises. For writing an SRS, and for reading one for the first time, the suggested order of sections is:

- Goal Statement
- Instance Models
- Requirements
- Introduction
- Specific System Description

—TPLT]

[Guiding principles for the SRS document:

- Do not repeat the same information at the same abstraction level. If information is repeated, the repetition should be at a different abstraction level. For instance, there will be overlap between the scope section and the assumptions, but the scope section will not go into as much detail as the assumptions section.

—TPLT]

[The template description comments should be disabled before submitting this document for grading. —TPLT]

[You can borrow any wording from the text given in the template. It is part of the template, and not considered an instance of academic integrity. Of course, you need to cite the source of the template. —TPLT]

[When the documentation is done, it should be possible to trace back to the source of every piece of information. Some information will come from external sources, like terminology. Other information will be derived, like General Definitions. —TPLT]

[An SRS document should have the following qualities: unambiguous, consistent, complete, validatable, abstract and traceable. —TPLT]

[The overall goal of the SRS is that someone that meets the Characteristics of the Intended Reader (Section 2.3) can learn, understand and verify the captured domain knowledge. They should not have to trust the authors of the SRS on any statements. They should be able to independently verify/derive every statement made. —TPLT]

2 Introduction

The McMaster Baja engineering team is facing challenges in tuning their Continuous Variable Transmission (CVT). The current process of tuning the CVT is extensive, time consuming and requires testing of multiple physical components leading to possible inconsistencies due to weather and wear. These inconsistences then complicate the tuning process of the CVT's torque transfer directly affecting the performance of the Baja vehicles. To address these issues, our team aims to develop a CVT simulation tool that uses mathematical models, a rendering engine and a user-friendly interface. Our solution aims to streamline the optimization process allowing Baja members to simulate real-world factors and virtually test various tuning parameters. Our simulation tool will be validated against data collected by the McMaster Baja Data Acquisition team to ensure reliability and accuracy.

This introduction will outline the key objectives of this document to provide a structured guide for the development of the CVT tool and outline of the system's requirements. Additionally, the scope will explore technical and functional assumptions and will intend to streamline the complex tuning process of a CVT. A detailed overview of this document will provide guidance to the reader regarding the structure and flow of the document.

2.1 Purpose of Document

The purpose of this Software Requirements Specification (SRS) is to provide an outline of the system requirements given for the development of a CVT simulation tool. This ensures that all stakeholders including members of the Baja team, Dr. Smith and the developers understand the projects objectives and constraints. This document will serve as a communication tool to align expectations, provide guidance during the design of this tool and aid in the development and testing phases of this project. This SRS will be referenced throughout the development of the system to ensure the tool is within the scope and meets the outlined requirements.

2.2 Scope of Requirements

[Modelling the real world requires simplification. The full complexity of the actual physics, chemistry, biology is too much for existing models, and for existing computational solution techniques. Rather than say what is in the scope, it is usually easier to say what is not. You can think of it as the scope is initially everything, and then it is constrained to create the actual scope. For instance, the problem can be restricted to 2 dimensions, or it can ignore the effect of temperature (or pressure) on the material properties, etc. —TPLT]

[The scope section is related to the assumptions section (Section 4.2.4). However, the scope and the assumptions are not at the same level of abstraction. The scope is at a high level. The focus is on the "big picture" assumptions. The assumptions section lists, and describes, all of the assumptions. —TPLT]

[The scope section is relevant for later determining typical values of inputs. The scope should make it clear what inputs are reasonable to expect. This is a distinction between scope and context (context is a later section). Scope affects the inputs while context affects how the software will be used. —TPLT]

This model will simulate as a closed system and will not consider environmental factors such as temperature, humidity, and wind. It also will not consider any unintended damage, forces or effects on the components of the CVT system. The system will not account for any relative displacement between the components of the CVT system due to driver input or external forces.

2.3 Characteristics of Intended Reader

Readers or reviewers of this SRS document should have a solid understanding of the principles of newtoninan physics and basic calculus. Knowledge of the components in a Continuous Variable Transmission (CVT) and a basic understanding of how these parts work together will also be beneficial. Proficiency in these areas will enough to comprehend the material in this document.

2.4 Organization of Document

This document is structured as follows:

- Section 3 discusses the general context and description of the system
- Section 4 details the specific system description, goals, and definitions
- Section 5 covers the system requirements
- Section 6 outlines the likely changes for the system
- Section 7 discusses the unlikely changes for the system
- Section 8 covers the traceability of the requirements

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Figure 1 shows the system context. A circle represents an external entity outside the software. A rectangle represents the software system itself (CVT Simulator). Arrows are used to show the data flow between the system and its environment.

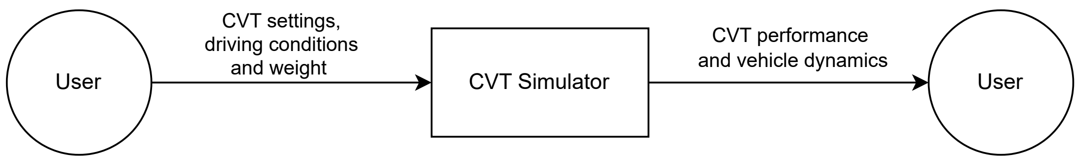


Figure 1: System Context

- User Responsibilities:
  - Provide the input data related to the tunable components of the CVT system.
  -
- CVT Simulator Responsibilities:
  - Accept user input data to simulate the CVT system, validating against set constraints.
  - Calculate the output data based on the input data and the CVT system model.
  - Display the results of the simulation.

3.2 User Characteristics

The expected user of this application is someone who is very familiar with drivetrains and CVTs but not necessarily familiar with software. The software expects the user to know all of the components of a CVT and how they can be modified or adjusted. The user should also have enough of an understanding of undergraduate physics and calculus to be able to make sense of the outputs of the software. On the software side, the user should be able to navigate a GUI and know how to input data into the software.

3.3 System Constraints

The system should be able to run on any machine that supports Python 3.9 or later and Unity projects.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

The CVT simulator is intended to address the challenges faced by the McMaster Baja racing team when they tune there CVT. The current tuning process is time-consuming and requires physical testing of multiple components and configurations. This leads to inconsistencies in the tuning process due to weather and wear. The CVT simulator will allow the Baja team to virtually test different tuning parameters and simulate real-world factors that affect the CVT's performance.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- RPM - Revolutions Per Minute, a measure of rotational speed.
- CVT - Continuous Variable Transmission
- Torque - The rotational force produced by the engine or seen as load by the wheels.
- Primary RPM - The RPM of the primary CVT.
- Secondary RPM - The RPM of the secondary CVT.
- Belt - The rubber V-belt that connects the primary and secondary CVT.
- Primary ramp - The ramp that the fly weights in the primary CVT push against. Highlighted green in Figure 2.
- Flyweight - The weights that experience centrifugal force and push against the primary ramp. Highlighted yellow in Figure 2.
- Helix - The helical ramp that the secondary CVT pushes against. Highlighted bronze in Figure 4.

- CVT System - This is the entire system that includes the primary and secondary CVT and the belt. The primary CVT is connected to the engine and the secondary CVT is connected to the wheels.

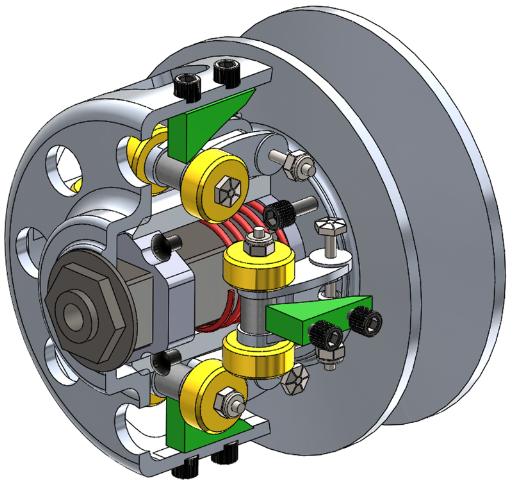


Figure 2: Primary CVT

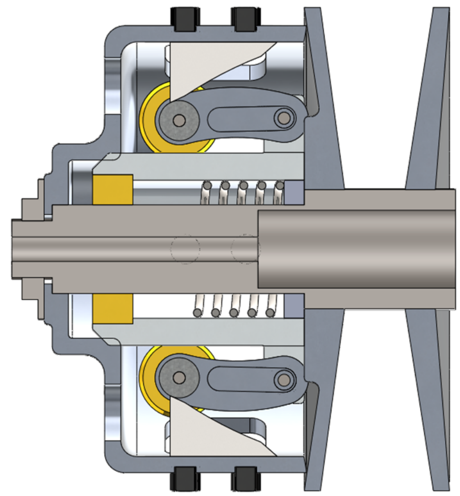


Figure 3: Primary CVT Side View

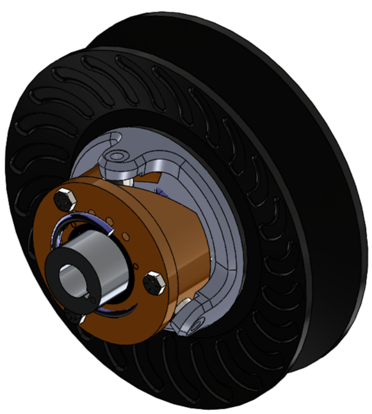


Figure 4: Secondary CVT

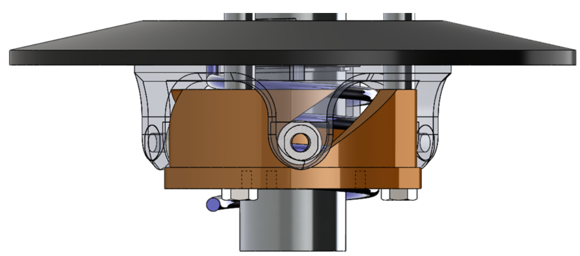


Figure 5: Secondary CVT Side View

Note: While the McMaster Baja Team's CVT differs slightly from these figures, its mechanics remain widely equivalent.

4.1.2 Physical System Description

The physical system of CVT Simulator, as shown in Figures 2-5, includes the following elements:

PS1: Flyweights

- PS1a: mass (m_{fly}) in kg.
- PS1b: initial radius (r_{fly}) in m.

PS2: Primary spring

- PS2a: spring constant (k_{prim}) in N/m.
- PS2b: pre-compression distance (d_{prim}) in m.

PS3: Primary ramp geometry function ($f_{\text{prim\_angle}}$)

- PS3a: shift distance (d_{shift}) in m.
- PS3b: output angle (θ_{prim}) in radians.

PS4: Primary ramp height function ($f_{\text{prim\_height}}$)

- PS4a: shift distance (d_{shift}) in m.
- PS4b: output height (h_{prim}) in m.

PS5: Secondary spring

- PS5a: compression spring constant ($k_{\text{sec\_comp}}$) in N/m.
- PS5b: torsional spring rate ($k_{\text{sec\_tor}}$) in Nm/rad.
- PS5c: pre-compression distance (d_{sec}) in m.
- P55d: pre-torsional rotation (θ_{sec}) in radians.

PS6: Secondary ramp geometry function ($f_{\text{sec\_ramp}}$)

- PS6a: shift distance (d_{shift}) in m.
- PS6b: output angle (θ_{ramp}) in radians.

PS7: Secondary ramp rotation function ($f_{\text{sec\_shift}}$)

- PS7a: shift distance (d_{shift}) in m.
- PS7b: output angle (θ_{shift}) in radians.

PS8: Weight

PS8a: driver weight (m_{d}) in kg.

PS8b: vehicle weight (m_{v}) in kg.

PS9: Engine torque function (f_{engine})

PS9a: input angular velocity (ω_{engine}) in rad/s.

PS9b: output torque (T_{engine}) in Nm.

PS10: Belt

PS10a: angle with sheave (θ_{belt}). I think this does not need to exist, it matches the sheave angle?

PS10b: friction coefficient (μ_{belt}), unitless.

PS11: Total reduction

PS11a: gearbox reduction ratio (r_{gearbox}), unitless.

PS11b: wheel radius (r_{wheel}) in m.

PS12: Coefficient of air resistance (μ_{air}), unitless.

PS13: Center to Center distance (d_{center}) in m.

PS14: Angle on Incline (θ_{inc}) in radians.

PS15: Secondary Ramp Radius ($r_{\text{sec\_ramp}}$) in m.

PS16: Traction (μ_{traction}), unitless.

We might also want belt length, belt width, belt weight?, other properties

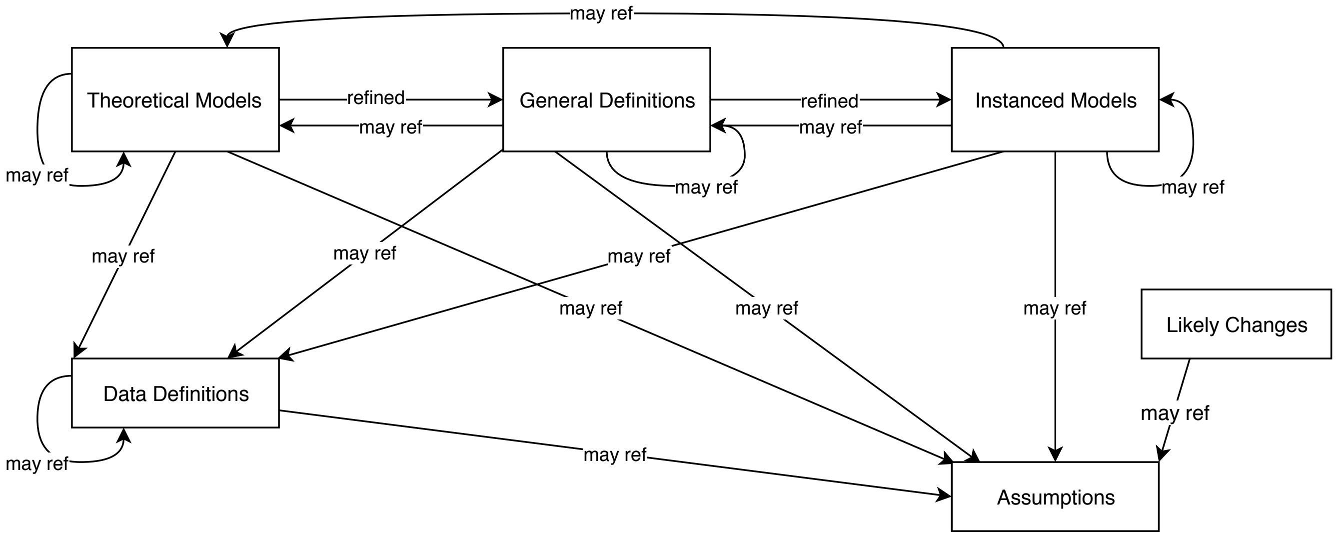
4.1.3 Goal Statements

GS1: Simulates the kinematics of the vehicle

GS2: Simulates the output of the engine

GS3: Simulates the CVT system over time

4.2 Solution Characteristics Specification



The instance models that govern CVT Simulator are presented in Subsection 4.2.9. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Types

N/A

4.2.2 Scope Decisions

N/A

4.2.3 Modelling Decisions

N/A

4.2.4 Assumptions

A1: **Temperature Variations** Ignore the effect of temperature on the material properties, excluding the belt.

A2: **Negligible Gravitational Influence**: The orientation of the CVT system and gravitational effects are considered negligible due to the high rotational speeds involved in the system's operation.

A3: **Elasticity of Belts**: Elasticity of belts are assumed negligible through its length, and the components are considered ideally rigid under normal operating conditions. The belt may still rotate about the pulleys.

A4: **Vibrational Effects**: Vibrational effects from the vehicle or external environment are assumed to be minimal and are not factored into the transmission's operation in this model.

A5: **Wear and Tear**: Component wear over time, including belt/chain wear and pulley degradation, is assumed negligible in the short-term operational model.

A6: **Frictional Losses in Non-critical Components**: Frictional losses in non-critical components (e.g., bearings, shafts) are assumed negligible, focusing only on friction relevant to the CVT system's core mechanics.

A7: **Engine Behavior Assumption:** The engine is modeled to operate at full throttle with consistent power curves and peak performance, maintaining a uniform response throughout the analysis. Load feedback will be integrated to adjust RPM and power output as necessary.

A8: **Axis Alignment:** The CVT's axis are aligned with one another.

A9: **Belt Slippage:** The CVT belt slippage is negligible.

4.2.5 Theoretical Models

This section focuses on the general equations and laws that CVT Simulator is based on.

General Theories

This portion will specify the existing theories that are used in the development of the CVT simulation tool.

RefName: **TM:HL**

Label: Hookes Law

Equation: $F = k\Delta x$

Description: F is the force (N)
k is the spring constant (N/m)
Δx is the displacement (m)

Notes: In the context of this project Hookes law is used in both a torsional and linear way.

Source: https://www.britannica.com/science/Hookes-law

Ref. By:

Preconditions for TM:HL: None

Derivation for TM:HL: Not Applicable

RefName: **TM:CF**

Label: Centrifugal Force

Equation: $F_c = m\omega^2 r$

Description: F is the force (N)
m is the mass (kg)
ω is the angular velocity (rad/s)
r is the radius (m)

Notes: None.

Source: https://www.omnicalculator.com/physics/centrifugal-force

Ref. By:

Preconditions for TM:CF: None

Derivation for TM:CF: Not Applicable

RefName: **TM:AR**

Label: Air Resistance

Equation: $F_d = \frac{1}{2}\rho v^2 C_d A$

Description: F is the force (N)
ρ is the density of the fluid (kg/m^3)
v is the velocity of the object (m/s)
C_d is the drag coefficient
A is the cross-sectional area of the object (m^2)

Notes: None.

Source: https://softschools.com/formulas/physics/air_resistance_formula/85/

Ref. By:

Preconditions for TM:AR: None

Derivation for TM:AR: Not Applicable

RefName: **TM:ET**

Label: Engine Torque

Equation: $\tau = \frac{P}{\omega}$

Description: τ is the torque (N-m)
P is the power (W)
ω is the angular velocity (rad/s)

Notes: None.

Source: https://powertestdyno.com/how-to-calculate-horsepower/

Ref. By:

Preconditions for TM:ET: None

Derivation for TM:ET: Not Applicable

RefName: **TM:GR**

Label: Gearing

Equation: $\tau_{\text{output}} = \tau_{\text{input}} \times \text{gear reduction}$

Description: τ_{output} is the output torque (N-m)
τ_{input} is the input torque (N-m)

Notes: None.

Source: https://science.howstuffworks.com/transport/engines-equipment/gear-ratio.htm

Ref. By:

Preconditions for TM:GR: None

Derivation for TM:GR: Not Applicable

RefName: **TM:FR**

Label: Friction Formula

Equation: $F_f = \mu F_n$

Description: $\mu_{\text{effective}}$ is the effective coefficient of friction
μ_{nominal} is the nominal coefficient of friction
ϕ is the angle of wrap of the belt (radians)

Notes: None.

Source: https://en.wikipedia.org/wiki/Friction

Ref. By:

Preconditions for TM:FR: None

Derivation for TM:FR: Not Applicable

RefName: **TM:DE**

Label: Density

Equation: $\rho = \frac{m}{V}$

Description: ρ is the density (kg/m^3)
m is the mass (kg)
V is the volume (m^3)

Notes: None.

Source: https://www.britannica.com/science/density-physics

Ref. By:

Preconditions for TM:DE: None

Derivation for TM:DE: Not Applicable

RefName: **TM:CE**

Label: Capstan Equation

Equation: $T_1 = T_2 e^{\mu\theta}$

Description: T_1 is the tension on the first side of the belt (N)
T_2 is the tension on the second side of the belt (N)
μ is the coefficient of friction
θ is the angle of wrap of the belt (radians)

Notes: None.

Source: https://hackaday.com/2021/01/26/cable-mechanism-maths-designing-against-the-capstan-equa

Ref. By:

Preconditions for TM:CE: None

Derivation for TM:CE: Not Applicable

RefName: **TM:N2**

Label: Newtons 2nd Law

Equation: $F = ma$

Description: F is the force (N)
m is the mass (kg)
a is the acceleration (m/s^2)

Notes: Newtons 2nd law is used repeatedly in several Instance models

Source: https://www.physicsclassroom.com/class/newtlaws/lesson-3/newton-s-second-law

Ref. By:

Preconditions for TM:N2: None

Derivation for TM:N2: Not Applicable

Helper Theories

This portion will define auxiliary theories to aid in further derivation below.
Add here stuff like: - in: shift distance, out: radius of pulley
- in: shift distance, out: angle of ramp
- in: shift distance, out: angle of helix
- in: shift distance, out: shift radians

4.2.6 General Definitions

We do not have any general definitions at this time.

Number	GD1
Label	**Helix Side Force**
SI Units	$\mathrm{N} = \mathrm{kg\,m\,s^{-2}}$
Equation	$F_H = \frac{T_{\text{eng}} CVT_{\text{ratio}}}{2r\tan(\beta)}$
Description	F_H is the side force due to the helix (N) T_{eng} is the engine torque (N-m) CVT_{ratio} is the CVT ratio (unitless) r is the radius of the ramps from the shaft (m) β is the helix angle (rad)
Source	
Ref. By	IM5

Number	GD2
Label	**Spring Side Force**
SI Units	$\mathrm{N} = \mathrm{kg\,m\,s^{-2}}$
Equation	$F_S = \frac{k_{\text{tor}}(\theta_0+\theta)}{2r\tan(\beta)} + k_{\text{comp}}(x_0 + x)$
Description	F_S is the side force due to the spring (N) k_{tor} is the torsional spring constant (N m/rad) θ_0 is the initial torsion (rad) θ is the amount of angular torsion (rad) k_{comp} is the compression spring constant (N/m) x_0 is the initial compression (m) x is the amount of compression (m) r is the radius of the ramps from the shaft (m) θ is the amount of angular torsion (rad)
Source	
Ref. By	IM5

4.2.7 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Density**
Symbol	ρ
SI Units	kg/m^3
Equation	$\rho = \frac{m}{V}$
Description	ρ is the density (kg/m^3) m is the mass (kg) V is the volume (m^3)
Sources	None.
Ref. By	None.

Number	DD2
Label	**Coefficient of Friction**
Symbol	μ
SI Units	unitless
Equation	$F_f = \mu F_n$
Description	μ is the coefficient of friction F_f is the force of friction F_n is the normal force
Sources	None.
Ref. By	None.

Number	DD3
Label	**Compressionional Spring Constant**
Symbol	k
SI Units	N/m
Equation	$F = k\Delta x$
Description	k is the spring constant (N/m) F is the force (N) Δx is the displacement (m)
Sources	None.
Ref. By	None.

Number	DD4
Label	**Torsional Spring Constant**
Symbol	k
SI Units	N m/rad
Equation	$\tau = k\Delta\theta$
Description	k is the torsional spring constant (N m/rad) τ is the torque (N m) θ is the angle of rotation (radians)
Sources	None.
Ref. By	None.

Number	DD5
Label	**Gravitational Constant**
Symbol	g
SI Units	m/s^2
Equation	$F_g = m \cdot g$
Description	g is the gravitational constant (m/s^2) F_g is the force of gravity m is the mass
Sources	None.
Ref. By	None.

Number	DD6
Label	**Drag Coefficient**
Symbol	C_d
SI Units	unitless
Equation	$F_d = \frac{1}{2}\rho v^2 C_d A$
Description	C_d is the drag coefficient F_d is the force of drag ρ is the density of the fluid v is the velocity of the object A is the cross-sectional area of the object
Sources	None.
Ref. By	None.

Number	DD7
Label	**Rolling Resistance Coefficient**
Symbol	C_{rr}
SI Units	unitless
Equation	$F_{\text{rr}} = C_{\text{rr}} F_n$
Description	C_{rr} is the rolling resistance coefficient F_{rr} is the force of rolling resistance F_n is the normal force
Sources	None.
Ref. By	None.

4.2.8 Data Types

N/A

4.2.9 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.7 to replace the abstract symbols in the models identified in Sections 4.2.5 and 4.2.6.

The goal statment GS1 are solved by Instance Models 1, 2 and 3
The goal statement GS2 are solved by Instance Models 8
The goal statement GS3 are solved by Instance Models 4, 5, 6 and 7

Number	IM1
Label	**Acceleration**
Input	$r_{\text{gear}}, r_{\text{wheel}}, \rho, C_D, A, \theta, T_{\text{CVT}}, C_{\text{rr}}, m_v, m_d$
Equation	$a = \frac{\left(\frac{T_{\text{CVT}}}{r_{\text{gear}}+r_{\text{wheel}}}\right)-(C_{\text{rr}}F_g \sin(\theta))-\left(\frac{1}{2}\rho v^2 C_D A\right)-(F_g \cos(\theta))}{m_v+m_d}$
Output	acceleration a
Description	r_{gear} is the radius of the gear (m) r_{wheel} is the radius of the wheel (m) ρ is the density of the fluid (kg/m^3) C_D is the drag coefficient A is the cross-sectional area of the object (m^2) θ_{inc} is the angle of incline (rad) T_{CVT} is the torque transferred to the wheel (N-m) C_{rr} is the coefficient of rolling resistance m_v is the mass of the vehicle (kg) m_d is the mass of the driver(kg)
Sources	-
Ref. By	

Derivation of Acceleration

The acceleration of the vehicle is derived as follows:

$$T_W - R_r - F_D - F_g \cos(\theta) = ma$$

Where T_W is the torque of the wheel, R_r is the rolling resistance, F_D is the drag force, F_g is the gravitational force, and m is the mass of the vehicle.

$$T_W = \frac{T_{\text{CVT}}}{r_{\text{gear}} + r_{\text{wheel}}}$$

$$R_r = C_{\text{rr}} F_g \sin(\theta_{inc})$$

$$F_D = \frac{1}{2}\rho v^2 C_D A$$

$$m = m_v + m_d$$

Substituting the above equations into the original equation, and then solving for acceleration gives us the final equation:

$$a = \frac{\left(\frac{T_{\text{CVT}}}{r_{\text{gear}} + r_{\text{wheel}}}\right) - (C_{\text{rr}} F_g \sin(\theta)) - \left(\frac{1}{2}\rho v^2 C_D A\right) - (F_g \cos(\theta))}{m_v + m_d}$$

Number	IM2
Label	**Velocity**
Input	$a(t)$
Equation	$v(t) = \int a(t)dt$
Output	$v(t)$
Description	$a(t)$ is the function of acceleration over time
Sources	-
Ref. By	

Number	IM3
Label	**Distance**
Input	$v(t)$
Equation	$d(t) = \int v(t)dt$
Output	$d(t)$
Description	$v(t)$ is the function of velocity over time
Sources	-
Ref. By	

Number	IM4
Label	**Primary Clamping Force**
Input	m, r_{fly}, $f_{\text{prim\_height}}(d_{\text{shift}})$, $f_{\text{prim\_angle}}(d_{\text{shift}})$, k_{prim}, d_{prim}, d_{shift}
Equation	$F_{FW} - F_S = m(r_{\text{fly}} + f_{\text{prim\_height}}(d_{\text{shift}})) \cdot \tan(f_{\text{prim\_angle}}(d_{\text{shift}})) + k_{\text{prim}}(d_{\text{prim}} + d_{\text{shift}})$
Output	$F_{FW} - F_S$ where F_{FW} is the flywheel force and F_S is the spring force, representing the total primary clamping force.
Description	m: Mass of the system. r_{fly}: Radius related to the flywheel. $f_{\text{prim\_height}}(d_{\text{shift}})$: Function that represents the height of the primary system. $f_{\text{prim\_angle}}(d_{\text{shift}})$: Primary function that represents the force dependent on the shifting distance. k_{prim}: Stiffness or coefficient related to the primary system. d_{prim}: Pre-compression or pre-load distance. d_{shift}: Displacement or shift in the system.
Sources	
Ref. By	IM6

Derivation of Primary Clamping Force

The flywheel force is derived as follows:

$$F_{FW} = mr\omega^2 \cdot \tan(f_{\text{prim\_angle}}(d_{\text{shift}}))$$

Where r can be represented by $r_{\text{fly}} + f_{\text{prim\_height}}(d_{\text{shift}})$. Then F_{FW} becomes

$$F_{FW} = m(r_{\text{fly}} + f_{\text{prim\_height}}(d_{\text{shift}}))\omega^2 \cdot \tan(f_{\text{prim\_angle}}(d_{\text{shift}}))$$

The spring force is derived as follows:

$$-F_S = +k_{\text{prim}}(d_{\text{prim}} + d_{\text{shift}})$$

Combining these two forces gives the total primary clamping force:

$$F_{FW} - F_S = m(r_{\text{fly}} + f_{\text{prim\_height}}(d_{\text{shift}}))\omega^2 \cdot \tan(f_{\text{prim\_angle}}(d_{\text{shift}})) + k_{\text{prim}}(d_{\text{prim}} + d_{\text{shift}})$$

Number	IM5
Label	**Secondary Clamping Force**
Input	$k_{\text{sec\_tor}}$, θ_{sec}, $r_{\text{sec\_ramp}}$, $f_{\text{sec\_ramp}}(d_{\text{shift}})$, $f_{\text{sec\_shift}}(d_{\text{shift}})$, $k_{\text{sec\_comp}}$, d_{sec}, d_{shift}
Equation	$F_{\text{sec\_clamp}} = \frac{k_{\text{sec\_tor}}(\theta_{\text{sec}}+f_{\text{sec\_shift}}(d_{\text{shift}}))+T_{eng}CVT_{ratio}}{2r_{\text{sec\_ramp}}\tan(f_{\text{sec\_ramp}}(d_{\text{shift}}))} + k_{\text{sec\_comp}}(d_{\text{sec}} + d_{\text{shift}})$
Output	Secondary clamping force $F_{\text{sec\_clamp}}$
Description	$k_{\text{sec\_tor}}$: Torsional spring rate. θ_{sec}: Initial torsional rotation. $f_{\text{sec}}(d_{\text{shift}})$: Amount of torsional rotation based on the distance shifted. $r_{\text{sec\_ramp}}$: Radius of the ramps from the shaft. $k_{\text{sec\_comp}}$: Compression spring rate. d_{sec}: Initial compression distance. d_{shift}: Displacement or shift in the system. T_{eng}: Torque of the engine. CVT_{ratio}: Ratio of the CVT system.
Sources	
Ref. By	IM6

Derivation of Secondary Clamping Force

The secondary clamping force is derived as follows:

$$F_{\text{sec\_clamp}} = F_H + F_S$$

The helix force is derived as follows:

$$F_H = \frac{T_{\text{eng}}CVT_{\text{ratio}}}{2r\tan(\beta)}$$

Substituting in our variables gives us the following equation:

$$F_H = \frac{T_{eng}CVT_{ratio}}{2r_{\text{sec\_ramp}}\tan(f_{\text{sec\_ramp}}(d_{\text{shift}}))}$$

The spring force is derived as follows:

$$F_S = \frac{k_{\text{tor}}(\theta_0 + \theta)}{2r\tan(\beta)} + k_{\text{comp}}(x_0 + x)$$

Substituting in our variables gives us the following equation:

$$F_S = \frac{k_{\text{sec\_tor}}(\theta_{\text{sec}} + f_{\text{sec\_shift}}(d_{\text{shift}}))}{2r_{\text{sec\_ramp}} \tan(f_{\text{sec\_ramp}}(d_{\text{shift}}))} + k_{\text{sec\_comp}}(d_{\text{sec}} + d_{\text{shift}})$$

Combining these two forces gives the total secondary clamping force:

$$F_H + F_S = \frac{T_{eng} CVT_{ratio}}{2r_{\text{sec\_ramp}} \tan(f_{\text{sec\_ramp}}(d_{\text{shift}}))} + \frac{k_{\text{sec\_tor}}(\theta_{\text{sec}} + f_{\text{sec\_shift}}(d_{\text{shift}}))}{2r_{\text{sec\_ramp}} \tan(f_{\text{sec\_ramp}}(d_{\text{shift}}))} + k_{\text{sec\_comp}}(d_{\text{sec}} + d_{\text{shift}})$$

The final equation can be reached through simplifying the above equation by combining like terms:

$$F_{\text{sec\_clamp}} = \frac{k_{\text{sec\_tor}}(\theta_{\text{sec}} + f_{\text{sec\_shift}}(d_{\text{shift}})) + T_{eng} CVT_{ratio}}{2r_{\text{sec\_ramp}} \tan(f_{\text{sec\_ramp}}(d_{\text{shift}}))} + k_{\text{sec\_comp}}(d_{\text{sec}} + d_{\text{shift}})$$

Number	IM6
Label	**Sheave Acceleration**
Input	$F_{\text{clamp\_prim}}, F_{\text{clamp\_sec}}, \mu, T_T, T_s, m$
Equation	$a = \frac{F_{\text{clamp\_prim}} - F_{\text{clamp\_sec}} - \mu(T_T - T_s)}{m}$
Output	Sheave acceleration a
Description	$F_{\text{clamp\_prim}}$: Primary clamping force $F_{\text{clamp\_sec}}$: Secondary clamping force μ: Coefficient of friction T_T: Torque of the transmission T_s: Torque of the sheave m: Mass of the system
Sources	
Ref. By	-

-

Number	IM7
Label	**Belt Acceleration**
Input	$T_{\text{eng}}, r_{\text{prim}}, F_F, m$
Equation	$a_{\text{belt}} = \frac{T_{\text{eng}}}{m \cdot r_{\text{prim}}} - F_F$
Output	Belt Acceleration a_{belt}
Description	T_{eng}: Engine torque r_{prim}: Primary radiu F_F: Friction force m: Mass of the system
Sources	
Ref. By	

Derivation of Belt Acceleration

The torque of the engine equation is derived as such:

$$T_{\text{eng}} = (T_T - T_s) \cdot r_{\text{prim}} \quad \Rightarrow \quad T_T - T_s = \frac{T_{\text{eng}}}{r_{\text{prim}}}$$

Similarly, for the load torque:

$$T_{\text{load}} = (T_T - T_s) \cdot r_{\text{sec}} \quad \Rightarrow \quad T_T - T_s = \frac{T_{\text{load}}}{r_{\text{sec}}}$$

Equating the two torque equations:

$$\frac{T_{\text{eng}}}{r_{\text{prim}}} = \frac{T_{\text{load}}}{r_{\text{sec}}}$$

The equation for the total torque is derived as:

$$T_T = T_s \cdot e^{\mu\theta}$$

Substituting this into the torque balance equation:

$$\frac{T_{\text{eng}}}{r_{\text{prim}}} + T_s = T_s \cdot e^{\mu\theta}$$

Solving for T_s:

$$T_s = \frac{T_{\text{eng}}}{r_{\text{prim}} \cdot (e^{\mu\theta} - 1)}$$

The final equation for belt acceleration is derived as:

$$m \cdot a_{\text{belt}} = \frac{T_{\text{eng}} \cdot e^{\mu\theta}}{r_{\text{prim}} \cdot (e^{\mu\theta} - 1)} - \frac{T_{\text{eng}}}{r_{\text{prim}} \cdot (e^{\mu\theta} - 1)}$$

Simplifying:

$$a_{\text{belt}} = \frac{T_{\text{eng}}}{m \cdot r_{\text{prim}} \cdot (e^{\mu\theta} - 1)} \cdot (e^{\mu\theta} - 1)$$

Thus, the final equation becomes:

$$a_{\text{belt}} = \frac{T_{\text{eng}}}{m \cdot r_{\text{prim}}} - F_F$$

Number	IM8
Label	**RPM and Torque of Engine**
Input	
Equation	
Output	
Description	
Sources	
Ref. By	-

4.2.10 Input Data Constraints

Table 2 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 2 are listed in Table 4.

Table 2: Input Variables

Var	**Physical Constraints**	**Software Constraints**	**Typical Value**	**Uncertainty**
L	$L > 0$	$L_{\text{min}} \leq L \leq L_{\text{max}}$	1.5 m	10%
m_{fly}	$0 \leq m_{fly}$	$m_{fly} \leq 1$	0.2 kg	
ω_{engine}	$0 \leq \omega_{engine}$	$\omega_{engine} \leq \omega_{\text{max}}$	400 rad/s	
d_{shift}	$0 \leq d_{shift} \leq w_{belt}$		0.1 m	
θ_{prim_out}	$0 \leq \theta_{prim\_out} \leq \frac{\pi}{2}$		$\frac{\pi}{4}$ rad	
d_{prim}	$0 \leq d_{\text{prim}} \leq d_{\text{max\_prim\_pre}}$		0.1 m	
d_{sec}	$0 \leq d_{\text{sec}} \leq d_{\text{max\_sec\_pre}}$		0.1 rad	
m_{driver}	$0 \leq m_{\text{driver}}$	$m_{\text{driver}} \leq m_{\text{max\_human}}$	75 kg	
m_{car}	$0 \leq m_{\text{car}}$	$m_{\text{car}} \leq m_{\text{max\_car}}$	225 kg	
k_{prim}	$0 \leq k_{\text{prim}}$	$k_{\text{prim}} \leq k_{\text{max\_comp\_spring}}$	100 N/m	
$k_{\text{sec\_comp}}$	$0 \leq k_{\text{sec\_comp}}$	$k_{\text{sec\_comp}} \leq k_{\text{max\_comp\_spring}}$	100 N/m	
$k_{\text{sec\_tor}}$	$0 \leq k_{\text{sec\_tor}}$	$k_{\text{sec\_tor}} \leq k_{\text{max\_tor\_spring}}$	50 Nm/rad	

Table 4: Specification Parameter Values

Var	**Value**
L_{min}	0.1 m
ω_{max}	600 rad/s
w_{belt}	UNKNOWN, see belt spec m
$d_{\text{max\_prim\_pre}}$	UNKNOWN, see CAD m
$d_{\text{max\_sec\_pre}}$	UNKNOWN, see CAD rad
$m_{\text{max\_human}}$	200 kg
$m_{\text{max\_car}}$	350 kg
$k_{\text{max\_comp\_spring}}$	1000 N/m
$k_{\text{max\_tor\_spring}}$	750 Nm/rad
v_{max}	UNKNOWN, do gear calcs m/s
$f_{\text{belt\_max}}$	UNKNOWN, see belt spec N

4.2.11 Properties of a Correct Solution

Table 6: Output Variables

Var	**Physical Constraints**	**unit**
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A??)	unitless
$v(t)$	$0 \leq v(t) \leq v_{\text{max}}$	m/s
Sum of clamping forces	Clamp $\leq f_{\text{belt\_max}}$	N

[This section is not for test cases or techniques for verification and validation. Those topics will be addressed in the Verification and Validation plan. —TPLT]

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: The system shall simulate the behavior of the McMaster Baja Team's Continuous Variable Transmission(CVT) using mathematical models.

R2: The system shall allow users to adjust the following CVT tunning parameters: primary weight, primary ramp geometry, primary spring rate, primary spring pretension, secondary helix geometry, secondary spring rate, secondary spring pretension.

R3: The system shall allow users to adjust vehicle and driver weight, traction and angle of incline in addition to the CVT tuning parameters.

R4: The system shall provide users with an interface to input tuning parameters.

R5: The system shall display graphs of simulation output based on the user inputted tuning parameters.

R6: The system shall allow users to compare results of simulations with different CVT tuning parameters.

R7: The system shall calculate the CVT ratio, clamping force, RPM, torque and belt slippage as functions of time. Combined with engine input, the system shall calculate distance, speed and acceleration as functions of time.

R8: The system shall allow users to export outputted simulated graphical data.

R9: The system shall allow users to access the application through the use of a personal computer or laptop.

R10: The system shall be compatible with Windows, MacOS and Linux.

5.2 Nonfunctional Requirements

NFR1: **Accuracy** The system shall achieve an accuracy of at least 95 % in correctly predicting CVT outputs.

NFR2: **Usability** 85 % of a representative user group shall be able to successfully input CVT parameters and receive CVT outputs on their first use.

NFR3: **Maintainability** If a likely change is made to the finished software, it will take at most 10 % of the original development time, assuming the same development resources are available.

NFR4: **Verifiability** The system shall output the resulting tunned CVT outputs which must able to be cross-verified against data available through the McMaster Baja Team with a consistency rate of 95 %.

NFR5: **Understandability** The system shall be understood by at least 90 % of a representative user group with no more than 15 minutes of initial training, as measured by assessing ability to generate and export tunned CVT output where users must score at least 90 %

NFR6: **Reusability** The system's components shall be designed for reuse in the case the McMaster Baja team acquires a new CVT. These modifications should be made with minimal modifications, defined as requiring modification to less than 30 % of the code to configure these changes.

5.3 Rationale

The assumptions made for the CVT (Continuously Variable Transmission) system model aim to simplify its dynamics, making it easier to analyze and develop a solution. A key assumption is the exclusion of temperature effects on material properties, aside from the belt, which allows for a more straightforward focus on the mechanics of the interactions (A1). The negligible influence of gravitational forces, further simplifies the problem by removing the need to eliminating the need to consider variations in position. Additionally, the elasticity of belts is considered insignificant, treating components as ideally rigid, allowing the development of a model without accounting for material deformations (A3). Vibrational effects from external sources are also assumed to be minimal, avoiding the complexity of vibration-induced disturbances in the transmission behavior (A4). Wear and tear of components, including the belt and pulley are disregarded for short-term operational modeling, allowing for an idealized performance assumption (A5). Frictional losses in non-critical components, are deemed negligible, allowing the focus to frictional effects that directly impact the CVT's core mechanics (A6). Finally, the engine is assumed to operate at full throttle with consistent power delivery, simplifying the model by maintaining uniform engine behavior (A4). These simplifications affect the process of formulating equations governing velocity, distance, primary clamping force, secondary clamping force, CVT Ratio, Torque transfer and RPM and Torque of the Engine(IM1-IM8)

6 Likely Changes

LC1: The effect of temperature on material properties might be included in the future, (A1)

LC2: The elasticity of the belt might be considered in the future, (A3)

LC3: Frictional losses in non-critical components might be considered in the future, (A6)

LC4: Some of the math models will most likely change as the design is refined.

7 Unlikely Changes

None

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 9 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 11 shows the dependencies of instance models, requirements, and data constraints on each other. Table 13 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —TPLT]

[The traceability matrix is not generally symmetric. If GD1 uses A1, that means that GD1's derivation or presentation requires invocation of A1. A1 does not use GD1. A1 is "used by" GD1. —TPLT]

[The traceability matrix is challenging to maintain manually. Please do your best. In the future tools (like Drasil) will make this much easier. —TPLT]

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent

	TM??	TM??	TM??	GD??	GD??	DD??	DD??	DD??	DD??	IM??	IM??	IM?
TM??												
TM??			X									
TM??												
GD??												
GD??	X											
DD??				X								
DD??				X								
DD??												
DD??								X				
IM??					X	X	X				X	
IM??					X		X		X	X		
IM??		X										
IM??		X	X				X	X	X		X	

Table 8: Traceability Matrix Showing the Connections Between Items of Different Sections

	TM1	TM2	TM3	TM4	TM5	TM6	TM7	TM8	TM9	IM1	IM2	IM3	IM4	IM5
TM1														
TM2														
TM3														
TM4														
TM5														
TM6														
TM7														
TM8														
TM9														
IM1														
IM2														
IM3														
IM4														
IM5														
IM6														
IM7														
IM8														

Table 9: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM??	IM??	IM??	IM??	4.2.10	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R??	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R??			X	X			
R??		X					
R??		X					

Table 10: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	IM??	IM??	IM??	IM??	4.2.10	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R??	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R??			X	X			
R??		X					
R??		X					

Table 11: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
TM??	X																		
TM??																			
TM??																			
GD??		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 12: Traceability Matrix Showing the Connections Between Assumptions and Other Items

	A1	A2	A3	A4	A5	A6	A7
TM1							
TM2							
TM3							
TM4							
TM5							
TM6							
TM7							
TM8							
TM9							
IM1							
IM2							
IM3							
IM4							
IM5							
IM6							
IM7							
IM8							
LC1							
LC2							
LC3							
LC4							

Table 13: Traceability Matrix Showing the Connections Between Assumptions and Other Items

dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure **??** shows the dependencies of instance models, requirements, and data constraints on each other.

9 Development Plan

N/A

[This section is optional. It is used to explain the plan for developing the software. In particular, this section gives a list of the order in which the requirements will be implemented. In the context of a course this is where you can indicate which requirements will be implemented as part of the course, and which will be "faked" as future work. This section can be organized as a prioritized list of requirements, or it could should the requirements that will be implemented for "phase 1", "phase 2", etc. —TPLT]

10 Values of Auxiliary Constants

[Show the values of the symbolic parameters introduced in the report. —TPLT]

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —TPLT]

[The value of FRACTION, for the Maintainability NFR would be given here. —TPLT]

References

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

W. Spencer Smith and Nirmitha Koothoor. A document-driven method for certifying scientific computing software for use in nuclear safety analysis. *Nuclear Engineering and Technology*, 48(2):404–418, April 2016. ISSN 1738-5733. doi: http://dx.doi.org/10.1016/j.net.2015.11.008. URL http://www.sciencedirect.com/science/article/pii/S1738573315002582.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ȧgerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.

[The following is not part of the template, just some things to consider when filing in the template. —TPLT]
[Grammar, flow and LaTeXadvice:

- For Mac users `*.DS_Store` should be in `.gitignore`
- LaTeX and formatting rules
  - Variables are italic, everything else not, includes subscripts (link to document)
    - Conventions
    - Watch out for implied multiplication
  - Use BibTeX
  - Use cross-referencing
- Grammar and writing rules
  - Acronyms expanded on first usage (not just in table of acronyms)
  - "In order to" should be "to"

—TPLT]
[Advice on using the template:

- Difference between physical and software constraints
- Properties of a correct solution means *additional* properties, not a restating of the requirements (may be "not applicable" for your problem). If you have a table of output constraints, then these are properties of a correct solution.
- Assumptions have to be invoked somewhere
- "Referenced by" implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract

—TPLT]

Appendix — Reflection

[Not required for CAS 741 —SS]

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning.

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
   We were successful in splitting up the work that could be done individually and then coming together to discuss and combine our work. We also did a good job in playing to each team members strengths and dividing the work accordingly.

2. What pain points did you experience during this deliverable, and how did you resolve them?
   The main pain point of this deliverable was in trying to understand the mathematical models and physics behind the CVT system. We resolved this by meeting together as a team and having a group discussion on the topic. We went through the math and derived the equations together, which helped us understand the material better. We also reached out to our clients for clarifications and they provided us with help and resources to better understand the topic.

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?
   An example of a requirement coming from our clients is the Verifiability requirement. When we had our first capstone meeting with Dr. Smith he stressed the importance of verifying our results. This resulted in us adding the requirement to our document. Additionally, the majority of our functional requirements were inspired by the needs of the McMaster Baja team who we meet with regularly.

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.
   Many of the courses we have taken so far will prove helpful in completing our capstone project. SFWRENG 4HC3, will help us with the design of the GUI. SFWRENG 2AA4 and SFWRENG 3A04 will help us architect our code base. PHYSICS 1D03,

SFWRENG 3MX3 and SFWRENG 3DX34, will help us understand and model the physics and math behind the CVT system. SFWRENG 3XB3 and SFWRENG 4X03, will help with creating efficient and reliable implementations. SFWRENG 3S03, will help us verify and validate our software.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.
   We will need to learn a lot about Unity projects and how to create our mathematical models in python. For Unity we need to learn how to make a GUI interface, how to add and animate a 3D model, how to create and show graphs, and how to interface with a python script to run the mathematical models. For python we need to learn how to create and implementation these complex mathematical models and how to make it interface with Unity.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?
   For getting familiar with Unity projects, we will watch youtube tutorials and read guides provided by Unity. The team members who chose to do the videos did so because they are more visual learners and prefer to see the process in action. Those that chose to read the guides did so because they prefer to read and also prefer that it is from an official source. For learning the python required, we will do research on libraries and functions that will help us implement our mathematical models. The team members who chose to do research did so because they are more comfortable with researching and learning on their own.