Software Requirements Specification for Damped Harmonic Ocsillator: Damped Harmonic Oscillator Illustrated by Online Calculator

Muhammad Waqar Ul Hassan Awan

February 5, 2024

Contents

Revision History

Date	Version	Notes
February 2, 2024	1.0	Initial Document Release

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
N	force	Newton
Hz	frequency	hertz (s^{-1})
v	velocity	metre per second ($\mathrm{m\,s^{-1}}$)
a	acceleration	metre per second square ($\mathrm{m\,s^{-2}}$)
g	gravity	metre per second square ($\mathrm{m\,s^{-2}}$)

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**
k	Newton per metre ($\mathrm{N\,m^{-1}}$)	Spring Constant
c	kilogram per second ($\mathrm{kg\,s^{-1}}$)	Damping Coefficient
x	metre (m)	Displacement
w	radians per second ($\mathrm{rad\,s^{-1}}$)	Angular Frequency

1.3 Abbreviations and Acronyms

symbol	**description**
SRS	Software Requirement Specification
ODE	Ordinary Differential Equation
DE	Differential Equation
DHO	Damped Harmonic Oscillator
SHM	Simple Harmonic Motion

1.4 Mathematical Notation

- **Greek symbols** represent constants and parameters, such as ω for angular frequency.
- **Subscripts** are used to denote specific instances or components, e.g., x_t for the position at time t.
- **Mathematical operations** and functions are denoted as follows: sin and cos for trigonometric functions, e for the exponential function, and d/dt for differentiation with respect to time.

2 Introduction

In physics and engineering, the concept of harmonic oscillators gives us a way to understand how things move back and forth or oscillate. It offering insights into systems' behavior under periodic forces. Understanding system dynamics of a simple pendulum or a mass attached to a spring is crucial for advancements in various scientific and engineering fields. This document introduces a software project aimed at modeling and analyzing harmonic oscillators, providing a tool for educational, research, and industrial applications to observe the effect of damping on oscillating bodies.

This introduction serves as a roadmap to the document, outlining its structure and guiding the reader through the subsequent sections. Following this introduction, the document details the purpose of the SRS, the scope of requirements, characteristics of the intended reader, and the organization of the document itself.

2.1 Purpose of Document

The purpose of this Software Requirement Specification (SRS) document is to outline the functional and non-functional requirements for a software project focused on the simulation and analysis of harmonic oscillators. This document is intended to serve as a comprehensive guide for the development team, ensuring that the software meets the specific needs of its users. Additionally, this SRS aims to facilitate clear communication among stakeholders, provide a basis for estimating costs and timelines, and implementation phases of the project.

2.2 Scope of Requirements

This project is the simulation of harmonic oscillators, a fundamental concept in physics. To manage the complexity in modeling real-world phenomena, the scope of this software will be constrained. Specifically, the project will focus on:

- Modeling in two dimensions to simplify visualizations and computations.
- Ignoring environmental factors such as temperature and pressure variations that might affect the system's properties.
- Only considering linear restoring force.

These constraints are chosen to make the problem tractable while still providing valuable insights and educational utility.

2.3 Characteristics of Intended Reader

This document is written for the people who will help build, review, and maintain the software. They should know their way around basic physics and be comfortable with the mathematics, especially the oscillators and how they work. They don't need to be physicists, but they should understand the science behind what I am trying to simulate.

2.4 Organization of Document

The document is organized to facilitate easy navigation and understanding of the software requirements. After this section, the document is structured as follows:

- Section 3: General System Description
- Section 4: Specific System Description
- Section 5: Requirements

Readers are encouraged to refer to Section 3 for general system description, Section 4 for specific system description, and Section 5 for requirements.

3 General System Description

This section outlines the overall framework of the software designed to simulate and analyze harmonic oscillators, both simple and damped. It aims to provide a basic understanding of how the system interacts with its environment, detailing user interaction, external interfaces, and inherent system constraints. The purpose here is to establish a broad context that will make the specific requirements outlined in subsequent sections clearer and more meaningful. By describing the system at a general level, this section remains applicable even as specific functionalities evolve or expand within the project's scope.

3.1 System Context

[Your system context will include a figure that shows the abstract view of the software. Often in a scientific context, the program can be viewed abstractly following the design pattern of Inputs → Calculations → Outputs. The system context will therefore often follow this pattern. The user provides inputs, the system does the calculations, and then provides the outputs to the user. The figure should not show all of the inputs, just an abstract view of the main categories of inputs (like material properties, geometry, etc.). Likewise, the outputs should be presented from an abstract point of view. In some cases the diagram will show other external entities, besides the user. For instance, when the software product is a library, the user will be another software program, not an actual end user. If there are system constraints that the software must work with external libraries, these libraries can also be shown on the System Context diagram. They should only be named with a specific library name if this is required by the system constraint. —TPLT]

[For each of the entities in the system context diagram its responsibilities should be listed. Whenever possible the system should check for data quality, but for some cases the

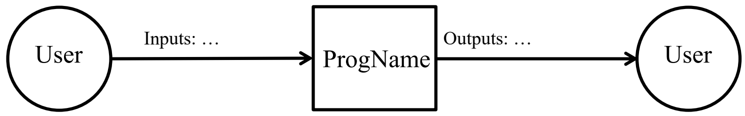

Figure 1: System Context

user will need to assume that responsibility. The list of responsibilities should be about the inputs and outputs only, and they should be abstract. Details should not be presented here. However, the information should not be so abstract as to just say "inputs" and "outputs". A summarizing phrase can be used to characterize the inputs. For instance, saying "material properties" provides some information, but it stays away from the detail of listing every required properties. —TPLT]

- User Responsibilities:
  -
- Damped Harmonic Ocsillator Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  -

[Identify in what context the software will typically be used. Is it for exploration? education? engineering work? scientific work?. Identify whether it will be used for mission-critical or safety-critical applications. —TPLT] [This additional context information is needed to determine how much effort should be devoted to the rationale section. If the application is safety-critical, the bar is higher. This is currently less structured, but analogous to, the idea to the Automotive Safety Integrity Levels (ASILs) that McSCert uses in their automotive hazard analyses. —TPLT]

[The —SS]

3.2 User Characteristics

Users of the Damped Harmonic Oscillator software are expected to have a basic understanding of calculus and physics at an undergraduate level. Specifically, they should be familiar with the concepts of differential equations as they apply to motion and be able to apply basic physics principles to interpret the simulation results. This foundational knowledge is crucial for effective interaction with the software, enabling users to input realistic parameters and accurately interpret simulation outcomes.

3.3 System Constraints

Several key constraints influence the design and deployment of this software:

- Platform Independence: It should run on common operating systems and web browsers without special hardware requirements.
- Libraries and APIs: Where applicable, the software will rely on standard, open-source libraries for mathematical computations and graphical displays (e.g., NumPy, Matplotlib). The choice of these libraries is constrained by their availability, documentation, and compatibility with the target operating systems.
- User Interface: Aimed at users with a basic understanding of physics and calculus, the software interface will prioritize simplicity to facilitate learning and exploration without overwhelming users with unnecessary complexity.

4 Specific System Description

This section delves into the high-level overview of the problem that the Damped Harmonic Oscillator software aims to solve. Following this, the solution characteristics specification elaborates on the assumptions, theories, definitions, and instance models that explains the solution proposed by this project.

4.1 Problem Description

Damped Harmonic Ocsillator targets the problem of accurately modeling and simulating the behavior of a damped harmonic oscillator. Such systems are essential in understanding phenomena where an object oscillates and gradually loses energy due to resistance or damping forces. This simulation is vital in fields such as mechanical engineering, where understanding the damping of oscillatory systems can lead to better designs and predictions.

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:

- **Isolated System:** Neglects external forces other than the damping and restoring forces.
- **Harmonic Motion:** Assumes that in the absence of damping, the system would exhibit simple harmonic motion.
- **Linear Damping:** A damping force that is directly proportional to the velocity of the oscillating object.

- **Nonlinear Damping:** A damping force that varies with velocity in a non-proportional manner, often depending on the velocity's magnitude or other factors.

4.1.2 Physical System Description

The system comprises key elements crucial for its analysis and simulation:

- **Mass (m):** Represents the inertia of the oscillating object.
- **Spring Constant (k):** Indicates the force needed to displace the system from its equilibrium position.
- **Damping Coefficient (b):** For linear damping, quantifies the proportionality between the damping force and velocity.
- **Damping Function (f(v)):** For nonlinear damping, represents the relationship between damping force and velocity, which may vary based on the system's specific characteristics.

The system's behavior is significantly influenced by the nature of the damping force. The model must account for both linear and nonlinear damping scenarios, with the interactions between the mass, spring constant, and damping forces defining the system's dynamic response.

4.1.3 Goal Statements

With the inclusion of both linear and nonlinear damping forces, the software aims to:

GS1: Create a simulation model that accurately represents the behaviour of a damped harmonic oscillator in various scenarios.

GS2: Facilitate the understanding of damping effects on oscillatory systems through interactive and visual tools.

GS3: Simulate of mathematical derivation.

GS4: Design tools to allow users to create custom scenarios and do comparative analysis.

GS5: Enhance educational understanding of damped oscillatory systems.

4.2 Solution Characteristics Specification

This section elaborates on the mathematical and physical principles that form the basis of the solution, including assumptions, theoretical models, general definitions, and the derivation of instance models.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [TM], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: **Constant Mass [TM, DD]:** The mass (m) of the oscillator is constant throughout the motion.

A2: **Homogeneous Medium [TM, GD]:** Assumes the oscillator moves through a homogeneous medium, affecting the damping forces uniformly across the system.

A3: **No External Forces [TM]:** External forces, other than damping and restoring forces, are neglected.

A4: **Initial Conditions Known [IM]:** The initial position and velocity of the oscillator are known and specified.

4.2.2 Theoretical Models

Theoretical models provide the mathematical foundation necessary for understanding and solving the dynamics of damped harmonic oscillators. These models incorporate physical laws and constitutive equations that describe how the system behaves under various conditions, including both linear and non-linear damping forces. The models are crucial for predicting system behavior and formulating solutions that can be applied in practical scenarios.

RefName: **TM:EOM-LDHO**

Label: Equation of Motion for Linearly Damped Harmonic Oscillators

Equation:

$$m * d^2x/dt^2 + c * dx/dt + kx = 0$$

Description: This equation models the motion of a damped harmonic oscillator, where m is the mass of the oscillator, dx/dt, d^2x/dt^2 represent the velocity and acceleration, respectively, c is the damping coefficient for linear damping, k is the spring constant.

Notes: None.

Source: None

Ref. By: None

Preconditions for TM:EOM-LDHO: Assumes that the oscillator is subject to a restoring force proportional to displacement x from equilibrium and a damping force that is linearly proportional to velocity.

Derivation for TM:EOM-LDHO: Not Applicable

RefName: **TM:EOM-NDHO**

Label: Equation of Motion for Non-Linearly Damped Harmonic Oscillators

Equation:

$$m * d^2x/dt^2 + f(dx/dt) + kx = 0$$

Description: This equation models the motion of a damped harmonic oscillator, where m is the mass of the oscillator, dx/dt, d^2x/dt^2 represent the velocity and acceleration, respectively, $f(dx/dt)$ represents the function describing non-linear damping forces, k is the spring constant.

Notes: None.

Source: None

Ref. By: None

Preconditions for TM:EOM-NDHO: Assumes that the oscillator is subject to a restoring force proportional to displacement x from equilibrium and a damping force that follows non-linear relationship.

Derivation for TM:EOM-NDHO: Not Applicable

4.2.3 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Damping Force**
SI Units	N(Newton)
Equation	$F_{\text{damping}} = -c * dx/dt$ – Linear Damping $F_{\text{damping}} = -f(dx/dt)$ – Non-Linear Damping
Description	This definition differentiates between the damping forces in linear and non-linear systems. For linear systems, the force is proportional to the velocity(dx/dt) with a constant$c$. In contrast, non-linear systems exhibit a damping force represented by $f(dx/dt)$, indicating a dependency on the velocity's magnitude or other factors.
Source	Hadley, Mark. "Damped Harmonic Oscillator" Department of Physics, Graz University of Technology
Ref. By	DD**??**, DD**??**

Number	GD2
Label	**Natural Frequency of Oscillation**
SI Units	Hz(Hertz)
Equation	$\omega_0 = \sqrt{k/m}$
Description	This equation provides the natural frequency(w_0) of an undamped harmonic oscillator, derived from the spring constant(k) and mass(m). It signifies the oscillator's inherent vibrational frequency absent damping forces.
Source	Hadley, Mark. "Damped Harmonic Oscillator" Department of Physics, Graz University of Technology
Ref. By	DD**??**, DD**??**, DD**??**

Number	GD3
Label	**Energy Dissipation**
SI Units	J(Joules)
Equation	**Total Energy:** $E = 1/2 * kx^2 + 1/2 * mv^2$ **Linear Damping Dissipation Rate:** $dE/dt = -c * dx/dt$ **Non-Linear Damping:** Energy Dissipation rate varies with $f(dx/dt)$
Description	Outlines how the oscillator's total mechanical energy is influenced by damping. In linear damping scenarios, energy dissipation is directly proportional to the square of velocity, while in non-linear scenarios, it is dictated by a function of velocity, $f(dx/dt)$.
Source	Hadley, Mark. "Damped Harmonic Oscillator" Department of Physics, Graz University of Technology
Ref. By	DD**??**, DD**??**, DD**??**

4.2.4 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	**Damping Coefficient**
Symbol	c (linear damping), $f(dx/dt)$ (non-linear damping)
SI Units	$\mathrm{kg\,s^{-1}}$
Equation	-
Description	For linear damping, c represents the constant proportionality factor between the damping force and the velocity of the oscillator. For non-linear damping, $f(dx/dt)$ represents a function defining the relationship between the damping force and the velocity, which varies with the velocity's magnitude or other system-specific factors.
Sources	"Damped Harmonic Oscillator" – Wikipedia
Ref. By	IM2

Number	DD2
Label	**Mass**
Symbol	m
SI Units	kg
Equation	-
Description	Represents the mass of the harmonic oscillator. The mass is a crucial parameter in determining the system's natural frequency and the dynamics of its motion under damping forces.
Sources	"Damped Harmonic Oscillator" – Wikipedia
Ref. By	IM**??**

Number	DD3
Label	**Spring Constant**
Symbol	k
SI Units	$\mathrm{N\,m^{-1}}$
Equation	-
Description	The spring constant k quantifies the stiffness of the spring in the harmonic oscillator. It is a key factor in defining the natural frequency of the oscillator and its response to applied forces.
Sources	"Damped Harmonic Oscillator" – Wikipedia
Ref. By	IM**??**

Number	DD4
Label	**Natural Frequency**
Symbol	ω_0
SI Units	$\mathrm{rad\,s^{-1}}$
Equation	-
Description	The natural frequency ω_0 of an undamped harmonic oscillator is determined by its mass m and spring constant k. This frequency is foundational for analyzing the oscillator's behavior in the absence of damping forces.
Sources	"Damped Harmonic Oscillator" – Wikipedia
Ref. By	IM**??**

Number	DD5
Label	**Displacement**
Symbol	x
SI Units	m
Equation	-
Description	Displacement x from the equilibrium position provides a measure of how far the oscillator moves over time. It is a fundamental quantity for describing the oscillator's motion and is affected by both damping forces and the system's natural dynamics.from the equilibrium position provides a measure of how far the oscillator moves over time. It is a fundamental quantity for describing the oscillator's motion and is affected by both damping forces and the system's natural dynamics.
Sources	"Damped Harmonic Oscillator" – Wikipedia
Ref. By	IM**??**(Damping Force for Linear and Non-Linear Systems), IM**??** (Displacement and Velocity Over Time)

4.2.5 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 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

Number	IM1
Label	**Equation of Motion**
Input	Mass of the oscillator m, damping coefficient c or $f(dx/dt)$, spring constant k, initial displacement x_0, initial velocity v_0.
Output	Displacement xt as a function of time t, Velocity $v(t)$ as a function of time t.
Description	This model describes the motion of a damped harmonic oscillator by incorporating linear or non-linear damping forces. The equation of motion for linear damping is given by $m * d^2x/dt^2 + f(dx/dt) + kx = 0$, and for non-linear damping, it is modified to include a non-linear damping term, represented by $f(dx/dt)$, affecting the velocity term.
Sources	Derived from the principles of mechanics as discussed in the provided Wikipedia link and the Graz University of Technology's resource on damped harmonic oscillators.
Ref. By	IM**??**

Derivation of Equation of Motion

The equation is derived from Newton's second law of motion, $F = ma$, with force contributions from both the spring ($-kx$) and the damping force ($-c * dx/dt$ or $-f(dx/dt)$).

Number	IM2
Label	**Energy Dissipation**
Input	Spring constant k, mass m, damping coefficient c or $f(dx/dt)$ initial displacement x_0, initial velocity v_0.
Output	Total energy $E(t)$ of the system as a function of time t.
Description	This model quantifies the energy in the oscillator system and its rate of dissipation due to damping. For linear damping, the rate of energy dissipation is directly proportional to the square of the velocity, whereas, for non-linear damping, the dissipation rate is a function of velocity $f(dx/dt)$.
Sources	Based on energy conservation principles and the work-energy theorem, adapted to include damping forces.
Ref. By	IM**??**

Derivation of Energy Dissipation

The total mechanical energy $E(t)$ is given by the sum of kinetic and potential energy. The rate of change of this energy (dE/dt) reflects energy dissipation due to damping.

4.2.6 Input Data Constraints

Table 1 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 1 are listed in Table 2.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
m	$m > 0$	$m_{\text{min}} \leq m \leq m_{\text{max}}$	1 kg	5%
k	$k > 0$	$k_{\text{min}} \leq k \leq k_{\text{max}}$	200 N/m	5%
c	$c \geq 0$	$c_{\text{min}} \leq c \leq c_{\text{max}}$	5 N s/m	10%
x_0	$x_0 \geq 0$	$x_{0_{min}} \leq x_0 \leq x_{0_{max}}$	0.1 m	5%
v_0	No constraint	$v_{0_{min}} \leq v_0 \leq v_{0_{max}}$	0 m/s	5%

(*) The software constraints are set to allow a wide range of scenarios, facilitating experimentation with both common and unusual situations. The specified typical values and uncertainties offer a baseline for standard simulations and uncertainty quantification exercises.

Table 2: Specification Parameter Values

Var	**Value**
m_{min}	0.1 kg
m_{max}	10 kg
k_{min}	50 $\text{N}\,\text{m}^{-1}$
k_{max}	500 $\text{N}\,\text{m}^{-1}$
c_{min}	0 $\text{N}\,\text{s}\,\text{m}^{-1}$
c_{max}	50 $\text{N}\,\text{s}\,\text{m}^{-1}$
$x_{0_{min}}$	0 m
$x_{0_{max}}$	1 m
$v_{0_{min}}$	-10 $\text{m}\,\text{s}^{-1}$
$v_{0_{max}}$	10 $\text{m}\,\text{s}^{-1}$

4.2.7 Properties of a Correct Solution

A correct solution must exhibit several fundamental physical principles and constraints. These criteria ensure that the model accurately reflects the behavior of real-world systems under both linear and non-linear damping forces. Below is an overview of these essential properties, along with a table summarizing the output variables and their physical constraints.

Essential Properties

- **Conservation of Energy (when applicable):** In the absence of external forces, the total mechanical energy (kinetic plus potential) of the system should be conserved over a complete cycle of motion in an undamped oscillator. For damped oscillators, the solution must show a decrement in total energy over time, corresponding to the energy dissipated due to damping.

- **Harmonic Motion:** The solution should exhibit characteristics of harmonic motion, with periodicity and amplitude consistent with the input parameters for the mass, spring constant, and damping coefficients.

- **Damping Behavior:** For linear damping, the solution must reflect an exponential decay of motion amplitude over time. In the case of non-linear damping, the solution should demonstrate amplitude decay behavior that aligns with the non-linear damping force characteristics.

- **Equilibrium State:** The long-term behavior of the oscillator should converge to a state of equilibrium, where the net force acting on the oscillator is zero, especially under damping conditions.
- **Physical Realism:** The model's outputs must remain within physically plausible ranges, considering the initial conditions and system parameters.

Table 3: Output Variables

Var	**Physical Constraints**
$x(t)$	Must satisfy boundary conditions
$v(t)$	Consistent with $x(t)$ and damping
$E(t)$	Decreases over time for damped cases

5 Requirements

[The requirements refine the goal statement. They will make heavy use of references to the instance models. —TPLT]

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: [Requirements for the inputs that are supplied by the user. This information has to be explicit. —TPLT]

R2: [It isn't always required, but often echoing the inputs as part of the output is a good idea. —TPLT]

R3: [Calculation related requirements. —TPLT]

R4: [Verification related requirements. —TPLT]

R5: [Output related requirements. —TPLT]

[Every IM should map to at least one requirement, but not every requirement has to map to a corresponding IM. —TPLT]

5.2 Nonfunctional Requirements

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —TPLT] [The goal is for the nonfunctional requirements to be unambiguous, abstract and verifiable. This isn't easy to show succinctly, so a good strategy may be to give a "high level" view of the requirement, but allow for the details to be covered in the Verification and Validation document. —TPLT] [An absolute requirement on a quality of the system is rarely needed. For instance, an accuracy of 0.0101 % is likely fine, even if the requirement is for 0.01 % accuracy. Therefore, the emphasis will often be more on describing now well the quality is achieved, through experimentation, and possibly theory, rather than meeting some bar that was defined a priori. —TPLT] [You do not need an entry for correctness in your NFRs. The purpose of the SRS is to record the requirements that need to be satisfied for correctness. Any statement of correctness would just be redundant. Rather than discuss correctness, you can characterize how far away from the correct (true) solution you are allowed to be. This is discussed under accuracy. —TPLT]

NFR1: **Accuracy** [Characterize the accuracy by giving the context/use for the software. Maybe something like, "The accuracy of the computed solutions should meet the level needed for <engineering or scientific application>. The level of accuracy achieved by Damped Harmonic Ocsillator shall be described following the procedure given in Section X of the Verification and Validation Plan." A link to the VnV plan would be a nice extra. —TPLT]

NFR2: **Usability** [Characterize the usability by giving the context/use for the software. You should likely reference the user characteristics section. The level of usability achieved by the software shall be described following the procedure given in Section X of the Verification and Validation Plan. A link to the VnV plan would be a nice extra. —TPLT]

NFR3: **Maintainability** [The effort required to make any of the likely changes listed for Damped Harmonic Ocsillator should be less than FRACTION of the original development time. FRACTION is then a symbolic constant that can be defined at the end of the report. —TPLT]

NFR4: **Portability** [This NFR is easier to write than the others. The systems that Damped Harmonic Ocsillator should run on should be listed here. When possible the specific versions of the potential operating environments should be given. To make the NFR verifiable a statement could be made that the tests from a given section of the VnV plan can be successfully run on all of the possible operating environments. —TPLT]

- Other NFRs that might be discussed include verifiability, understandability and reusability.

5.3 Rationale

[Provide a rationale for the decisions made in the documentation. Rationale should be provided for scope decisions, modelling decisions, assumptions and typical values. —TPLT]

6 Likely Changes

LC1: [Give the likely changes, with a reference to the related assumption (aref), as appropriate. —TPLT]

7 Unlikely Changes

LC2: [Give the unlikely changes. The design can assume that the changes listed will not occur. —TPLT]

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 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 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 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.

	TM??	TM??	TM??	GD3	GD??	DD5	DD??	DD??	DD??	IM2	IM??	IM??
TM??												
TM??			X									
TM??												
GD3												
GD??	X											
DD5				X								
DD??				X								
DD??												
DD??								X				
IM2					X	X	X				X	
IM??					X		X		X	X		
IM??		X										
IM??		X	X				X	X	X		X	

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM2	IM??	IM??	IM??	4.2.6	R??	R??
IM2		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R2	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 5: 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??																			
GD3		X																	
GD??			X	X	X	X													
DD5							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM2											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 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

9 Development Plan

[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]

[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

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

1. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

2. 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.

3. 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?