System Verification and Validation Plan for Double Pendulum

Zhi Zhang

November 12, 2019

1 Revision History

Date	Version	Notes
Oct.30	1.0	Initial Draft

Contents

List of Tables

List of Figures

2 Symbols, Abbreviations and Acronyms

symbol	description
VnV	Verification and Validation
SRS	Software Requirements Specification
FR	Functional Requirements
NFR	Nonfunctional Requirements

This document provides an overview of the Verification and Validation(VnV) plan for Double Pendulum. The general information is introduced in section General Information. SRS, design, implementation verification plan and software validation plan is introduced in section Plan. In section System Test Description, tests for both FR and NFR of Double Pendulum will be discussed, with the traceability between test cases and requirements.

3 General Information

3.1 Summary

The software to be tested is Double Pendulum. This software allows user to input the initial conditions of double pendulum, calculates the equations for the movements of the two pendulums, and then output the result to a text file.

3.2 Objectives

The objective of the VnV plan is to verify the FR and NFR, as found in the SRS titled Double Pendulum, has been met. The most important requirement is the correctness of the software, the goal is to build confidence in the software correctness by comparing the outputs of the software with the Double Pendulum tool from myPhysicsLab.com Neumann. Adequate usability also needs to be achieved since the software is designed for any user who is interested in the motion of a double pendulum.

3.3 Relevant Documentation

The relevant documentations are:

- SRS
- Unit VnV Plan

4 Plan

4.1 Verification and Validation Team

- Zhi Zhang

4.2 SRS Verification Plan

The SRS verification plan will include feedback from author's professor and classmates, mainly the domain expert.

4.3 Design Verification Plan

The design verification plan will include feedback from author's professor and classmates, mainly the domain expert.

[Plans for design verification —SS]

4.4 Implementation Verification Plan

The implementation verification plan will include the followings:

- Code walkthroughs: both the developer and the domain expert will inspect the code to ensure all the requirements of the SRS are met.
- Unit testing: all modules are to be unit tested to ensure correctness, more details of unit testing can be found in the Unit Testing Plan.
- Dynamic testing: the software will be executed and all its functions will be tested mannually by test team.

4.5 Software Validation Plan

N/A

5 System Test Description

5.1 Tests for Functional Requirements

There are five functional requirements described in section 5.1 of the SRS.

R1: Input the required values into the corresponding area.

R2: Check the entered input values to ensure that they do not exceed the data constraints.

R3: Calculate the equation for the following values: θ_1(t) and θ_1(t).

R4: Output the results to a file.

R5: Output graphs of θ_1(t) and θ_1(t).

Double Pendulum shall verify that the inputs are valid, shall guarantee the calculation is correct and the outputs are in the correct form.

Detailed test plan for the five functional requirements will be covered in the next five subsections.

5.1.1 Input Verification

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
m_1	$m_1 > 0$	-	kg	10%
m_2	$m_2 > 0$	-	kg	10%
L_1	$L_1 > 0$	-	m	10%
L_2	$L_2 > 0$	-	m	10%
θ_1	$\theta_1 \neq 0$	-	°	10%
θ_2	$\theta_2 \neq 0$	-	°	10%
g	$g > 0$	-	N/kg	10%

Table 1: Input Variables

The above table is the data constraints table from SRS. The test in this section covers requirements R.1 and R.2.

Invalid Inputs

1. Non-positive Mass

   Control: Manual

   Initial State: Double Pendulum is started and running

   Input: m_1 = -10 m_2 = 0 L_1 = 10 L_2 = 10 θ_1 = 10 θ_2 = 10 g = 9.8 Output: An error message of "Please input positive value for the mass".

   Test Case Derivation: Check if the error message pops up.

   How test will be performed: Manually input the above the Input data to the software by test team.

2. Non-positive Length of Rods

   Control: Manual

   Initial State: Double Pendulum is started and running

   Input: m_1 = 10 m_2 = 20 L_1 = -10 L_2 = 0 θ_1 = 10 θ_2 = 10 g = 9.8

   Output: An error message of "Please input positive value for the the length of rods".

   Test Case Derivation: Check if the error message pops up.

   How test will be performed: Manually input the above the Input data to the software by test team.

3. Zero Starting Angle Control: Manual

   Initial State: Double Pendulum is started and running

   Input: m_1 = 10 m_2 = 10 L_1 = 10 L_2 = 10 θ_1 = 0 θ_2 = 0 g = 9.8 Output: An error message of "Please input non-zero value for the starting angle".

   Test Case Derivation: Check if the error message pops up.

   How test will be performed: Manually input the above the Input data to the software by test team.

Valid Inputs

1. Valid Values

   Control: Manual

   Initial State: Double Pendulum is started and running

   Input: $m_1 = 10$ $m_2 = 10$ $L_1 = 10$ $L_2 = 10$ $\theta_1 = 10$ $\theta_2 = 10$ $g = 9.8$
   Output: The output file is generated and the graphs are displayed.

   Test Case Derivation: Check if there is output file and the graphs.

   How test will be performed: Manually input the above the Input data to the software by test team.

5.1.2 Outputs Verification

This section covers requirements R.3, R.4 and R.5.

Outputs Correctness Tests

1. File Created

   Type: Manual

   Initial State: Double Pendulum takes valid inputs

   Input/Condition: Press Start button

   Output/Result: An output file created

   How test will be performed: Test team will go to the directory of Double Pendulum to check the existence of the output file.

2. Graph Generated

   Type: Dynamic, Manual

   Initial State: Double Pendulum takes valid inputs and the output file is generated

   Input/Condition: Press Graph button

   Output/Result: Graphs of θ_1 and θ_2 displayed on the screen of the software

   How test will be performed: Test team will manually press the Graph button and check if the graph is displayed

3. Correct Output

   Type: Dynamic, Manual

   Initial State: The graphs are generated

   Input/Condition: Compare the graphs to the graphs generated on myPhysicsLab.com Neumann with the same inputs.

   Output/Result: Check if the graphs match.

   How test will be performed: Test team will manually input the same data into myPhysicsLab.com Neumann and compare the graphs.

5.2 Tests for Nonfunctional Requirements

There are five nonfunctional requirements described in section 6.1 of SRS.

NFR6: The outputs of the code have the properties described in table Output Variables.

NFR7: The code is tested with complete verification and validation plan.

NFR8: The code is modularized.

NFR9: The traceability between requirements, assumptions, theoretical models, general definitions, data definitions, instance models, likely changes, unlikely changes, and modules is completely recorded in traceability matrices in teh SRS and module guide.

NFR10: The code is able to be run in different environments.

Var	**Physical Constraints**
${\theta_1}''$	${\theta_1}'' \neq 0$
${\theta_2}''$	${\theta_2}'' \neq 0$

Table 2: Output Variables

5.2.1 Correctness and Verifiability

The correctness test covers the NFR 6, and the verifiability test covers the NFR 7. Both tests are to ensure that the software meets the SRS, and they are been tested in the section 5.1.2.

5.2.2 Maintainability

The maintainability test covers the NFR 9.

Maintainability Testing

Type: Manual, Nonfunctional, Dynamic
Initial State: N/A
Input/Condition: Existing Double Pendulum software
Output/Result: Update Double Pendulum software
How test will be performed: Test team will try to add new features to the software or modify some parts of the software.

5.2.3 Re-usability

The re-usability test covers the NFR 8.

Re-usability Testing

Type: Manual, Nonfunctional
Initial State: N/A
Input/Condition: Existing Double Pendulum system
Output/Result: N/A
How test will be performed: Test team will perform code walk through to ensure that all codes are modularized.

5.2.4 Portability

The portability test covers the NFR 10.

Portability Testing

1. Portability on Windows System Type: Manual, Dynamic

   Initial State: Double Pendulum has been successfully installed on a Windows system

   Input/Condition: Operate the basic functions of the software

   Output/Result: The software performs all functions

   How test will be performed: The test will be performed by test team manually.

2. Portability on MacOS system

   Type: Manual, Dynamic

   Initial State: Double Pendulum has been successfully installed on a MacOS system

   Input/Condition: Operate the basic functions of the software

   Output/Result: The software performs all functions

   How test will be performed: The test will be performed by test team manually.

5.3 Traceability Between Test Cases and Requirements

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
5.1.1	x	x								
5.1.2			x	x	x					
5.2.1				x	x	x	x			
5.2.2									x	
5.2.3								x		
5.2.4										x

Table 3: Traceability Between Test Cases and Requirements

References

Erik Neumann. Double pendulum. URL https://www.myphysicslab.com/pendulum/double-pendulum-en.html.