CAS 741: Unit Verification and Validation Report

Dynamical Systems: MPSim

Karol Serkis
serkiskj@mcmaster.ca
GitHub: karolserkis

December 19, 2018

1 Revision History

Date	Version	Notes
Dec. 18, 2018	1.0	First full draft for submission

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at:
https://github.com/karolserkis/CAS-741-Pendula/blob/master/docs/SRS/SRS.pdf
The symbols are listed in alphabetical order.

symbol	description
A	Assumption
DD	Data Definition
FT	Functional Test
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
MG	Module Guide
MIS	Module Interface Specification
NF	Non-Functional Requirement
R	Requirement
SRS	Software Requirements Specification
T	Test
VnV	Verification and Validation

Contents

List of Tables

List of Figures

This document is a review of the system tests that have been performed on MPSim . This report is a partner to the Unit VnV Report that is located in the docs/VnVReport/UnitVnVReport directory.

3 Functional Requirements Evaluation

FR1 was: MPSim program should be rendered with 3D Cartesian coordinates.
This requirement has been successfully met.

FR2 was: MPSim program will take the following inputs:

1. The number of pendulum rods in the simulation.
2. Toggle the ground plane to introduce friction to the pendulum rod initial moment of inertia
3. Toggle the plot of Potential & Kinetic Energy

This requirement has been successfully met.

FR3 was: MPSim program will calculate the kinetic and potential energy after the user has set the initialization parameters of input from the user have been entered.
This requirement has been successfully met.

FR4 was: MPSim program will calculate the Lagrangian after the user has set the initialization parameters of input have been entered and the kinetic and potential energy of the system as whole has been calculated. This requirement has been somewhat met. A generic use of OpenGL and GLUT has allowed for other ways around this.

FR5 was: MPSim program will ensure that the inputs do not violate the constraints specified in the Data Constraints section:

1. MPSim program will generate diagrams with and plot lines and time-line of logged movement.
2. The time-line of swings of the pendulum will be logged and eventually return to a resting state in equilibrium

This requirement has been somewhat successfully met.

4 Nonfunctional Requirements Evaluation

NFR1 and subsequent requirements for the software to be portable have been met.
The simulation software can run on any computer system and is contained in one Python file for convenience.

4.1 Usability

To make the program easy to use, the interaction with the GUI is minimal.
The user can provide a variety of input types, but because of exception handling the program will still run or provide feedback to let the user try the correct input parameters into the command-line.

4.2 Performance

The performance bottleneck is most likely the running of the program on a local computer. The simulation requires some relative graphics card for calculations. Performance otherwise is not a focus of the program.

5 Comparison to Existing Implementation

Comparing Existing Implementation resulted in very comparable results thus the simulation was quite capable and true to life.

6 Unit Testing

The unit testing will be covered in more details in the UnitVnVReport. The actual implementation of the unit tests can be found in the src/ or test/ directory on the github repo https://github.com/karolserkis/CAS-741-Pendula.

7 Changes Due to Testing

The actual Unit VnV Plan tests will be discussed henceforth. All the functional and most of the non-functional requirements have been meet.

8 Automated Testing

Unit-based scripts can be created for testing the modules in MPSim and be tested automatically with scripts written in pytest. The test scripts using pytest will cycle through an array of data and check an assert statement. See section on verification tools for more information.

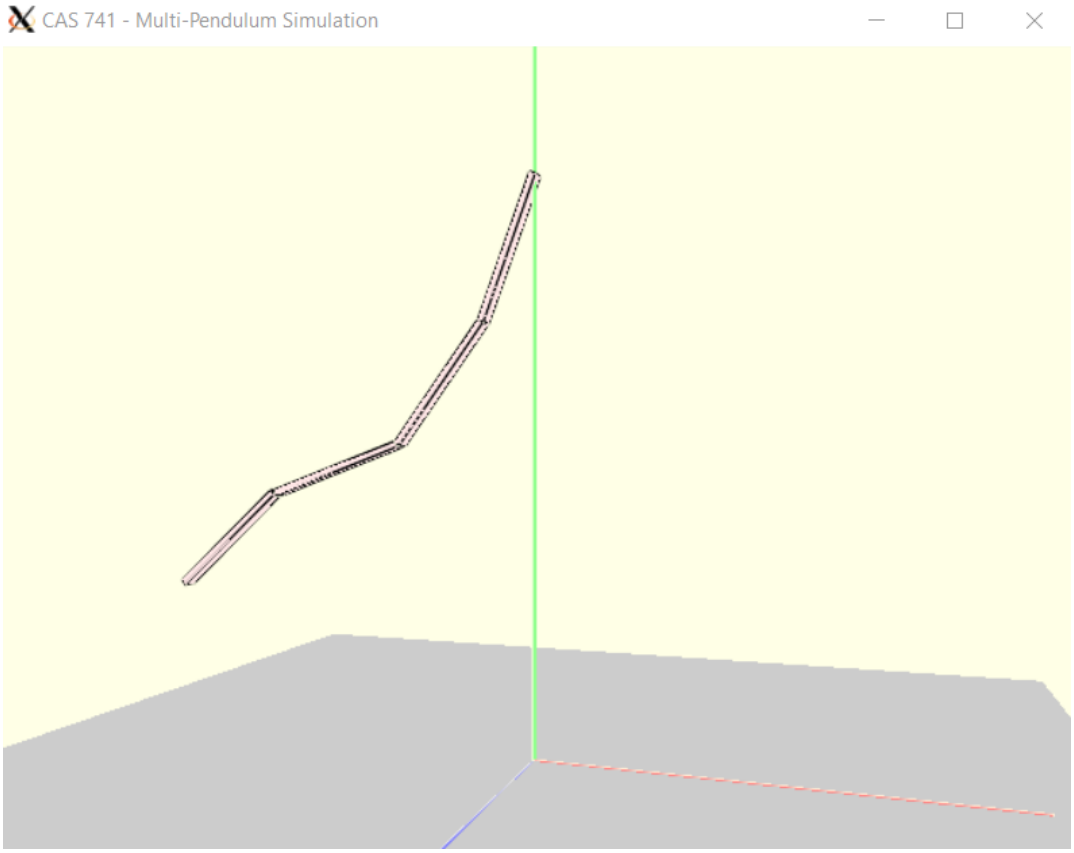

Figure 1: Kinetic and potential energy and simulation of multi-rod pendulum chain (**?**)

9 Trace to Requirements

All functional requirements met.

Req.	**Modules**
R1	MPSim.py
R2	MPSim.py
R3	MPSim.py
R4	MPSim.py
R5	MPSim.py, MPSim_inputargparse_test.py
NFR1	MPSim.py

Table 1: Trace between requirements and modules, with the specific python file names given (as found in the kaplan directory). A similar table is found in the module guide document.

10 Trace to Modules

Module	Test File	Comments
M1 (Hardware Hiding)	-	Out of scope
M2 (MPSim Control)	MPSim.py, MPSim_inputargparse_test.py	complete
M3 (Generic GUIl)	MPSim.py, MPSim_inputargparse_test.py	complete
M3 (User Input)	MPSim.py, MPSim_inputargparse_test.py	complete
M5 (Lagrangian Module)	MPSim.py, MPSim_inputargparse_test.py	complete
M6 (Data Structure)	MPSim.py	incomplete
M7 (Generic Plot)	MPSim.py	complete for current spec
M8 (Generic Trajectory)	MPSim.py	complete for current spec

Table 2: Trace between modules and tests, with the specific python file names given (as found in the kaplan/test directory), or (if no python test was written) a comment or reference to a jupyter notebook (located in the kaplan/test/jupyter-notebooks directory).

11 Code Coverage Metrics

Current code coverage metrics can be found in the test scripts using pytest will cycle through an array of data and check an assert statement. See section on verification tools for more information.

```
Name                                                    Cover
-------------------------------------------------------------------
src/MPSim.py                                            1/3 tests
src/MPSim_inputargparse_test.py                         7/8 tests
test/sim_inputargparse_test.py                          7/8 tests
-------------------------------------------------------------------
```

```
$ pytest-3
============= test session starts ==============================
platform linux -- Python 3.6.7, pytest-3.3.2, py-1.5.2, pluggy-0.6.0
rootdir: /mnt/c/Users/Karol/CAS-741-Pendula/src, inifile:
collected 8 items

MPSim_inputargparse_test.py .......x
[100%]

============= 7 passed, 1 xfailed in 2.19 seconds ==============
```