Test Report: PD Controller

Naveen Ganesh Muralidharan

December 14, 2020

1 Revision History

Date	Version	Notes
14-Dec-2020	1.0	First Draft of the VnV Report

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

All the units, abbreviations, and symbols recorded in the Software Requirement Specification Muralidharan (2020a) and the Verification and Validation Plan Muralidharan (2020b) apply to this document as well.

Contents

List of Tables

This document provides the summary of the Verification and Validation of the PD Controller software.

Sections 3 and 4 summarize the results of the functional and non-functional requirements respectively. Subsequent sections summarizes the test results in detail.

3 Functional Requirements Evaluation

In the VnV Plan Muralidharan (2020b) the test cases TC-PD-1, TC-PD-2, and TC-PD-14 test the functional requirements. The test results and the corresponding artifacts can be found here,

https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/functional_requirements

4 Nonfunctional Requirements Evaluation

4.1 Portability

The PD Controller software was successfully executed on both Windows and Linux Operating Systems. The test results and the corresponding artifacts can be found here,

https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/nonfunctional_requirements/portability

4.2 Maintenance

Modularity -

The PD Controller software, generated by the Drasil Carette and Smith (2020) software, was inspected, and it was found that the resultant code is modularized as follows,

- Calculations.py - Provides functions for calculating the outputs.
- Constants.py - Provides the structure for holding constant values.
- Control.py - Controls the flow of the program.

- InputParameters.py - Provides the function for reading inputs and the function for checking the physical constraints on the input.
- OutputFormat.py - Provides the function for writing outputs.

Linting - Linting check of the PD Controller software against the PEP-8 standards did not reveal any significant issues. Minor issues such a white-space before ':', or long line lengths were identified. These are negligible, and hence overall the Linting test was successful. The test results and the corresponding artifacts can be found here,
https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/nonfunctional_requirements/linting

Documented - The source code of the PD Controller software has been adequately captured in the Doxygen pdf file. There are few minor issues like missing names for a few functions but they are negligible.

The test results and the corresponding artifacts can be found here,
https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/nonfunctional_requirements/linting

4.3 Security

Memory Leak Check - Valgrind analysis of the PD Controller software indicated that there is about 576 Bytes confirmed leak, and 148 MB possible leak. However this data is inconclusive, as the leak maybe in Python itself.

The test results and the corresponding artifacts can be found here,
https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/nonfunctional_requirements/memory_leak

Negative square root check - The source code of the PD Controller software was inspected, and there are no square root calls in the program.

Divide by Zero error - The source code of the PD Controller software was inspected, and it was confirmed that the divide by zero error is averted by the checks for the input constraints.

Table 1

Module1	Function1	Module2	Function2
Control	main	InputParameters	get_input
Control	main	Calculations	func_y_t
Control	main	OutputFormatter	write_output

Table 2

Module1	Function1	Module2	Function2
Control	main	InputParameters	input_constraints

4.4 Verifiability

Tracing - All the functional and non functional requirements have at-least one test case in the VnV plan and report.

Statement Coverage - Automated testing proved that 100% statement coverage has been achieved,

The artifacts of the Statement Coverage analysis are found here,
https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/nonfunctional_requirements/code_coverage

Data Coupling - Inspection of the source code revealed that the functions listed in Table-1 are coupled by data flow,

The log file generated by the source code was inspected, and the DataFlow between the functions was confirmed.

Control Coupling - Inspection of the source code revealed that the functions listed in Table-2 are coupled by control flow,

The log file generated by the source code was inspected, and the control flow in the function was confirmed.

The artifacts of the DCCC analysis are found here,
https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/nonfunctional_requirements/DCCC

5 Comparison to Existing Implementation

Not applicable to the PD Controller software.

6 Unit Testing

The Unit test case, TC-PD-14, passed successfully. The results are found here,
https://github.com/muralidn/CAS741-Fall20/tree/master/test/results/functional_requirements

7 Changes Due to Testing

The following bugs were caught during verification and were addressed,

- The maximum physical constraint of the Step Time (t_{step}) was updated to always be less than the Simulation Time (t_{sim}). This way the User is free to choose the step size.

8 Trace to Requirements

See section 5.3 of the VnV plan Muralidharan (2020b).

9 Trace to Modules

See section 6.3 of the VnV plan Muralidharan (2020b).

10 Code Coverage Metrics

The statement coverage metric is summarized in Table-3.

References

Jacques Carette and Spencer Smith. Drasil, 2020. URL https://jacquescarette.github.io/Drasil/.

Table 3

Module	Statement Coverage
Calculations.py	100%
Constants.py	100%
Control.py	100%
InputParameters.py	100%
OutputFormat.py	100%

Naveen Ganesh Muralidharan. System requirements specification for pd controller. https://github.com/muralidn/CAS741-Fall20/blob/master/docs/SRS/SRS.pdf, 2020a.

Naveen Ganesh Muralidharan. Verification and validation plan for the pd controller. https://github.com/muralidn/CAS741-Fall20/blob/master/docs/VnVPlan/VnVPlan.pdf, 2020b.