Software Requirements Specification for PD Controller

Naveen Ganesh Muralidharan

December 14, 2020

Contents

1 Reference Material **3**
- 1.1 Table of Units . . . 3
- 1.2 Table of Symbols . . . 3
- 1.3 Abbreviations and Acronyms . . . 4

2 Introduction **4**
- 2.1 Purpose of Document . . . 5
- 2.2 Scope of Requirements . . . 5
- 2.3 Characteristics of Intended Reader . . . 5
- 2.4 Organization of Document . . . 5

3 General System Description **5**
- 3.1 System Context . . . 5
- 3.2 User Characteristics . . . 6
- 3.3 System Constraints . . . 6

4 Specific System Description **6**
- 4.1 Problem Description . . . 7
  - 4.1.1 Terminology and Definitions . . . 7
  - 4.1.2 Physical System Description . . . 8
  - 4.1.3 Goal Statements . . . 8
- 4.2 Solution Characteristics Specification . . . 9
  - 4.2.1 Assumptions . . . 9
  - 4.2.2 Theoretical Models . . . 10
  - 4.2.3 General Definitions . . . 12
  - 4.2.4 Data Definitions . . . 13
  - 4.2.5 Instance Models . . . 18
  - 4.2.6 Data Constraints . . . 20

5 Requirements **21**
5.1 Functional Requirements 21
5.2 Non-Functional Requirements 21

6 Likely Changes **22**

7 Traceability Matrices and Graphs **22**

8 References **24**

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

The unit system used throughout is SI (Système International d'Unités). In addition to the basic units, several derived units are also used. For each unit, Tab: ToU lists the symbol, a description and the SI name.

Symbol	Description	SI Name
s	time	second

Table 1: Table of Units

1.2 Table of Symbols

The symbols used in this document are summarized in Tab: ToS along with their units. The symbols are listed in alphabetical order.

Symbol	Description	Units
$-\infty$	Negative Infinity	–
$AbsTol$	Absolute Tolerance	–
C_{s}	Control Variable in the frequency domain	–
c_{t}	Control Variable in the time domain	–
D_{s}	Derivative control in the frequency domain	–
E_{s}	Process Error in the frequency domain	–
e_{t}	Process Error in the time domain	–
F_{s}	Laplace Transform of a function	–
f_{t}	Function in the time domain	–
H_{s}	Transfer Function in the frequency domain	–
K_{d}	Derivative Gain	–
K_{DC}	DC gain of the First Order system.	–
K_{p}	Proportional Gain	–
$L^{-1}[F(s)]$	Inverse Laplace Transform of a function	–
P_{s}	Proportional control in the frequency domain	–
R_{s}	Set Point in the frequency domain	–
r_{t}	Set Point	–
$RelTol$	Relative Tolerance	–
s	Complex frequency-domain parameter	–
t	Time	s
t_{sim}	Simulation Time	s

Symbol	Description	Units
t_{step}	Step Time	s
Y_{s}	Process Variable in the frequency domain	–
y_{t}	Process Variable	–
∞	Infinity	–
$\boldsymbol{\tau}$	Time Constant of the First Order system.	s

Table 2: Table of Symbols

1.3 Abbreviations and Acronyms

Abbreviation	Full Form
A	Assumption
D	Derivative
DD	Data Definition
GD	General Definition
GS	Goal Statement
I	Integral
IM	Instance Model
P	Proportional
PD	Proportional Derivative
PID	Proportional Integral Derivative
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model
Uncert.	Typical Uncertainty

Table 3: Abbreviations and Acronyms

2 Introduction

Automatic process control with a controller (P/PI/PD/PID) is used in a variety of applications such as thermostats, automobile cruise-control, etc. The gains of a controller in an application must be tuned before the controller is ready for production. Therefore a simulation of the PD Controller with a First Order System is created in this project that can be used to tune the gain constants.

The following section provides an overview of the Software Requirements Specification (SRS) for PD Controller. This section explains the purpose of this document, the scope of the requirements, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

The purpose of this document is to capture all the necessary information including assumptions, data definitions, constraints, models, and requirements to facilitate an unambiguous development of the PD controller software and test procedures.

2.2 Scope of Requirements

The scope of the requirements includes a PD Control Loop with three subsystems namely a PD Controller , a Summing Point , and a Power Plant . Only the Proportional and Derivative controllers are used here; the Integral controller is beyond the scope of this software. Additionally, this software is intended to aid with the manual tuning of the PD Controller.

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have an understanding of control systems (control theory and controllers) at a fourth year undergraduate level and engineering mathematics at a second year undergraduate level. The users of PD Controller can have a lower level of expertise, as explained in Section: User Characteristics.

2.4 Organization of Document

The sections in this document are based on [5]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the data definitions in Section: Instance Models and trace back to find any additional information they require.

The goal statements (Section: Goal Statements) are refined to the theoretical models and the theoretical models (Section: Theoretical Models) to the instance models (Section: Instance Models). The instance model referred as IM: pdEquationIM provides an Ordinary Differential Equation (ODE) that models the PD Controller.

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

Fig:systemContextDiag shows the system context. The circle represents an external entity outside the software, the user in this case. The rectangle represents the software system

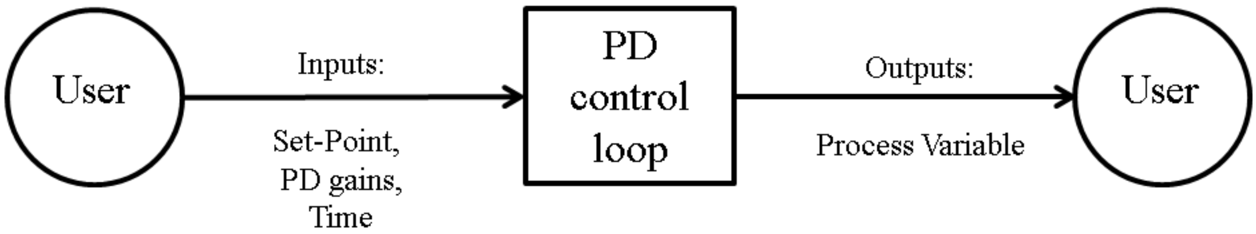

Figure 1: System Context

itself, PD Controller in this case. Arrows are used to show the data flow between the system and its environment.

PD Controller is self-contained. The only external interaction is with the user. The responsibilities of the user and the system are as follows:

- User Responsibilities
  - Feed inputs to the model
  - Review the response of the Power Plant
  - Tune the controller gains
- PD Controller Responsibilities
  - Check the validity of the inputs
  - Calculate the outputs of the PD Controller and Power Plant

3.2 User Characteristics

The end user of PD Controller is expected to have taken a course on Control Systems at an undergraduate level.

3.3 System Constraints

There are no system constraints.

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, and definitions that are used.

4.1 Problem Description

A system is needed to provide a model of a PD Controller that can be used for the tuning of the gain constants before the deployment of the controller.

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.

- PD Control Loop: Closed loop control system with PD Controller, Summing Point and Power Plant.
- PD Controller: Proportional-Derivative Controller.
- Summing Point: Control block where the difference between the Set-Point and the Process Variable is computed.
- Power Plant: A first order system to be controlled.
- First Order System: A system whose input-output relationship is denoted by a first order differential equation.
- Process Error: Input to the PID controller. Process Error is the difference between the Set Point and the Process Variable.
- Simulation Time: Total execution time of the PD simulation.
- Process Variable: The output value from the power plant.
- Set Point: The desired value that the control system must reach. This also knows as reference variable.
- Proportional Gain: Gain constant of the proportional controller.
- Derivative Gain: Gain constant of the derivative controller.
- Frequency Domain: The analysis of mathematical functions with respect to frequency, instead of time.
- Laplace transform: An integral transform that converts a function of a real variable t (often time) to a function of a complex variable s (complex frequency).
- Control Variable: The Control Variable is the output of the PD controller.
- Step Time: Simulation step time.
- Absolute Tolerance: Absolute tolerance for the integrator.

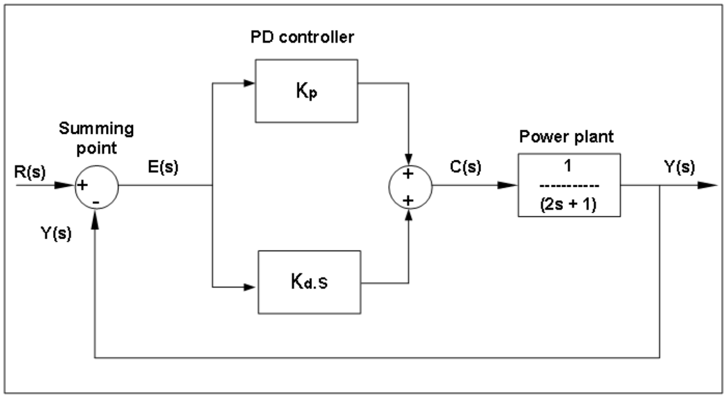

Figure 2: The physical system

- Relative Tolerance: Relative tolerance for the integrator.
- DC Gain: DC Gain is the ratio of the steady state output to the input signal.
- Time Constant: Time Constant is a measure of the response of a First Order System to a step input.
- Transfer Function: Transfer Function of a system is the ratio of the output to the inputfunctions in the Frequency Domain.

4.1.2 Physical System Description

The physical system of PD Controller, as shown in Fig:pidSysDiagram, includes the following elements:

PS1: The Summing Point.

PS2: The PD Controller.

PS3: The Power Plant.

4.1.3 Goal Statements

Given Set Point, Simulation Time, Proportional Gain, Derivative Gain, and Step Time, the goal statements are:

Process-Variable: Calculate the output of the Power Plant (Process Variable) over time.

4.2 Solution Characteristics Specification

The instance models that govern PD Controller are presented in Section: Instance Models. 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 Assumptions

This section simplifies the original problem and helps in developing the theoretical models by filling in the missing information for the physical system. The assumptions refine the scope by providing more detail.

Power plant: The Power Plant and the Sensor are coupled as a single unit. (RefBy: A: Transfer Function and A: Power Plant Parameters.)

ecoupled equation: The decoupled form of the PD Controller equation used in this simulation. (RefBy: DD: ddCtrlVar.)

Set-Point: The Set Point is a constant throughout the simulation. (RefBy: DD: ddProcessError and IM: pdEquationIM.)

ternal disturbance: There are no external disturbances to the Power Plant during the simulation. (RefBy: GD: gdPowerPlant.)

Initial Value: The initial value of the Process Variable is assumed to be zero. (RefBy: DD: ddProcessError.)

Parallel Equation: The Parallel form of the equation is used for the PD Controller. (RefBy: DD: ddCtrlVar.)

filtered Derivative: A pure derivative function is used for this simulation; there are no filters applied. (RefBy: DD: ddDerivCtrl.)

Transfer Function: The combined Power Plant and Sensor (from A: Power plant) are characterized by a First Order System. (RefBy: TM: tmFOSystem, LC: Second Order Power Plant, and A: Power Plant Parameters.)

Plant Parameters: The DC Gain and Time Constant of the First Order System (from A: Power plant and A: Transfer Function) are assumed to be 1, and 2 seconds respectively. There are no physical signifiance to the values chosen. (RefBy: LC: DC Gain and Time Constant and GD: gdPowerPlant.)

4.2.2 Theoretical Models

This section focuses on the general equations and laws that PD Controller is based on.

Refname	**TM:laplaceTransform**
Label	Laplace Transform
Equation	$$F_{\text{s}} = \int_{-\infty}^{\infty} f_{\text{t}} e^{-st} \, dt$$
Description	F_{s} is the Laplace Transform of a function (Unitless) f_{t} is the Function in the time domain (Unitless) s is the Complex frequency-domain parameter (Unitless) t is the time (s)
Notes	Bilateral Laplace Transform. The Laplace transforms are typically inferred from a pre-computed table of Laplace Transforms ([2]).
Source	[2]
RefBy	DD: ddPropCtrl, DD: ddProcessError, DD: ddDerivCtrl, and GD: gdPowerPlant

Refname	**TM:invLaplaceTransform**
Label	Inverse Laplace Transform
Equation	$$f_{\mathrm{t}} = L^{-1}[F(s)]$$
Description	f_{t} is the Function in the time domain (Unitless) $L^{-1}[F(s)]$ is the Inverse Laplace Transform of a function (Unitless)
Notes	Inverse Laplace Transform of F(S). The Inverse Laplace transforms are typically inferred from a pre-computed table of Laplace Transforms ([2]).
Source	[2]
RefBy	IM: pdEquationIM

Refname	TM:tmFOSystem
Label	First Order System
Equation	$$K_{\text{DC}}\frac{1}{\boldsymbol{\tau}s+1}$$
Description	K_{DC} is the DC gain of the First Order system. (Unitless) $\boldsymbol{\tau}$ is the Time Constant of the First Order system. (s) s is the Complex frequency-domain parameter (Unitless)
Notes	The Transfer Function of a First Order System is characterised by this equation (from A: Transfer Function).
Source	[6]
RefBy	GD: gdPowerPlant

4.2.3 General Definitions

This section collects the laws and equations that will be used to build the instance models.

Refname	GD:gdPowerPlant
Label	Transfer function of the Power Plant.
Equation	$$\frac{1}{2s+1}$$
Description	s is the Complex frequency-domain parameter (Unitless)
Notes	The Transfer Function of the First Order System is reduced to this equation by substituting the DC Gain (K_{DC}) to 1, and the Time Constant ($\boldsymbol{\tau}$) to 2 seconds (from TM: tmFOSystem and A: Power Plant Parameters). The equation is converted to frequency domain by applying the Laplace transform (from TM: laplaceTransform). Additionally, there are no external disturbances to the power plant (from A: External disturbance).
Source	[3] and [6]
RefBy	IM: pdEquationIM

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models.

Refname	**DD:ddProcessError**
Label	Process Error in the frequency domain
Symbol	E_{s}
Units	Unitless
Equation	$$E_{\text{s}} = R_{\text{s}} - Y_{\text{s}}$$
Description	E_{s} is the Process Error in the frequency domain (Unitless) R_{s} is the Set Point in the frequency domain (Unitless) Y_{s} is the Process Variable in the frequency domain (Unitless)
Notes	Process Error is the difference between the Set-Point and Process Variable. The equation is converted to frequency domain by applying the Laplace transform (from TM: laplaceTransform). The Set-Point is assumed to be constant throughout the simulation (from A: Set-Point). The initial value of the Process Variable if assumed to be zero (from A: Initial Value).
Source	[4]
RefBy	DD: ddPropCtrl, DD: ddDerivCtrl, and IM: pdEquationIM

Refname	DD:ddPropCtrl
Label	Proportional control in the frequency domain
Symbol	P_{s}
Units	Unitless
Equation	$$P_{\text{s}} = K_{\text{p}} \cdot E_{\text{s}}$$
Description	P_{s} is the Proportional control in the frequency domain (Unitless) K_{p} is the Proportional Gain (Unitless) E_{s} is the Process Error in the frequency domain (Unitless)
Notes	Proportional controller is the product of the Proportional Gain and the Process Error (from DD: ddProcessError) The equation is converted to frequency domain by applying the Laplace transform (from TM: laplaceTransform).
Source	[4]
RefBy	DD: ddCtrlVar

Refname	**DD:ddDerivCtrl**
Label	Derivative control in the frequency domain
Symbol	D_{s}
Units	Unitless
Equation	$$D_{\mathrm{s}} = K_{\mathrm{d}} \cdot E_{\mathrm{s}} \cdot s$$
Description	D_{s} is the Derivative control in the frequency domain (Unitless) K_{d} is the Derivative Gain (Unitless) E_{s} is the Process Error in the frequency domain (Unitless) s is the Complex frequency-domain parameter (Unitless)
Notes	Derivative controller is the product of the Derivative Gain and the differential of the Process Error (from DD: ddProcessError) The equation is converted to frequency domain by applying the Laplace transform (from TM: laplaceTransform) A pure form of the Derivative controller is used in this application (from A: Unfiltered Derivative).
Source	[4]
RefBy	DD: ddCtrlVar

Refname	DD:ddCtrlVar
Label	Control Variable in the frequency domain
Symbol	C_{s}
Units	Unitless
Equation	$$C_{\text{s}}=K_{\text{p}}\cdot E_{\text{s}}+K_{\text{d}}\cdot E_{\text{s}}\cdot s$$
Description	C_{s} is the Control Variable in the frequency domain (Unitless) K_{p} is the Proportional Gain (Unitless) E_{s} is the Process Error in the frequency domain (Unitless) K_{d} is the Derivative Gain (Unitless) s is the Complex frequency-domain parameter (Unitless)
Notes	The control variable is the output of the controller. In this case, it is the sum of the Proportional (from DD: ddPropCtrl) and Derivative (from DD: ddDerivCtrl) controllers The parallel (from A: Parallel Equation) and de-coupled (from A: Decoupled equation) form of the PD equation is used in this document.
Source	[4]
RefBy	IM: pdEquationIM

4.2.5 Instance Models

This section transforms the problem defined in Section: Problem Description into one which is expressed in mathematical terms. It uses concrete symbols defined in Section: Data Definitions to replace the abstract symbols in the models identified in Section: Theoretical Models and Section: General Definitions.

Refname	IM:pdEquationIM
Label	Computation of the Process Variable as a function of time.
Input	r_{t}, K_{p}, K_{d}
Output	y_{t}
Input Constraints	$r_{\mathrm{t}} > 0$ $K_{\mathrm{p}} > 0$ $K_{\mathrm{d}} > 0$
Output Constraints	$y_{\mathrm{t}} > 0$
Equation	$(2 + K_{\mathrm{d}}) \frac{dy_{\mathrm{t}}}{dt} + \left(1 + K_{\mathrm{p}}\right) y_{\mathrm{t}} - r_{\mathrm{t}} K_{\mathrm{p}} = 0$
Description	K_{d} is the Derivative Gain (Unitless) t is the time (s) y_{t} is the Process Variable (Unitless) K_{p} is the Proportional Gain (Unitless) r_{t} is the Set Point (Unitless)
Source	[1] and [4]
RefBy	FR: Output-Values and FR: Calculate-Values

Detailed derivation of Process Variable: The Process Variable (Y(S)) in a PD Control Loop is the product of the Process Error (from DD: ddProcessError), Control Variable (from DD: ddCtrlVar), and the Power-Plant (from GD: gdPowerPlant).

$$Y_{\text{s}} = (R_{\text{s}} - Y_{\text{s}}) \left(K_{\text{p}} + K_{\text{d}} s\right) \frac{1}{2s+1}$$

Rearranging the equation.

$$(2 + K_{\text{d}})\, Y_{\text{s}} s + \left(1 + K_{\text{p}}\right) Y_{\text{s}} - R_{\text{s}} s K_{\text{d}} - R_{\text{s}} K_{\text{p}} = 0$$

Computing the Inverse Laplace Transform of a function (from TM: invLaplaceTransform) of the equation.

$$(2 + K_{\text{d}}) \frac{dy_{\text{t}}}{dt} + \left(1 + K_{\text{p}}\right) y_{\text{t}} - K_{\text{d}} \frac{dr_{\text{t}}}{dt} - r_{\text{t}} K_{\text{p}} = 0$$

The Set point (r(t)) is a step function, and a constant (from A: Set-Point). Therefore the differential of the set point is zero. Hence the equation reduces to,

$$(2 + K_{\text{d}}) \frac{dy_{\text{t}}}{dt} + \left(1 + K_{\text{p}}\right) y_{\text{t}} - r_{\text{t}} K_{\text{p}} = 0$$

4.2.6 Data Constraints

Table:InDataConstraints shows the data constraints on the input variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. 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 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.

Var	Physical Constraints	Typical Value	Uncert.
K_{d}	$K_{\text{d}} \geq 0$	1.0	10%
K_{p}	$K_{\text{p}} > 0$	20.0	10%
r_{t}	$r_{\text{t}} > 0$	1.0	10%
t_{sim}	$1 \leq t_{\text{sim}} \leq 60$	10.0 s	10%
t_{step}	$\frac{1}{100} \leq t_{\text{step}} < t_{\text{sim}}$	0.01 s	10%

Table 4: Input Data Constraints

5 Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete, and the non-functional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete.

Input-Values: Input the values from Table:ReqInputs.

Verify-Input-Values: Ensure that the input values are within the limits specified in Section: Data Constraints.

Calculate-Values: Calculate the Process Variable (from IM: pdEquationIM) over the simulation time.

Output-Values: Output the Process Variable (from IM: pdEquationIM) over the simulation time.

Symbol	Description	Units
K_{d}	Derivative Gain	–
K_{p}	Proportional Gain	–
r_{t}	Set Point	–
t_{sim}	Simulation Time	s
t_{step}	Step Time	s

Table 5: Required Inputs following FR: Input-Values

5.2 Non-Functional Requirements

This section provides the non-functional requirements, the qualities that the software is expected to exhibit.

Portable: The code shall be portable to multiple Operating Systems.

Secure: The code shall be immune to common security problems such as memory leaks, divide by zero errors, and the square root of negative numbers.

Maintainable: The dependencies among the instance models, requirements, likely changes, assumptions and all other relevant sections of this document shall be traceable to each other in the trace matrix.

Verifiable: The code shall be verifiable against a Verification and Validation plan.

6 Likely Changes

This section lists the likely changes to be made to the software.

Order Power Plant: The Power Plant maybe changed into a second order system (from A: Transfer Function).

nd Time Constant: The DC Gain and Time Constant maybe changed to be supplied by the User (from A: Power Plant Parameters).

7 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" should be modified as well. Table:TraceMatAvsA shows the dependencies of assumptions on the assumptions. Table:TraceMatAvsAll shows the dependencies of data definitions, theoretical models, general definitions, instance models, requirements, likely changes, and unlikely changes on the assumptions. Table:TraceMatRefvsRef shows the dependencies of data definitions, theoretical models, general definitions, and instance models with each other. Table:TraceMatAllvsR shows the dependencies of requirements, goal statements on the data definitions, theoretical models, general definitions, and instance models.

	A: Power plant	**A: Decoupled equation**	**A: Set-Point**	**A: Exter**
A: Power plant				
A: Decoupled equation				
A: Set-Point				
A: External disturbance				
A: Initial Value				
A: Parallel Equation				
A: Unfiltered Derivative				
A: Transfer Function	X			
A: Power Plant Parameters	X			

Table 6: Traceability each other.

	A: Power plant	**A: Decoupled equation**	**A: Set-Point**	**A:**
DD: ddProcessError			X	
DD: ddPropCtrl				
DD: ddDerivCtrl				

	A: Power plant	A: Decoupled equation	A: Set-Point	A:
DD: ddCtrlVar		X		
TM: laplaceTransform				
TM: invLaplaceTransform				
TM: tmFOSystem				
GD: gdPowerPlant				X
IM: pdEquationIM			X	
FR: Input-Values				
FR: Verify-Input-Values				
FR: Calculate-Values				
FR: Output-Values				
NFR: Portable				
NFR: Secure				
NFR: Maintainable				
NFR: Verifiable				
LC: Second Order Power Plant				
LC: DC Gain and Time Constant				

Table 7: Traceab
Items

	DD: ddProcessError	DD: ddPropCtrl	DD: ddDerivCtrl	DD: d
DD: ddProcessError				
DD: ddPropCtrl	X			
DD: ddDerivCtrl	X			
DD: ddCtrlVar		X	X	
TM: laplaceTransform				
TM: invLaplaceTransform				
TM: tmFOSystem				
GD: gdPowerPlant				
IM: pdEquationIM	X			X

Table 8: Traceability Matr

	DD: ddProcessError	DD: ddPropCtrl	DD: ddDerivCtrl	DD: dd
GS: Process-Variable				
FR: Input-Values				
FR: Verify-Input-Values				
FR: Calculate-Values				
FR: Output-Values				

	DD: ddProcessError	DD: ddPropCtrl	DD: ddDerivCtrl	DD: dd
NFR: Portable				
NFR: Secure				
NFR: Maintainable				
NFR: Verifiable				

8 References

[1] Nasser M. Abbasi. *A differential equation view of closed loop control systems*. https://www.12000.org/my_notes/connecting_systems/report.htm. Nov. 2020.

[2] Wikipedia Contributors. *Laplace transform*. https://en.wikipedia.org/wiki/Laplace_transform. Nov. 2020.

[3] Wikipedia Contributors. *PID controller*. https://en.wikipedia.org/wiki/PID_controller. Oct. 2020.

[4] Michael A. Johnson and Mohammad H. Moradi. *PID Control: New Identification and Design Methods, Chapter 1*. Springer Science and Business Media, 2006.

[5] W. Spencer Smith and Lei Lai. "A new requirements template for scientific computing". In: *Proceedings of the First International Workshop on Situational Requirements Engineering Processes - Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*. Ed. by PJ Agerfalk, N. Kraiem, and J. Ralyte. In conjunction with 13th IEEE International Requirements Engineering Conference, Paris, France, 2005, pp. 107–121.

[6] Electical4U team. *First Order Systems*. https://www.electrical4u.com/first-order-control-system. Nov. 2020.