Software Requirements Specification for PD Controller

Naveen Ganesh Muralidharan

June 3, 2023

Contents

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, the Table of Units lists the symbol, a description and the SI name.

Symbol	Description	SI Name
kg	mass	kilogram
s	time	second

Table 1: Table of Units

1.2 Table of Symbols

The symbols used in this document are summarized in the Table of Symbols along with their units. The symbols are listed in alphabetical order.

Symbol	Description	Units
$AbsTol$	Absolute Tolerance	–
C_{s}	Control Variable in the frequency domain	–
c	Damping coefficient of the spring	–
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_{p}	Proportional Gain	–
k	Stiffness coefficient of the spring	s
$L^{-1}[F(s)]$	Inverse Laplace Transform of a function	–
m	Mass	kg
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	–

Symbol	Description	Units
t	Time	s
t_{sim}	Simulation Time	s
t_{step}	Step Time	s
Y_{s}	Process Variable in the frequency domain	–
y_{t}	Process Variable	–

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
RefBy	Referenced by
Refname	Reference Name
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 Second 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 in this software; the Integral controller is beyond the scope of this project. 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 the 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 Sec:User Characteristics.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [5], [7], [8], and [6]. 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 and trace back to find any additional information they require.

The goal statements are refined to the theoretical models and the theoretical models to the 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.

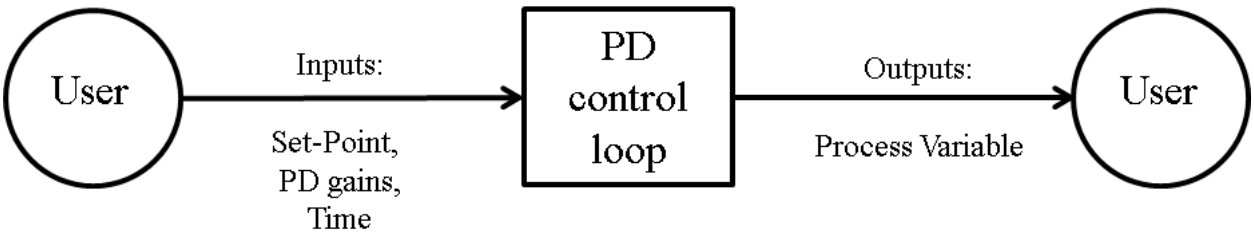

Figure 1: System Context

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 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 second order system to be controlled.
- Second Order System: A system whose input-output relationship is denoted by a second-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 the reference variable.
- Proportional Gain: Gain constant of the proportional controller.
- Derivative Gain: Gain constant of the derivative controller.
- Proportional control: A linear feedback control system where correction is applied to the controlled variable which is proportional to the difference between desired and measured values.
- Derivative control: Monitors the rate of change of the error signal and contributes a component of the output signal (proportional to a derivative of the error signal).
- Frequency domain: The analysis of mathematical functions in terms of frequency, instead of time.

- Time domain: The analysis of mathematical functions in terms 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.
- Relative Tolerance: Relative tolerance for the integrator.
- Transfer Function: The Transfer Function of a system is the ratio of the output to the input functions in the frequency domain.
- Damping Coefficient: Quantity that characterizes a second order system's oscillatory response.
- Stiffness Coefficient: Quantity that characterizes a spring's stiffness.

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 statement is:

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 the Instance Model Section. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

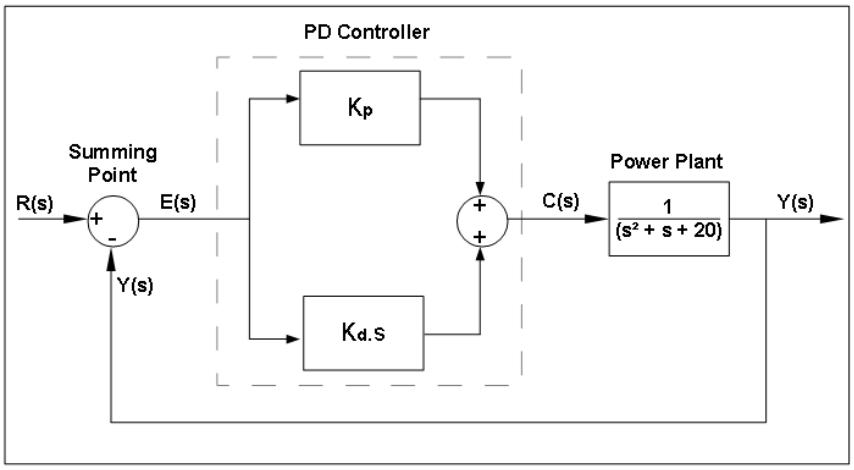

Figure 2: The physical system

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:Spring Stiffness Coefficient, A:Transfer Function, A:Spring Mass, and A:Spring Damping Coefficient.)

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

Set-Point: The Set-Point is 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.)

iltered 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 (A:Power plant) are characterized by a Second Order mass-spring-damper System. (RefBy: TM:tmSOSystem.)

Spring Mass: The mass of the spring in the mass-spring-damper system (A:Power plant) is assumed to be 1 kilogram. (RefBy: LC:DC Gain and Time Constant and GD:gdPowerPlant.)

amping Coefficient: The Damping Coefficient of the spring in the mass-spring-damper system (A:Power plant) is assumed to be 1. (RefBy: LC:DC Gain and Time Constant and GD:gdPowerPlant.)

tiffness Coefficient: The Stiffness Coefficient of the spring in the mass-spring-damper system (A:Power plant) is assumed to be 20. (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	$$F_{\mathrm{s}} = \int_{-\infty}^{\infty} f_{\mathrm{t}} e^{-st}\, dt$$
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	$$f_t = L^{-1}[F(s)]$$
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:tmSOSystem**
Label	Second Order Mass-Spring-Damper System
Equation	$$\frac{1}{ms^2 + cs + k}$$
Description	m is the mass (kg) s is the Complex frequency-domain parameter (Unitless) c is the Damping coefficient of the spring (Unitless) k is the Stiffness coefficient of the spring (s)
Notes	The Transfer Function (from A:Transfer Function) of a Second Order System (mass-spring-damper) is characterized by this equation.
Source	[1]
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	The Transfer Function of the Power Plant
Equation	$$\frac{1}{s^2 + s + 20}$$
Description	s is the Complex frequency-domain parameter (Unitless)
Notes	The Transfer Function of the Second Order System (from TM:tmSOSystem) is reduced to this equation by substituting the mass (m) to 1 Kg (from A:Spring Mass), the Damping Coefficient (c) to 1 (from A:Spring Damping Coefficient), and the Stiffness Coefficient (k) to 20 (from A:Spring Stiffness Coefficient). The equation is converted to the 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 [1]
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	$$E_s = R_s - Y_s$$
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	The Process Error is the difference between the Set-Point and Process Variable. The equation is converted to the 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 is 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	$$P_s = K_p E_s$$
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	The Proportional Controller is the product of the Proportional Gain and the Process Error (from DD:ddProcessError). The equation is converted to the 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	$$D_s = K_d E_s s$$
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	The Derivative Controller is the product of the Derivative Gain and the differential of the Process Error (from DD:ddProcessError). The equation is converted to the 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	$$C_s = E_s \left(K_p + K_d s\right)$$
Description	C_s is the Control Variable in the frequency domain (Unitless) E_s is the Process Error in the frequency domain (Unitless) K_p is the Proportional Gain (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 the problem description into one which is expressed in mathematical terms. It uses concrete symbols defined in the data definitions to replace the abstract symbols in the models identified in theoretical models and 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	$r_t > 0$ $K_p > 0$ $K_d > 0$
Output Constraints	$y_t > 0$
Equation	$\frac{d^2 y_t}{dt^2} + (1 + K_d)\frac{dy_t}{dt} + (20 + K_p)\, y_t = r_t K_p$
Description	t is the time (s) y_t is the Process Variable (Unitless) K_d is the Derivative Gain (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).

$$Y_s = (R_s - Y_s)\left(K_p + K_d s\right)\frac{1}{s^2 + s + 20}$$

Substituting the values and rearranging the equation.

$$s^2 Y_s + (1 + K_d) Y_s s + (20 + K_p) Y_s - R_s s K_d - R_s K_p = 0$$

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

$$\frac{d\frac{dy_t}{dt}}{dt} + (1 + K_d) \frac{dy_t}{dt} + (20 + K_p) y_t - K_d \frac{dr_t}{dt} - r_t K_p = 0$$

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

$$\frac{d\frac{dy_t}{dt}}{dt} + (1 + K_d) \frac{dy_t}{dt} + (20 + K_p) y_t - r_t K_p = 0$$

4.2.6 Data Constraints

The Data Constraints Table 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	$K_d \geq 0$	1	10%
K_p	$K_p > 0$	20	10%
r_t	$r_t > 0$	1	10%
t_{sim}	$1 \leq t_{sim} \leq 60$	10 s	10%
t_{step}	$\frac{1}{1000} \leq t_{step} < t_{sim}$	0.001 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 Tab:ReqInputs.

erify-Input-Values: Ensure that the input values are within the limits specified in the 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.

nd Time Constant: The mass, Damping Coefficient and the Stiffness Coefficient may be changed to be supplied by the user (from A:Spring Mass, A:Spring Damping Coefficient, and A:Spring Stiffness Coefficient).

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. Tab:TraceMatAvsA shows the dependencies of the assumptions on each other. Tab:TraceMatAvsAll shows the dependencies of the data definitions, theoretical models, general definitions, instance models, requirements, likely changes, and unlikely changes on the assumptions. Tab:TraceMatRefvsRef shows the dependencies of the data definitions, theoretical models, general definitions, and instance models on each other. Tab:TraceMatAllvsR shows the dependencies of the requirements and 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:Spring Mass	X			
A:Spring Damping Coefficient	X			
A:Spring Stiffness Coefficient	X			

	A:Power plant	**A:Decoupled equation**	**A:Set-Point**	**A:Ext**
DD:ddProcessError			X	
DD:ddPropCtrl				
DD:ddDerivCtrl				
DD:ddCtrlVar		X		
TM:laplaceTransform				
TM:invLaplaceTransform				
TM:tmSOSystem				
GD:gdPowerPlant				X
IM:pdEquationIM			X	
FR:Input-Values				
FR:Verify-Input-Values				

	A:Power plant	A:Decoupled equation	A:Set-Point	A:Ext
FR:Calculate-Values				
FR:Output-Values				
NFR:Portable				
NFR:Secure				
NFR:Maintainable				
NFR:Verifiable				
LC:DC Gain and Time Constant				

	DD:ddProcessError	DD:ddPropCtrl	DD:ddDerivCtrl	DD:ddCt
DD:ddProcessError				
DD:ddPropCtrl	X			
DD:ddDerivCtrl	X			
DD:ddCtrlVar		X	X	
TM:laplaceTransform				
TM:invLaplaceTransform				
TM:tmSOSystem				
GD:gdPowerPlant				
IM:pdEquationIM	X			X

Table 8: Traceability Matrix S

	DD:ddProcessError	DD:ddPropCtrl	DD:ddDerivCtrl	DD:ddCtrl
GS:Process-Variable				
FR:Input-Values				
FR:Verify-Input-Values				
FR:Calculate-Values				
FR:Output-Values				
NFR:Portable				
NFR:Secure				
NFR:Maintainable				
NFR:Verifiable				

The purpose of the traceability graphs is also to provide easy references on what has

Figure 3: TraceGraphAvsA

Figure 4: TraceGraphAvsAll

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. Fig:TraceGraphAvsA shows the dependencies of assumptions on each other. Fig:TraceGraphAvsAll shows the dependencies of data definitions, theoretical models, general definitions, instance models, requirements, likely changes, and unlikely changes on the assumptions. Fig:TraceGraphRefvsRef shows the dependencies of data definitions, theoretical models, general definitions, and instance models on each other. Fig:TraceGraphAllvsR shows the dependencies of requirements and goal statements on the data definitions, theoretical models, general definitions, and instance models. Fig:TraceGraphAllvsAll shows the dependencies of dependencies of assumptions, models, definitions, requirements, goals, and changes with each other.

For convenience, the following graphs can be found at the links below:

- TraceGraphAvsA
- TraceGraphAvsAll
- TraceGraphRefvsRef
- TraceGraphAllvsR
- TraceGraphAllvsAll

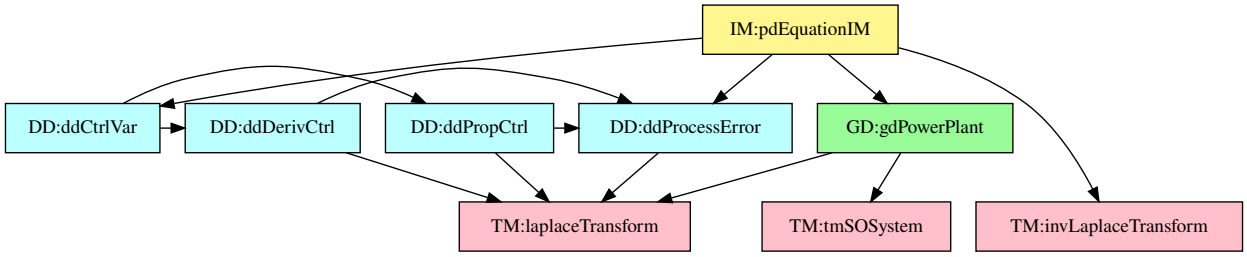

Figure 5: TraceGraphRefvsRef

Figure 6: TraceGraphAllvsR

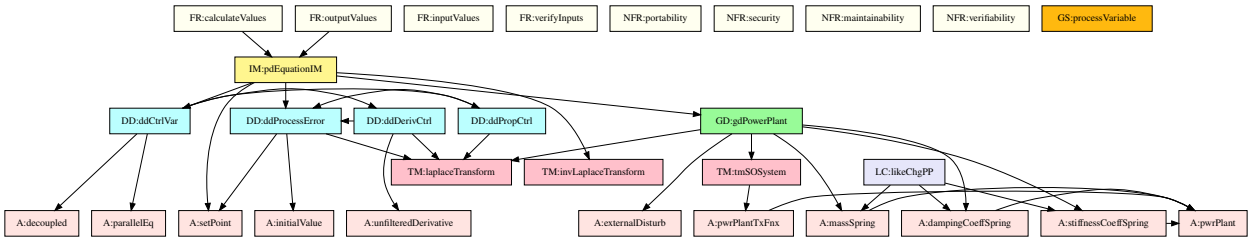

Figure 7: TraceGraphAllvsAll

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] Nirmitha Koothoor. "A Document Driven Approach to Certifying Scientific Computing Software". MA thesis. Hamilton, ON, Canada: McMaster University, 2013.

[6] W. Spencer Smith and Nirmitha Koothoor. "A Document-Driven Method for Certifying Scientific Computing Software for Use in Nuclear Safety Analysis". In: *Nuclear Engineering and Technology* 48.2 (Apr. 2016), pp. 404–418.

[7] 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.

[8] W. Spencer Smith, Lei Lai, and Ridha Khedri. "Requirements Analysis for Engineering Computation: A Systematic Approach for Improving Software Reliability". In: *Reliable Computing, Special Issue on Reliable Engineering Computation* 13.1 (Feb. 2007), pp. 83–107.