

Software Requirements Specification for PD Controller

Naveen Muralidharan

December 2, 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	6
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	8
4.2.1	Assumptions	8
4.2.2	Theoretical Models	9
4.2.3	General Definitions	11
4.2.4	Data Definitions	11
4.2.5	Instance Models	16
4.2.6	Data Constraints	19

5	Requirements	20
5.1	Functional Requirements	20
5.2	Non-Functional Requirements	20
6	Likely Changes	21
7	Traceability Matrices and Graphs	21
8	References	23

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	—
A_{tol}	Absolute tolerance	—
C_s	Control-Variable in frequency domain	—
c_t	Control-Variable in time domain	—
D_s	Derivative Control in frequency domain	—
E_s	Error Signal in the frequency domain	—
e_t	Error Signal in the time domain	—
F_s	Laplace Transform of a function $f(t)$	—
f_t	Function in the time domain	—
H_s	Transfer function of the Power-Plant in frequency domain	—
K_p	Proportional Gain	—
$L^{-1}F(s)$	Inverse Laplace Transform of $F(s)$	—
P_s	Proportional Control in frequency domain	—
R_s	Set Point in the frequency domain	—
R_{tol}	Relative tolerance	—
r_t	Set-Point	—
s	Complex frequency domain parameter	—
t	Time	s

Capitalization seems inconsistent in this table. (This may be a consequence of the Ordo code. I haven't looked into it)

Symbol	Description	Units
t_{sim}	Simulation time	s
t_{step}	Step time	s
Y_s	Process Variable in the frequency domain	—
y_t	Process Variable	—
∞	Infinity	—

Table 2: Table of Symbols

1.3 Abbreviations and Acronyms

Abbreviation	Full Form
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
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. This software is intended to aid with the manual tuning of the PD Controller.

Monitor that integral is being left out.

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 [smithLai2005]. 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](#)).

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 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:

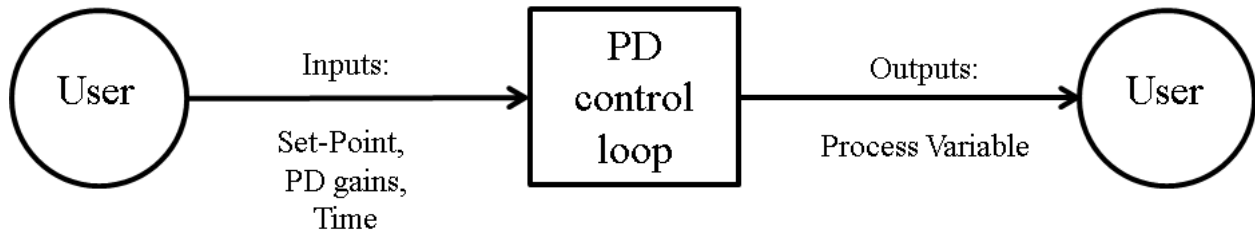


Figure 1: System Context

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

Is the user or the system responsible for checking the validity of the inputs?

3.2 User Characteristics

The end user of PD Controller is expected to have completed at least the equivalent of the second year of an undergraduate degree in electrical engineering.

3.3 System Constraints

There are no system constraints.

↳ Isn't possible to get the knowledge as a chemical engineer, or software engineer? Maybe focus on the likely title of the required course.

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 This program intends 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.
- Error Value: Input to the PID controller. Error Value 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.

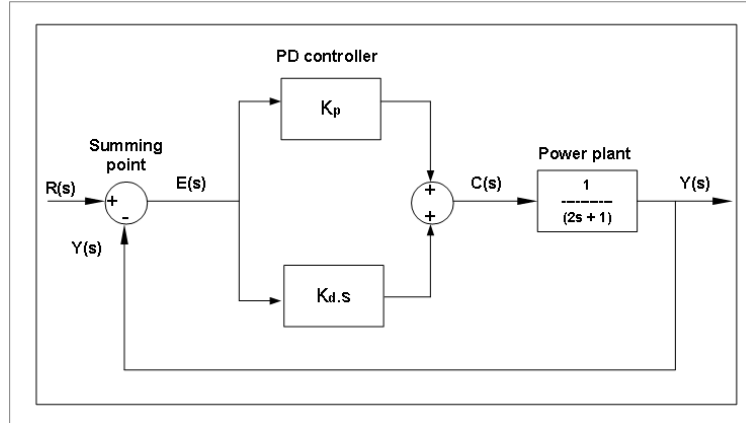


Figure 2: The physical system

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 a single unit with transfer function, $1/(2s + 1)$. (RefBy: **DD: ddPowerPlant** and **LC: Second Order Power Plant**.)
- Decoupled 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: ddErrorSignal** and **IM: pdEquationIM**.)
- External disturbance: There are no external disturbances to the Power Plant during the simulation. (RefBy: **DD: ddPowerPlant**.)
- Initial value: The initial value of the Process Variable is assumed to be zero. (RefBy: **DD: ddErrorSignal**.)
- Parallel equation: The Parallel form of the equation is used for the PD Controller. (RefBy: **DD: ddCtrlVar**.)

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	$F_s = \int_{-\infty}^{\infty} f_t e^{-st} dt$
Description	<p> F_s is the Laplace Transform of a function $f(t)$ (Unitless) f_t is the Function in the time domain (Unitless) s is the Complex frequency domain parameter (Unitless) t is the time (s) </p>
Notes	<p> Bilateral Laplace Transform. The Laplace transforms are usually inferred from the Laplace Transform table in section 4 of the [2]. </p>
Source	[2]
RefBy	DD: ddPowerPlant , DD: ddPropCtrl , DD: ddErrorSignal , and DD: ddDerivCtrl


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 F(s) (Unitless)
Notes	Inverse Laplace Transform of F(S). The Inverse Laplace transforms are usually inferred from the Laplace Transform table in section 4 of the [2].
Source	[2]
RefBy	IM: pdEquationIM

4.2.3 General Definitions

There are no general definitions.

4.2.4 Data Definitions

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

Refname	DD:ddErrorSignal
Label	Error Signal in the frequency domain
Symbol	E_s
Units	Unitless
Equation	$E_s = R_s - Y_s$
Description	<p>E_s is the Error Signal in the frequency domain (Unitless)</p> <p>R_s is the Set Point in the frequency domain (Unitless)</p> <p>Y_s is the Process Variable in the frequency domain (Unitless)</p>
Notes	<p>Error Signal 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 is assumed to be zero (from A: Initial value) </p>
Source	[4]
RefBy	DD: ddPropCtrl, DD: ddDerivCtrl, and IM: pdEquationIM



Refname	DD:ddPropCtrl
Label	Proportional Control in frequency domain
Symbol	P_s
Units	Unitless
Equation	$P_s = K_d \cdot E_s$
Description	<p>P_s is the Proportional Control in frequency domain (Unitless)</p> <p>K_d is the Proportional Gain (Unitless)</p> <p>E_s is the Error Signal in the frequency domain (Unitless)</p>
Notes	Proportional controller is the product of the Proportional Gain and the Error Signal (from DD: ddErrorSignal) 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 frequency domain
Symbol	D_s
Units	Unitless
Equation	$D_s = K_d \cdot E_s \cdot s$
Description	<p>D_s is the Derivative Control in frequency domain (Unitless)</p> <p>K_d is the Proportional Gain (Unitless)</p> <p>E_s is the Error Signal in the frequency domain (Unitless)</p> <p>s is the Complex frequency domain parameter (Unitless)</p>
Notes	Derivative controller is the product of the Derivative Gain and the differential of the Error Signal (from DD: ddErrorSignal) The equation is converted to frequency domain by applying the Laplace transform (from TM: laplaceTransform).
Source	[4]
RefBy	DD: ddCtrlVar


Refname	DD:ddPowerPlant
Label	Transfer function of the Power-Plant in frequency domain
Symbol	H_s
Units	Unitless
Equation	$H_s = \frac{1}{2s + 1}$
Description	<p>H_s is the Transfer function of the Power-Plant in frequency domain (Unitless)</p> <p>s is the Complex frequency domain parameter (Unitless)</p>
Notes	<p>The power plant is represented by a first order system (from A: Power plant) 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).</p> <p><i>Is there a physical meaning associated with this transfer function?</i></p>
Source	[3]
RefBy	IM: pdEquationIM

Refname	DD:ddCtrlVar
Label	Control-Variable in frequency domain
Symbol	C_s
Units	Unitless
Equation	$C_s = K_d \cdot E_s + K_d \cdot E_s \cdot s$
Description	<p> C_s is the Control-Variable in frequency domain (Unitless) K_d is the Proportional Gain (Unitless) E_s is the Error Signal in the frequency domain (Unitless) s is the Complex frequency domain parameter (Unitless) </p>
Notes	<p>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.</p>
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	$r_t > 0$ $K_d > 0$ $K_d > 0$		
Output Constraints	$y_t > 0$		
Equation	$(2 + K_d) \frac{dy_t}{dt} + (1 + K_d) y_t - r_t K_d = 0$		
Description	K_d is the Proportional Gain (Unitless) t is the time (s) y_t is the Process Variable (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 Error signal (from **DD: ddErrorSignal**), Control Variable (from **DD: ddCtrlVar**), and the Power-Plant (from **DD: ddPowerPlant**).

$$Y_s = (R_s - Y_s) (K_d + K_d s) \frac{1}{2s + 1}$$

Rearranging the equation.

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

Computing the Inverse Laplace Transform of $F(s)$ (from **TM: invLaplaceTransform**) of the equation.

$$(2 + K_d) \frac{dy_t}{dt} + (1 + K_d) y_t - K_d \frac{dr_t}{dt} - r_t K_d = 0$$

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,

$$(2 + K_d) \frac{dy_t}{dt} + (1 + K_d) y_t - r_t K_d = 0$$

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	$K_d > 0$	1.0	10%
K_d	$K_d > 0$	100.0	10%
r_t	$r_t > 0$	1.0	10%
t_{sim}	$1 \leq t_{sim} \leq 60$	10.0 s	10%
t_{step}	$\frac{1}{100} \leq t_{step} \leq 1$	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
A_{tol}	Absolute tolerance	—
K_d	Proportional Gain	—
K_d	Proportional Gain	—
R_{tol}	Relative tolerance	—
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 code shall be thoroughly documented with appropriate User Guides.

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

This is how you intend to achieve maintainability—it isn't actually a req. for maintainability. Maybe you can say that the time for redevelopment is a FACTOR of the original development time, where you set a reasonable value for FACTOR

Quality: The code shall be written with high-quality standards. The code should adhere to good coding standards and should not contain any dead, or unreachable statements. *Same comment as above*
this isn't a software standard NFK.

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: Power plant**).

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: External disturbance
A: Power plant				
A: Decoupled equation				
A: Set-Point				
A: External disturbance				
A: Initial value				
A: Parallel equation				

Table 6: Traceability Matrix Showing the Connections Between Assumptions and each other.

	A: Power plant	A: Decoupled equation	A: Set-Point	A: External disturbance
DD: ddErrorSignal			X	
DD: ddPropCtrl				
DD: ddDerivCtrl				
DD: ddPowerPlant	X			X
DD: ddCtrlVar		X		
TM: laplaceTransform				

	A: Power plant	A: Decoupled equation	A: Set-Point	A: Error
TM: invLaplaceTransform				
IM: pdEquationIM			X	
FR: Input-Values				
FR: Verify-Input-Values				
FR: Calculate-Values				
FR: Output-Values				
NFR: Portable				
NFR: Secure				
NFR: Maintainable				
NFR: Verifiable				
NFR: Quality				
LC: Second Order Power Plant	X			

Table 7: Traceability Matrix Showing the Connections Between Items

	DD: ddErrorSignal	DD: ddPropCtrl	DD: ddDerivCtrl	DD: ddPowerPlant
DD: ddErrorSignal				
DD: ddPropCtrl	X			
DD: ddDerivCtrl	X			
DD: ddPowerPlant				
DD: ddCtrlVar		X	X	
TM: laplaceTransform				
TM: invLaplaceTransform				
IM: pdEquationIM	X			X

Table 8: Traceability Matrix Showing the Connections Between Items

	DD: ddErrorSignal	DD: ddPropCtrl	DD: ddDerivCtrl	DD: ddPowerPlant
GS: Process-Variable				
FR: Input-Values				
FR: Verify-Input-Values				
FR: Calculate-Values				
FR: Output-Values				
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.