Software Requirements Specification for PID Controller: Simulation of a PID control loop.

Naveen Ganesh Muralidharan

October 7, 2020

Contents

Revision History

Date	Version	Notes
28-Sep-2020	1.0	First draft of the SRS

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
s	time	second

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
$e(t)$	N/A	Error Signal at time t
K_d	N/A	Derivative Gain
K_i	N/A	Integral Gain
K_p	N/A	Proportional Gain
T_{sim}	s	Total Simulation Time
t_{step}	s	Simulation step time
$c(t)$	N/A	Control Variable at time t
$y(t)$	N/A	Process Variable at time t

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PID	Proportional Integral Derivative
PID Controller	Proportional Integral Derivative Controller
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
T	Theoretical Model

2 Introduction

A closed loop control system with a PID controller is used in a wide range of applications such as a thermostat, or cruise-control of an automobile. The gains of a PID controller in an application have to be tuned before the controller is ready for production. Therefore, a software model is necessary for the simulation of PID controller. This model can be used with the simulation of the Power Plant to calibrate the gains of the PID controller.

In this section the purpose of the document is discussed, followed by the scope of the requirements, characteristics of intended reader, and finally 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, data constrains, theory and instance models, and the requirements to provide the reader with a comprehensive understanding of the PID controller, and to facilitate an unambiguous development of the software and test procedures for the software model of a PID controller.

2.2 Scope of Requirements

The scope of the requirements is limited to the simulation of a closed control loop control system with three subsystems, namely, a summer, a PID controller and a power plant.

2.3 Characteristics of Intended Reader

The reader of this document is assumed to have a prerequisite knowledge of control theory at a second year undergraduate level, and high school calculus.

2.4 Organization of Document

The sections in this document is are based on Smith and Lai (2005); Smith et al. (2007). Subsequent sections of this document will breakdown and present the problem in a less abstract manner than the previous section. For example, the context for the mathematical equations required to solve the problem is set in the theoretical models, and are refined in general statements and data definitions, and the final form of the equations are derived in the instance models.

The suggested order of reading this document for a reader who is not familiar with the domain would be to start with the problem description, review the goals, and proceed downwards to the subsequent sections. Reader with knowledge of the domain experts can review any section of this document and and choose proceed in either direction.

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

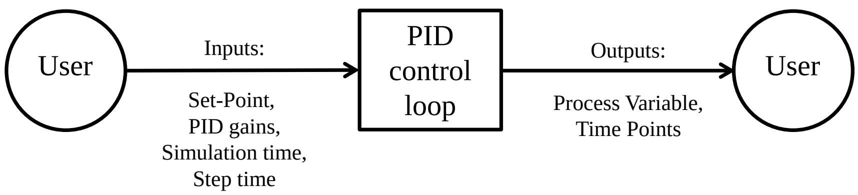


Figure 1: System Context

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

3.2 User Characteristics

The end user of PID Controller should have an understanding of undergraduate Level 1 Control Engineering and High school Calculus.

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, definitions and finally the instance models.

4.1 Problem Description

This program intends to provide a model of a PID 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:

Control Variable: The output from the PID controller.

Error Value: Input to the PID controller. Error Value is the difference between the Set Point and the Process Variable.

Power Plant: The system to be controlled.

Process Variable: The output value from the power plant.

Set-Point: The desired value that the control system much reach.

Simulation Time: Total execution time of the PID loop.

Step Time: Optimal time for one iteration of the control loop.

Summing Point: Control block where the difference between the Set-Point and the Process Variable is computed.

4.1.2 Physical System Description

The physical system of PID Controller, as shown in Figure-1, includes the following elements:

PS1: Summing Point.

PS2: PID Controller.

PS3: Power Plant.

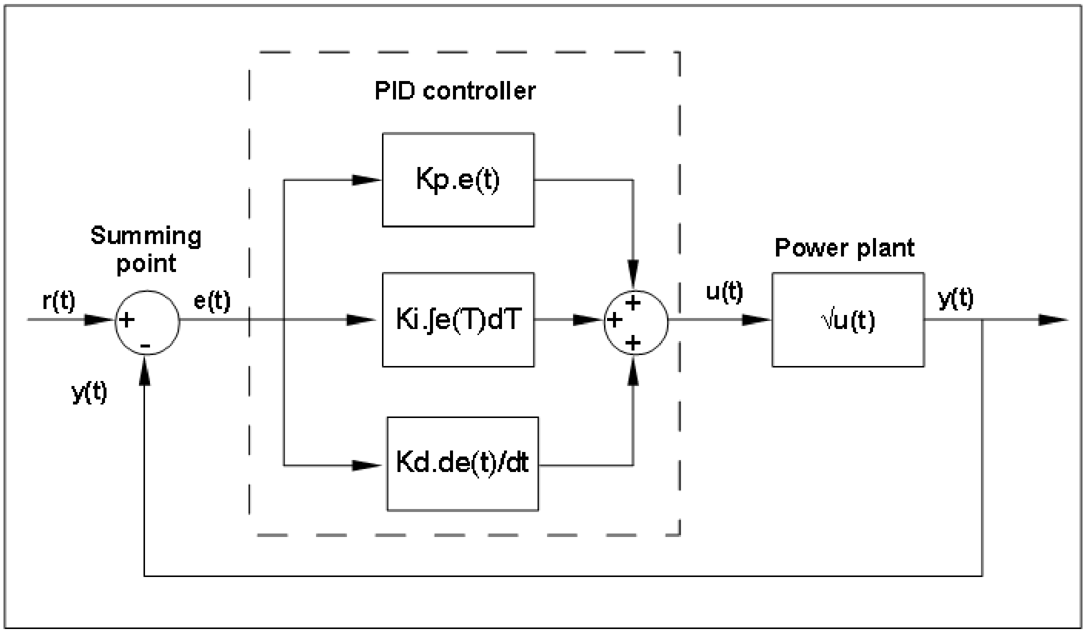


Figure 2: PID Control Loop

4.1.3 Goal Statements

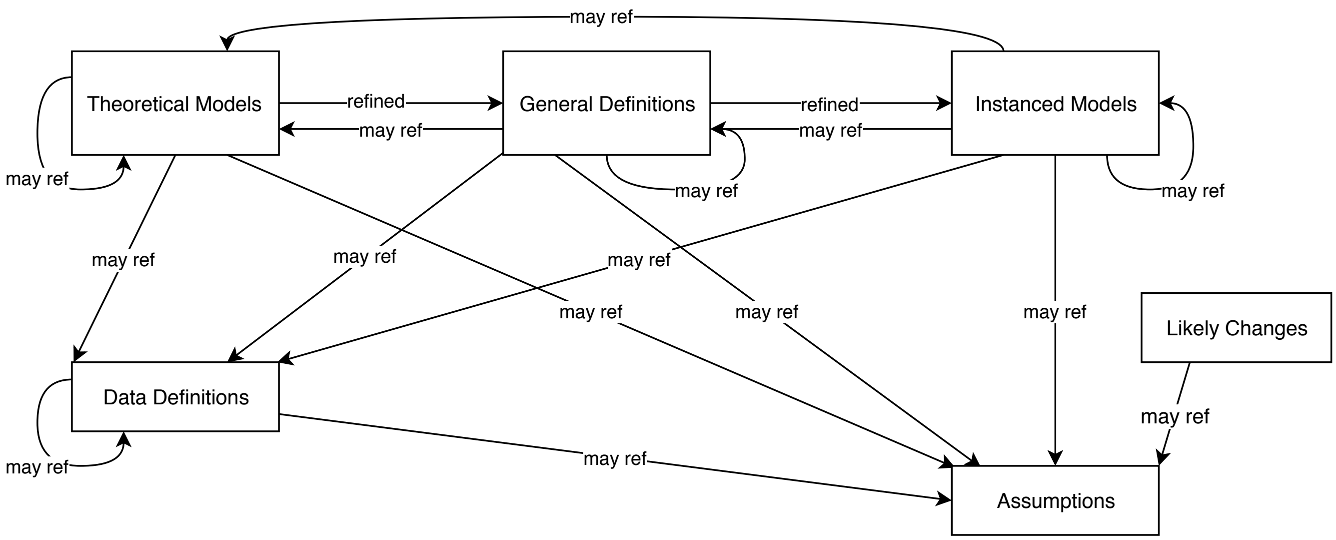
Given the Set-Point, Proportional Gain, Integral Gain, Derivative Gain, Total Simulation time and Step Time, the goal statements are:

GS1: Calculate the Error-Value at every step.

GS2: Compute the output of the PID-controller at every step.

GS3: Compute the output of the Power-Plant at every step.

4.2 Solution Characteristics Specification



The instance models that govern PID Controller are presented in Subsection 4.2.5. 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 model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: The Power Plant and the Sensor are coupled into a single system, with a response of $f(x) = \sqrt{x}$.

A2: The Set-Point does not vary during the simulation.

A3: There are no external disturbances during the simulation.

A4: There are no delays in the signal propagation.

A5: All analysis are in the time domain.

A6: This model will be used for Manual tuning of the contol gains.

A7: There is no residual error, aka. inital value.

A8: There is no Dead Time in the response of the Power Plant.

A9: There is no Dead Band in the response of the controller.

4.2.2 Theoretical Models

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

Number	T1
Label	**Calculus - Definite Integrals**
Equation	$\int_a^b f(x)dx$
Description	Real valued function f(x) integrated over closed limits [a,b].
Source	https://en.wikipedia.org/wiki/Integral
Ref. By	GD**??**

Number	T2
Label	**Calculus - Differentiation**
Equation	$\frac{dy}{dx}$
Description	Rate of change of y with respect to x
Source	https://en.wikipedia.org/wiki/Derivative
Ref. By	GD**??**

4.2.3 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Integration over time**
SI Units	$\mathrm{W\,m^{-2}}$
Equation	$q(t) = h\Delta T(t)$
Description	Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings. $q(t)$ is the thermal flux ($\mathrm{W\,m^{-2}}$). h is the heat transfer coefficient, assumed independent of T (A**??**) ($\mathrm{W\,m^{-2}\,°C^{-1}}$). $\Delta T(t) = T(t) - T_{\text{env}}(t)$ is the time-dependent thermal gradient between the environment and the object (°C).
Source	Citation here
Ref. By	DD1, DD**??**

Detailed derivation of simplified rate of change of temperature

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Heat flux out of coil**
Symbol	q_C
SI Units	$\mathrm{W\,m^{-2}}$
Equation	$q_C(t) = h_C(T_C - T_W(t))$, over area A_C
Description	T_C is the temperature of the coil (°C). T_W is the temperature of the water (°C). The heat flux out of the coil, q_C ($\mathrm{W\,m^{-2}}$), is found by assuming that Newton's Law of Cooling applies (A**??**). This law (GD1) is used on the surface of the coil, which has area A_C ($\mathrm{m^2}$) and heat transfer coefficient h_C ($\mathrm{W\,m^{-2}\,°C^{-1}}$). This equation assumes that the temperature of the coil is constant over time (A**??**) and that it does not vary along the length of the coil (A**??**).
Sources	Citation here
Ref. By	IM1

4.2.5 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goals are solved by .

Number	IM1
Label	**Energy balance on water to find T_W**
Input	m_W, C_W, h_C, A_C, h_P, A_P, t_{final}, T_C, T_{init}, $T_P(t)$ from IM**??** The input is constrained so that $T_{\text{init}} \leq T_C$ (A**??**)
Output	$T_W(t)$, $0 \leq t \leq t_{\text{final}}$, such that $\frac{dT_W}{dt} = \frac{1}{\tau_W}[(T_C - T_W(t)) + \eta(T_P(t) - T_W(t))]$, $T_W(0) = T_P(0) = T_{\text{init}}$ (A**??**) and $T_P(t)$ from IM**??**
Description	T_W is the water temperature (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^o$C, where 0^oC and 100^oC are the melting and boiling points of water, respectively (A**??**, A**??**).
Sources	Citation here
Ref. By	IM**??**

Derivation of ...

4.2.6 Input Data Constraints

Table 1 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. 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. 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 specification parameters in Table 1 are listed in Table 2.

(*)

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\text{min}} \leq L \leq L_{\text{max}}$	1.5 m	10%

Table 2: Specification Parameter Values

Var	Value
L_{min}	0.1 m

4.2.7 Properties of a Correct Solution

A correct solution must exhibit .

Table 3: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A**??**)

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1:

R2:

R3:

R4:

R5:

5.2 Nonfunctional Requirements

6 Likely Changes

LC1:

7 Unlikely Changes

LC2:

8 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" may have to be modified as well. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

The purpose of the traceability graphs is also to provide easy references on what has 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. Figure **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure **??** shows the dependencies of instance models, requirements, and data constraints on each other.

9 Values of Auxiliary Constants

	T2	T??	T??	GD1	GD??	DD1	DD??	DD??	DD??	IM1	IM??	IM??	IM??
T2													
T??			X										
T??													
GD1													
GD??	X												
DD1				X									
DD??				X									
DD??													
DD??								X					
IM1					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM??	IM??	IM??	4.2.6	R??	R??
IM1		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R2	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 5: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T2	X																		
T??																			
T??																			
GD1		X																	
GD??			X	X	X	X													
DD1							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM1											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

References

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ȧgerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.