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

Naveen Ganesh Muralidharan

October 12, 2020

Contents

Revision History

Date	Version	Notes
09-Oct-2020	1.0	First draft of the SRS
12-Oct-2020	1.1	Addressed the following issues, • Elaborated Intended reader and user characteristics. • Fixed Physical system figure reference. • Added reference to assumptions A4 and A7. • Fixed assumption A8. • Added explanation for system context. • Added cross reference in assumptions for readability.

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
$e(t)$	Mathematical value (dimensionless)	Error Signal at time t
K_d	Constant (dimensionless)	Derivative Gain
K_i	Constant (dimensionless)	Integral Gain
K_p	Constant (dimensionless)	Proportional Gain
SP	Mathematical value (dimensionless)	Set Point
T_{sim}	s	Total Simulation Time
T_{elapsed}	s	Elapsed time
T_{now}	s	Time point now
T_{prev}	s	Previous time point
t_{step}	s	Simulation step time
$u(t)$	Mathematical value (dimensionless)	Control Variable at time t
$y(t)$	Mathematical value (dimensionless)	Process Variable at time t

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
API	Application Program Interface
CV	Control Variable
D	Derivative term of the PID controller
DD	Data Definition
etc	Et cetra
GD	General Definition
GS	Goal Statement
I	Integral term of the PID controller
IM	Instance Model
LC	Likely Change
P	Proportional term of the PID controller
PID	Proportional Integral Derivative
PID Controller	Proportional Integral Derivative Controller
PS	Physical System Description
PV	Process Variable
R	Requirement
SP	Set Point
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, automobile cruise-control etc. The gains of a PID controller in an application must be tuned before the controller is ready for production. Therefore, a software model is necessary for the simulation of PID controller.

In this section, the purpose of the document is first 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, constraints, models, and requirements to facilitate an unambiguous development of the PID controller software and test procedures.

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 System (Control theory and controllers) at an undergraduate level 4 engineering level, and knowledge of calculus at high school level.

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 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 goals, and proceed to the subsequent sections. Reader with knowledge of the domain can directly review the Instance Models and choose to 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 shows the system context. A circle represents an external entity outside the software, the user in this case. A rectangle represents the software system itself. Arrows are used to show the data show between the system and its environment.

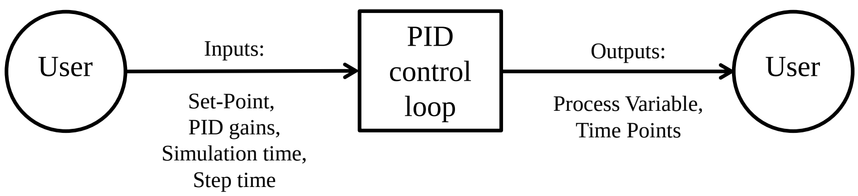


Figure 1: System Context

The PID 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.
- PID Controller Responsibilities:
  - Calculate the outputs of the PID controller, and Power Plant.

3.2 User Characteristics

The end user of PID Controller should at the minimum have knowledge of Control Systems at a second year undergraduate engineering level, and knowledge of High school Calculus.

3.3 System Constraints

No system constraints identified for PID Controller.

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 which 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.

Elapsed Time: Time elapsed between subsequent iterations of the control loop.

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 application.

Step Time: Optimal wait 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 a PID control loop, as shown in Figure-2, includes the following elements:

PS1: Summing Point

PS2: PID Controller

PS3: Power Plant

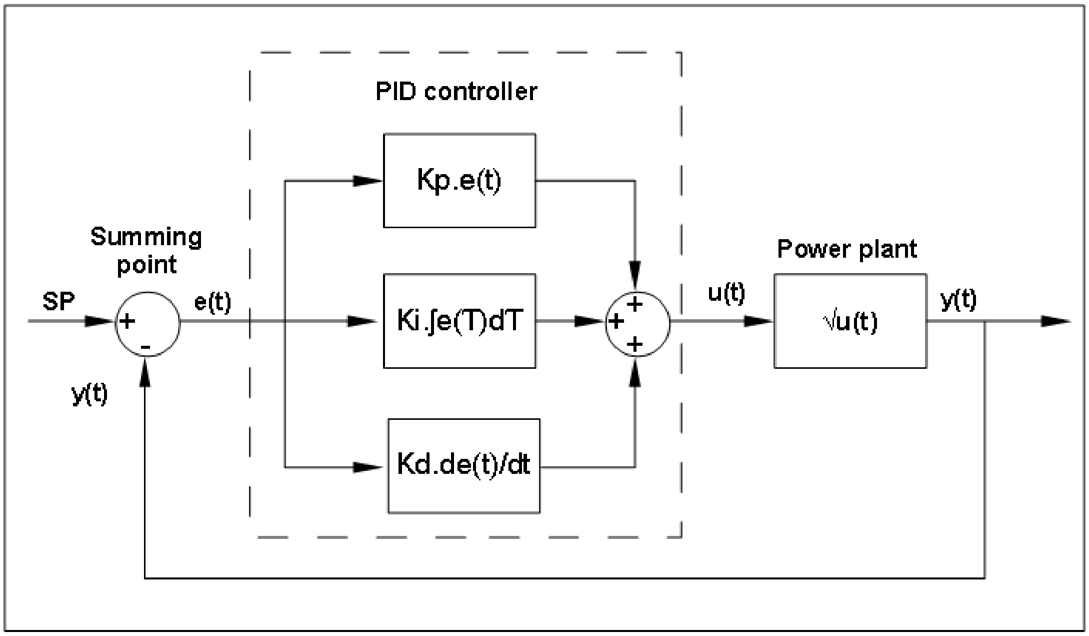


Figure 2: PID Control Loop

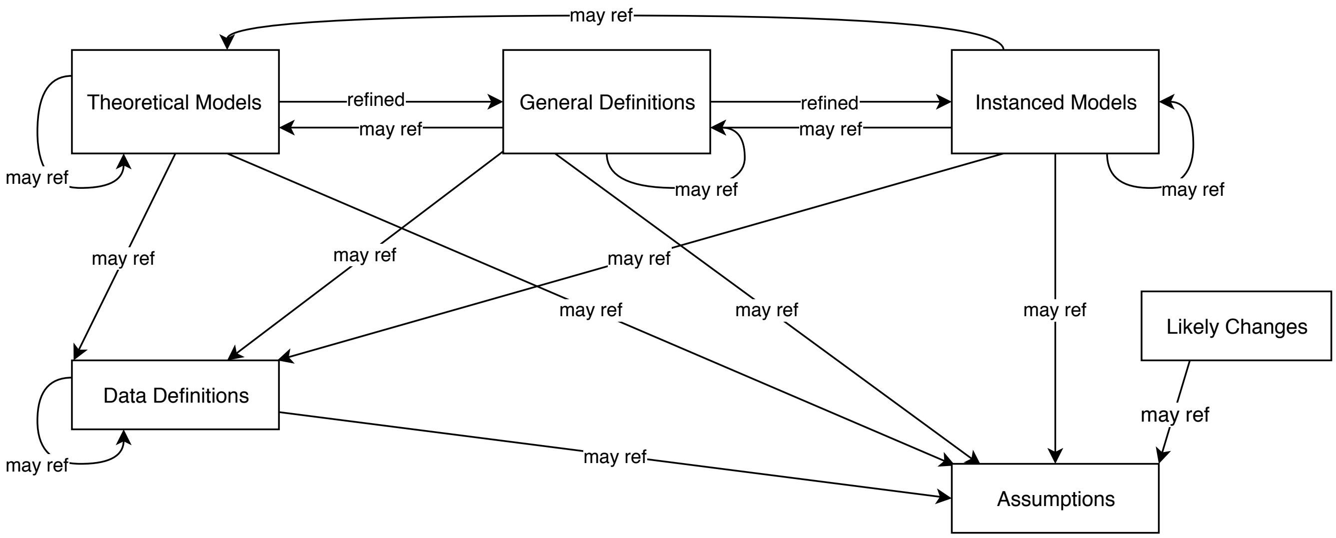
4.1.3 Goal Statements

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

GS1: Compute the output of the Power-Plant at every step.

GS2: Calculate the elapsed time 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: Contrary to the real world, the Power Plant and the Sensor are coupled into a single system, with a response of $f(x) = \sqrt{x}$ [Ref By: IM1, LC1].

A2: This program uses the decoupled form of the equation of the PID controller [Ref By: IM2, LC2].

A3: The Set-Point is a constant throughout the execution of the program [Ref By: DD1, LC4].

A4: There are no external disturbances to the Power plant during the simulation [contributors (2002)] [Ref By: IM3].

A5: There are no delays in the signal propagation, other than the step time, during the simulation [contributors (2002)] [Ref By: DD6].

A6: All analysis are performed in the time domain [Ref By: LC5].

A7: This model will be used for the Manual tuning of the control gains by the user [Ref By: DD2, DD3, DD4].

A8: The initial value (at the start of the simulation) of the Process Variable is assumed to be zero [Ref By: IM1, LC3].

A9: The Parallel form of the equation is used for the PID controller [Ref By: IM2, LC6].

4.2.2 Theoretical Models

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

PID Controller is not based on any general equation or laws.

4.2.3 General Definitions

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

This section is not applicable for PID Controller.

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	**Set Point**
Symbol	SP
SI Units	-
Equation	-
Description	The Set Point (SP) is the expected output that the control loop must achieve. For this simulation, the set-point is assumed to be a constant throughout the simulation (A3).
Source	Michael A Johnson (2006)
Ref. By	IM1

Number	DD2
Label	**Proportional term**
Symbol	P
SI Units	-
Equation	$P = K_p \ . \ x(t)$
Description	The proportional term of the PID controller is the product of the input ($x(t)$) and the proportional gain K_p. K_p is tuned by the user (A7).
Source	Michael A Johnson (2006)
Ref. By	IM2

Number	DD3
Label	**Integral term**
Symbol	I
SI Units	-
Equation	$I = K_i \ . \ \int_0^t x(\mathrm{T}) d\mathrm{T}$
Description	The integral term of the PID controller is a product of the accumulated input over time and the integral gain K_i. K_i is tuned by the user (A7).
Source	Michael A Johnson (2006)
Ref. By	IM2

Number	DD4
Label	**Derivative term**
Symbol	D
SI Units	-
Equation	$D = K_d \ . \ \frac{d(x(t))}{dt}$
Description	The derivative term of the PID controller is a product of the rate of change of the input and the derivative constant K_d. K_d is tuned by the user (A7).
Source	Michael A Johnson (2006)
Ref. By	IM2

Number	DD5
Label	**Total Simulation time**
Symbol	T_{sim}
SI Units	second
Equation	-
Description	T_{sim} is the total time that the control loop must execute. This is is required to terminate the execution in case the Power-Plant is in an unstable state (oscillation).
Source	-
Ref. By	R4

Number	DD6
Label	**Simulation step time**
Symbol	t_{step}
SI Units	second
Equation	-
Description	t_{step} is a time between consecutive iterations of the control loop. This is equivalent to the total time taken for the response of the system (A5).
Source	-
Ref. By	R4

Number	DD7
Label	**Time Point Now**
Symbol	T_{now}
SI Units	second
Equation	-
Description	T_{now} is the time elapsed since epoch in seconds.
Source	-
Ref. By	IM4 DD8

Number	DD8
Label	**Previous Time Point**
Symbol	T_{prev}
SI Units	second
Equation	-
Description	T_{prev} is the last calculated value of Time Point (from DD7).
Source	-
Ref. By	IM4

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 1 and 2 are solved by IM3 and IM4 respectively.

Number	IM1
Label	**Calculate the error signal,** $e(t)$
Input	SP (DD1), $y(t)$ (IM3) The initial value of the process variable, $y(t)$, is assumed to be zero (A8).
Output	$e(t)$, such that $e(t) = SP$ - $y(t)$
Description	The error term is the difference between the Set-Point (required value), and Process Variable, (the measured value). The error term is calculated at each step of the simulation.
Sources	Michael A Johnson (2006)
Ref. By	IM2 R3

Number	IM2
Label	**Calculate the control variable,** $u(t)$
Input	$e(t)$ (IM1)
Output	$u(t)$, such that $u(t) = K_p \ . \ e(t) + K_i \ . \ \int_0^t e(\mathrm{T})d\mathrm{T} + K_d \ . \ \frac{d(e(t))}{dt}$
Description	The Control Variable, $u(t)$ is the output of the PID controller. The Control Variable is the sum of the present (Proportional), past (Integral) and future (Derivative) values of the error. The control variable is calculated at each step of the simulation. The equation used for the PID controller is parallel (A9), that is, the P,I,D components are parallel to each other, and decoupled (A2), that is the gains of the P, I, D component are not interdependent.
Sources	Michael A Johnson (2006)
Ref. By	IM3 R3

Derivation of Control Variable, $u(t)$

Control Variable = Proportional term (from DD2) + Integral term (from DD3) + Derivative term (from DD4)

Therefore, given the input, $e(t)$ (from IM1), the Control Variable, $u(t)$ is,

$u(t) = K_p \ . \ e(t) + K_i \ . \ \int_0^t e(\mathrm{T})d\mathrm{T} + K_d \ . \ \frac{d(e(t))}{dt}$

Number	IM3
Label	**Calculate the process variable, $y(t)$**
Input	$u(t)$ (from IM2)
Output	$y(t)$, such that $y(t) = \sqrt{u}(t)$.
Description	The Process Variable is the output of the power plant (A1). The Process Variable is calculated at each step of the simulation. Additionally, it is assumed that the the power plant does not have any external disturbances (A4).
Sources	-
Ref. By	IM1 R3 R5

Number	IM4
Label	**Calculate the Elapsed time, $T_{\textbf{Elapsed}}$**
Input	T_{now} (DD7), T_{prev} (DD8)
Output	T_{Elapsed}, such that $T_{\text{Elapsed}} = T_{\text{now}} - T_{\text{prev}}$
Description	Time elapsed is the difference between the current time and the previously calculated time. The Time Elapsed is calculated at each step of the simulation.
Sources	-
Ref. By	R3 R5

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
SP	$SP > 0$	$SP_{\text{min}} \leq SP \leq SP_{\text{max}}$	15	10%
K_p	$K_p \geq 0$	$K_{\text{pMin}} \leq K_p \leq K_{\text{pMax}}$	5	10%
K_i	$K_i \geq 0$	$K_{\text{iMin}} \leq K_i \leq K_{\text{iMax}}$	3	10%
K_d	$K_d \geq 0$	$K_{\text{dMin}} \leq K_d \leq K_{\text{dMax}}$	3	10%
T_{sim}	$T_{\text{sim}} > 0$	$T_{\text{simMin}} \leq T_{\text{sim}} \leq T_{\text{simMax}}$	60 s	10%
t_{step}	$t_{\text{step}} > 0$	$t_{\text{stepMin}} \leq t_{\text{step}} \leq t_{\text{stepMax}}$	0.01 s	10%

(*) There is no ideal value for the Set-Point. However, setting it to the mid point of SP_{min} and SP_{max} may help with tuning.

4.2.7 Properties of a Correct Solution

The output response of the power plant must be stable (without oscillation) and must reach the Set-Point.

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: Input the values specified in Table-1.

R2: Ensure that the inputs supplied in R1 are within the limits specified in Table-1.

Table 2: Specification Parameter Values

Var	Value
SP_{min}	0.1
SP_{max}	30
K_{pMin}	0
K_{pMax}	10
K_{iMin}	0
K_{iMax}	10
K_{dMin}	0
K_{dMax}	10
T_{simMin}	1 s
T_{simMax}	120 s
t_{stepMin}	0.01 s
t_{stepMax}	1 s

R3: Calculate e(t) (from IM1), u(t) (from IM2), y(t) (from IM3) and T_{elapsed} (from IM4).

R4: Repeat R3 at t_{step} seconds until T_{sim} seconds (from DD5) has elapsed.

R5: Output the y(t) (from IM3) and T_{elapsed} (from IM4) computed at every iteration (from R3) after T_{sim} seconds (from DD5) has elapsed.

5.2 Nonfunctional Requirements

NFR1: The code shall be portable across multiple Operating Systems platforms.

NFR2: The code shall be importable as a module by another program.

NFR3: The code shall protect the inputs against unconventional input values from users. Unconventional in this context refers to a value of a different type, or unexpectedly large or small values.

NFR4: The application shall be void of memory leaks.

NFR5: The application shall have protection for divide-by-zero error.

NFR6: The code shall be thoroughly documented with Design Specifications and User Guides.

NFR7: The code shall be verifiable against a Verification and Validation plan.

NFR8: The code shall meet the properties of correct solution specified in 4.2.7.

6 Likely Changes

LC1: The transfer function of the Power-plant could be updated to model a more complex system (A1).

LC2: The equation of the PID controller could be changed to industrial form (A2).

LC3: A non-zero value of process variable could be used at the start of the simulation (A8).

LC4: The Set Point could be made a variable during the simulation (A3).

7 Unlikely Changes

LC5: The time domain analysis of a PID controller is unlikely to change (A6).

LC6: The parallel form of equation for PID controller is unlikely to change (A9).

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 3 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 4 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. Table 5 shows the dependencies of Data Definition and Requirements with each other.

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.

9 Values of Auxiliary Constants

No new constants were defined in this report.

	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	IM1	IM2	IM3	IM4
DD1												
DD2												
DD3												
DD4												
DD5												
DD6												
DD7												
DD8							X					
IM1	X										X	
IM2		X	X	X					X			
IM3										X		
IM4							X	X				

Table 3: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	IM3	IM4	R1	R2	R3	R4	R5
IM1			X						
IM2	X								
IM3		X							
IM4									
R1									
R2									
R3	X	X	X	X					
R4							X		
R5			X	X			X		

Table 4: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	R1	R2	R3	R4	R5
DD1													
DD2													
DD3													
DD4													
DD5													
DD6													
DD7													
DD8							X						
R1													
R2													
R3													
R4					X	X					X		
R5					X						X		

Table 5: Traceability Matrix Showing the Connections Between Requirements and Data Definitions

	A1	A2	A3	A4	A5	A6	A7	A8	A9
DD1			X						
DD2							X		
DD3							X		
DD4							X		
DD5									
DD6					X				
DD7									
DD8									
IM1								X	
IM2		X							X
IM3	X			X					
IM4									
LC1	X								
LC2		X							
LC3								X	
LC4			X						
LC5						X			
LC6									X

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

References

Wikipedia contributors. Pid controller — wikipedia, the free encyclopedia, 2002. URL https://en.wikipedia.org/wiki/PID_controller. [Online; accessed 09-Oct-2020].

Mohammad H. Moradi Michael A Johnson, editor. *PID Control: New Identification and Design Methods*. Springer Science and Business Media, 2006.

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.