Module Interface Specification for IP Simulator

Mina Mahdipour

March 18, 2023

1 Revision History

Date	Version	Notes
March 2, 2023	1.0	First version, created and added introduction parts
March 10, 2023	1.1	Added three modules
March 12, 2023	1.2	Added the rest of modules
March 17, 2023	1.3	Updated ODE modules

2 Symbols, Abbreviations and Acronyms

The reader can refer to SRS.

Contents

List of Tables

3 Introduction

The following document details the Module Interface Specifications for IP Simulator software. This document specifies how every module is interfacing with every other part of the program based on "module state machine" approach.
Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at the github repository for the IP Simulator. The author uses [3] as a reference to write this document.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper [2], with the addition that template modules have been adapted from [1]. The mathematical notation comes from Chapter 3 of Hoffman and Strooper [2]. For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | ... | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by IP Simulator.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)
non-negative integer	$\mathbb{N}_0$	a number without a fractional component in [0, ∞)

The specification of IP Simulator uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, IP Simulator uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	Level 2
Hardware-Hiding Module	
Behaviour-Hiding Module	Input Parameters Module Output Parameters Module Constant Parameter Module The Equation of Motion ODE Module IP Control Module
Software Decision Module	ODE Solver Module Data Structure Module Plotting Module

Table 1: Module Hierarchy

6 MIS of Input Parameters Module

This module is responsible for reading the input parameters from a file and storing them in the data structures. It also verifies them using Constant Parameter Module.

6.1 Module

InputM

6.2 Uses

ConstantM (Section 8)

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
load_inputs	string	-	FileError
verify_inputs	-	-	InvalidInput Error
m_p	-	$\mathbb{R}$	
l_p	-	$\mathbb{R}$	
m_c	-	$\mathbb{R}$	
$friction$	-	$\mathbb{R}$	
$f\_external(t)$	$\mathbb{R}$	$\mathbb{R}$	InvalidFunction
i_p	-	$\mathbb{R}$	

6.4 Semantics

6.4.1 State Variables

#From R1:
m_p: $\mathbb{R}$
l_p: $\mathbb{R}$
m_c: $\mathbb{R}$
$friction$: $\mathbb{R}$
$f\_external(t)$: $\mathbb{R} \to \mathbb{R}$
i_p: $\mathbb{R}$

6.4.2 Environment Variables

In the case of this module, the environment variable is the input file which is containing the input data.

6.4.3 Assumptions

- load_inputs will be called before the values of any state variables will be accessed.
- The file contains the expected inputs data in order, each on a new line. The order of the input data is as below:
  - line 1: the mass of the pendulum
  - line 2: the length of the pendulum
  - line 3: the mass of the cart
  - line 4: the friction of the cart
  - line 5: the external force as a function of time

6.4.4 Access Routine Semantics

The value of each state variable can be accessed through its name (getter). An access program is available for each state variable. There are no setters for the state variables, since the values will be set and checked by load params and not changed for the life of the program.

load_inputs($fName$: string):

- transition: The filename $fName$ is associated with the inputFile. The state variables are modified with the following procedures:
  1. Read data from inputFile to populate the state variables from R1.
  2. Store the data parameters.
  3. Calculate the derived quantity as follows (from DD3 in SRS):
     $i_p = \frac{1}{12}(m_p)(l_p^2)$
  4. Verify the inputs through verify_inputs().
- exception: exc := a file name $fName$ cannot be found OR the format of inputFile is incorrect $\Rightarrow$ FileError

verify_inputs():

- out: out := none
- exception: exc :=

$\neg(m_{\text{pmin}} \leq m_p \leq m_{\text{pmax}})$	$\Rightarrow$ Invalid Mass of the Pendulum
$\neg(l_{\text{pmin}} \leq l_p \leq l_{\text{pmax}})$	$\Rightarrow$ Invalid Length of the Pendulum
$\neg(m_{\text{cmin}} \leq m_c \leq m_{\text{cmax}})$	$\Rightarrow$ Invalid Mass of the Cart
$\neg$(friction>= 0)	$\Rightarrow$ Invalid Value of the Car Friction
$f\_external$ is not a function of t	$\Rightarrow$ InvalidFunction

etc. See Appendix (Section 13) for the complete list of exceptions and associated error messages.

InputM.m_p:

- output: $out := m_p$
- exception: none

InputM.l_p:

- output: $out := l_p$
- exception: none

InputM.m_c:

- output: out :=m_c
- exception: none

InputM.$friction$:

- output: $out := friction$
- exception: none

/InputM.$f\_external(t)$:

- output: $out := f\_external(t)$
- exception: none

6.4.5 Local Functions

None

7 MIS of Output Parameters Module

7.1 Module

OutputM

7.2 Uses

ConstantM (Section 8)

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
output	$fOName$: string, t: $\mathbb{R}$, x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$	-	-
verify_out	[x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$]	-	InvalidOutput

7.4 Semantics

7.4.1 State Variables

None

7.4.2 Environment Variables

The environment variable is the output file which includes the angle of the pendulum and position of the cart with regards to the data inputs.

7.4.3 Assumptions

All of the fields of the input parameters structure have been assigned a value.

7.4.4 Access Routine Semantics

output ($fOName$: string, t: $\mathbb{R}$, x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$):

- transition: The filename $fOName$ is associated with the outputFile. The following procedure:

  1. verify_out (x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$)

2. write the calculated cart position, cart velocity, pendulum angle, and pendulum velocity for a specified time in the output file.

- exception: none

verify_out (x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$):

- output: none
- exception: exec:=

$$\begin{array}{ll} \theta \leq 0 & \Rightarrow \text{Invalid Angle the Pendulum} \\ \dot{\theta} \leq 0 & \Rightarrow \text{Invalid Velocity of the Pendulum} \end{array}$$

7.4.5 Local Functions

None

8 MIS of Constant Parameter Module

8.1 Module

ConstantM

8.2 Uses

Not Applicable

8.3 Syntax

8.3.1 Exported Constants

From Table 2 in SRS

Name	Type	Value	Description
g	$\mathbb{R}$	$9.81\ \mathrm{m\,s^{-2}}$	the gravity of the earth
m_{pmin}	$\mathbb{R}$	0.01 kg	minimum value of the pendulum mass
m_{pmax}	$\mathbb{R}$	50 kg	maximum value of the pendulum mass
l_{pmin}	$\mathbb{R}$	0.01 m	minimum value of the length of the pendulum
l_{pmax}	$\mathbb{R}$	10 m	maximum value of the length of the pendulum
m_{cmin}	$\mathbb{R}$	0.01 kg	minimum value of the mass of the cart
m_{cmax}	$\mathbb{R}$	50 kg	maximum value of the mass of the cart
x	$\mathbb{R}$	0	the initial condition of the position of the cart
$\dot{x}$	$\mathbb{R}$	$0\ \mathrm{m\,s^{-2}}$	the initial condition of the velocity of the cart
θ	$\mathbb{R}$	π rad	the initial condition of the angle of the pendulum
$\dot{\theta}$	$\mathbb{R}$	$0\ \mathrm{m\,s^{-2}}$	the initial condition of the velocity of the pendulum
t_{start}	$\mathbb{R}$	0 min	the start time of simulation
t_{span}	$\mathbb{R}$	100 min	the duration time of simulation

8.3.2 Exported Access Programs

None

8.4 Semantics

Not Applicable

9 MIS of Equation of Motion ODE Module

9.1 Module

EqMo

9.2 Uses

ConstantM (Section 8)

9.3 Syntax

9.3.1 Exported Constants

None

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
ODE_Motion	m_p: $\mathbb{R}$, l_p: $\mathbb{R}$, m_c: $\mathbb{R}$, $friction$: $\mathbb{R}$, $f\_external(t)$, i_p: $\mathbb{R}$, x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$)	$\dot{x}$: $\mathbb{R}$, $\ddot{x}$: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$, $\ddot{\theta}$: $\mathbb{R}$	-

9.4 Semantics

9.4.1 State Variables

None

9.4.2 Environment Variables

None

9.4.3 Assumptions

None

9.4.4 Access Routine Semantics

ODE_Motion(m_p: $\mathbb{R}$, l_p: $\mathbb{R}$, m_c: $\mathbb{R}$, $friction$: $\mathbb{R}$, $f\_external(t)$):

$$\ddot{x} = \frac{friction}{(m_c + m_p)}\dot{x} + \frac{f\_external(t) - m_p l_p \ddot{\theta}\cos\theta + m_p l_p (\dot{\theta})^2 \sin\theta}{(m_c + m_p)}$$

$$\ddot{\theta} = \frac{m_p g l_p}{(i_p + m_p l_p^2)}\sin\theta - \frac{m_p l_p \ddot{x}}{(i_p + m_p l_p^2)}\cos\theta$$

- output: $[\dot{x}, \ddot{x}, \dot{\theta}, \ddot{\theta}]$
- exception: none

9.4.5 Local Functions

None

10 MIS of IP Control Module

10.1 Module

Control

10.2 Uses

InputM (Section 6), OutputM (Section 7), EqMo (Section 9), ODE Solver (Section 11), Plot (Section 12),

10.3 Syntax

10.3.1 Exported Constants

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
control	-	-	-

10.4 Semantics

The Control Module is designed to control the process flow in the software. It organizes all other modules to satisfy all the requirements and also helps maintainability and expandability of IP Simulator by classifying different parts of the code.

10.4.1 State Variables

None

10.4.2 Environment Variables

None

10.4.3 Assumptions

None

10.4.4 Access Routine Semantics

control():

- transition: Control the flow input data, calculation, and the output data by following below steps:
  Get ($fName$: string) from user
  load_inputs($fName$)

verify_inputs()
ODE_Motion(m_p: $\mathbb{R}$, l_p: $\mathbb{R}$, m_c: $\mathbb{R}$, $friction$: $\mathbb{R}$, $f\_external(t)$, i_p: $\mathbb{R}$, x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$))
solveODE($\mathbf{f} : (\mathbb{R}^8 \rightarrow \mathbb{R}^4), t_{\text{start}} : \mathbb{R}, \mathbf{y}_{\text{iniCond}} : \mathbb{R}^6, t_{\text{span}} : \mathbb{R}$)
verify_out(x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$)
output($fOName$, x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$)
plot(t: $\mathbb{R}$, x: $\mathbb{R}$, θ: $\mathbb{R}$)

- exception: none

10.4.5 Local Functions

None

11 MIS of ODE Solver Module

11.1 Module

ODE Solver

11.2 Uses

None

11.3 Syntax

11.3.1 Exported Constants

11.3.2 Exported Access Programs

Name	In	Out	Except.
solveODE	$\mathbf{f} : (\mathbb{R}^8 \rightarrow \mathbb{R}^4), t_{\text{start}} : \mathbb{R}, \mathbf{y}_{\text{iniCond}} : \mathbb{R}^6, t_{\text{span}} : \mathbb{R}$	$\mathbf{y}$: [x: $\mathbb{R}$, $\dot{x}$: $\mathbb{R}$, θ: $\mathbb{R}$, $\dot{\theta}$: $\mathbb{R}$]	ODE_Error

11.4 Semantics

11.4.1 State Variables

None

11.4.2 Environment Variables

None

11.4.3 Assumptions

None

11.4.4 Access Routine Semantics

solveODE($\mathbf{f}$, t_{start}, $\mathbf{y}_{\text{iniCond}}$, t_{span}):

- output: $out := \mathbf{y}(t)$ where

$$\mathbf{y}(t) = \mathbf{y}_{\text{iniCond}} + \int_{t_{\text{start}}}^{t_{\text{span}}} \mathbf{f}(s, \mathbf{y}(s)) ds$$

  $y(t)$ is calculated from $t = t_{\text{start}}$ to $t = t_{\text{span}}$.
  We have two coupled ODE in the IP which s is θ the first one and x in the second one.

- exception: exc := (ODE Solver Fails $\Rightarrow$ ODE_ERR)

11.4.5 Local Functions

None

12 MIS of Plotting Module

12.1 Module

Plot

12.2 Uses

Not Applicable

12.3 Syntax

12.3.1 Exported Constants

None

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
plot	t: $\mathbb{R}$, x: $\mathbb{R}$, θ: $\mathbb{R}$	Graph	-

12.4 Semantics

12.4.1 State Variables

None

12.4.2 Environment Variables

win: 2D diagram displayed on the screen

12.4.3 Assumptions

12.4.4 Access Routine Semantics

plot(t: $\mathbb{R}$, x: $\mathbb{R}$, θ: $\mathbb{R}$):

- transition: Modify win to display a plot where the vertical axis is the angle of the pendulum and the position of the cart. The time should run from t_{start} to t_{span}.
- exception: none

12.4.5 Local Functions

References

[1] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

[2] Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

[3] W. Spencer Smith. A rational document driven design process for scientific computing software. In Jeffrey C. Carver, editor, *Software Engineering for Science*, chapter Section I – Examples of the Application of Traditional Software Engineering Practices to Science. Submitted 2016. 30 pp.

13 Appendix

Table 2: Possible Exceptions

Message ID	Error Message
FileError	Error: The expected file does not exist.
InvalidFunction	Error: The function of external force must be only the function of time.
Invalid Mass of the Pendulum	Error: The mass of the pendulum should be positive.
Invalid Length of the Pendulum	Error: The length of the pendulum should be positive.
Invalid Mass of the Cart	Error: The mass of the cart should be positive.
Invalid Value of the Car Friction	Error: The $friction$ must be greater that zero.
InvalidOutput	Error: The angle and velocity of the pendulum must be greater that zero.
ODE_Error	Error: When fails to solve the ODE.