CAS 741: Module Interface Specification

Dynamical Systems: Multi-Pendulum Simulation

Karol Serkis
serkiskj@mcmaster.ca
GitHub: karolserkis

November 24, 2018

1 Revision History

Date	Version	Notes
November 22, 2018	1.0	First full draft for submission

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at CAS-741-Pendula SRS [give url —SS]
[Also add any additional symbols, abbreviations or acronyms —SS]
The symbols are listed in alphabetical order.

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
NF	Non-Functional Requirement
MIS	Module Interface Specification
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
T	Theoretical Model

Contents

3 Introduction

The following document details the Module Interface Specifications for the Multi-Pendulum Simulation project [Fill in your project name and description —SS]

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at: https://github.com/karolserkis/CAS-741-Pendula. [provide the url for your repo —SS]

4 Notation

[You should describe your notation. You can use what is below as a starting point. —SS]

The structure of the MIS for modules comes from ?[HoffmanAndStrooper1995 —SS], with the addition that template modules have been adapted from ?[GhezziEtAl2003 —SS]. The mathematical notation comes from Chapter 3 of ?[HoffmanAndStrooper1995 —SS]. For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

(Karol) I am also using this link for now.

The following table summarizes the primitive data types used by Multi-Pendulum Simulation and the symbols used in this document. The choice of symbols was made to be consistent with calculus, ordinary differentials (ODE), the Lagrangian, kinematics etc. The standard mathematical spaces are used (e.g. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, etc.)

Data Type	Notation	Description
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of Multi-Pendulum Simulation 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, Multi-Pendulum Simulation 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	
Behaviour-Hiding	Multi-Pendulum Simulation Control Module Multi-Pendulum Simulation GUI Module User Input Parameters Module Data Structure Module
Software Decision	Generic Trajectory Simulation GUI Module Generic GUI/Plot Module Lagrangian Module Hamiltonian Module

Table 1: Module Hierarchy

6 MIS of [Module Name —SS]

[Use labels for cross-referencing —SS]

[You can reference SRS labels, such as R??. —SS]

[It is also possible to use LaTeXfor hypperlinks to external documents. —SS]

6.1 Module

[Short name for the module —SS]

6.2 Uses

6.3 Syntax

6.3.1 Exported Access Programs

Name	In	Out	Exceptions
[accessProg —SS]	-	-	-

6.4 Semantics

6.4.1 State Variables

6.4.2 Access Routine Semantics

[accessProg —SS]():

- transition: [if appropriate —SS]
- output: [if appropriate —SS]
- exception: [if appropriate —SS]

7 MIS of [Module Name —SS]

[Use labels for cross-referencing —SS]

[You can reference SRS labels, such as R??. —SS]

[It is also possible to use LaTeXfor hypperlinks to external documents. —SS]

7.1 Module

[Short name for the module —SS]

7.2 Uses

7.3 Syntax

7.3.1 Exported Access Programs

Name	In	Out	Exceptions
[accessProg —SS]	-	-	-

7.4 Semantics

7.4.1 State Variables

7.4.2 Access Routine Semantics

[accessProg —SS]():

- transition: [if appropriate —SS]
- output: [if appropriate —SS]
- exception: [if appropriate —SS]

References

[You shouldn't do this manually, especially since you use BibTeX in the next section. The new references can be added to your .bib file and everything will be generated automatically. —SS]

- [1] Dynamics of multiple pendula
  http://wmii.uwm.edu.pl/~doliwa/IS-2012/Szuminski-2012-Olsztyn.pdf
- [2] Pendulum
  https://en.wikipedia.org/wiki/Pendulum
- [3] Pendulum (mathematics)
  https://en.wikipedia.org/wiki/Pendulum_(mathematics)
- [4] Double Pendulum
  https://en.wikipedia.org/wiki/Double_pendulum
- [4] Differential-Algebraic Equations by Taylor Series
  http://www.cas.mcmaster.ca/~nedialk/daets/
- [5] Multi-body Lagrangian Simulations
  https://www.youtube.com/channel/UCCuLchOx0W0yoNE9KOCYlVQ
- [6] The double pendulum: Lagrangian formulation
  https://diego.assencio.com/?index=1500c66ae7ab27bb0106467c68feebc6
- [7] Poincaré map
  https://en.wikipedia.org/wiki/Poincar%C3%A9_map
- [8] D. L. Parnas, "On the criteria to be used in decomposing systems into modules", Comm. ACM, vol. 15, pp. 1053-1058, December 1972.
- [9] D. Parnas, P. Clement, and D. M. Weiss, "The modular structure of complex systems", in International Conference on Software Engineering, pp. 408-419, 1984.
- [10] D. L. Parnas, "Designing software for ease of extension and contraction," in ICSE '78: Proceedings of the 3rd international conference on Software engineering, (Piscataway, NJ, USA), pp. 264-277, IEEE Press, 1978.
- [11] Smith and Lai(2005); Smith et al. (2007).
  See bib file that doesn't work for me for full citation.

8 Appendix

[Extra information if required —SS]