Module Interface Specification for 3D-H3C

Reyhaneh Norouziani

March 24, 2024

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/rnorouziani/3D-H3C/blob/main/docs/SRS/SRS.pdf

Contents

3 Introduction

The following document details the Module Interface Specifications for 3D-H3C

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/rnorouziani/3D-H3C

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). 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 3D-H3C.

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

The specification of 3D-H3C 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, 3D-H3C 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	Input Parameters Output Format Module Output Verification Module Magnetic Core Module CoilT Module Control Module
Software Decision	Vector Module

Table 1: Module Hierarchy

6 MIS of Input Format Module

6.1 Module

Input

6.2 Uses

Hardware Hiding Modules

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
load_coils	string	[CoilT, CoilT, CoilT, [ℝ,ℝ,ℝ]]	FileError
load_params	string	[[ℝ,ℝ,ℝ], [ℝ,ℝ,ℝ]]	FileError

6.4 Semantics

6.4.1 State Variables

None.

6.4.2 Environment Variables

command line, file

6.4.3 Assumptions

None.

6.4.4 Access Routine Semantics

load_coils(file_name):

- output: The output is composed of an array containing three instances of CoilT, with each instance representing a pair of coils, and an array representing the magnetic moment of the obeject subject to the specified target magnetic torque and force .

- exception: FileError if the file name file_name cannot be found OR the format of inputFile is incorrect.

load_params(input_str):

- output: The output is an array composed of tow arrays. One representing the target force vector, and the other representing the target torque vector.
- exception: FileError if the format of input_str is incorrect.

6.4.5 Local Functions

None.

7 MIS of CoilT module

7.1 Module

coil

7.2 Uses

None.

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
CoilT	$\mathbb{R}$, $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}$	CoilT	ValueError
get_R	-	$\mathbb{R}$	-
get_L	-	$\mathbb{R}$	-
get_N	-	$\mathbb{N}$	-
get_maxI	-	$\mathbb{R}$	-
get_magnetic_force	$\mathbb{R}$, [$\mathbb{R}$,$\mathbb{R}$,$\mathbb{R}$]	[$\mathbb{R}$,$\mathbb{R}$,$\mathbb{R}$]	-
get_magnetic_torque	$\mathbb{R}$, [$\mathbb{R}$,$\mathbb{R}$,$\mathbb{R}$]	[$\mathbb{R}$,$\mathbb{R}$,$\mathbb{R}$]	-

7.4 Semantics

7.4.1 State Variables

R, l, N, Max_I

7.4.2 Environment Variables

None.

7.4.3 Assumptions

None.

7.4.4 Access Routine Semantics

CoilT(R,l,N,Max_I):

- transition: The state variables are initialized with their values by invoking the constructor.
- exception: ValueError if the inputs do not adhere to the Input Data Constraints. Detailed messages corresponding to each type of violation can be found in the appendix.

get_R():

- output: The radius of the coil is returned as the output.

get_R():

- output: The radius of the coil is returned as the output.

get_L():

- output: The distance between the pair of coils is returned as the output.

get_N():

- output: The number of turns of wire in each coil is returned as the output.

get_maxI():

- output: The maximum acceptable electric current of the coil is returned as the output.

get_magnetic_fild(I, isMaxwell):

- output: Calculate the magnetic field generated in the coils by the current, I. The parameter isMaxwell determines whether the coil configuration is that of a Helmholtz coil or a Maxwell coil. The equation mentioned in GD2 and GD3 of the System Requirements Specification (SRS) is used.

get_magnetic_force(I, [m_x,m_y,m_z]):

- output: The magnetic force exerted on an object, characterized by its magnetic moment vector m, is calculated based on the magnetic field produced by the current I.

get_magnetic_torque(I, [m_x,m_y,m_z]):

- output: The magnetic torque exerted on an object, characterized by its magnetic moment vector m, is calculated based on the magnetic field produced by the current I.

7.4.5 Local Functions

None.

8 MIS of Magnetic Core Module

8.1 Module

mCore

8.2 Uses

Vector Module, CoilT Module [7]

8.3 Syntax

8.3.1 Exported Constants

None

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
MagneticCore	CoilT, CoilT, CoilT	MagneticCore	-
calculate_current_of_target_force	-	$\mathbb{N}$	-
calculate_current_of_target_torque	[$\mathbb{R}$,$\mathbb{R}$,$\mathbb{R}$]	$\mathbb{R}$	-

8.4 Semantics

8.4.1 State Variables

CoilTx, CoilTy, CoilTz , [m_x, m_y, m_z]

8.4.2 Environment Variables

None.

8.4.3 Assumptions

None.

8.4.4 Access Routine Semantics

MagneticCore (CoilT, CoilT, CoilT, [ℝ,ℝ,ℝ]):

- transition: The state variables are initialized with their values by invoking the constructor.

calculate_current_of_target_force([ℝ,ℝ,ℝ]):

- output: The current is calculated by using the equation mentioned in IM1 in the System Requirements Specification (SRS) document. The resulting calculated current is then returned as the output.

calculate_current_of_target_torque([ℝ,ℝ,ℝ]):

- output: The current is calculated by using the equation mentioned in IM2 in the System Requirements Specification (SRS) document. The resulting calculated current is then returned as the output.

8.4.5 Local Functions

None.

9 MIS of Output Verification Module

9.1 Module

Input

9.2 Uses

Hardware Hiding Modules

9.3 Syntax

9.3.1 Exported Constants

None

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
OutputVerification	CoilT,CoilT,CoilT,- [ℝ,ℝ,ℝ], [ℝ,ℝ,ℝ], [ℝ,ℝ,ℝ], ℝ,ℝ		OutputVerification
is_currents_within_range	-	bool	-
calculate_accuracy	-	bool	-

9.4 Semantics

9.4.1 State Variables

CoilTx, CoilTy, CoilTz , $[m_x, m_y, m_z]$,$[f_x, f_y, f_z]$, $[t_x, t_y, t_z]$, I1, I2

9.4.2 Environment Variables

9.4.3 Assumptions

None.

9.4.4 Access Routine Semantics

OutputVerification(CoilTx, CoilTy, CoilTz , $[m_x, m_y, m_z]$,$[f_x, f_y, f_z]$, $[t_x, t_y, t_z]$, I1, I2):

- output: The state variables are initialized with their values by invoking the constructor.

is_currents_within_range():

- output:Returns true if all the currents are smaller than the maximum acceptable electric current for the corresponding coils.

calculate_accuracy():

- output:Returns true if the accuracy of the calculated current I exceeds a specified constant value.

9.4.5 Local Functions

None.

10 MIS of Control Module

10.1 Module

control

10.2 Uses

Input Format Module [6], Output Format Module, Output Verification Module [9], Magnetic Core Module [8], CoilT Module [7]

10.3 Syntax

10.3.1 Exported Constants

None

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
main	-	-	-

10.4 Semantics

10.4.1 State Variables

CoilTx, CoilTy, CoilTz , inputF, magneticCore, verification, OutputF

10.4.2 Environment Variables

None.

10.4.3 Assumptions

None.

10.4.4 Access Routine Semantics

main():

- transition: The state variables are initialized with their values through the invocation of constructors, and these state variables are modified during the execution by these steps:

  [coilTx, coilTy, coilTz, [m_x, m_y, m_z]] = load_coils(file_name)

  The following steps are repeatable, enabling the calculation of currents multiple times.

  [[f_x, f_y, f_z], [t_x, t_y, t_z]] = load_params(file_name)

  MagneticCore(coilTx, coilTy, coilTz)

  I1 = calculate_current_of_target_force()

  I2 = calculate_current of_target_torque()

  OutputVerification(CoilTx, CoilTy, CoilTz , [m_x, m_y, m_z],[f_x, f_y, f_z], [t_x, t_y, t_z], I1, I2)

  is_currents_within_range()

calculate_accuracy()

10.4.5 Local Functions

None.

References

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

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.

11 Appendix

Table 2: Possible Exceptions

Exception Type	**Error Message**
ValueError	Error: Radius must be > 0
ValueError	Error: Distance between the coils must be > 0
ValueError	Error: The number of turns must be elements of the set of natural numbers