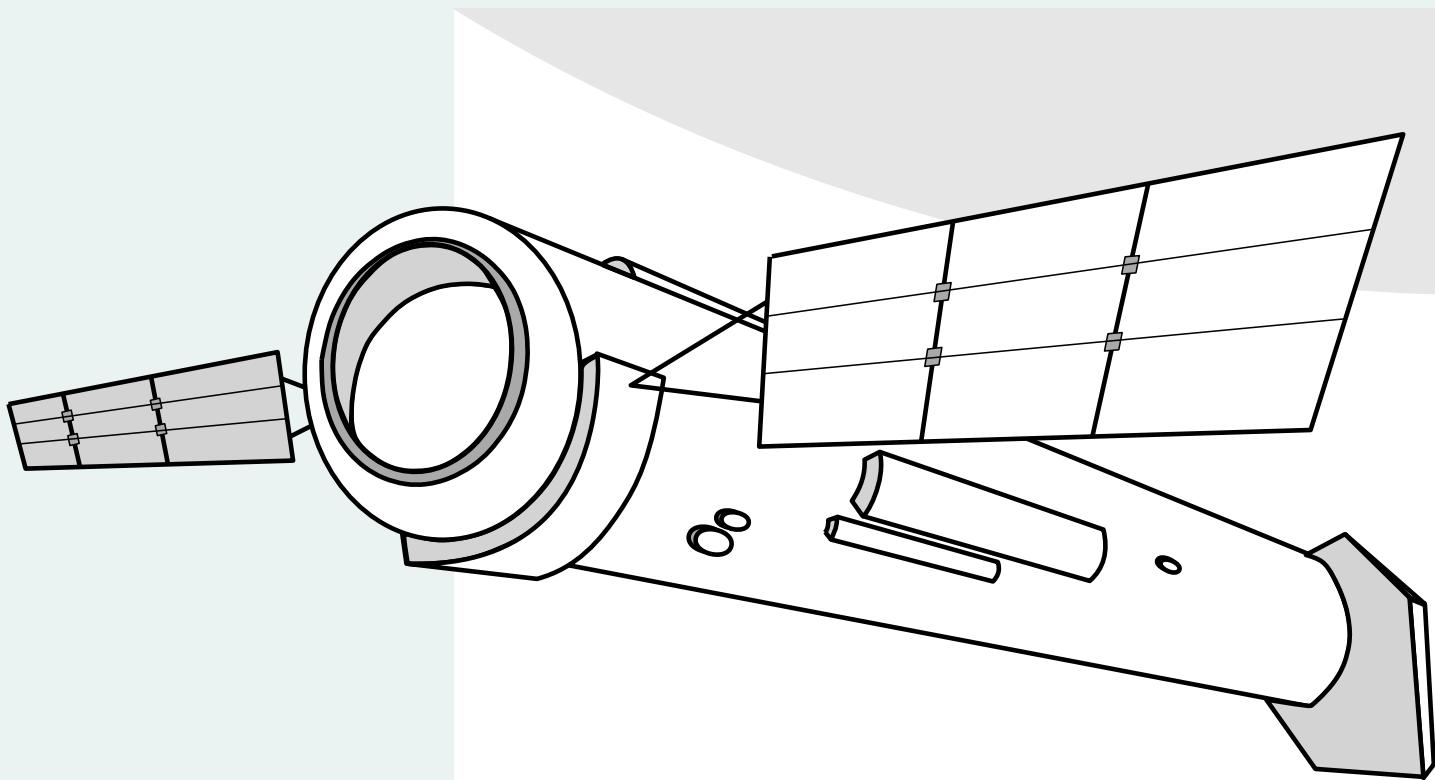


P. Iannelli
F. Sanfedino
D. Alazard

European Satellite Benchmark for Control Education & Industrial Training

User Manual
v1.2
2025



Version 1.2
January 15, 2025
Copyright (c) ISAE-SUPAERO, All Rights Reserved
ESA 4000141060/23/NL/MGu
ESA Public Licence - v2.4 - Permissive (Type 3)

*To Samir Bennani,
who first believed in this project
and paved the way
for modern GNC practice*

Note for the reader

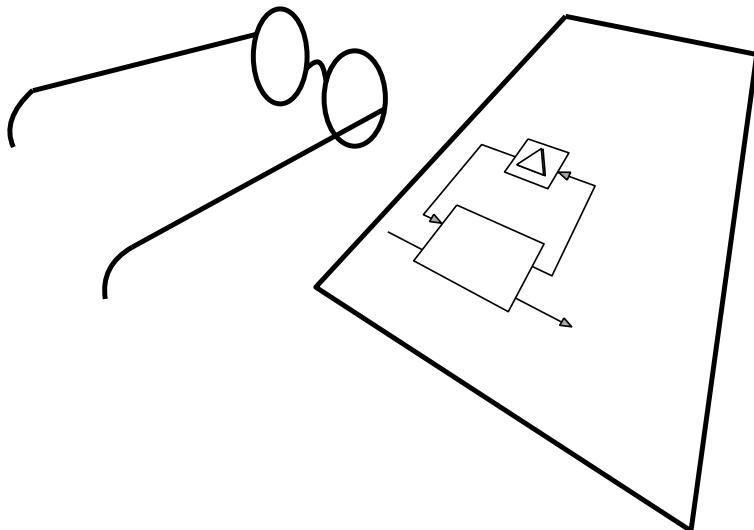
For any use of the present document and related software or part of them (i.g. for presentations, official documents, scientific article or report) please mention the following references:

- 1) "*Iannelli, P. and Sanfedino, F. and Alazard, D., European Satellite Benchmark: for Control Education and Industrial Training - User manual, Version 1.2, 2025*"
- 2) "*Francesco Sanfedino, Paolo Iannelli, Daniel Alazard, Emilie Pelletier, European satellite benchmark for control education & industrial training — Lesson learned in the control design of a fine pointing mission. Mechanical Systems and Signal Processing, Volume 242, 2026, 113684. <https://doi.org/10.1016/j.ymssp.2025.113684>"*

Contacts For any questions regarding the benchmark fell free to contact:

francesco.sanfedino@isae.fr

daniel.alazard@isae.fr



Project Acronym	GNC V&V Benchmark
Project Title	Generic Satellite Simulation & Analysis Tool: Dynamics Modelling, Controls and Validation of Complex Satellite Systems with Guaranteed Robust Pointing Performances
Duration	12 months
Deliverable Number	UM 02
Deliverable Title	User Manual 02 for the partners
Version	1.1
Status	Final
Deliverable Type	Document/Report
Actual Date of Deliverable	16/12/2024

Document Prepared by

Name And Organization	Position and title
Paolo IANNELLI ; ISAE-SUPAERO	PostDoc
Francesco SANFEDINO; ISAE-SUPAERO	Ass. Prof.
Daniel ALAZARD; ISAE-SUPAERO	Prof.

Document Verified by

Name And Organization	Position and title
Émilie PELLETIER ; ESA-ESTEC	Technical Officer

Contents

Nomenclature	xv
1 Introduction	1
1.1 Context and Motivation	1
1.2 Scope and purpose	3
1.3 Applicable Documents (ADs)	6
1.4 Document Outline	7
2 Description of Simulator-02	9
2.1 Getting started	9
2.1.1 Installation and requirements	9
2.2 Directory structure	10
3 High pointing benchmark study definition	13
3.1 Definition of the Spacecraft model	14
3.1.1 AOCS actuators and sensors	14
3.1.2 Actuators/Sensors at payload level	16
3.1.3 AOCS and LOS stabilization architectures	16
3.1.4 Benchmark timeline	18
3.2 Objectives and constraints	20
3.2.1 Constraints on actuators	21
3.3 Disturbances	22
4 Tutorial I: Spacecraft dynamical model	23

4.1	Overview of SDTLIB spacecraft model	24
4.1.1	Reaction Wheel System (RWS)	28
4.1.2	Sloshing equivalent mechanical model	29
4.1.3	Reaction Control System (RCS)	29
4.1.4	Micro-Propulsion System (MPS)	31
4.1.5	Payload (Telescope structure, FSM, PMA, Isolator)	32
4.2	Optical model	34
4.2.1	Telescope model	34
4.2.2	Telescope finite element model	35
4.2.3	Ray-tracing algorithm for linear optical sensitivities	38
4.3	LFT/LPV model and data provided to the user	39
5	Synthesis model	43
5.1	Code for synthesis model: <code>Main_Open_Loop_Dynamic_Model.m</code>	45
5.1.1	GUI for selection of phase, requirements and actuators/sensors . .	45
6	Non-linear simulator	51
6.1	Mission timeline and conditions for phase transition	52
6.2	Overview of the simulator	53
6.2.1	Simulator subsystem: DYNAMICAL SYSTEM	55
6.2.2	Simulator subsystem: SENSORS	78
6.2.3	Simulator subsystem: OUTPUT TO WORKSPACE	81
6.2.4	Simulator subsystem: CHECK ON APE REQUIREMENTS FOR PHASE TRANSITION	83
6.3	Tutorial: Setting up a time-domain analysis	86
6.3.1	Time domain initialization	87
6.3.2	Tutorial: Setting up worst-case analysis	90
6.3.3	Tutorial: Setting up Monte Carlo analysis	91
6.4	Tutorial: How to modify the controllers	94
6.5	Tutorial: How to post-process simulation data	97
6.6	Computational aspects of the simulator	100

A Overview of SDTlib and TITOP formulation	101
A.1 Rotation matrix of angle θ around z -axis	102
B Overview of pointing metrics in time and frequency domain	105
C Parametric Uncertainties	107

List of Figures

1.1	Integrated Design Logic. Hubble Image: ©ESA/NASA.	2
1.2	High-fidelity simulation environment schematics	4
1.3	Schematics of an incremental complexity approach applied for modelling reaction wheels	4
2.1	Content of the main directory	10
2.2	Content of the directory <i>Utilities</i>	11
3.1	Schematics of the proposed spacecraft architecture for the benchmark problem	15
3.2	Timeline sequence for the benchmark problem	18
4.1	Visualization of the SDTlib model	25
4.2	View of the assembled Simscape model of the spacecraft	27
4.3	Single reaction wheel (left) and equivalent TITOP model (right)	28
4.4	Equivalent LPV model of the RWS scheduled according to the 4 RW speeds	28
4.5	Configuration with 12 thruster in parallelepiped configuration.	30
4.6	Visualization of the SDTlib model for the payload system	32
4.7	Payload plate with the PMA system. The figure shows also the position of the FSM and of the detector	33
4.8	Fast steering mirror (left) and equivalent TITOP model (right)	33
4.9	Proof-mass actuator view (left) and equivalent TITOP model (right)	34
4.10	Ritchey-Chrétien two-mirror telescope system	35
4.11	FE model of the telescope	36

4.12 Internal supporting structure of the primary mirror (left); Rigid body connection of the supporting structure to the interface node with RBE2	36
4.13 First 6 modes of vibration of the telescope structure	37
5.1 Generalized closed-loop model for synthesis (with weighting filters in cyan and without in light yellow)	44
5.2 GUI for Inputs/outputs selection	45
5.3 Simulink file <code>General_DL_model_NoFilters.slx</code>	48
6.1 Non-linear simulator in simulink file: <code>Simscape_Benchmark_SIM1.slx</code>	54
6.2 View of the simulator subsystem: DYNAMICAL SYSTEM	56
6.3 Flexible Appendages (Solar arrays + HGA) subsystem	58
6.4 Flexible Appendages (Solar arrays + HGA) subsystem	60
6.5 Flywheel static (a) and dynamic (b) imbalance models.	61
6.6 Non-linear simulator in simulink file: <code>Spacecraft_LFT_LPV_generation.mlx</code>	62
6.7 Reaction wheel waterfall plot of the disturbance torque measured in the wheel reference frame. Note the nutation modes of the wheel isolator varying with the wheel speed.	63
6.8 Friction model, Saturation and friction compensation filter	63
6.9 Torque spike generation function	65
6.10 Time history of the Torque spike (top figure); Time of the counter and the spike duration compared to show that a torque spike is initiated when these two quantities are equal (bottom figure)	66
6.11 Distribution of torque spike magnitude (top figure); Cumulative density function (bottom figure)	67
6.12 Distribution of the repetition time (top figure); Cumulative density function (bottom figure)	67
6.13 Reaction wheel friction and friction estimate	69
6.14 Reaction wheel friction and friction estimate with RW-1 presenting 2 rate reversals	69
6.15 RCS thrusters model	70
6.16 Signal modulated with PWM in a sampling period T	72
6.17 MPS thrusters model	74
6.18 Tank and sloshing equivalent mechanical model	75

6.19 Payload System (Telescope structure, FSM, PMA system, Isolator)	77
6.20 Simulator subsystem: SENSORS	80
6.21 Check for completion of the slew phase	83
6.22 Check for initializing the transition between coarse transient and coarse pointing phase	84
6.23 Check for coarse pointing completion	84
6.24 Check for initializing the transition between the fine transient and the fine pointing phase	85
6.25 Non-linear simulator: CONTROLLERS subsystem	94
6.26 Subsystem SLEW	95
6.27 Simulink bus for outputting only specific measurements of the <i>meas</i> bus	95
6.28 Subsystem TRANSITIONSLEW2COARSE	97
A.1 (a) <i>i</i> -th flexible appendage of a complex sub-structured body; (b) Compact representation of plant $\mathcal{M}_{PC}^{\mathcal{L}_i}$ (image credit: [44]).	102
B.1 Time dependency of pointing errors - Image credit: [AD-03]	105

List of abbreviations and acronyms

The following table provides the list of acronyms and abbreviations used in this document.

Acronym	Description
AAT	Analytical Analysis Tool
AOCS	Attitude and Orbital Control System
APE	Absolute Pointing error
APM	Antenna Pointing Mechanism
CMG	Control Moment Gyro
CMS	Component Mode Synthesis
DAE	Differential-Algebraic Equation
ESA	European Space Agency
EMM	Equivalent Mechanical Model
FE	Finite Element
FEEP	Field Emission Electric Propulsion
FSM	Fast Steering Mirror
GA	Genetic Algorithm
GNC	Guidance Navigation and Control
HGA	High Gain Antenna
IQC	Integral Quadratic Constraint
LFT	Linear Fractional Transformation
LMI	Linear Matrix Inequality
LOS	Line-of-Sight

Acronym	Description
LPV	Linear Parameter Varying
LTI	Linear Time Invariant
LTV	Linear Time Varying
MC	Monte-Carlo
MIMO	Multi-Input Multi-Output
NINOP	N-Input N-Output Port
PDE	Performance Drift Error
PMA	Proof-Mass Actuators
PWM	Pulse-Width Modulation
RCS	Reaction Control System
RMS	Root Mean Square
ROA	Region of Attraction
ROM	Reduced Order Model
RP	Robust Performance
RPE	Relative Pointing Error
RS	Robust Stability
RW	Reaction Wheel
RWA	Reaction Wheels Assembly
SA	Solar Array
SADM	Solar Array Drive Mechanism
SAT	Simulation Analytical Tool
SDP	SemiDefinite Program
SISO	Single-Input Single-Output
SOS	Sum of Squares
SRP	Solar Radiation Pressure
TITOP	Two-Input Two-Output Port
TM	Transfer Matrix
ToC	Target of Opportunity

Nomenclature

Unless stated otherwise, symbols are used according to the following conventions: scalars are represented by lower-case italics like a ; vectors by bold-italics lower-case letters like \mathbf{a} ; matrices by bold capital letters like \mathbf{A} ; and sets or specific operators are denoted by calligraphic letters like \mathcal{F} .

Math notation

\mathbf{A}^\dagger	Moore-Penrose pseudoinverse of \mathbf{A}
$\text{diag}(\dots)$	block diagonal matrix
Im / Ker	image/kernel of a linear transformation (matrix)
\dim	dimension of a vector space
$\text{rank}(\mathbf{A})$	rank of a matrix \mathbf{A}
$\text{sign}(x)$	signum function of a real number x
\otimes, \odot	Kronecker, quaternion product
$\mathbf{P} \succ 0 (\mathbf{P} \prec 0)$	means that \mathbf{P} is a real symmetric and positive (negative) definite matrix
$\ \cdot\ $	2-norm of a vector or the induced matrix 2-norm
$\ \cdot\ _{lp}$	\mathcal{L}_p -norm of a signal, $p \in \{1, 2, \infty\}$
$ \cdot $	absolute value of a scalar
$\{\dots\}$	set
$E\{\cdot\}$	Expected value
$\mathbf{a} \cdot \mathbf{b}$	dot (scalar) product of vectors \mathbf{a} and \mathbf{b}
$\mathbf{a} \times \mathbf{b}$	cross (vector) product of vectors \mathbf{a} and \mathbf{b}
δ_{ij}	Kronecker's delta
$\mathcal{U}(a, b)$	is the uniform distribution with boundaries a and b
$\mathcal{N}(\mu, \sigma^2)$	normal (Gaussian) distribution with mean value μ and variance σ^2
$(\mathbf{u})_\times$	skew matrix of vector \mathbf{u}
τ_u	$\begin{bmatrix} \mathbf{I}_3 & (\mathbf{u})_\times \\ 0 & \mathbf{I}_3 \end{bmatrix}$
$[\mathbf{v}]_{\mathcal{R}_b} = \mathbf{P}_{a/b} [\mathbf{v}]_{\mathcal{R}_a}$	The direct cosine matrix between two frames $\mathcal{R}_a = (O_a, x_a, y_a, z_a)$ and $\mathcal{R}_b = (O_b, x_b, y_b, z_b)$ contains the coordinates of the vectors (O_a, x_a, y_a, z_a) expressed in the reference frame \mathcal{R}_b

Chapter 1

Introduction

1.1 Context and Motivation

As reported by [1], guidance, navigation, and control of aerospace systems is growing more complex together with a need of increased system performance. Over the last decades, academic research developed several solutions to address this need in modeling, control and validation and verification (V&V) areas. What is lacking nowadays is: a systematic transfer of these technologies to the industrial world, the scaling up of these advanced analysis and control algorithms to complex industrial benchmarks and the training of the future control engineers to deal with control problems of industrial complexity. High pointing space missions [2] represent the perfect scenario in which GNC design encounters the hard task of coping with challenging high-level specifications and limitations imposed by the other spacecraft sub-systems (structure, thermal, optics, propulsion, etc.). In order to push the overall system to the limits of achievable performance, it is in fact necessary to predict in a preliminary design phase the worst-case configurations, due both to uncertainty in the system knowledge and mission operation conditions. NASA was a pioneer in the field of the integrated modeling philosophy. [3] were one of the first to investigate the fully coupled thermal/structure/control analysis problem. This philosophy was then employed in several NASA projects, from the Space Interferometry Mission [4], to the James Webb Space Telescope [5]. European Space Agency and industry investigated as well multi-disciplinary model-based system representation for end-to-end pointing performance characterization and optimization. Several missions, from the SILEX experience [6] to the recent Euclid Mission [7], benefited of this approach. The difficulty in practice of correctly conducting worst-case analysis for multi-disciplinary projects lies in their sequential approach, which is commonly followed in industry. Different tools are in fact employed for each area of expertise (GNC, structures, optics, etc.) and worst-case analyses are done in parallel in each field before reaching the final design. This results in many sub-iterations for the exchange

of evolving inputs/outputs and trade-offs among different departments and consequent re-validation [8]. A unique user-friendly tool able to incorporate all possible plant configurations (dictated by the particular mission) is then needed to control engineers in order to shortcut this process, synthesize robust control strategies and fast validate system stability and performance.

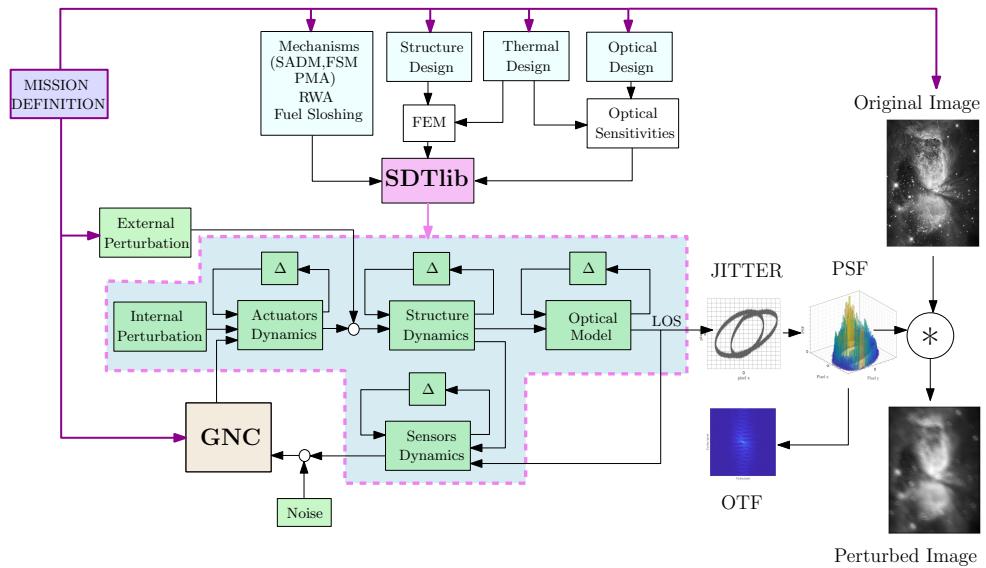


Figure 1.1: Integrated Design Logic. Hubble Image: ©ESA/NASA.

With this philosophy in mind, a complex multi-body spacecraft model was developed with the Satellite Dynamics Toolbox library (SDTLIB) [9, 10] as a Linear Fractional Transformation (LFT) [11] model by analytically including any uncertain and variable physical parameter. By having a look to Fig. 1.1, structural components coming from analytical models or numerical Finite Element Model (FEM) analysis (imported from NASTRAN) can be directly connected together with mechanism models (like Solar Array Drive Mechanism (SADM), Reaction Wheel Assembly (RWA), Fast Steering Mirror (FSM), etc.), sloshing phenomena, chemical propulsion and analytical optical models. Variation and uncertainties of the overall system can be easily taken into account to model:

- uncertainties coming from preliminary hypotheses on mission design parameters,
- uncertainties coming from not guaranteed FEM validation on Earth in non-operative conditions (no microgravity, simulation of space environment on Earth),
- uncertainties coming from misknowledge of sensors and actuators models,
- varying parameters like orientation of flexible appendages or speed of rotating elements (to correctly take into account gyroscopic effects).

For the mechanisms, it is also possible to take into account internal disturbances like static and dynamic imbalances of reaction wheels and harmonic perturbations due to SADM stepper commands, which have to be taken into account for microvibration compensation. Another feature available in SDTLIB is the possibility to build models with the minimum number of occurrences of uncertain and variable parameter by construction and to pre-process the final model (ready for control synthesis and analysis) in order to eliminate non-physical states and reduce the model complexity. This functionalities are particularly interesting when the model has to be used for robust control synthesis (H_2/H_∞ - synthesis) or formal robust stability/performance analysis (μ or IQC analysis).

1.2 Scope and purpose

This document constitutes the user manual of the GENERIC SATELLITE SIMULATION & ANALYSIS TOOL and aims at providing the user with the guidelines for the software usage, including the definition of the spacecraft LFT/LPV model according to the toolbox SDTLIB, the generation of the synthesis models and the set-up of the non-linear time domain simulation and results analysis.

The software and the benchmark problem provided with it are constituted of:

- A scalable analysis and synthesis model, specifically an LFT model of a complex telescope space mission generated with SDTLIB [12], which enables to perform tasks at various levels of granularity.
- an open equivalent non-linear high-fidelity simulator developed in Simscape and validated with SDTLIB for time-domain V&V analysis. It provides a validation platform to analyze proposed controlled system laws in a more complex scenario by including transitory behaviours, complex phenomena like friction and strong nonlinearities like saturations or PWM signals. An overview of the elements composing the high-fidelity model, which will enable the simulation of the mission scenario and the assessment of the user's proposed control architecture, as well as support other potential activities, is presented in Fig. 1.2.

The benchmark follows a multi-physics modelling approach allowing multi-granularity modelling of various substructure elements (these range from high fidelity, medium fidelity, low fidelity model). The user is required to set the desired granularity of the model. Based on the user's inputs, the corresponding linear and Simscape non-linear models will be automatically generated. For example, users can select a model for synthesizing either slew/coarse or fine pointing controller, choose from available sensors and actuators, and set the fidelity level for the nonlinear simulator. For the reaction wheels, one can choose to include static and dynamic imbalances, saturation, friction, and/or friction spikes (see for instance this approach for the RW model implementation in Fig. 1.3). To achieve

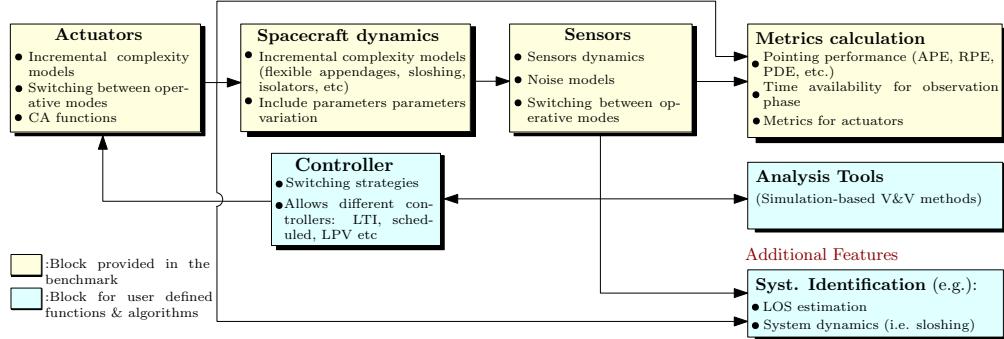


Figure 1.2: High-fidelity simulation environment schematics

modularity and allow the activation/deactivation of functions in the simulation environment, the simulator will feature the use of simulink *variant subsystems* and/or *switch* blocks.

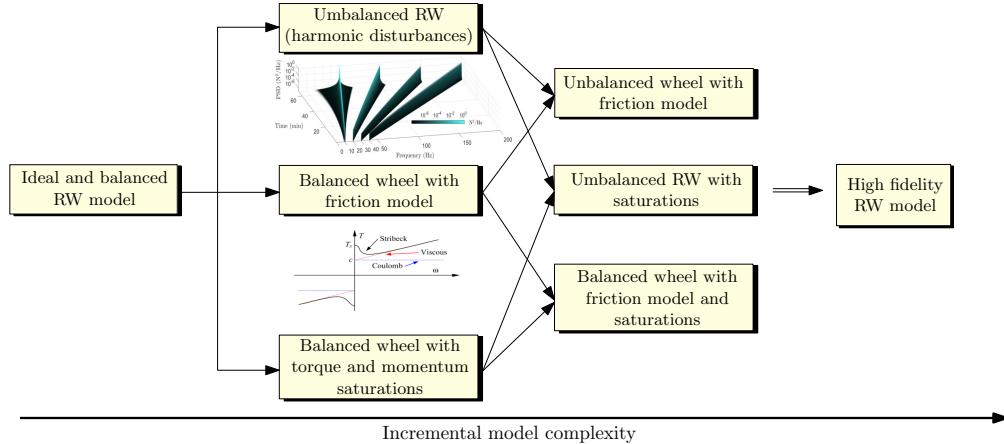


Figure 1.3: Schematics of an incremental complexity approach applied for modelling reaction wheels

The ambition of this work is to establish a reliable benchmark of industrial complexity and a robust simulation infrastructure, fostering innovation in V&V approaches and stimulating research and industry to push into the development and industrialization of new GNC V&V methods, tools, and techniques, addressing the complex challenges in Satellite GNC Design and V&V.

The use case is centered around a high-performance telescope Science mission. The benchmark problem is detailed in Chapt. 3.

Potential applications: The optical benchmark can be used for several research problems and applications as listed below (not exhaustive list):

- End-to-end control synthesis and analysis process for stringent pointing requirements as proposed in [1];
- Parametric sensitivity analysis as proposed in [13] in order to reduce the number of most impacting uncertainties on closed-loop stability and performance analysis and better understand the physical meaning of possible worst-case configurations;
- Deterministic or probabilistic μ -analysis as proposed respectively by [14] and [15]; and with application to flight data in [16];
- Integral Quadratic Constraints (IQC) analysis as in [17] and [18];
- Non-linear validation algorithms based on global optimization techniques as proposed by [19];
- Testing controllers and V&V strategies for complex set of physical uncertainties, varying parameters and strong non-linearities like saturations, PWM signals or RW friction spikes;
- Classical fully non-linear Monte-Carlo V&V campaigns;
- Multi-disciplinary optimization as proposed by [8];
- Line-of-sight estimation with different combinations of sensors/actuators as in [20];
- Rigorous analysis ([21]) of transition between two different control modes (switching);
- Strategies to reduce time overheads, such as minimizing tranquilization periods and shortening transition phases;
- Exploration of new architectures for actively controlling the spacecraft during Science phase without stopping disturbance sources (like reaction wheels) in order to optimize the science time window as shown in [22];
- Performance analyses of hybrid continuous/discrete time linear multi-rate control systems in the presence of parametric uncertainties [23];
- Generation of time domain data for any other applications (i.e. data driven control, system identification).

1.3 Applicable Documents (ADs)

The following documents, listed in order of priority, contain requirements applicable to the activity:

Number	Title	Reference	Issue
AD-01	Satellite dynamics toolbox for preliminary design phase	Alazard D., Sanfedino F., In: 43rd Annual AAS Guidance and Control Conference. 2020. p. 1461-1472.	1.0
AD-02	Satellite Dynamics Toolbox library (SDTlib) - User's Guide	Alazard D., Sanfedino F., Satellite dynamics toolbox library (sdplib)-user's guide. 2021.	1.0
AD-03	ESA Pointing error engineering handbook	ESA, ESA Pointing error engineering handbook, ESSB-HB-E-003, European Space Agency, 2011	1.0
AD-04	Space Engineering – Control Performance	ECSS-E-ST-60-10C, 15/11/2008	1.0

1.4 Document Outline

This user manual (UM-02), attached to simulator-02, provides external users with the guidelines for the software usage, including the generation of synthesis models (extracted from provided spacecraft LFT/LPV model) and the set-up of the non-linear time domain simulation and analysis of the results. This document is organized as a step-by-step guide through the various features of the software to help users quickly get started.

The document is organized as follows:

- Chapt. 2 briefly describes the content of the benchmark folder and the required MATLAB configuration and package to run the software.
- Chapt. 3 details the definition of the benchmark problem, including the space-craft model, the specific timeline sequence of the problem, available actuators and sensors, and mission requirements.
- Chapt. 4 briefly presents how the spacecraft LFT/LPV model, which the user will use to synthesize and test controllers, was built using the SDTLIB toolbox, along with the non-linear model defined with SIMULINK and the SIMSCAPE MULTIBODY library.
- Chapt. 5 covers the description of the codes used for the generation of the synthesis models.
- Finally, Chapt. 6 provides a guide on setting up time-domain analysis (with specific parameter configurations or Monte Carlo runs), modifying the simulator to accommodate controllers developed by the user, and analyzing the benchmark results.

Chapter 2

Description of Simulator-02

2.1 Getting started

The selection of Matlab/Simulink as the main modeling toolbox for the definition of the benchmark is to make it practical and available to the majority of the GNC workforce. Indeed, according to the results of the questionnaire proposed during the inter-agency workshop on GNC V&V, half of respondents indicated they mainly use MATLAB/Simulink (48%) for GNC design and verification, with in-house tools (26%), open-source software (15%), agency-owned legacy software (5%), and other tools (5%) trailing behind [24].

2.1.1 Installation and requirements

The required MATLAB configuration and package for the software are the following:

- MATLAB R2021A (or later); the software has been tested on R2021A and, although less extensively, on MATLAB R2023A.
- CONTROL TOOLBOX
- ROBUST TOOLBOX
- SIMULINK
- SIMULINK CONTROL DESIGN
- OPTIMIZATION TOOLBOX
- AEROSPACE TOOLBOX
- PARALLEL COMPUTING TOOLBOX (if Monte-Carlo Analysis is run)
- REPORT GENERATOR (Optional)

Once the `USERS_BENCHMARKSAT_V01` folder is unzipped on the computer at `UserDirectory→Users_BenchmarkSat_V01`, the user can launch MATLAB and directly use the software.

It is possible, although not necessary, to add the folder and its subfolders to the path of MATLAB by:

1. right-clicking, while on MATLAB, on the folder and selecting:
Add to path → Selected Folders and Subfolders
2. type the following command in the Command Window:
`addpath(genpath('Users_BenchmarkSat_V01'))`

2.2 Directory structure

An illustrative representation of the directory structure of the toolbox is provided through the tree diagram in Fig. 2.1. The main functions of the software are underlined and placed hierarchically under the main folder. A brief description of these functions is outlined in Table 2.2 with detailed explanations to follow in subsequent sections.

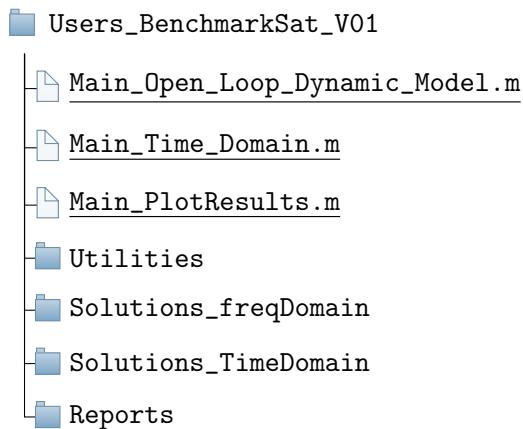


Figure 2.1: Content of the main directory

These main functions call for the routines and simulink files contained in the other subfolders.

The folder *Solution_freqDomain* is used to save open-loop dynamic models generated by the user with the script *Main_Open_Loop_Dynamic_Model.m*. The folder *Solution_TimeDomain* is used to save results of non-linear time simulations generated by the user with the script *Main_Time_Domain.m*. The folder *Reports* is used to save results in PDF format both for open-loop models generated by the user with the script *Main_Open_Loop_Dynamic_Model.m*. and non-linear time simulations generated by the user with the script *Main_Time_Domain.m*.

Finally the Utilities folder is organised as shown in Fig. 2.2.

Table 2.1: Description of the main functions of the software

Name	Brief Description
<i>Main_Open_Loop_Dynamic_Model.m</i>	Matlab script that generates the LFT/LPV open-loop dynamic model (that can be used for control synthesis) according to the selected phase of the missions and the actuators/sensors selected.
<i>Main_TimeDomain.m</i>	Matlab code for launching time-domain simulation (in nominal configuration, with a provided parameter configuration or Monte Carlo run) for the complete timeline of the benchmark problem.
<i>Main_PlotResults.m</i>	This code provides the results of the time-domain analysis. According to the provided inputs the codes manages how the results are visualized. The user can provide the name of the .mat containing the results (up to 2 models with a single model config.) and 1 Monte Carlo run.

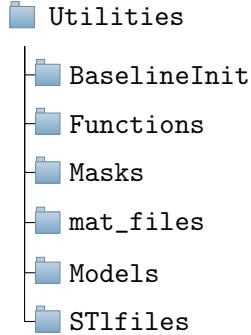
Figure 2.2: Content of the directory *Utilities*

Table 2.2: Description of the main functions of the software

Name	Brief Description
<i>BaselineInit</i>	This folder contains all data and files to run simulations with the ISAE-SUPAERO baseline controller. The user can plug its own control solution by modifying the function <i>Baseline_Controller.m</i> . It is not advised to modify any other script, function or data file in this folder.
<i>Functions</i>	This folder contains all core routines used for the benchmark. It is advised to make any modification in this folder.
<i>Masks</i>	This folder contains all icons used for in the Simscape simulator. It is advised to make any modification in this folder.
<i>mat_files</i>	This folder contains various data used in the benchmark. It is advised to make any modification in this folder.
<i>Models</i>	This folder contains the LFT/LPV model built with SDTlib and its Simscape non-linear simulator counterpart. It is advised to make any modification in this folder.
<i>STLfiles</i>	This folder contains the STL files used by Mechanics Explorer for visualization. It is advised to make any modification in this folder.

Chapter 3

High pointing benchmark study definition

Contents

3.1	Definition of the Spacecraft model	14
3.1.1	AOCS actuators and sensors	14
3.1.2	Actuators/Sensors at payload level	16
3.1.3	AOCS and LOS stabilization architectures	16
3.1.4	Benchmark timeline	18
3.2	Objectives and constraints	20
3.2.1	Constraints on actuators	21
3.3	Disturbances	22

This chapter covers the general description of the use case, centered around a high-performance optical payload. The benchmark is formulated specifically to comply with:

- testing different control strategies of coarse/fine pointing control by playing with different sets of actuators and sensors or their arrangement in the system architecture;
- investigating switching control modes (i.e. coarse to pointing transition which could exhibit resonance at the controller switch) and analysing reduction strategies of time overheads such as long tranquilization time and duration of transition phases;
- testing controllers and V&V strategies for complex set of physical uncertainties, varying parameters and strong non-linearities like saturations, PWM signals or RW friction spikes.

3.1 Definition of the Spacecraft model

A schematic of the proposed spacecraft model is shown in Fig. 3.1. It comprises of a central bus \mathcal{B} connected to two rotating flexible solar arrays $\{\mathcal{A}_1, \mathcal{A}_2\}$ and a High-Gain Antenna (HGA) flexible boom \mathcal{A}_3 at the interface point A_1 , A_2 and A_3 respectively and an optical payload system \mathcal{P} . The optical payload is composed of a flexible structure connecting the optical elements: the two mirrors M_1 and M_2 , the Charge-Coupled Device (CCD) and a Fast Steering Mirror (FSM). This payload is linked to the main body through a passive/active isolator¹. The isolation assembly (\mathcal{IA}) interfaces the payload with the spacecraft bus with the objective of reducing the micro-vibration transmission from disturbance sources (RWs and SADMs) to the instruments. The type of isolation architecture (passive or active via PMAs) is a design choice that the end user of the benchmark will have to address in order to effectively tackle the fine pointing requirements of the observation phase. Finally, a tank is supposed to be connected to the spacecraft hub at the interface note I_s to account for the effect of sloshing on the system's dynamics via an equivalent mechanical model.

3.1.1 AOCS actuators and sensors

For AOCS, the system is equipped with:

1. *A set of 4 reaction wheels in pyramidal configuration:* for simplicity, the scheme in Fig. 3.1 reports the RWs clustered together and connected via a single point to the spacecraft bus \mathcal{B} . However, in the benchmark study each RW will be connected to body \mathcal{B} in a different point (each with its own passive isolation platform). The benchmark will also offer the opportunity to exclude any passive isolation system between RWs and the spacecraft bus.
2. *A Reaction Control System (RCS):* The RCS is constituted by a clusters of thrusters that typically provide higher control authority over the spacecraft attitude with respect to RWs. The use of thruster can prevent/reduce the impact of microvibrations produced by RWs. The pointing levels that can be achieved depend on the thruster control authority and accuracy which is driven by the minimum impulse bit level; furthermore, these are often operated using PWM strategies thus introducing nonlinearities and limit cycle oscillations.

¹Passive isolation typically makes use of visco-elastic materials, springs or hydraulic dampers isolators. The main drawback of using only a passive solution is that the problem of microvibrations is mitigated at very high frequency while consistent perturbations stay in the mid-frequency range. Moreover, isolators introduce supplementary underdamped flexible modes at low frequency. Active isolation solutions rely on the use of actuators that directly provide the action to actively counteract microvibrations in the middle-range frequency band [25, 26]. Hybrid passive/active control solutions can result thus mandatory for mission where a broadband pointing accuracy is demanded.

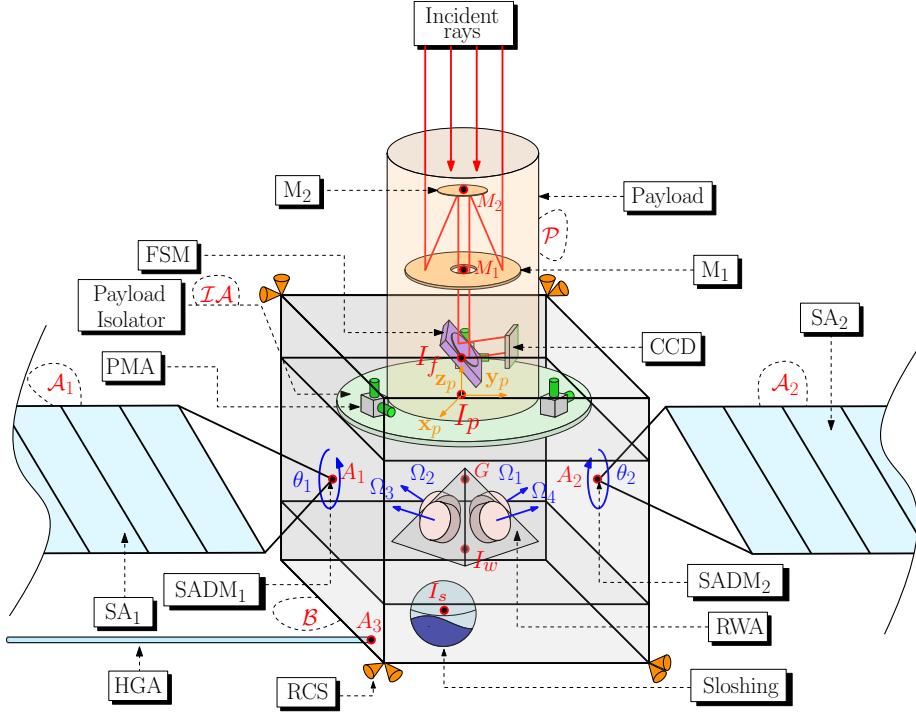


Figure 3.1: Schematics of the proposed spacecraft architecture for the benchmark problem

3. *A Micro-Propulsion System (MPS)*: MPS architectures are ideally suited for a wide range of precise micro-propulsion applications, spanning from a few micro-Newton (μN) to a few milli-Newton (mN) with precise thrust control resolutions ($< 0.1 \mu\text{N}$).

The AOCS sensors suite comprises:

- a *star tracker (STR)* system for attitude measurements.
- a *gyro (GYR)* system for angular velocity measurements. Two gyro assemblies are made available in the benchmark: a coarse pointing gyro (denominated as GYR-c) and an high performance one (GYR-f) that could be used during fine pointing operations.
- a *Fine Guidance Sensor (FGS)* which provides very precise attitude measurements necessary to cope with the tight pointing requirements needed during observations. The FGS is generally positioned near the main instruments and shares its LOS in order to limit as possible thermo-elastic deformations that can worsen the LOS alignment between the instrument and FGS. The FGS is characterized by a reduced

field of view and long image exposure time to provide a sufficient Signal-to-Noise Ratio in the measurements for star detection.

Note that Coarse and Fine Sun Sensors are not included in the model since they are generally used to acquire and maintain sun pointing for the solar panels, avoid forbidden zone of the telescope during non-nominal operations and anomaly detection.

3.1.2 Actuators/Sensors at payload level

At the payload level the system is equipped with the following actuators:

1. A *Fast-Steering Mirror (FSM)*: These actuators are composed of a lightweight mirror attached to a steerable base that can rotate around two axes [27] and are placed on the optical path of the payload, directly acting on the degraded LOS.
2. A system of 6 *Proof-Mass Actuators (PMA)*: Another possible micro-vibration mitigation architecture relies on proof-mass actuators (PMA). PMAs are mechanisms that can produce a periodic force along their principal axis. Multiple PMA can be placed and oriented in a particular configuration to provide controllability over the desired number of DoFs of a body as in [28].

and sensors:

1. Three 6-DoFs *accelerometers units* are considered situated at the payload level (one on the primary mirror, one on the secondary and one in the proximity of the detector), Each accelerometer is supposed to provide both linear and angular acceleration measurements.
2. *FSM x/y deflection measurements* (strain gauges) providing the two rotation angles of the FSM.

3.1.3 AOCS and LOS stabilization architectures

Throughout the entirety of the previous section, there has been voluntarily a notable absence of a detailed descriptions of the architectures employed for AOCS and LOS stabilization architectures in each of the 3 time-domain windows (slew, coarse and fine pointing); indeed, the benchmark will be established with the aim of providing the end user with a certain degree of autonomy in the selection of the architectural design, encompassing not only control laws but also the selection of actuators and sensors². Table 3.1

²A baseline architecture in terms of actuators/sensors, controllers and V&V analysis method will be provided by ISAE-SUPAERO.

and 3.2 respectively report the set of actuators and sensors that the end user will be able to activate during each phase of the benchmark.

As far as actuators are concerned, RWs are available for providing attitude control during the first two phases of the mission, while the RCS can be activated only during slew maneuvers. In the last phase of the mission, due to the stringent pointing requirements only the Micro-Propulsion system or the RWs can be activated for AOCS potentially augmented with LOS stabilization systems such as PMAs for active payload isolation and FSMs. During the last phase, it is also imposed that the RW must be kept at constant non-zero speed. It is imposed that they cannot be spun down to a stand still since this operation negatively impact the total time allocated for the science observation. Furthermore, this constrain stimulates the implementation of different architectures and control strategies.

Table 3.1: Available actuators for AOCS and LOS stabilization during benchmark phases

Actuators	Slew	Coarse Pointing	Fine Pointing
Reaction Wheels (RWs)	✓	✓	✓
Reaction Control System (RCS)	✓	✗	✗
Micro-Propulsion System (MPS)	✗	✗	✓
Fast Steering Mirrors (FSMs)	✗	✗	✓
Proof Mass Actuators (PMAs)	✗	✗	✓

Table 3.2: Available sensors for AOCS and LOS stabilization during benchmark phases

Sensors	Slew	Coarse Pointing	Fine Pointing
Star Tracker (STR)	✓	✓	✓
Gyroscope (GYR)	✓	✓	✓
Fine Guidance Sensor (FGS)	✗	✗	✓
FSM strain gauges	✗	✗	✓
Accelerometers at payload level	✗	✗	✓

As far as the gyroscopes' selection is concerned, two possible gyroscopes with different characteristics can be selected in the GNC system: 1) a coarse gyro, indicated as GYR-c, and a high-performance one, denoted as GYR-f. Typically, if requirements can be met with the GYR-c, such option should be preferred over the GYR-f due to the generally

much higher cost of the second device.

An additional set of sensors is added to the previous list of Table 3.2 and it is the reaction wheel tachometer which is used to monitor the RW rate and to implement a friction compensation filter. The data associated to the various sensors (Noise characteristic and bandwidth) are reported in Sect. 6.2.2.

3.1.4 Benchmark timeline

Within the various operational modes of the spacecraft, a particular problem is specifically designed as a use case. Fig. 3.2 depicts the timeline for the proposed use case, outlining a sequence of operations designed to reach the final science phase which requires precise pointing accuracy and stability to enable observations of a specific target.

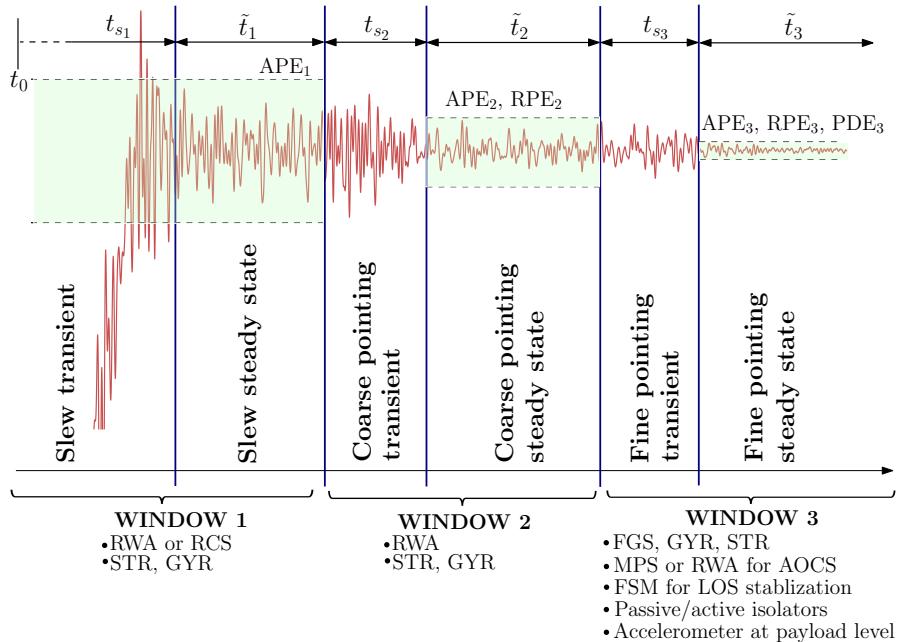


Figure 3.2: Timeline sequence for the benchmark problem

An overview of the phases is reported here:

- **Window 1 - slew transient:** The timeline of the proposed benchmark starts at time t_0 and it is initially assumed that the spacecraft performs a slew maneuver to point the payload LOS to the target. Such transient shall be optimized in terms of agility, reduction of the tranquilization time and final attitude error (APE_1).
- **Window 1 - slew steady state:** A first APE requirement (APE_1) has to be met during this phase in order to provide the conditions for the mode switching and

the transition to the coarse pointing phase. The duration of the slew steady state is considered fixed and equal to \tilde{t}_1 .

- **Window 2 - Coarse pointing transient:** After the specification APE₁ is verified at the end of the slew, the mode switching to the control architecture for coarse pointing control is initiated. This transient ends when the spacecraft fulfill the requirements for coarse pointing steady state operations.
- **Window 2 - Coarse pointing steady state:** The coarse pointing phase is envisioned as an intermediate phase in order to provide the sufficient pointing conditions for the initial hand over from star trackers to the FGS and subsequent transition from coarse to fine pointing control architecture. In this phase requirements are provided both in terms of instantaneous and window error indexes (APE₂ and RPE₂). The duration of the coarse pointing steady state is considered fixed and equal to \tilde{t}_2 .
- **Window 3 - Fine pointing transient:** In this phase the hand over to the fine pointing control architecture is initiated. In this scenario a combination of actuators and sensors at payload level can be incorporated in the control architecture to guarantee the required pointing performance for the optimal execution of the science observation.
- **Window 3 - Fine pointing steady state:** The fine pointing phase (science observation phase) presents the most challenging aspects for the AOCS and LOS stabilization system in terms of pointing requirements and complexity of the overall control architecture. Indeed, besides more stringent instantaneous and windowed requirements (APE₃ and RPE₃) with respect to the previous phase, a windowed-stability requirements in the form of Performance Drift Error (PDE) is imposed during science observation (PDE₃). The duration of the fine pointing steady state is considered equal to \tilde{t}_3 (not fixed).

CHALLENGE YOURSELF!

The total time for the previous sequence is fixed and a baseline controller is provided to the user, who can directly plug his solution into the simulator instead. The challenge for the user will be to maximize the science phase, i.e the fine pointing steady state duration \tilde{t}_3 . This means that the user has to propose a solution that optimizes the slew maneuver and the transitions between modes by coping with all uncertainties of the system and by facing non-linearities like saturations, reaction wheels imbalances, Stribek and spike frictions, SADM microstepping signals, PWM signals for thrusters and all measurement noises. How the transition is implemented in the non-linear simulator between the various phases (Slew to Coarse pointing and then Coarse to fine pointing) is discussed in Sect 6.4. Besides the proposed problem, users have the ability to explore with this benchmark several other topics and applications as discussed in 1.2.

3.2 Objectives and constraints

The pointing requirements and objectives of the benchmark problem are given as follows:

- POINTING REQUIREMENTS: the pointing requirements, according to the definition given in Appendix. B, are listed in Table 3.3.

Phase	Error Index	ϵ_{metric}	Δt [s]	Δt_s [s]
Slew Maneuver	APE ₁	[0.2 , 0.2 , 0.2] deg	-	-
	APE ₂	[10 , 10 , 60] arcsec	-	-
Coarse Pointing	RPE ₂	[3 , 3 , 30] arcsec	10	-
	APE ₃	[3.5 , 3.5 , 50] arcsec	-	-
	RPE ₃	[0.3 , 0.3 , 10] arcsec	10	-
Fine Pointing	PDE ₃	[0.5 , 0.5 , 40] arcsec	10	150

Table 3.3: Pointing requirements

Throughout both transient phases, the design must ensure that the pointing errors do not exceed the requirements established in the previous phase. For instance, the APE₁ requirement should be kept throughout the switching between slew and coarse control architecture.

- STABILITY REQUIREMENT: The stability requirements are specified in terms of Gain Margin (GM) and Phase Margin (PM) and must be guaranteed for all combinations of uncertain and varying parameters inside the parameter space. The two stability requirements are reported as follows:

$$\begin{aligned} \text{GM} &\geq 6 \text{ dB} \\ \text{PM} &\geq 30 \text{ deg} \end{aligned} \tag{3.1}$$

Furthermore, the robust stability should also be assessed during transitions between controllers and sets of actuators/sensors.

In the baseline design proposed by ISAE-SUPAERO such requirements is addressed with an upper bound $\tilde{\gamma}$ given on the \mathcal{H}_∞ norm of the input sensitivity function. The selected value of $\tilde{\gamma}$ is 1.9 and ensures the following margin on each axis:

- a modulus margin $> 1/\tilde{\gamma} = 0.526$;
 - a gain margin $> \frac{\tilde{\gamma}}{\tilde{\gamma}-1} = 2.11$ (6.49 dB);
 - a phase margin $2 \arcsin \frac{1}{2\tilde{\gamma}} = 30.52$ deg
3. SCIENCE PHASE DURATION: In the evaluation of the proposed control solutions for the benchmark problem, another vital aspect to consider is the maximization of the allocated time available for the spacecraft to carry out its scientific observations. Supposing a fixed duration for the full timeline, this objective is achieved by reduction of the overall duration of the transients (t_{s_1} , t_{s_2} and t_{s_3}). The duration of the slew transient can be improved by increasing the agility and reducing the tranquilization time of the maneuver while coarse pointing transient and fine pointing transient duration (t_{s_2} and t_{s_3}) can be reduced by optimizing the controller switching strategy.

3.2.1 Constraints on actuators

Table 3.4 reports the constraints on the actuators selected for the benchmark problem.

Table 3.4: Actuators limits

Actuator	Max. command/Saturations
Reaction Wheels (RWs)	Max. torque: 0.2 Nm Max. wheel rate: 4000 RPM
Reaction Control System (RCS)	Max. force: 1 N
Micro-Propulsion System (MPS)	Max. force: $0.1 \mu\text{N}$
Fast Steering Mirrors (FSMs)	Max. rotation: 1 mrad
Proof Mass Actuators (PMAs)	Max. force: 50 N Max. displacement: 5 cm

3.3 Disturbances

The external/internal disturbances acting on the spacecraft are listed in Table 3.5 and are discussed in later sections of the document (the section the user has to refer is reported in the table).

Table 3.5: Disturbances contribution included in the non-linear simulator

Model	Contribution
Orbital disturbance	Sinusoidal signals (see Sect. 6.2)
	Imbalances (1 st harmonic disturbance) (see Sect. 6.2.1.2.1)
	Broadband disturbance (see Sect. 6.2.1.2.2)
Reaction wheels	Deterministic friction (See Sect.6.2.1.2.3)
	Friction spikes (See Sect.6.2.1.2.3)
	Torque noise (see Sect. 6.2.1.2.2)
	Saw-tooth signal (microstepping) (see Sect. 6.2.1.1)
SADM	Broadband disturbance (see Sect. 6.2.1.1)

Chapter 4

Tutorial I: Spacecraft dynamical model

Contents

4.1 Overview of SDTLIB spacecraft model	24
4.1.1 Reaction Wheel System (RWS)	28
4.1.2 Sloshing equivalent mechanical model	29
4.1.3 Reaction Control System (RCS)	29
4.1.4 Micro-Propulsion System (MPS)	31
4.1.5 Payload (Telescope structure, FSM, PMA, Isolator)	32
4.1.5.1 Payload isolator	33
4.1.5.2 Fast-Steering Mirror	33
4.1.5.3 Proof-Mass Actuator system (PMA)	34
4.2 Optical model	34
4.2.1 Telescope model	34
4.2.2 Telescope finite element model	35
4.2.2.1 Primary mirror: Centroid motion calculation	37
4.2.3 Ray-tracing algorithm for linear optical sensitivities	38
4.3 LFT/LPV model and data provided to the user	39

This chapter describes how the LFT/LPV model of the spacecraft, depicted in Fig. 3.1, has been built using the SDTLIB toolbox, along with the non-linear model defined with SIMULINK and the SIMSCAPE MULTIBODY library.

This chapters covers only the description of the mechanical system of the spacecraft both for the linear LFT/LPV model and its non-linear counterpart. The LFT/LPV model

will serve as the foundation for control synthesis and for the implementation and testing of Analytical Analysis Tools (AATs).

In order to follow this chapter the user shall refer to the following files:

- the simulink file:

`Simscape_Benchmark_Model.slx`

in folder:

`Users_BenchmarkSat_V01/Simscape_model`

with the non-linear SIMSCAPE MULTIBODY model.

- the .mat file:

`LFT_LPV_spacecraftModel_Users_2024-07-30_13h57m17s.mat`

in folder:

`Users_BenchmarkSat_V01/SDT_model`

containing the data for the generation of the spacecraft model and the LFT/LPV model in the form of an uncertain state space system `Gu.mat` (see Table C.1):

```
>> Gu
Uncertain continuous-time state-space model with 87 outputs, 39 inputs
, 220 states.
The model uncertainty consists of the following blocks:
CoMx: Uncertain real, nominal = 0, variability = [-0.01,0.01], 4
occurrences
CoMy: Uncertain real, nominal = 0, variability = [-0.01,0.01], 4
occurrences
...
w_Telescopestructure_5: Uncertain real, nominal = 2.32e+03,
variability = [-5,5]%, 2 occurrences
Model Properties

Type "Gu.NominalValue" to see the nominal value and "Gu.Uncertainty"
to interact with the uncertain elements.
>>
```

4.1 Overview of SDTLIB spacecraft model

The uncertain state space system `Gu.mat` is obtained by linearizing the simulink model in Fig. 4.1 which is built with the Satellite Dynamics Toolbox library (SDTLIB) [12] (NOTE: SDTlib is not provided with the software). Some details on the SDTLIB and the theoretical framework used in the library are reported in Appendix A.

The SDTLIB model of Fig. 4.1 is composed of the following sub-blocks (from the top to the bottom):

- the sub-block for the models of the flexible substructures (i.e. the two solar arrays and the HGA boom). In particular, each solar array is connected to the S/C hub

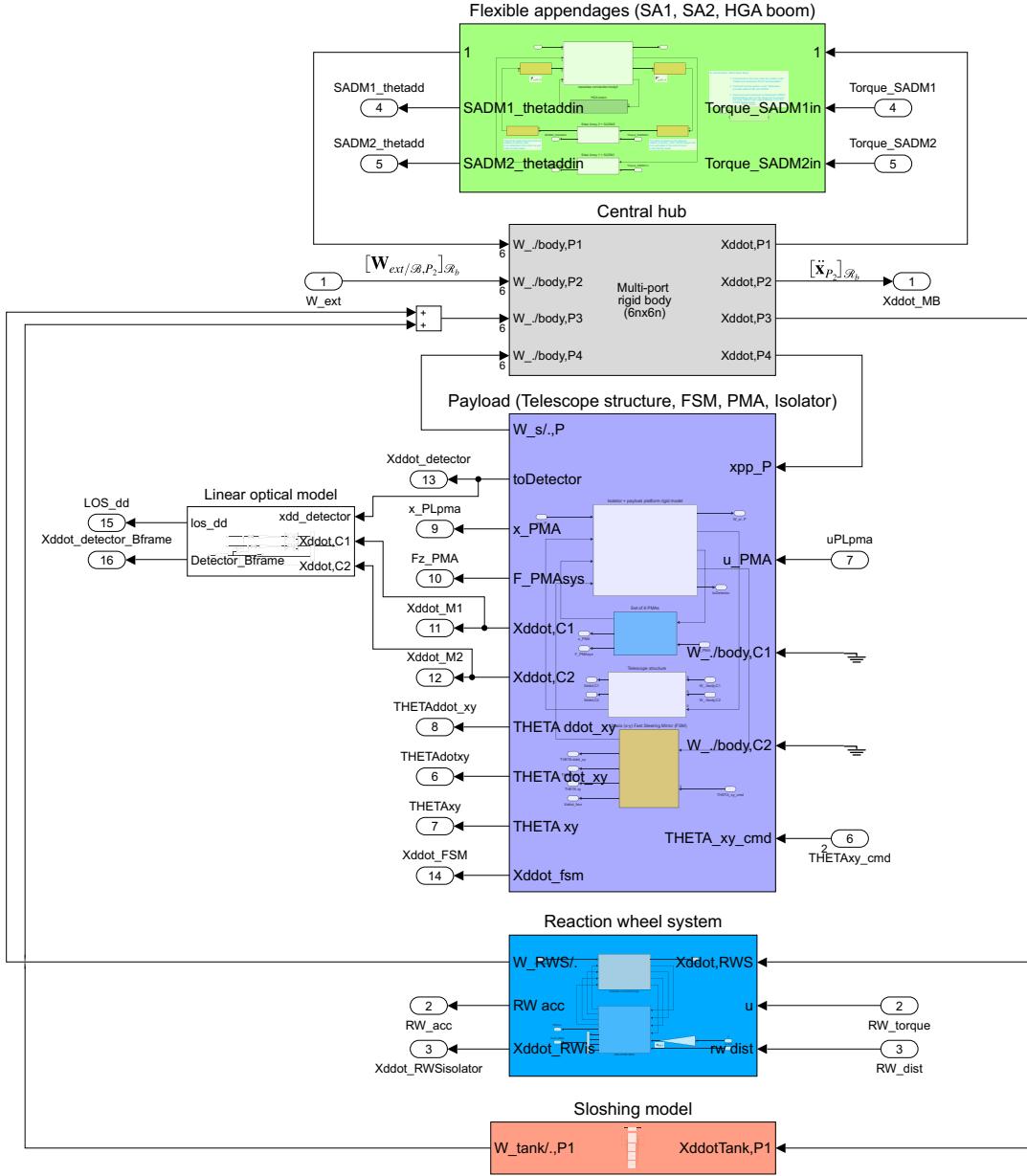


Figure 4.1: Visualization of the SDTlib model

via a SADM, which is modelled as a revolute joint with an input torque disturbance and a joint stiffness and damping. The angular configuration of the solar arrays constitutes one of the varying parameters of the LPV/LFT assembled spacecraft model (see Appendix A for further details on how rotations about a single axis are represented in SDTlib to obtain a minimal LFT). The telescope structure

(previously introduced in Sect. 4.2) is not included here as it is part of the payload subsystem.

- the sub-block for the rigid model of the S/C hub;
- the sub-block for the payload system which comprised of the flexible model of the telescope system, the payload isolator, the FSM and the PMA system actuator models (see Sect 4.1.5).
- the sub-block for the reaction wheel system. The model is composed of 4 RWs in a tetrahedral configuration with each RW connected to the S/C hub via an isolator.
- the sub-block for the sloshing model as discussed in Sect 4.1.2.

The equivalent SIMSCAPE MULTIBODY integrated model is shown in Fig. 4.2 and composed by:

- the sub-block **Flexible Appendages (Solar arrays + HGA)**;
- the sub-block **Bus** for the rigid model of the S/C hub.
- the sub-block **Payload System** includes the FEM flexible model of the telescope structure, the payload isolator, the FSM and the PMA system actuator models.
- the sub-block **WEMS / Reaction Wheels** for the reaction wheel system.
- the sub-block **Tank + Sloshing model** for the sloshing model.

Several tests have been conducted to cross-validate the SDTLIB LFT/LPV model against the non-linear simulator developed in Simscape.

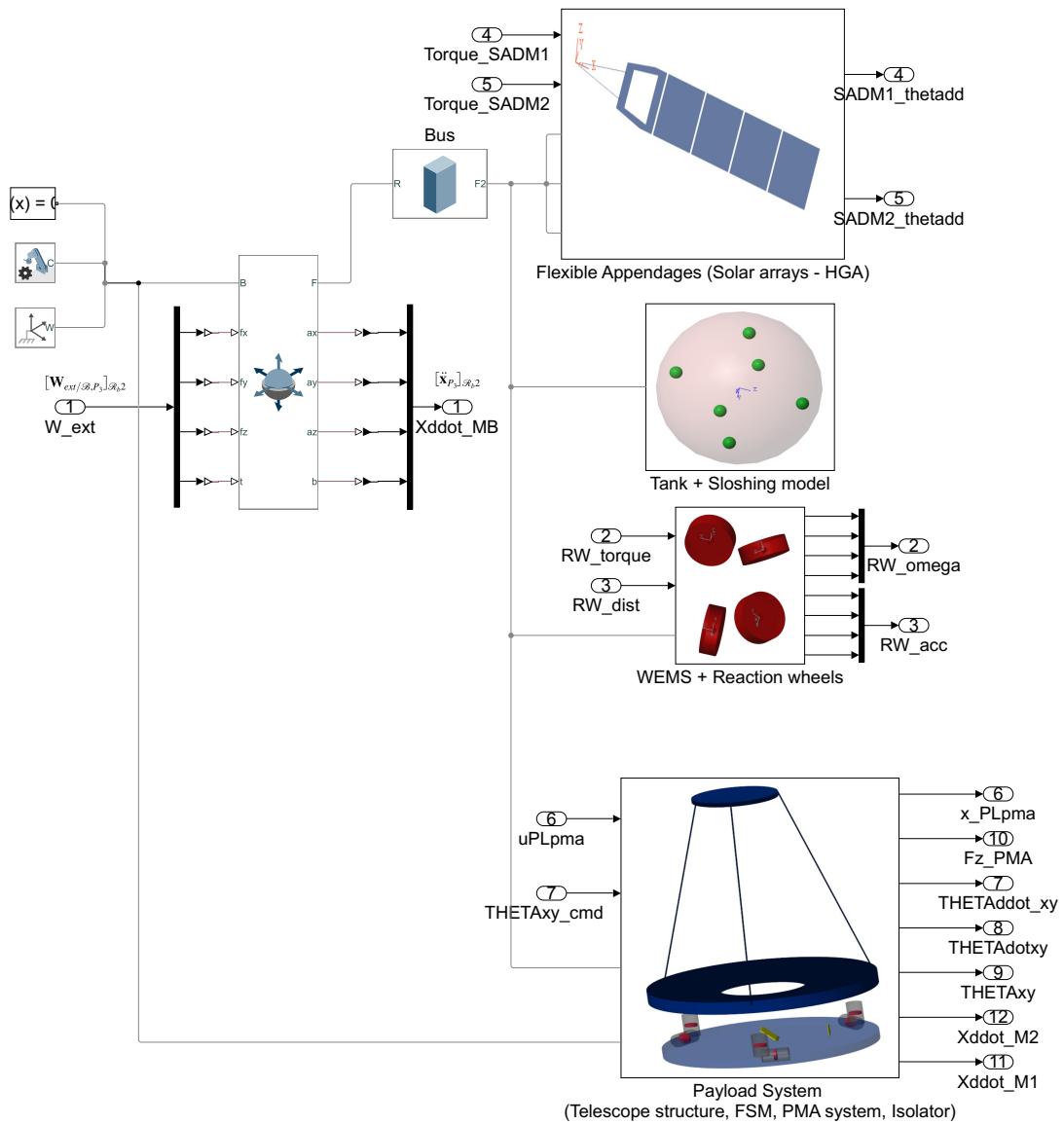


Figure 4.2: View of the assembled Simscape model of the spacecraft

4.1.1 Reaction Wheel System (RWS)

The RWS is composed 4 reaction wheels connected to the spacecraft hub by means of isolators. Each isolator is modelled as a massless spring-damper system with 6 DoFs.

The linear dynamic model of a reaction wheel, here defined as \mathcal{RW} , spinning at a given angular rate Ω and actuated by a torque input is briefly discussed. Let us consider the spinning wheel depicted in the Fig. 4.3 and characterized by the wheel mass m , the wheel axial and radial inertia I_w and I_r respectively, its spinning rate Ω around its \mathbf{z}_w , its centre of mass G and connection to a parent body at point P . Furthermore, the following reference frame are defined: 1) $\mathcal{R}_a = (O, \mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a)$ and 2) $\mathcal{R}_w = (G, \mathbf{x}_w, \mathbf{y}_w, \mathbf{z}_w)$ motionless with respect to \mathcal{R}_a . The input torque $-u$ is applied along the wheel axis by a driving mechanism. The wheel is assumed balanced, that is the inertia tensor $[\mathbf{J}_G^{\mathcal{RW}}]_{\mathcal{R}_w}$ is diagonal.

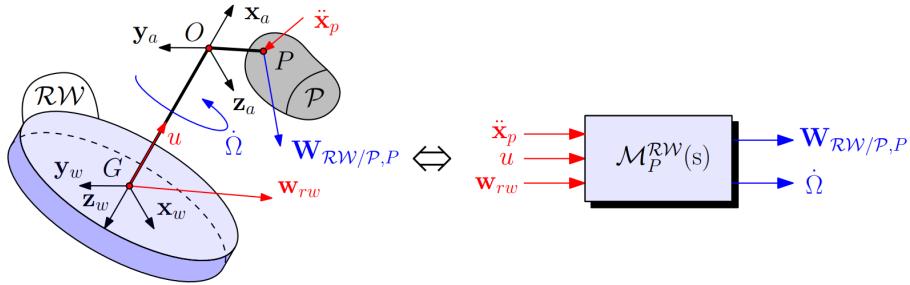


Figure 4.3: Single reaction wheel (left) and equivalent TITOP model (right)

To take into account the imbalances of the wheel responsible for harmonic disturbances, a vector of internal disturbances $\mathbf{w}_{rw} \in \mathbb{R}^5$, composed of the three components of the disturbing force (in the wheel frame \mathcal{R}_w) and two components of the radial disturbing torque, is considered acting at the center of mass of the wheel.

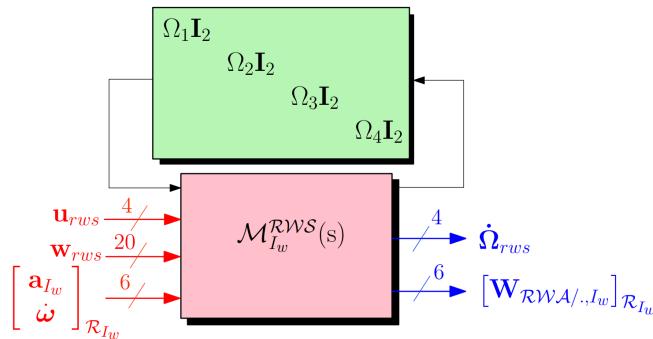


Figure 4.4: Equivalent LPV model of the RWS scheduled according to the 4 RW speeds

4.1.2 Sloshing equivalent mechanical model

The behaviour of fluid within a spherical tank is investigated by dividing the fuel in several particles according to the model developed in [29]. Each particle is connected to three sets of springs and dampers, where each of these sets is oriented along one of the three mutually perpendicular axes in three-dimensional space (x, y and z directions). The equilibrium positions of the six sloshing masses are the following:

$$\begin{aligned}\mathbf{x}_{s_1} &= [x_s \ 0 \ 0] & \mathbf{x}_{s_2} &= [-x_s \ 0 \ 0] \\ \mathbf{x}_{s_3} &= [0 \ x_s \ 0] & \mathbf{x}_{s_4} &= [0 \ -x_s \ 0] \\ \mathbf{x}_{s_5} &= [0 \ 0 \ x_s] & \mathbf{x}_{s_6} &= [0 \ 0 \ -x_s]\end{aligned}\quad (4.1)$$

Table 4.1 reports the data used for generating the sloshing model implemented in the benchmark study.

Table 4.1: Sloshing data

Parameter	Value
Particle mass	15 kg ($\pm 15\%$)
Tank + non-sloshing mass	150 kg
Stiffness	0.4 N/m
Damping	0.05 N s/m
Mode frequency	0.15 – 10.18 rad/s
Damping ratio	0.0095 – 0.011
Filling ratio	0.51 – 0.69

The sloshing model implemented in Simscape consists of 6 masses connected via cartesian joint (3 translational DoFs) with internal stiffness and damping along the 3 joint axis.

4.1.3 Reaction Control System (RCS)

In the live script the data for the building the thrusters' model and defining their position/orientation on the bus are provided. A baseline configuration in terms of number of thrusters and their position/orientation on the bus is provided by ISAE-SUPAERO. It consists of 12 thrusters as shown in Fig. 4.5. Data are provided in the matlab structure "thr" (see Table 4.2).

Users can also provide their own configuration. However, a different configuration in the non-linear simulator is not automatic and requires users to make some changes in the RCS model subsystem block and adjustments in the mapping of forces on the bus body through the External Force and Torque subsystem of SIMSCAPE MULTIBODY. This will be better documented in Sect. ??.

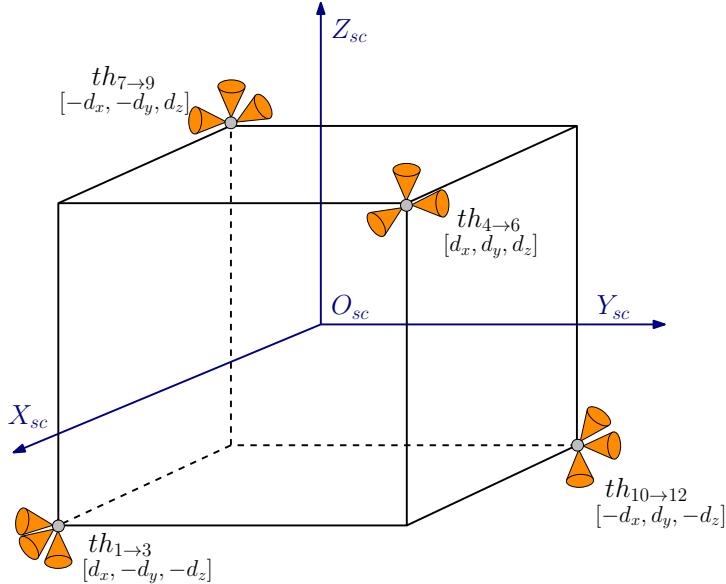


Figure 4.5: Configuration with 12 thruster in parallelepiped configuration.

Table 4.2: Input data structure for building the thrusters model

Name	Brief Description
thr.F	Scalar value - Thruster Force [N]
thr.MIB	Scalar value - Minimum Impulse Bit [N.s]
thr.nb_thr	Scalar value - Number of thrusters
thr.sc_thrust_pos	(3 x thr.nb_thr) - Thrusters position wrt the CoM of the spacecraft Hub [m]
thr.sc_thrust_dir	(3 x thr.nb_thr) Thrusters direction in spacecraft hub frame
thr.om	Scalar value - natural frequency of 2nd order TF [rad/sec]
thr.damp	Scalar value - Damping coefficient of 2nd order TF [-]

The data provided in the Table 4.2 allow then to compute the following parameters:

- `thr.u_min` : the minimum commanded force that activates the thrusters.
- `thr.sc_thrust_Hf` and `thr.sc_thrust_Ht`: the matrices mapping thruster forces to applied forces and torques on the spacecraft.
- `thr.FrameRot`: 1x12 cell containing the DCM of each thruster with respect to the bus reference frame.

4.1.4 Micro-Propulsion System (MPS)

The Micro-Propulsion System (MPS) is defined in a similar way to the RCS. The MPS is composed of thrusters capable of thrust forces in the micronewton (μN) range. The data provided here are used to build a simple model (used in the non-linear simulator) of MPS based on Cold-gas thrusters [30], which can generate a quasi-continuous thrust (differently for classical thrusters) in the range 0–1 mN, with a resolution of 0.1 μN . The list of input data for building the MPS model is given in Table 4.3.

Table 4.3: Input data structure for building the MPS thrusters

Name	Brief Description
MPS.F	Scalar value - MPS Maximum force [N]
MPS.res	Scalar value - Resolution [N]
MPS.nb_thr	Scalar value - Number of thrusters
MPS.sc_thrust_pos	(3 x MPS.nb_thr) - Thrusters position wrt the CoM of the spacecraft Hub [m]
MPS.sc_thrust_dir	(3 x MPS.nb_thr) Thrusters direction in spacecraft hub frame
MPS.om	Scalar value - natural frequency of 2nd order TF [rad/sec]
MPS.damp	Scalar value - Damping coefficient of 2nd order TF [-]

Based on these data, the following parameters are computed

- `MPS.sc_thrust_Hf` and `MPS.sc_thrust_Ht`: the matrices mapping the MPS thruster forces to applied forces and torques on the spacecraft.
- `MPS.FrameRot`: 1x12 cell containing the DCM of each MPS thruster with respect to the bus reference frame.

4.1.5 Payload (Telescope structure, FSM, PMA, Isolator)

At the payload level the system is composed of the telescope structure (and detector), the 2 actuator systems (FSM and PMAs) and the isolator between the payload and the spacecraft platform. The SDTLib is shown in Fig. 4.6.

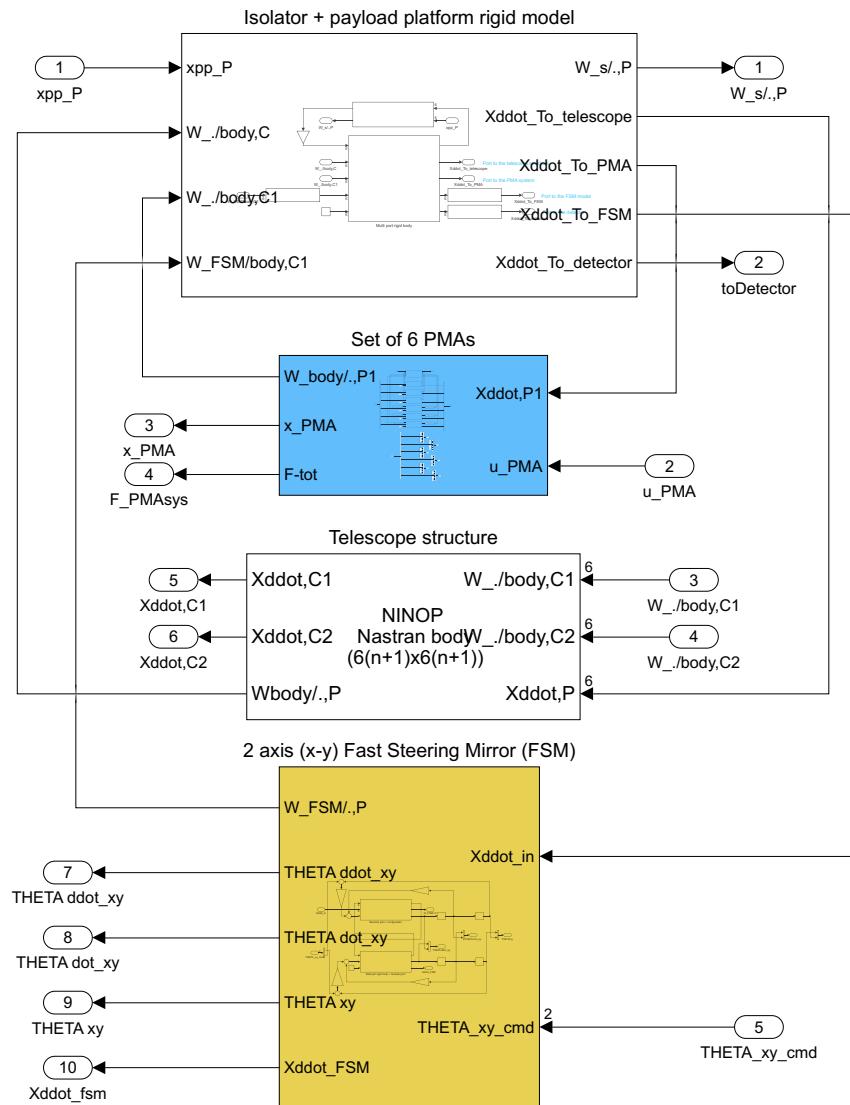


Figure 4.6: Visualization of the SDTlib model for the payload system

The assembled payload system (without the telescope) is better visualized thanks to the **Mechanics Explorer** in Fig. 4.7 which shows the relative position of the payload base plate, the FSM used to improve the LOS by acting on the path of the incoming

ray from the telescope secondary mirror to the detector, and the PMA system used to provide an active vibration control.

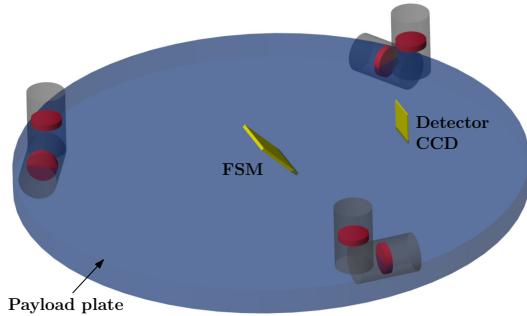


Figure 4.7: Payload plate with the PMA system. The figure shows also the position of the FSM and of the detector

4.1.5.1 Payload isolator

The payload isolator is modelled as a massless spring-damper system with 6 DoFs as done for the RW isolators. The mass, moment of inertia and position of the center of mass are associated to the isolator, payload mounting plate and other subsystem (not modelled) at payload level.

4.1.5.2 Fast-Steering Mirror

The FSM is a 2-axis scan mirror used to control and rapidly and accurately adjust the direction of a reflected optical beam. In the benchmark problem is positioned in the optical path between the secondary mirror and the detector. A view of the FSM implemented in SDTLIB is shown in Fig. 4.8.

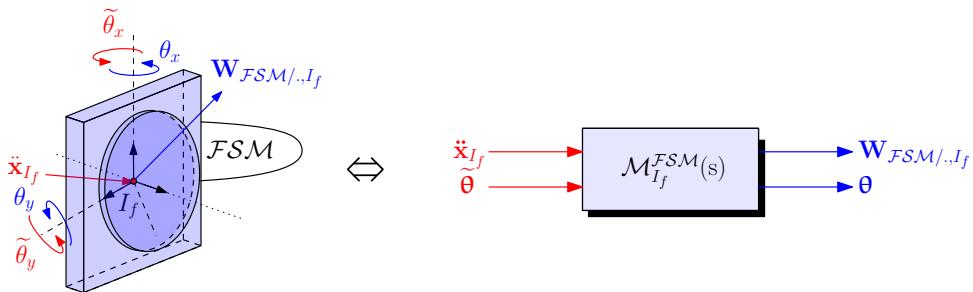


Figure 4.8: Fast steering mirror (left) and equivalent TITOP model (right)

4.1.5.3 Proof-Mass Actuator system (PMA)

PMAs are mechanisms that can produce a force along their principal axis. Multiple PMA can be placed and oriented in a particular configuration to provide controllability over the desired number of DOFs of a body as in [25].

Figure 4.9 shows a schematic view of a PMA and its equivalent SDTlib model model.

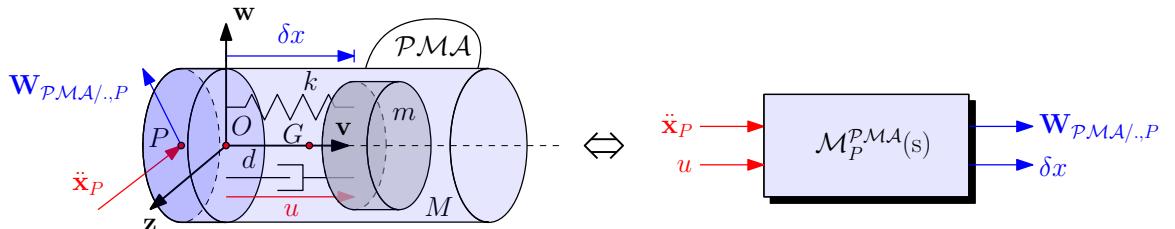


Figure 4.9: Proof-mass actuator view (left) and equivalent TITOP model (right)

The moving mass m of a single PMA is actuated by an input force u . The relative motion of the mass m with respect to the PMA caging produces a reaction force on the parent body that allows damping vibrational motion.

4.2 Optical model

The optical model implemented in the benchmark study (see Fig. 3.1) consists of a 2-mirror telescope system (in Ritchey-Chrétien), a fast steering mirror and the detector. The following section briefly covers the description of the telescope system and the data used in the benchmark. After that, the methodology used to compute the linear optical sensitivities (for reflective surfaces only) is presented to account for the flexibility of the telescope structure. Indeed, displacement and deformations of optical geometry cause deviation of the nominal beam train producing LOS and wavefront errors.

4.2.1 Telescope model

In the Ritchey-Chrétien (R-C) configuration, both the primary and secondary mirrors are hyperboloids. The R-C configuration is shown in Fig. 4.10. The main parameters of a Ritchey-Chrétien system with the numerical values used in the benchmark model are given in Table 4.4.

A Fast-Steering Mirror is then placed at half the back focal distance to rotate the beam towards the detector located on the focal plane. Considering the origin of the telescope reference frame in the primary mirror vertex with z -axis directed as the LOS

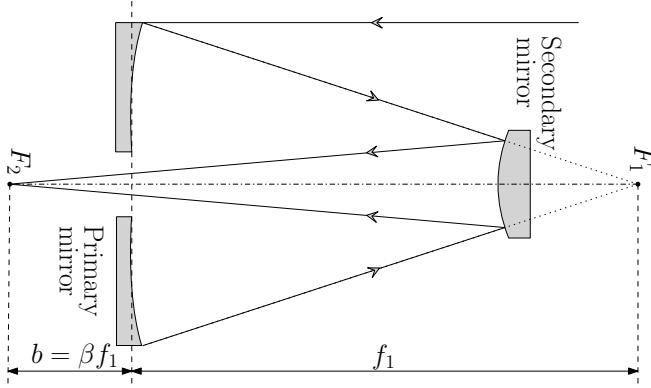


Figure 4.10: Ritchey-Chrétien two-mirror telescope system

Table 4.4: Telescope model and data

Parameter	Expression	Value
Primary mirror Diameter	D_1	0.85 m
Primary f-ratio	N_1	1.2
Primary mirror focal length	$f_1 = N_1 D_1$	1.02 m
Final f-ratio	N	12
Final focal length	f	10.2 m
Magnification of secondary mirror	m	10
Back-focal distance	b	0.437 m
Normalized back-focal distance	$\beta = b/f_1$	0.4284
Secondary mirror focal length	f_2	1.6189 m
Primary-secondary separation	$s = (f - b)/(m + 1)$	0.8875 m
Secondary mirror Diam.	$D_2 = D_1(f_1 + b)/(f + f_1)$	0.1104 m
Primary mirror conic constant	κ_1	-1.003546
Secondary mirror conic constant	κ_2	-1.531149
Obscuration ratio	D_2/D_1	0.13 m

then the position of the FSM and the detector are given as follow:

$$\begin{aligned}\mathbf{x}_{fsm} &= [0 \ 0 \ -b/2]^T \\ \mathbf{x}_{detector} &= [b/2 \ 0 \ -b/2]^T\end{aligned}\tag{4.2}$$

4.2.2 Telescope finite element model

This section briefly describes the finite element model of the telescope structures and discuss the procedure used for computing an approximation of primary mirror centroid

motion. The FE model of the structure is reported in Fig. 4.11 and it is composed of a primary and secondary mirror (with their supporting structure) and 3 truss elements connecting them. The model is design according to the data reported in Table 4.4 for the R-C configuration. A view of the supporting structure with the internal ribs' pattern of the primary mirror is shown in Fig. 4.12 (left).

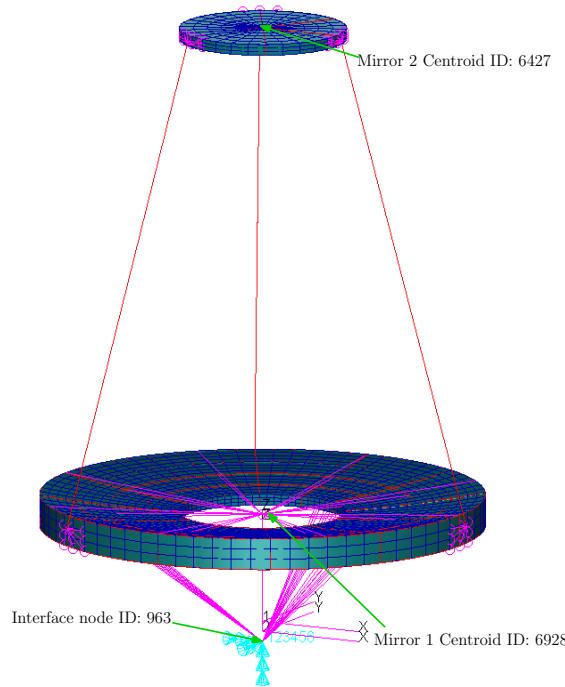


Figure 4.11: FE model of the telescope

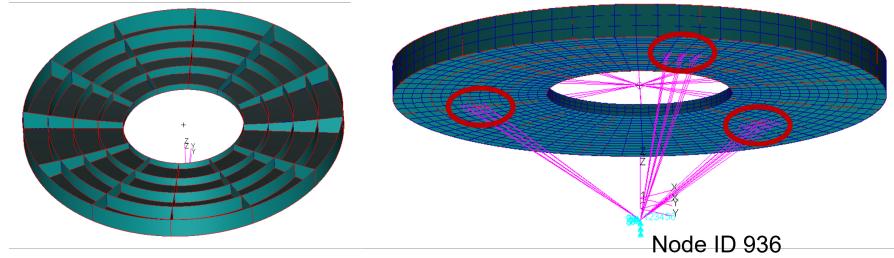


Figure 4.12: Internal supporting structure of the primary mirror (left); Rigid body connection of the supporting structure to the interface node with RBE2

Modes and natural frequencies of the telescope structure are computed with the systems clamped at the interface node 963 which is the interface with the isolator plate. The connection of the supporting structure with the interface node is realized with Rigid Body Elements (RBE2 from MSC Nastran) shown as purple lines in Fig. 4.12 (right) that

rigidly link the clamped interface node to three groups of nodes in the primary mirror structure (circled in red). Besides the interface node of the telescope with the isolator plate 2 more nodes are retrieved in matlab with SDTLIB function NASTRAN2ROFS:

- Node ID 6928: Primary mirror centroid computed with an Explicit MPC (MSC Nastran) to provide an averaged motion of the primary mirror (see Sect. 4.2.2.1);
- Node ID 6427: Secondary mirror centroid.

The telescope structure is introduced in the SDTLIB system with the **N-ports Nastran body** with 3 ports with node 963 as point P (connection to the parent body) and nodes 6928 and 6427 as points C for possible connections with other bodies or, as in this case, to retrieve the primary and secondary mirror accelerations.

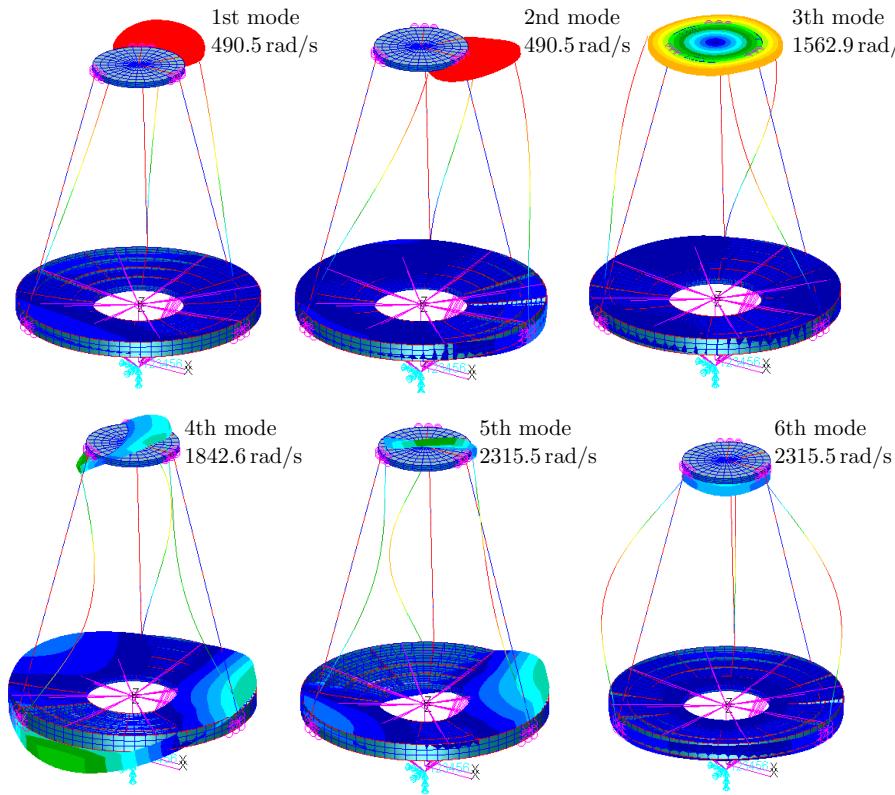


Figure 4.13: First 6 modes of vibration of the telescope structure

4.2.2.1 Primary mirror: Centroid motion calculation

An approximate computation of the primary mirror centroid motion can be provided by averaging the contribution of N_p nodes of the primary mirror structure. This operation

is done via the NASTRAN explicit Multi-Point Constraints that allows to write the DoFs of a dependent node as a linear combination of the Dofs of several independent nodes.

The 6 DoFs ($[u_c, v_c, w_c, \theta_c^x, \theta_c^y, \theta_c^z]$) of the centroid (Node ID: 6928) are computed with the following expressions (small displacements and small rotations are assumed):

$$\begin{aligned} u_c &= \frac{1}{N_p} \sum_{i=1}^{N_p} u_i, & v_c &= \frac{1}{N_p} \sum_{i=1}^{N_p} v_i, & w_c &= \frac{1}{N_p} \sum_{i=1}^{N_p} w_i \\ \theta_x &\cong \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{w_i}{x_i}, & \theta_y &\cong -\frac{1}{N_p} \sum_{i=1}^{N_p} \frac{w_i}{y_i}, & \theta_z &\cong \frac{1}{N_p} \sum_{i=1}^{N_p} \left(\frac{v_i}{x_i} - \frac{u_i}{y_i} \right) \end{aligned} \quad (4.3)$$

where x_i , y_i and z_i are the distances of the i-th node along the 3 axis. The independent nodes are 24 given in groups of 3 (three nodes arranged on the same line and placed at the inner, middle and external circumference of the mirror) equally spaced with 45 deg rotation angle about the mirror z-axis.

4.2.3 Ray-tracing algorithm for linear optical sensitivities

The linear optical model block shown in Fig. 4.1 contains the LOS linear optical sensitivities and it is derived via a ray-tracing algorithm derived from the work done in [31]. This allows to consider into our dynamical model (in the definition of the LOS) the perturbations induced by the motions on the optical elements on the incoming rays from the observed target. The ray tracing is performed with the approximation that light follows straight lines in vacuum or air without diffraction. The model is based on the paraxial (linear) approximation and thus the optical sensitivities provides a good model only for rays that are close to the exact nominal ray. The calculation of the sensitivities is then straightforward and it only requires the direction of the starting incident ray, the position of each optical element and its properties (i.e. eccentricity, focal length, principal axis)

Let us address the optical model of the benchmark. It is composed of 4 surfaces:

- Primary mirror ($i = 1$);
- Secondary mirror ($i = 2$);
- Fast steering mirror ($i = 3$);
- Detector ($i = 4$).

The first sensitivity is computed considering an incident ray entering the system along the LOS direction (z-axis of the telescope) and incident in the centroid of the primary mirror (see calculation provided in Sect. 4.2.2.1). Since computing the contribution of the optical surfaces' deformation to the Wave Front Error (WFE) is outside the scope of the benchmark, we compute an averaged motion of the primary mirror with a 'virtual' point positioned at the vertex of the primary mirror to address LOS errors.

4.3 LFT/LPV model and data provided to the user

The data provided to the user are listed in Table 4.5 and are saved into the .mat file:

- `LFT_LPV_spacecraftModel_Users_2024_07_30_13h_57m_12s.mat`

in folder: `Users_BenchmarkSat_V01/Utilities/Models/SDT_model`

The LFT/LPV model `Gum_I0selected` contains only the mechanical model of the space-craft and it is the base for constructing the synthesis model, which is the topic of the following chapter. The list of uncertain parameters with their nominal values and range of variation are reported in Appendix C. In Table 4.6 and Table 4.7 the inputs and outputs of the model are listed respectively. They can be accessed from the variable by typing `Gu_I0selected.InputName` and `Gu_I0selected.OutputName`.

Table 4.5: Data in .mat `LFT_LPV_spacecraftModel_Users_2024_07_30_13h_57m_12s.mat`

Structure name	Description
Benchmark	It contains the initial configuration data of the benchmark
Bus	Contains the MCI of the bus and the attachment points coordinates
FSM	It contains the FSM data (MCI, stiffness and damping, initial orientation and position of the FSM with respect to the payload's isolator)
Gu	LFT/LPV model in the form of an uncertain state space system. This linear model has only acceleration quantities as output.
Gum_IOselected	LFT/LPV model in the form of an uncertain state space system used for the generation of the synthesis model. It is retrieved from Gu after the addition of integrators and the reduction of non-minimal states.
HGA	It contains the data for building the HGA structural model (name of the .bdf and .f06 files, assumed damping factor and the reduced order mass, stiffness and damping matrices for the SIMSCAPE MULTIBODY block Reduced Order Flexible Solid)
ISOL	It contains the payload isolator data (MCI, stiffness and damping, port coordinates for the connection with the bus and the payload subsystem)
MPS	It contains the data for the Micro-Propulsion System
PAYLOAD	It contains the data for building the payload structural model (name of the .bdf and .f06 files, assumed damping factor and the reduced order mass, stiffness and damping matrices for the SIMSCAPE MULTIBODY block Reduced Order Flexible Solid . It also contains the data related to the telescope optical model (Mirrors configuration, detector position and optical sensitivity matrices associated to the FSM and the primary and secondary mirror)
PMA	It contains the PMA system data (MCI, stiffness and damping of the devices)
RWIsol	It contains the data for the 4 reaction wheel isolators (MCI, isolator stiffness and damping, port coordinates for the connection with the bus and the reaction wheel)
RWS	It contains the data for the reaction wheel system (MCI, Spin axis direction, maximum torque and angular speed, data for the static and dynamic imbalances and data for the friction models)
SA	It contains the data for building the solar array structural models (name of the .bdf and .f06 files, assumed damping factor, orientation and the reduced order mass, stiffness and damping matrices for the SIMSCAPE MULTIBODY block Reduced Order Flexible Solid)
SADM	It contains the stiffness and damping of the mechanism
slosh	It contains the data for the tank (mass and moment of inertia and its position with respect to the bus) and the equivalent mechanical model for the sloshing phenomena composed of 6 mass-spring-damper systems (equilibrium position of each mass, stiffness and damping factor)
thr	It contains the thruster data as discussed in Table 4.2

Table 4.6: List of inputs of the uncertain state-space `Gum_I0sselected`

Name	Description
'W_ext(1)';'W_ext(2)';'W_ext(3)'	Force vector applied at the spacecraft bus
'W_ext(4)';'W_ext(5)';'W_ext(6)'	Torque vector applied at the spacecraft bus
'RW_dist(1→5)'	Reaction wheel disturbance vector (RW-1)
'RW_dist(6→10)'	Reaction wheel disturbance vector (RW-2)
'RW_dist(11→15)'	Reaction wheel disturbance vector (RW-3)
'RW_dist(16→20)'	Reaction wheel disturbance vector (RW-4)
"Torque_SADM1";"Torque_SADM2"	Torque applied to the SADMs
THETAxycmd(1)';'THETAxycmd(2)'	Command (angle) to the FSM
'uPLpma(1→ 6)'	Force command vector to PMA system

Table 4.7: List of outputs of the uncertain state-space `Gum_I0sselected`

Name	Description
'Xddot_MB(1)';'Xddot_MB(2)';'Xddot_MB(3)'	Linear acceleration of the spacecraft bus
'X_MB(4)';'X_MB(5)';'X_MB(6)'	Attitude angle of the spacecraft bus
'Xdot_MB(4)';'Xdot_MB(5)';'Xdot_MB(6)'	Angular velocity of the spacecraft bus
"THETAxycmd(1)";"THETAxycmd(2)"	FSM angle
'x_PLpma(1→6)'	PMA displacement
'Xddot_M1(1→6)'	Acceleration at centre of primary mirror
'Xddot_M2(1→6)'	Acceleration at centre of secondary mirror
'Xddot_detector_Bframe(1→6)'	Acceleration at detector
'LOS(1)';'LOS(2)';'LOS(3)'	Line-Of-Sight (LOS)

Chapter 5

Synthesis model

Contents

5.1	Code for synthesis model: <code>Main_Open_Loop_Dynamic_Model.m</code> . . .	45
5.1.1	GUI for selection of phase, requirements and actuators/sensors . .	45
5.1.1.1	Output of the script	49

This chapter covers the description of the codes used for the generation of models that the user can use for control synthesis. The main code provided in Simulator-02 is:

- `Main_Open_Loop_Dynamic_Model.m` is used to build the synthesis model without the input/output weighting filters and it does not include the sections of the code dedicated for the synthesis of the baseline controllers presented by ISAE-SUPAERO.

Besides the main file, the following simulink model is used to generate the synthesis model:

- the simulink file `General_DL_model_NoFilters.slx` for the generation of the generalized open-loop model without input/output weighting filters.

Fig. 5.1 is provided as an example to show the synthesis model with and without weighting filters. The synthesis model without weighting filter is highlighted in yellow and adds to the spacecraft model spacecraft model $\mathbf{G}(s)$ (discussed in the previous chapter) the dynamics of actuators/sensors, delays and additional inputs/output to implement control objectives and constraints in optimization problem. The weighted problem (highlighted in cyan) added on top of this the weighted filters implemented by ISAE-SUPAERO to shape exogenous inputs and to impose requirements.

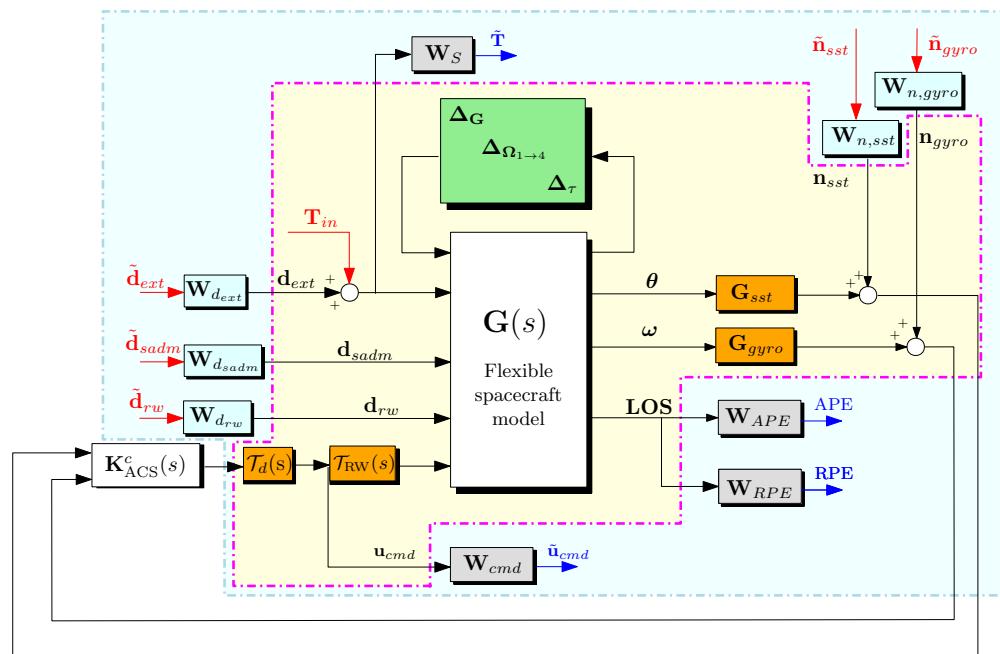


Figure 5.1: Generalized closed-loop model for synthesis (with weighting filters in cyan and without in light yellow).

5.1 Code for synthesis model: Main_Open_Loop_Dynamic_Model.m

This section shows the user how to use the matlab file `Main_Open_Loop_Dynamic_Model.m` for generating the LFT model for synthesis by using a GUI for selection of actuators/sensors and disturbances acting on the system.

5.1.1 GUI for selection of phase, requirements and actuators/sensors

The user is prompted to the GUI shown in Fig. 5.2, which initializes the automatic generation of the synthesis model. Based on the user selection, the code will create a model starting from the uncertain state-space `Gum_I0sselected` with inputs given by the selected disturbances, actuators and noises (associated to the selected sensors) and outputs determined by the sensors and the selected performance outputs. Furthermore, the code and the associated simulink model (see Fig. 5.3) append the actuators/sensors' dynamics to the spacecraft dynamical model.

Note that the last cell (**Performance output**) simply replicates the LOS output but with different names (APE, RPE, PDE). No weighting filter is applied to define the pointing requirements. This operation is done to simplify the application of pointing filters for users.

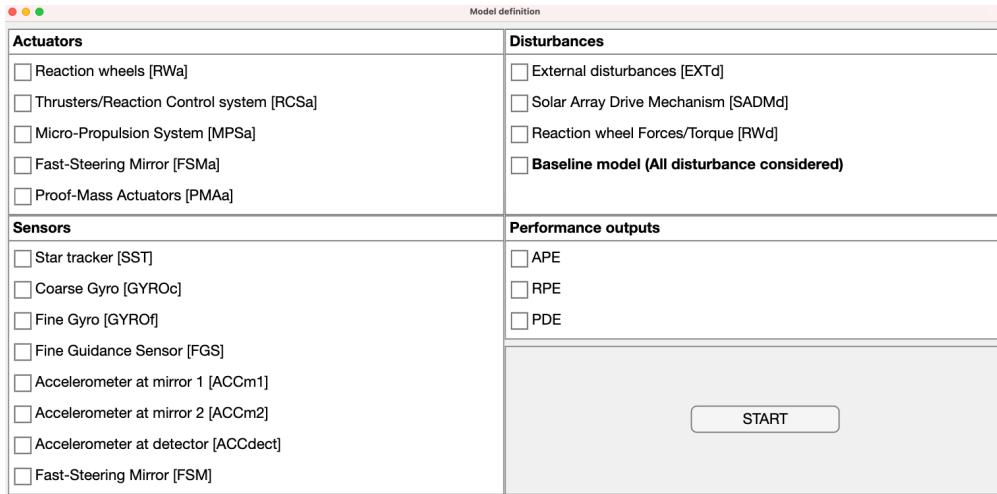


Figure 5.2: GUI for Inputs/outputs selection

Here we provide an example to show what is the output of the GUI selection. Referring to Fig. 5.2 the following check-boxes are selected:

- In DISTURBANCE: External disturbances
- In ACTUATORS: Reaction wheels

- In SENSORS: Star tracker and coarse gyro
- In PERFORMANCE OUTPUTS: APE.

After the selection procedure, the following message appears in the command line showing the name of the inputs/outputs in the LFT model (`ModelSyn_selected`).

```
-----  
Here is the list of selected inputs of the model:  
{'dn_EXTd(1)',      }  
{'dn_EXTd(2)',      }  
{'dn_EXTd(3)',      }  
{'Tin(1)',          }  
{'Tin(2)',          }  
{'Tin(3)',          }  
{'Nn_SST(1)',       }  
{'Nn_SST(2)',       }  
{'Nn_SST(3)',       }  
{'Nn_GYROc(1)',    }  
{'Nn_GYROc(2)',    }  
{'Nn_GYROc(3)',    }  
{'out_cmd_RWa(1)', }  
{'out_cmd_RWa(2)', }  
{'out_cmd_RWa(3)', }  
  
Note that the last elements (with the string starting with "  
out_cmd_" correspond to  
the outputs of the controller (i.e. the inputs of the system)  
-----  
  
-----  
Here is the list of selected outputs of the model:  
{'APE(1)',          }  
{'APE(2)',          }  
{'APE(3)',          }  
{'Tstab(1)',         }  
{'Tstab(2)',         }  
{'Tstab(3)',         }  
{'un_RWa(1)',        }  
{'un_RWa(2)',        }  
{'un_RWa(3)',        }  
{'in_cmd_theta_SST(1)', }  
{'in_cmd_theta_SST(2)', }  
{'in_cmd_theta_SST(3)', }  
{'in_cmd_thetad_GYROc(1)', }  
{'in_cmd_thetad_GYROc(2)', }  
{'in_cmd_thetad_GYROc(3)', }
```

```
Note that the last elements (with the string starting with "
in_cmd_"
correspond to
the inputs of the controller (i.e. the measurements of the system
)
-----
```

Note that the code automatically adds additional inputs/outputs to the system to address the presence of noise in the sensors and possible constraints (i.e. maximum actuator command) to be applied on the actuators. In this example, the additional inputs are the noise vectors associated to the star tracker and coarse gyro `Nn_SST(1→3)` and `Nn_GYROc(1→3)` and the additional output is the command to the RW system `un_RWa(1→3)`.

The model is obtained with the matlab function `ulinearize` applied on:

- the simulink file `General_DL_model_NoFilters.slx` for the synthesis model without weighting filters.

To better understand how the model is built, the definition of the inputs/outputs and how a closed-loop model can be built we can refer to the file `General_DL_model_NoFilters.slx` shown in Fig. 5.3. At the centre of the model we can find the uncertain state space model `Gum_I0selected`.

The inputs/outputs are color-coded to separate noises, disturbances, measurements and commands, and requirements/constraints:

- Disturbances and noises are reported in **orange**;
- the command to the system (output of the controller) are in **green**. The name of all these inputs is composed by the string "out_cmd" associated to the name of the actuator (i.e. for the reaction wheel "out_cmd_RWa").
- the measurements of the system (input of the controllers) are in **red**. The name of all these outputs is composed by the string "in_cmd" associated to the name of the sensor (i.e. for the coarse gyro "in_cmd_theta_GYROc").
- Requirements (for pointing metrics and actuators' limits) are given are in **yellow**. In the model with weighting filters, these outputs are shaped with transfer functions to account for actuators saturation and for pointing requirements (see Appendix B)

There is a specific input/output set (input: `Tin` - output: `Tstab`) which is used to compute the input sensitivity function and to impose the stability requirement (in Sect.3.2).

The models of the actuators are also added to the model (see light yellow blocks) via the following filters:

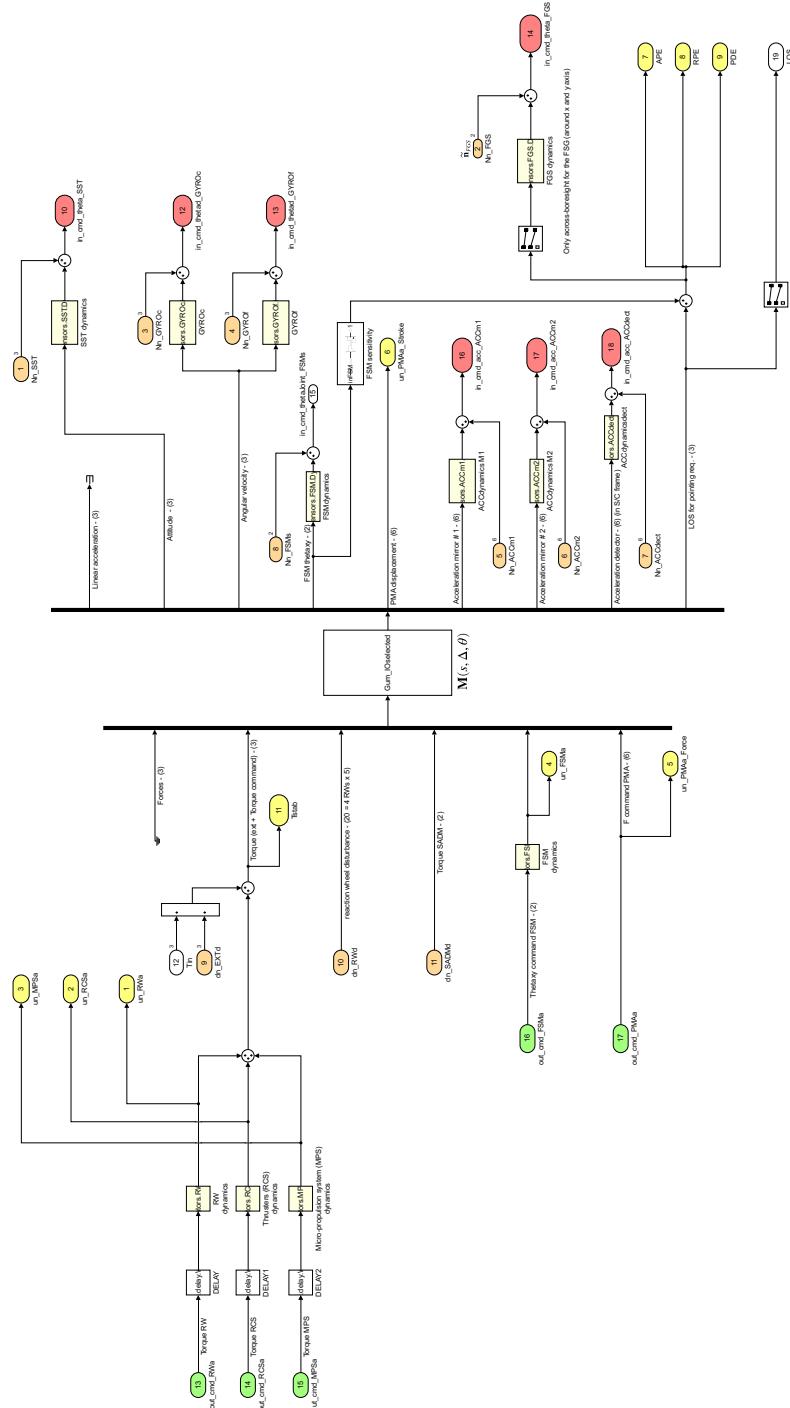


Figure 5.3: Simulink file General_OL_model_NoFilters.slx

- REACTION WHEELS: a reaction wheel system modeled as a second-order dynamics with natural frequency of 100 Hz (200π rad/s) and damping factor equal to 0.7:

$$\mathcal{T}_{RW}(s) = \frac{(200\pi)^2}{s^2 + 400 \cdot 0.7\pi s + (200\pi)^2} \mathbf{I}_3 \quad (5.1)$$

- THRUSTERS: the valve response of the thrusters is similarly modelled as a second-order dynamics with natural frequency of 200 Hz and damping factor equal to 0.7:

$$\mathcal{T}_{RCS}(s) = \frac{(400\pi)^2}{s^2 + 800 \cdot 0.7\pi s + (400\pi)^2} \mathbf{I}_3 \quad (5.2)$$

- MICRO-THRUSTERS: the response of the micro-thrusters is similarly modelled as a second-order dynamics with natural frequency of 4 Hz and damping factor equal to 0.7:

$$\mathcal{T}_{MPS}(s) = \frac{(8\pi)^2}{s^2 + 16 \cdot 0.7\pi s + (8\pi)^2} \mathbf{I}_3 \quad (5.3)$$

- FSM - PMA SYSTEM: their mechanical models are already incorporated within the general LFT/LPV model built with SDTLIB.

A 2nd order Padé approximation with a time constant $T_d = 0.05$ s is used to model the loop delay for the ACS:

$$\mathcal{T}_d(s) = \frac{T_d^2 s^2 - 6T_d s + 12}{T_d^2 s^2 + 6T_d s + 12} \mathbf{I}_3 \quad (5.4)$$

5.1.1.1 Output of the script

The outputs provided by the script `Main_Open_Loop_Dynamic_Model.m` are listed in Table 5.1.

Once the selection via the GUI is completed, the user can choose either to save or not save the data to the folder `Solutions_freqDomain` with the following name:

- `DYNAMICS_yyyy_mm_dd hh_min_sec.mat` when the user runs the code `Main_Open_Loop_Dynamic_Model.m`.

Furthermore, if the user has the MATLAB package REPORT GENERATOR installed, the code save a summary report with the name:

- `Report_yyyy_mm_dd hh_min_sec.pdf`

and a copy of the selection GUI with the name:

- `SelectionGUI_yyyy_mm_dd hh_min_sec.pdf`

in the folder "Reports".

Table 5.1: Output of the code `Main_Open_Loop_Dynamic_Model.m`

Structure/ variable name	Description
Actuators	It contains the information regarding the actuators in the system (RW, RCS, MPS, FSM, PMA);
Disturbance	It includes data about the disturbances affecting the system. However, this data represents only a subset of the disturbances impacting the system and is used exclusively for the baseline synthesis at ISAE-SUPAERO for defining the weighting filters. Table 3.5 provides the link to the sections where each disturbance signal is described.
Gu	LFT/LPV model in the form of an uncertain state space system. This linear model has only acceleration quantities as output
Model	It contains info about the model (ACS sampling, delays, and the corresponding indexes for actuators, sensors, noises and disturbances used to generate <code>ModelSyn_selected</code>).
Model_Name	It contains the file name of the initial LFT model Gu loaded at the start of the code.
ModelSyn_selected	Open-loop uncertain state-space for control synthesis
PHASE	It contains the pointing requirements of each phase of the mission.
Reduced_InputName	List of reduced set of inputs for uncertain state-space <code>ModelSyn_selected</code>
Reduced_OutputName	List of reduced set of outputs for uncertain state-space <code>ModelSyn_selected</code>
Sensors	It includes data about the noises affecting the system. However, this data represents only a subset of the noise impacting the system and is used exclusively for the baseline synthesis at ISAE-SUPAERO for defining the weighting filters. All noises are provided instead in Sect. 6.2.2.

Chapter 6

Non-linear simulator

Contents

6.1	Mission timeline and conditions for phase transition	52
6.2	Overview of the simulator	53
6.2.1	Simulator subsystem: DYNAMICAL SYSTEM	55
6.2.1.1	Flexible Appendages (Solar arrays + HGA)	57
6.2.1.2	WEMS / Reaction Wheels	59
6.2.1.3	Tank + Sloshing model	74
6.2.1.4	Payload System (Telescope structure, FSM, PMA system, Isolator)	76
6.2.2	Simulator subsystem: SENSORS	78
6.2.3	Simulator subsystem: OUTPUT TO WORKSPACE	81
6.2.4	Simulator subsystem: CHECK ON APE REQUIREMENTS FOR PHASE TRANSITION	83
6.3	Tutorial: Setting up a time-domain analysis	86
6.3.1	Time domain initialization	87
6.3.2	Tutorial: Setting up worst-case analysis	90
6.3.2.1	Worst-case analysis: how to provide the WC configuration	90
6.3.3	Tutorial: Setting up Monte Carlo analysis	91
6.3.3.1	Saving Monte Carlo results	92
6.4	Tutorial: How to modify the controllers	94
6.5	Tutorial: How to post-process simulation data	97
6.6	Computational aspects of the simulator	100

This chapter provides the guidelines for the use of the high-fidelity non-linear simulator. In order to follow this tutorial the user shall refer to the following files:

- the matlab code `Main_TimeDomain.m` which is used for running time domain simulations (Nominal, Worst-case and Monte Carlo analysis);
- the simulink file `Simscape_Benchmark_SIM1.slx` with the non-linear SIMSCAPE MULTIBODY model.
- `Config_Nominal` which contains the nominal parameters for the simulator (can be used to reset the simulator's parameters at their default value).
- `WC_template.m` used to generate the structure file to run the worst-case analysis (documented in Sect. 6.3.2.1).

The user will retain the ability to select the actuators/sensors from a proposed selection for the various phases of the scenario. Furthermore, whenever feasible, users will also have the option to adjust the complexity of the model employed in the simulations . To achieve modularity and allow the activation/deactivation of functions in the simulation environment, the simulator will feature the use of simulink *variant subsystems* and/or *switch* blocks.

6.1 Mission timeline and conditions for phase transition

The timeline of the mission is structured as presented in Table 6.1 and outlined in Fig. 3.2. At the start of the second (coarse) and third (fine) transient phases, the simulator activates the switching of the controllers and verifies if the conditions to enter the steady-state of the corresponding phases are met. These conditions are implemented as checks on the APE as follows:

$$\begin{aligned} \text{APE}(t) &\leq \text{APE}_2 \quad \forall t \in [\bar{t}, \bar{t} + \Delta t_{s \rightarrow c}] \\ \text{APE}(t) &\leq \text{APE}_3 \quad \forall t \in [\bar{t}, \bar{t} + \Delta t_{c \rightarrow f}] \end{aligned} \quad (6.1)$$

These conditions mean that the APE requirement has to be met for at least an interval of duration $\Delta t_{s \rightarrow c}$ for the transition to the coarse steady state and $\Delta t_{c \rightarrow f}$ for the transition to the fine pointing steady state. The values of these parameters are provided in Table 6.1.

Regarding the slew maneuver we suppose a single axis maneuver of 15° around x -axis. The reference signal for the attitude is given as step response of the following second-order filter:

$$\mathcal{T}_{ref} = \frac{\omega_{BW}^2}{s^2 + 2\zeta_{ref}\omega_{BW}s + \omega_{BW}^2} \quad (6.2)$$

where $\zeta_{ref} = 1$ and $\omega_{BW} = 0.0005$ rad/s.

IMPORTANT REMARK: Users can provide their own guidance law by modifying the code `Baseline_Controller.m` and the simulink file as documented in Sect. 6.4.

Name	Description	Value
T	Total duration of the mission	2700 s
t_{s1}	Duration of the slew maneuver (transient)	To be optimized
\tilde{t}_1	Duration of the slew steady-state	100 s
t_{s2}	Duration of the coarse transient	To be optimized
\tilde{t}_2	Duration of the coarse steady-state phase	600 s
t_{s3}	Duration of the fine transient	To be optimized
\tilde{t}_3	Duration of the fine steady-state phase	To be optimized
$\Delta t_{s \rightarrow c}$	Check transition to coarse pointing steady-state	30 s
$\Delta t_{c \rightarrow f}$	Check transition to fine pointing steady-state	30 s

Table 6.1: Benchmark phases and relevant parameters

6.2 Overview of the simulator

The spacecraft non-linear simulator is reported in Fig. 6.1. The simulator is composed of the following main blocks:

- The **Dynamical system** (already partially discussed in Chapt. 4 for the mechanical model but supplemented in the non-linear simulator with additional elements) - see Sect. 6.2.1
- The **Sensor** subsystem containing the model of the sensors.
- Various sub-blocks for the generation of disturbances for the attitude dynamics of the spacecraft.
- The **Controllers** subsystem containing the controllers for each phase of the mission.
- A subsystem for the initialization of the various phase transition (See Sect. 6.3.1)
- A subsystem to transfer outputs of the simulation to the workspace.

IMPORTANT REMARK: Some markup annotations are added to the simulator to provide explanations for certain functions or elements. If needed, the user can hide the markup in the model by going to the *Format* tab and then clicking on *Show Markup*.

IMPORTANT REMARK: Although the spacecraft has 6 rigid DoFs, the orbital dynamics is not modelled and the benchmark mainly focuses on the spacecraft attitude dynamics and the Line-of-Sight stabilization problem.

Orbital disturbance are modelled with as sinusoidal disturbance with:

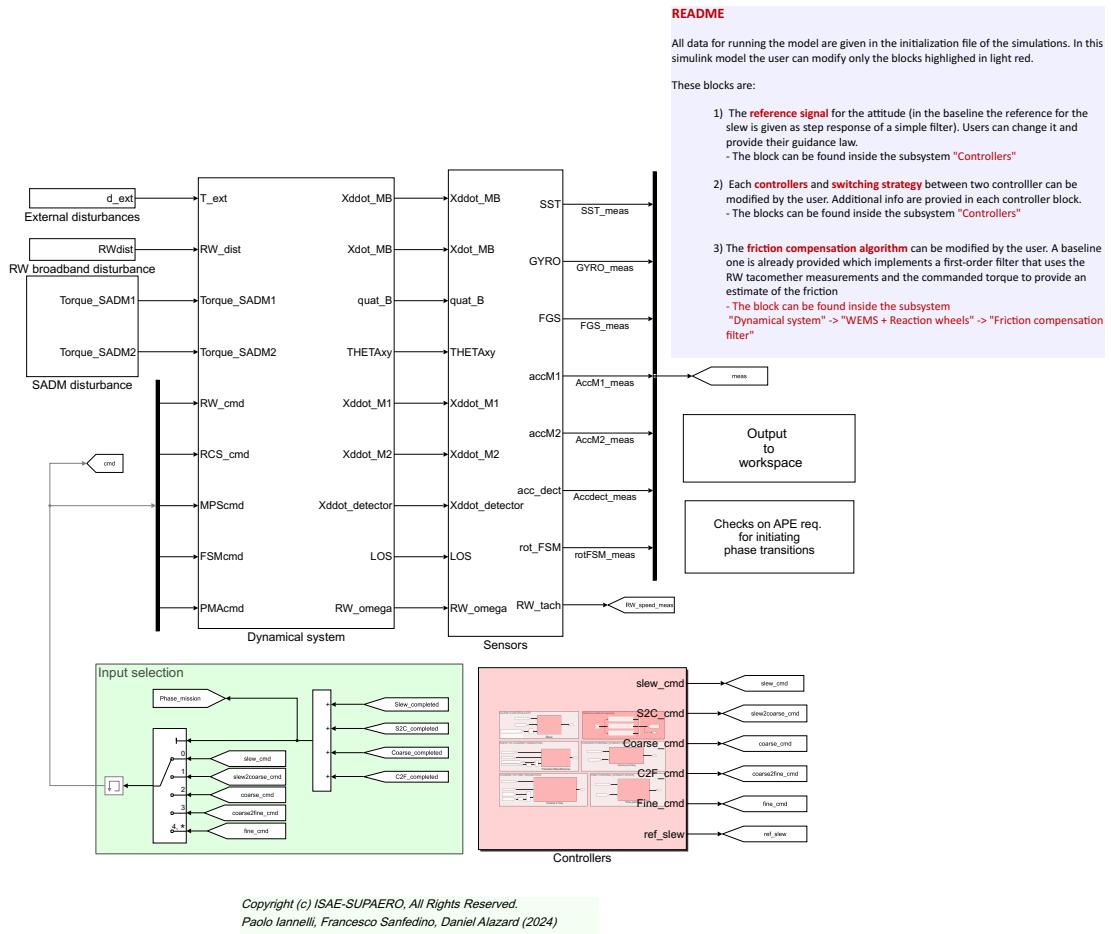


Figure 6.1: Non-linear simulator in simulink file: `Simscape_Benchmark_SIM1.slx`

- Amplitude equal to 1 mNm on each axis;
- Frequency equal to 0.0015 rad/s.

IMPORTANT REMARK: No real guidance or navigation blocks are implemented in the simulator.

6.2.1 Simulator subsystem: DYNAMICAL SYSTEM

The DYNAMICAL SYSTEM detailed in Fig. 6.2 block basically derives from the model developed in Sect. 4 with some additional components such as the thrusters' dynamics (RCS and MPS), friction models, LOS calculation in the payload block and command for the rotation of the solar arrays, which are discussed in the following sections.

A 6-dof joint is the starting pointing for modelling the spacecraft and it provides the 6 rigid DoF for the spacecraft translational and attitude dynamics. The spacecraft attitude is provided as output of this block and enters the SENSORS block to build the measurements from the star tracker and gyros.

The following subsystems are connected to the sub-block Bus for the rigid model of the S/C hub:

- the sub-block **Flexible Appendages** (Solar arrays + HGA);
- the sub-block **Payload System** includes the FEM flexible model of the telescope structure, the payload isolator, the FSM and the PMA system actuator models.
- the sub-block **WEMS / Reaction Wheels** for the reaction wheel system.
- the sub-block **Tank + Sloshing model** for the sloshing model.

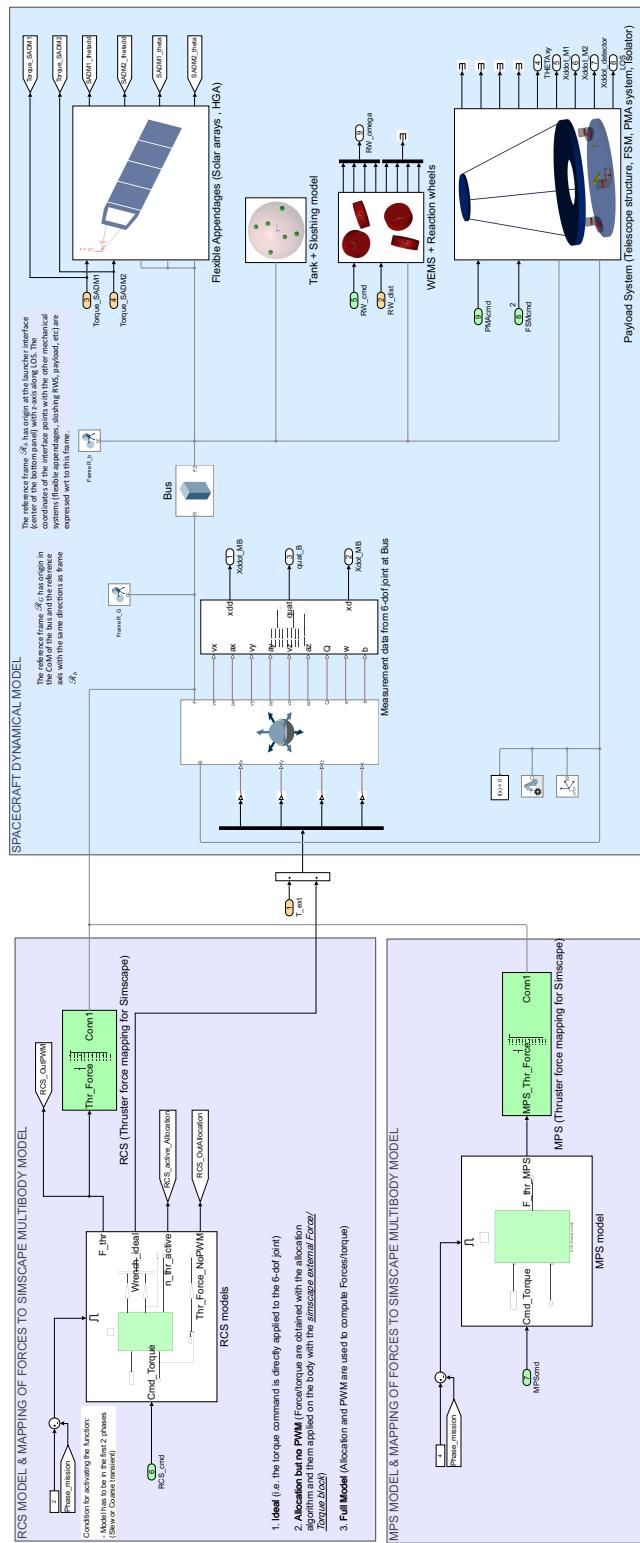


Figure 6.2: View of the simulator subsystem: DYNAMICAL SYSTEM

6.2.1.1 Flexible Appendages (Solar arrays + HGA)

This subsystem (depicted in Fig. 6.3) contains the model of the 2 flexible solar arrays and the High-Gain Antenna (HGA) boom. Flexible bodies are modelled with the block called REDUCED ORDER FLEXIBLE SOLID (ROFS), that allows the direct import of finite element models through the corresponding Craig-Bampton mass, stiffness and damping matrices.

Two revolute joints are used for each solar array mechanism. The first (show in green) is used to command in motion the solar array while the second revolute joint (in orange) is used as simplified model for the SADM. The second joint is characterized by an internal stiffness and damping and it is commanded in torque by the disturbance signal Torque_SADM1in (for SA-1). The real disturbance at the SADM is approximated as a saw-tooth signal with a certain frequency due to the micro-step as documented in [20]. The values are reported here:

- Saw-tooth amplitude: 0.1 Nm
- Saw-tooth frequency: 63.87 rad/s

Furthermore, an additional broadband disturbance is added to the SADM as a white noise:

- with a $\sigma_{sadm} = 0.0045$ Nm

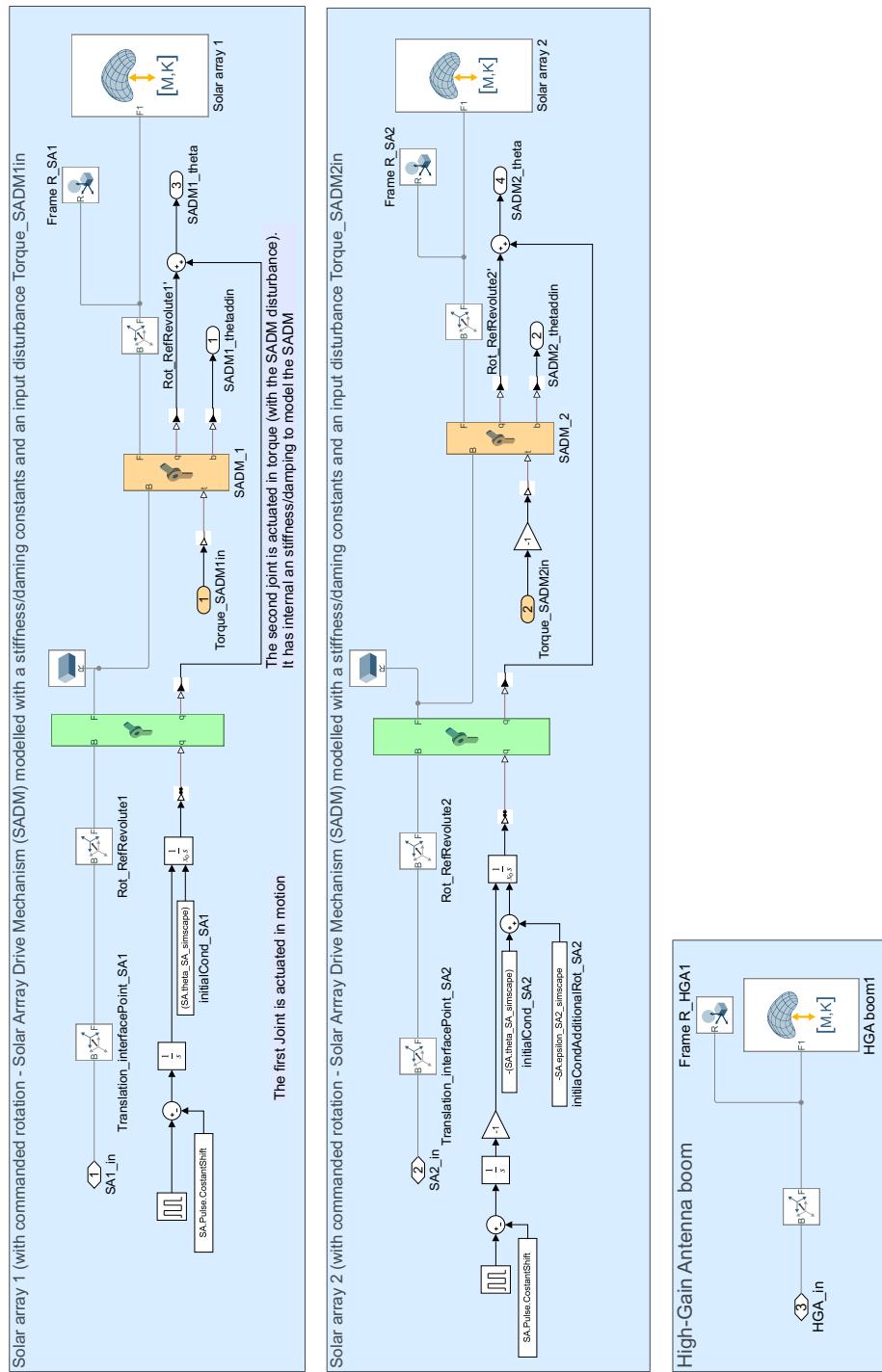


Figure 6.3: Flexible Appendages (Solar arrays + HGA) subsystem

6.2.1.2 WEMS / Reaction Wheels

The modelling of the reaction wheel (depicted in Fig. 6.4) in the simulator developed in Simscape follows an incremental complexity approach, meaning that the RWS can be simulated with different levels of detail, from the ideal configuration to the high-fidelity model or by considering intermediate configurations. This is done as discussed in the following section by properly activating/de-activating simulink VARIANT SUBSYSTEM block. This approach has also the benefit of allowing the separate assessment of the effect of each disturbance on the pointing performance of the system. The selection process is documented in Sect 6.3.1.

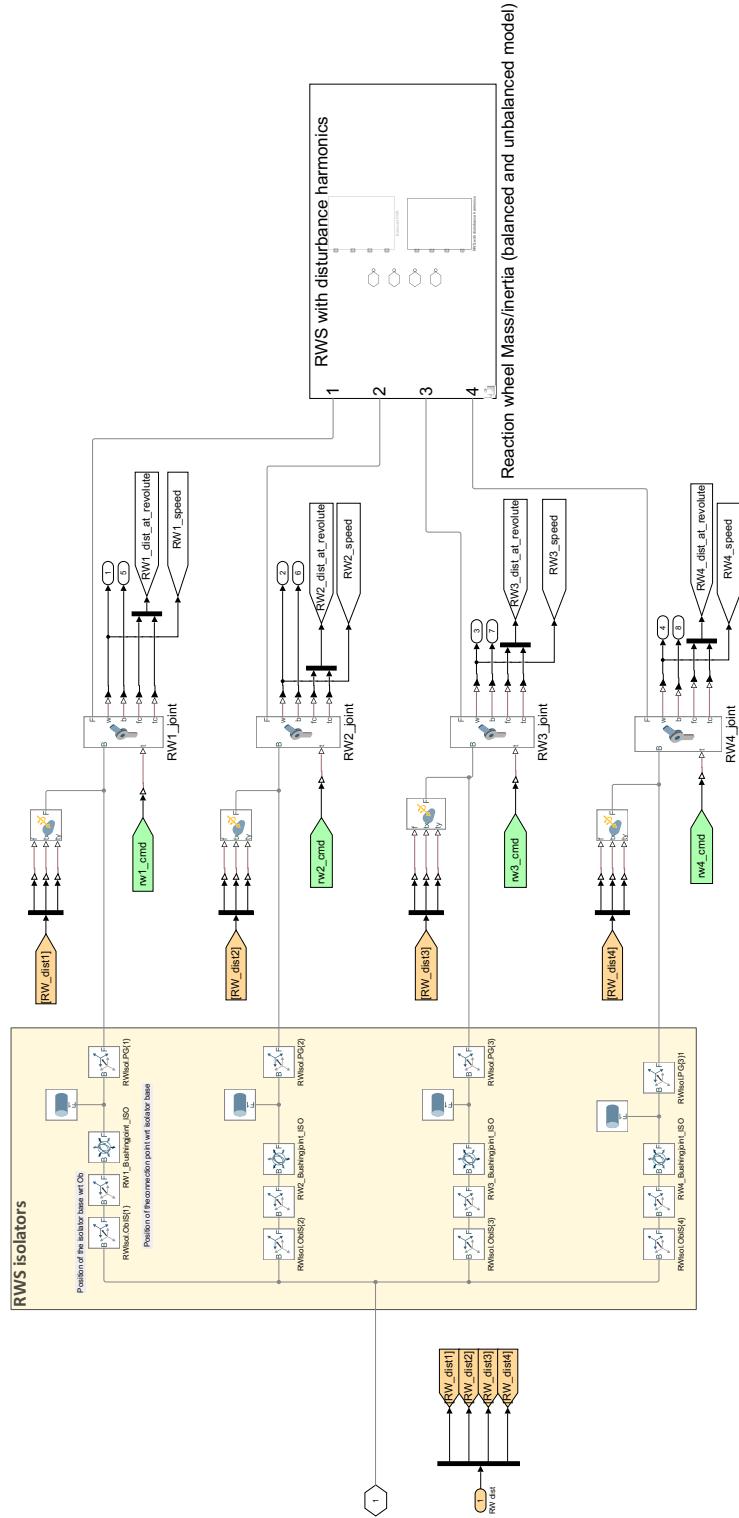


Figure 6.4: Flexible Appendages (Solar arrays + HGA) subsystem

6.2.1.2.1 Imbalanced reaction wheel model in Simscape

Flywheel imbalance is generally the largest disturbance source in the RWA and causes a disturbance force and torque at the wheel's spin rate Ω , that is referred to as the fundamental harmonic. Static imbalance results from the offset of the center of mass of the wheel from its spin axis, and dynamic imbalance is caused by the misalignment of the wheel's principle axis and the rotation axis. Figure 6.5 provides a schematic representation of the static and dynamic imbalances. The static and dynamic mass (m_s and m_d respectively) are used to construct an analytical model of the RW which captures the fundamental harmonic of the disturbance [32].

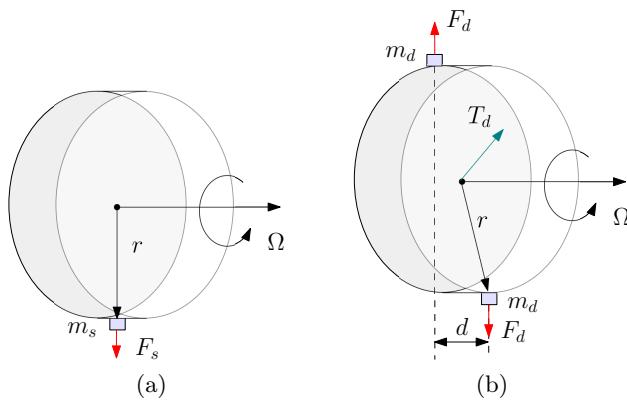


Figure 6.5: Flywheel static (a) and dynamic (b) imbalance models.

The first harmonic disturbances (at the fundamental speed Ω) are given by the following expressions:

$$\begin{bmatrix} F^x(t) \\ F^y(t) \\ F^z(t) \end{bmatrix} = \begin{bmatrix} a^f \Omega^2 \sin(\Omega t + \phi^f) \\ a^f \Omega^2 \cos(\Omega t + \phi^f) \\ a^{f_z} \Omega^2 \cos(\Omega t + \phi^{f_z}) \end{bmatrix} \quad \begin{bmatrix} T^x(t) \\ T^y(t) \end{bmatrix} = \begin{bmatrix} -a^t \Omega^2 \cos(\Omega t + \phi^t) \\ a^t \Omega^2 \sin(\Omega t + \phi^t) \end{bmatrix} \quad (6.3)$$

with a^f, a^{f_z}, a^t coefficients of the harmonic disturbance ($a^f = m_s r_s$, $a^{f_z} = 0$ and $a^t = 2m_d r d h_d$). Typically, a^f and a^t are defined with the terms static and dynamic imbalances (also generally denoted with the notation U_s and U_d). The imbalanced reaction wheel model in Simscape is reported in Fig. 6.6 where the static imbalance mass m_d and the two dynamic imbalance masses m_d are rigidly connected to the wheel at position $[r_s, 0, 0]$, $[r_d, 0, h_d]$ and $[-r_d, 0, -h_d]$ respectively in the rotating wheel frame.

6.2.1.2.2 Additional reaction wheel perturbation

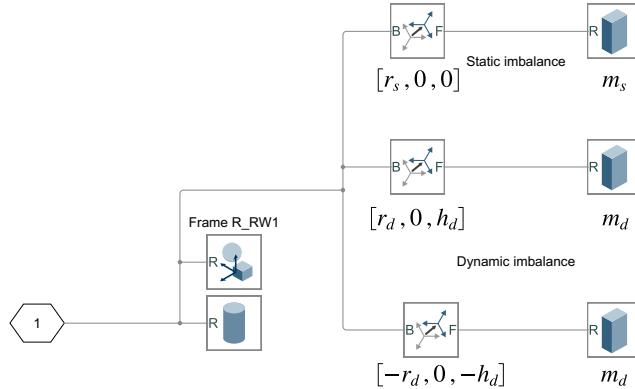


Figure 6.6: Non-linear simulator in simulink file: `Spacecraft_LFT_LPV_generation.mlx`

Additional perturbations (generally at integer or non-integer multiples of the wheel speed) are also present due to other causes (bearing irregularities, lubrication issue, etc.); these perturbations are not addressed in the provided benchmark since the main harmonics is considered the largest disturbance source as for the most of the real cases.

An additional broadband disturbance $[F_n^x(t), F_n^y(t), F_n^z(t), T_n^x(t), T_n^y(t)]$ is included in the model and enters at the joints (see Fig. orange inputs named RW_dist1→4. These broadband disturbances are modelled as:

- a white noise with $\sigma_f = 5 \cdot 10^{-4}$ N for the force along $\{x, y\}$ -axis of the RW.
- a white noise with $\sigma_t = 10^{-4}$ Nm for the torque about $\{x, y\}$ -axis of the RW.

The combination of the first harmonics and this broadband disturbance can be evaluated via the waterfall plot in Fig.6.7 at the variation of the wheel speed.

Finally a low-frequency torque disturbance (low-frequency "rubble" in the torque output) is modelled with:

- a white noise with ASD = 0.25 mNm/sqrt(Hz) up to 2 Hz.

6.2.1.2.3 Friction models and friction compensation

Figure 6.8 shows the block that introduces the friction model (if the user has selected it in the `Main_TimeDomain.m` file), the torque noise, and implements the friction compensation filter. The friction compensation filter is highlighted in red since it is one of the blocks (together with the controllers) that the user can modify to address the objectives of the proposed benchmark problem.

Deterministic friction model

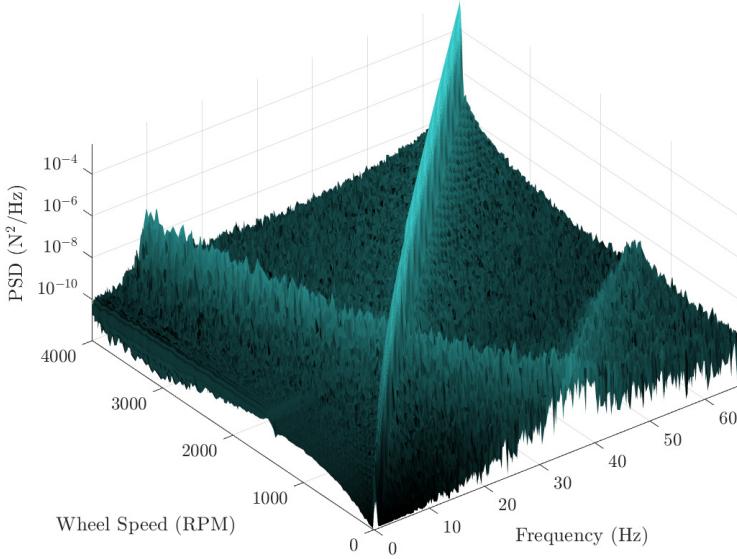


Figure 6.7: Reaction wheel waterfall plot of the disturbance torque measured in the wheel reference frame. Note the nutation modes of the wheel isolator varying with the wheel speed.

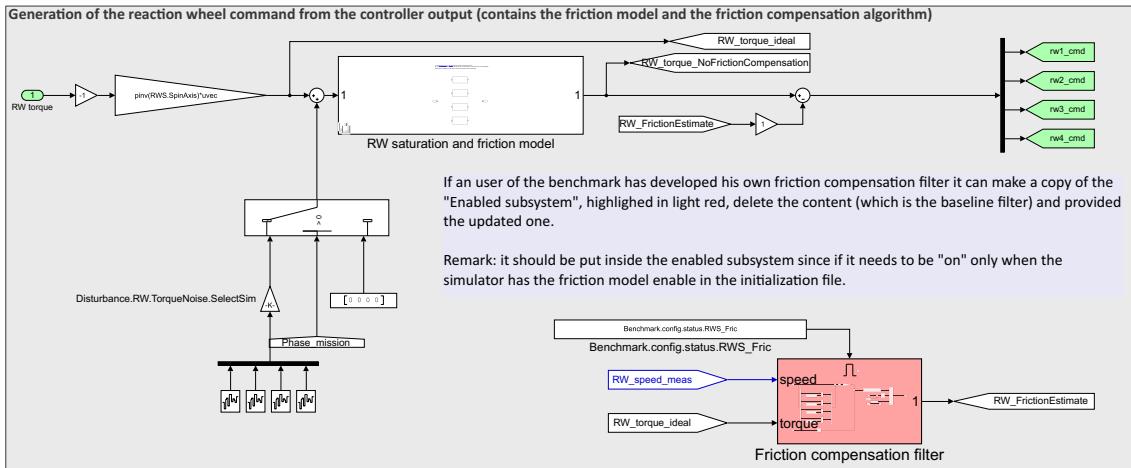


Figure 6.8: Friction model, Saturation and friction compensation filter

The friction phenomena is the results of many mechanisms, which are dependent on the contact geometry and topology, properties of the surface materials of the bodies in contact, relative displacement and velocity, presence of lubrication and etc. Different models, generally distinguished between static and dynamic model, have been developed to characterize these phenomena and a comprehensive review of such models is reported

in [33]. In the benchmark a static model is used to describe the RW friction torque. The model is derived as a simplified version of the Stribek friction model reported by the following equation:

$$f_f = \operatorname{sgn}(v)[f_C + (f_S - f_C) \exp\left\{-\left(\frac{v}{v_S}\right)^2\right\} + Cv] \quad (6.4)$$

where f_C and f_S are the Coulomb and stiction force respectively, C is the viscous friction coefficient, v is the relative velocity between the two surfaces and v_S is the critical Stribek velocity. The simplified model implemented in the benchmark is given by approximating the exponential term as follows:

$$f_f = \begin{cases} \operatorname{sgn}(v)[f_S + Cv] & \text{for } v \leq v_S \\ \operatorname{sgn}(v)[f_C + Cv] & \text{for } v > v_S \end{cases} \quad (6.5)$$

The results are produced with the following set of data for the RW and parameters for the friction model (see Table 6.2).

Table 6.2: RW data and friction model parameters

Parameter	Value
Coulomb friction	0.002 N m
Stiction Friction	0.0035 N m
Viscous friction constant	5e - 6 N m/rad/s
Stribek critical speed	1 rpm
Max. angular rate	4000 rpm

Friction spikes

Besides the deterministic model presented in the previous section to address the "classical" friction phenomena, a statistical representation of another friction perturbation, called torque friction instabilities, is based on the analysis conducted in ESA TRP program 'Mitigation of wheel friction torque instabilities' [34] of sudden torque variation phenomena happening during reaction wheels operation. The output of the study, based both on on-ground experimental testing with high temporal resolution and in-orbit using existing satellite telemetry, is a detailed statistical characterization of the phenomena and the definition of a model usable for satellite simulation. The model for the generation of the torque spike signal is briefly presented in this section.

Three parameters are used to characterize these disturbances:

- the Spike level [mNm] describing the magnitude of the spike;
- the repetition time [s] characterizing the occurrences of the spike torque phenomena;

- and the spike duration [s] describing how long the spike lasts.

Based on the analysis of the shape of torque level distribution on in-orbit data the following model are used to describe these parameters. An Erlang distribution with shape parameter $K = 2$ and rate parameter¹ $\lambda = 2.5$ describes the probability density function of the spike level. The random signal for the torque spike can be thus generated as follows:

$$T_{spike} = 0.63 - \frac{1}{2.5} \ln(U_1 U_2) \quad (6.6)$$

with U_1 and U_2 uniform distributed and independent random variables and 0.63 additional bias to account for the fact that the natural Erlang distribution starts from 0. Regarding the repetition time, this is given by:

$$t_{rep} = 125 - \frac{1}{0.0008} \ln(U_3) \quad (6.7)$$

with U_3 uniform distributed random variable. Finally, the duration is set to:

$$t_{duration} = 20 \text{ s} \quad (6.8)$$

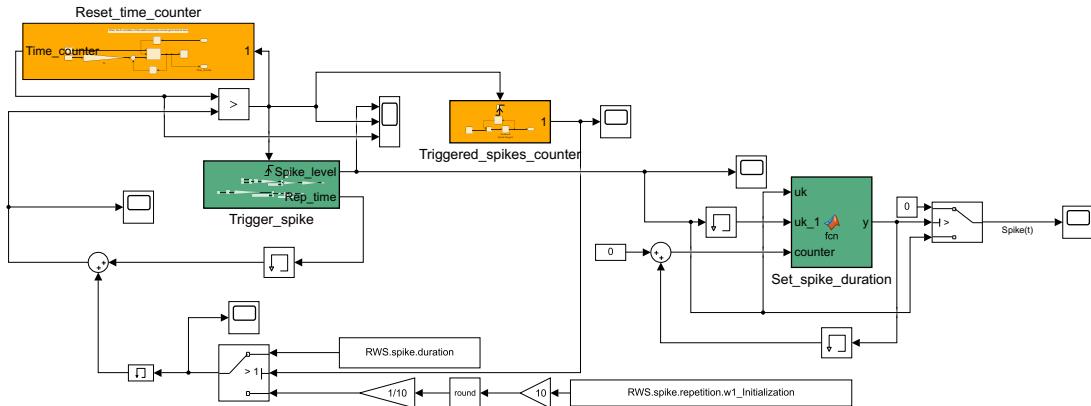


Figure 6.9: Torque spike generation function

The torque spike generation function is implemented in a Simulink block as shown in Fig. 6.9 and it is composed of the following elements:

- **RESET_TIME_COUNTER** block: this block contains a time counter that re-initializes once a specific condition is met. The condition provided by the relationship `time_counter > Repetition_time` allows the following events: 1) activates the spike signal, 2) triggers

¹The value has been modified with respect to the one provided in [34] (equal to 4.8) to increase the probability of having higher values of torque spikes.

the block TRIGGER_SPIKE (that generates the spike level and the repetition time for the next torque spike) and 3) resets the counter to the value time_counter = 0 in order to start the comparison with the repetition time just created by the TRIGGER_SPIKE;

- TRIGGER_SPIKE block: this subsystem internally computes the spike level (the output is in [Nm]) and the repetition time according to the expression given in Eq. (6.6) and 6.7 respectively;
- TRIGGERED_SPIKES_COUNTER block: the function computes how many spikes have been generated during the entire simulation. Furthermore, it provides a switching condition for the first spike generation; indeed, at the start of the simulation the trigger spike block has not generated any value for the repetition time and so this block allows to initialize its value with an externally generated repetition time (still computed according to Eq. (6.7));
- SET_SPIKE_DURATION block: It truncates the length of the constant spike signal with magnitude equal to the spike level to a certain time duration.

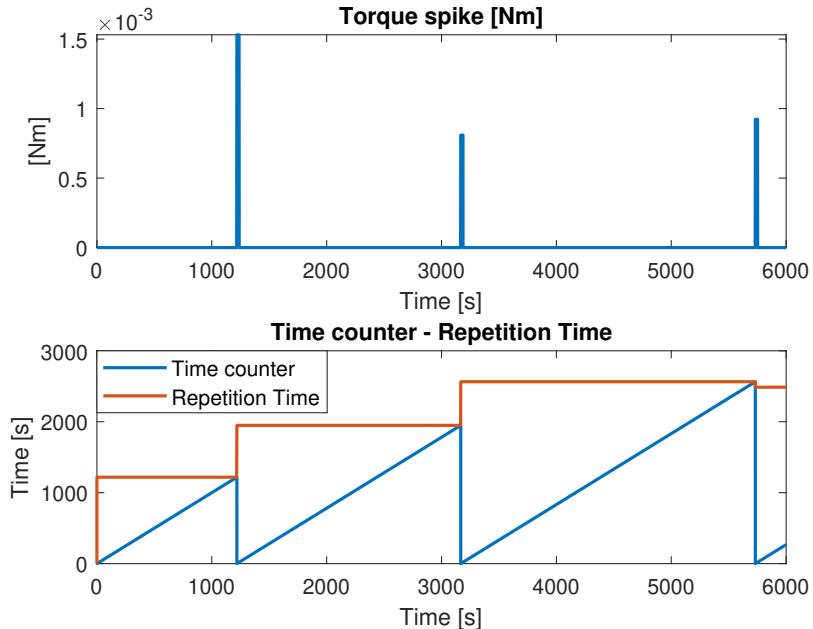


Figure 6.10: Time history of the Torque spike (top figure); Time of the counter and the spike duration compared to show that a torque spike is initiated when these two quantities are equal (bottom figure)

Figures 6.11 and Fig 6.12 also report the distributions and the cumulative density

functions for the spike magnitude and time repetition obtained by generating 10^6 samples from Eq. (6.6) and Eq. (6.7) respectively. The histograms indeed follow the Erlang distribution (as discussed in [34]). From the cumulative density functions, it is possible to determine that less than 1% of the spikes has a magnitude bigger than 3.2 mN m. Similarly, only in 1% of the instances the TRIGGER_SPIKE block is triggered the repetition time will be bigger than 5862 s.

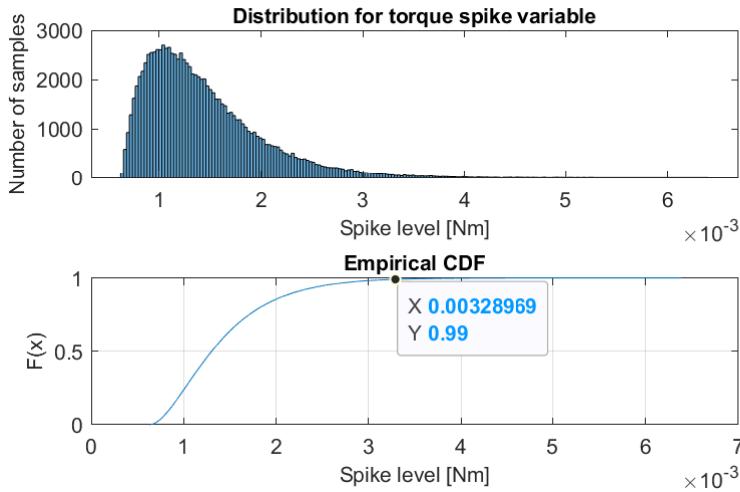


Figure 6.11: Distribution of torque spike magnitude (top figure); Cumulative density function (bottom figure)

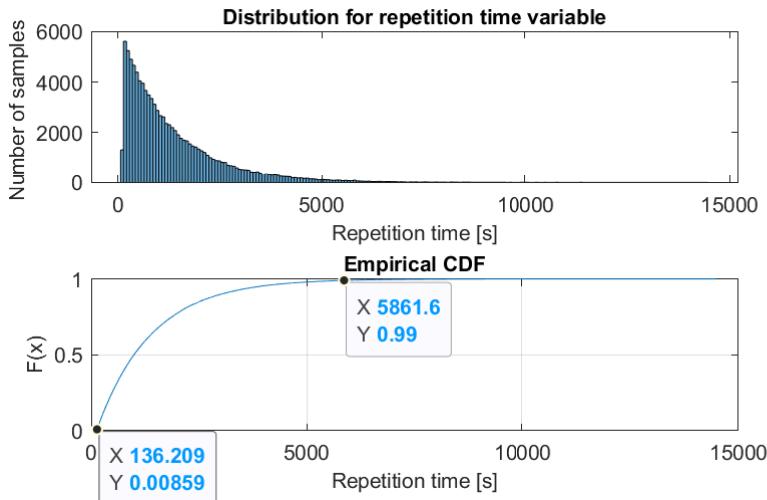


Figure 6.12: Distribution of the repetition time (top figure); Cumulative density function (bottom figure)

Friction compensation

A first-order filter to estimate the friction torque using the input torque command and the measurement of RW speed is implemented to compensate this disturbance. This is the red block reported in Fig. 6.8 The friction estimator is written as follow:

$$\tilde{T}_{frict} = \mathbf{K}_{f.e.}(s)(T_{cmd} - \Delta T_{rw}) \quad (6.9)$$

where \tilde{T}_{frict} is the estimate of the friction, T_{cmd} is the command from the controller to the RW, ΔT_{rw} is an estimate of torque variation and $\mathbf{K}_{f.e.}(s)$ is the compensation filter given by:

$$\mathbf{K}_{f.e.}(s) = \frac{k_{f1}}{s + k_{f2}} \quad (6.10)$$

with $k_{f1} = 0.02$ and $k_{f2} = 0.001$. The selection of the gain and bandwidth of the filter is based on a trade-off between RW tachometer noise rejection, steady state estimation error and tracking speed. The term ΔT_{rw} is approximated by backward difference as:

$$\Delta T_{rw}(t_k) = \frac{H_{rw}(t_k) - H_{rw}(t_{k-1})}{t_s} \quad (6.11)$$

where t_s is the ACS sample time and H_{rw} is the wheel angular momentum. In discrete time the friction estimation of Eq. (6.9) is given as:

$$\tilde{T}_{frict}(t_k) = \frac{1}{1 + k_{f2}t_s}\tilde{T}_{frict}(t_{k-1}) + \frac{k_{f1}t_s}{1 + k_{f2}t_s}(T_{cmd}(t_k) - \frac{H_{rw}(t_k) - H_{rw}(t_{k-1})}{t_s}) \quad (6.12)$$

The behaviour of the filter is shown in both nominal RW condition (Fig. 6.13) and in presence of RW rate reversal (Fig. 6.14)

European Satellite Benchmark

Copyright (c) ISAE-SUPAERO, All Rights Reserved

ESA 4000141060/23/NL/MGu

User Manual

ESA Public Licence - v2.4 - Permissive (Type 3)

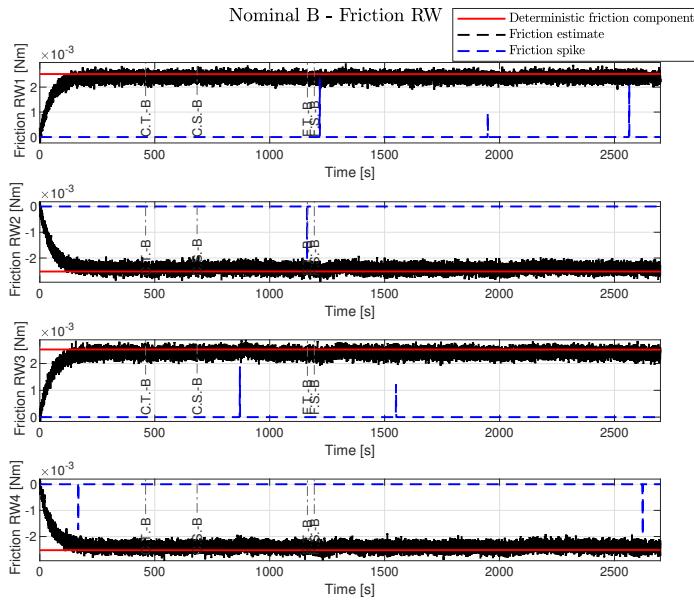


Figure 6.13: Reaction wheel friction and friction estimate

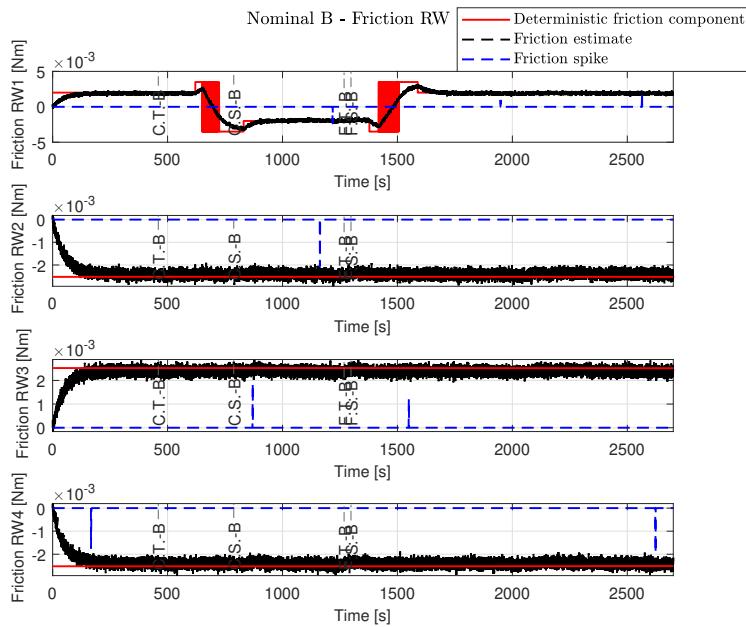


Figure 6.14: Reaction wheel friction and friction estimate with RW-1 presenting 2 rate reversals

6.2.1.2.4 Thrusters (RCS)

This section addresses the description of the non-linear model for the thrusters system implemented for the Simscape simulator. The control input (consisting of forces/torques in case of both position and attitude control) needs to be mapped into thrusters' forces. In the simulator this operation is accomplished by solving a linear programming problem. Furthermore, because of the thrusters' on-off nature, the output of the thruster mapping algorithm is the input of the pulse-width modulator that modulates a continuous thrust in a pulsed thrust aimed at minimizing the difference between the actual pulsed control and the designed controller. The Pulse-Width Modulation (PWM) function is build also considering thruster's characteristics such as the Minimum Impulse Bit (MIB). This PWM is implemented in the function 6.1.

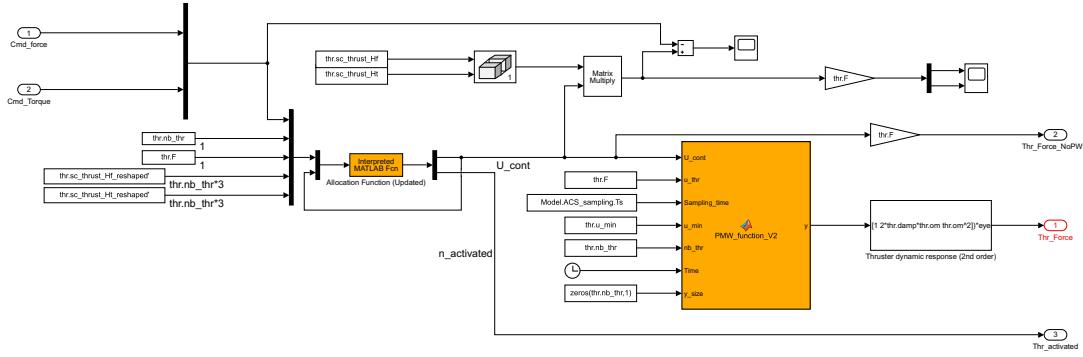


Figure 6.15: RCS thrusters model

Allocation function

Let us consider the vector $\mathbf{u}_{th} \in \mathbb{R}^n$ given by:

$$\mathbf{u}_{th} = u_a [\hat{u}_1 \quad \hat{u}_2 \quad \dots \quad \hat{u}_n]^T \quad (6.13)$$

with n number of thrusters, u_a the available force generated by the thruster and \hat{u}_i , binary vector, expressed as:

$$\hat{u}_i = \begin{cases} 0 & \text{if } i\text{-th thruster off} \\ 1 & \text{if } i\text{-th thruster on} \end{cases} \quad i = 1, 2, \dots, n \quad (6.14)$$

The forces/torques applied on the spacecraft are then given as follows:

$$\begin{bmatrix} \mathbf{F}_c \\ \mathbf{T}_c \end{bmatrix} = \begin{bmatrix} \mathbf{H}_F \\ \mathbf{H}_T \end{bmatrix} \mathbf{u}_{th} = \mathbf{H} \mathbf{u}_{th} \quad (6.15)$$

with $\mathbf{H} \in \mathbb{R}^{6 \times n}$ thrust distribution matrix related to the geometrical structure of the thrusters' placement on the spacecraft.

Given some commanded value of \mathbf{F}_c and \mathbf{T}_c a linear programming problem of the form (Eq. (6.16) and (6.17)) is solved by finding the vector $[\hat{u}_1 \ \hat{u}_2 \ \dots \ \hat{u}_n]$ such that:

$$\hat{u}^* = \min_{\hat{u}} \sum_{i=1}^n \hat{u}_i \quad (6.16)$$

with the following constraints

$$\begin{cases} u_a \mathbf{H}_F \hat{\mathbf{u}} = \mathbf{F}_c \\ u_a \mathbf{H}_T \hat{\mathbf{u}} = \mathbf{T}_c \\ 0 \leq \hat{u}_i \leq 1 \text{ for } i = 1, 2, \dots, n \end{cases} \quad (6.17)$$

Additional constraints are added to avoid residual forces applied to the spacecraft after the PWM command. Indeed, without the additional requirement, even in presence of a zero commanded force, there could be cases where some thrusters are not activated due to the MIB constraint and thus generating a residual force. The additional requirements sets a particular thrusters firing configuration (it sets equal the continuous output for pairs of thrusters) always allowing to have zero force when demanded.

PWM signal

Solved the constrained minimization problem in Eqs. (6.16) and (6.17)), we still have positive continuous variables which needs to be converted into a on-off command. Let us consider the k -th time step of duration T . The following three parameters are used to characterize the PWM signal and are derived as described in [35]:

- the discrete signal value $\bar{u}(kT)$:

$$\bar{u}(kT) = \text{sgn}(\hat{u}(kT)) u_a \quad (6.18)$$

- the duration of the pulse $\sigma(kT)$:

$$\sigma(kT) = T u_a \hat{u}(kT) / \bar{u}(kT) \quad (6.19)$$

- the shift applied to the pulse $\tau(kT)$ in order to center it with respect to the sampling period. The use of this additional parameter is motivated in [35] by showing that the residual error between the state controlled by the discrete time signal $\hat{u}(kT)$ and the one controlled by the PWM input signal $\bar{u}(kT)$ of duration $\sigma(kT)$ and starting at $\tau(kT)$ is $\propto T^3$.

$$\tau(kT) = [T - \sigma(kT)]/2 \quad (6.20)$$

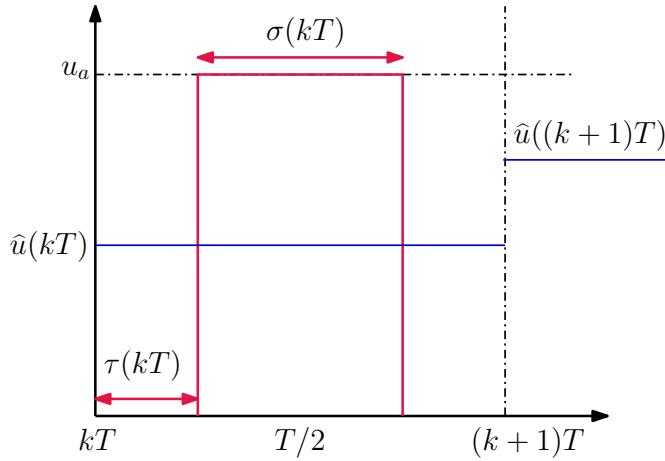


Figure 6.16: Signal modulated with PWM in a sampling period T

In order to account for the Minimum Impulse Bit (MIB) that thrusters can provide Eq. (6.19) is modified as follows:

$$\sigma(kT) = \begin{cases} 0 & \text{if } |\hat{u}(kT)| < u_{min} \\ T u_a \hat{u}(kT) / \bar{u}(kT) & \text{if } u_{min} \leq |\hat{u}(kT)| < u_a \\ T & \text{if } |\hat{u}(kT)| \geq u_a \end{cases} \quad (6.21)$$

where u_{min} is computed from the MIB time T_{MIB} as:

$$u_{min} = T_{MIB} u_a / T \quad (6.22)$$

```

function y = PMW_function_V2(U_cont,u_thr,Sampling_time,u_min,nb_thr,Time,
y_size)

%
% Copyright (c) ISAE-SUPAERO, All Rights Reserved.
% Paolo Iannelli, Francesco Sanfedino, Daniel Alazard (2024)
%

%
% Inputs:
% U_cont:           vector of the thrusts after simplex algorithm in the
%                   range [0, u_thr] - size: [1, nb_thr]
% u_thr:            Magnitude of the thrusters [N]
% Sampling_time:    Sampling_time [s]
% u_min:            Computed from the MIB [N]
% nb_thr:           Number of thrusters
% Time:             Simulation time [s]
% y_size:           Initialize the vector y (zeros(thr.nb_thr,1))

eps=2e-3;

```

```

y=y_size;                                     % Define the size of the output vector
K_sample=floor(Time/Sampling_time); % Time sample

%u_PWM=sign(U_cont)*u_thr;                  % Vector with values of the PWM signal
u_PWM=u_thr*ones(nb_thr,1);

% index_low: Find the thrusters that in, that time interval, cannot provide
%             the required force (since what is required is lower than the
%             specification given by the Minimum Impulse Bit)
index_low = find(abs(U_cont)<u_min);

% index_value: Find the thrusters that, in that time interval, can be
%               activated and the required force is lower than the
%               saturation
%               level (for this it is computed the start of the pulse and
%               its
%               duration)
index_value = find(abs(U_cont)>=u_min & abs(U_cont)<u_thr-eps);

% index_saturation: Find the thrusters that, in that time interval, are
%                   activated for the full duration of the interval since
%                   the required force value is over the saturation level
index_saturation = find(abs(U_cont)>=u_thr-eps);

% Duration of the pulse ( Vector of size: size: [1,nb_thr])
sigma_PWM=Sampling_time*(U_cont./u_PWM);

% Delay applied to the pulse (The pulse does not begin at the start of the
% sampling period). tau_PWM=0 when we reach saturation.
tau_PWM=(Sampling_time-sigma_PWM)/2;

for i=1:nb_thr

    if sum(i==index_low)==1 % lower than MIB = 0
        y(i)=0;
    elseif sum(i==index_saturation)==1 % saturation equal open for the
        full interval (there is a small activation time equal to 0.005 s)
        y(i)=u_PWM(i);
    elseif sum(i==index_value)==1 % The pulse is centered in the
        interval with the delay tau_PWM

        if Time-K_sample*Sampling_time-tau_PWM(i)<0
            y(i)=0;
        elseif Time-K_sample*Sampling_time-tau_PWM(i)>=0 & Time-
        K_sample*Sampling_time-tau_PWM(i)-sigma_PWM(i)<=0
            y(i)=u_PWM(i);
        elseif Time-K_sample*Sampling_time-tau_PWM(i)-sigma_PWM(i)>0
            y(i)=0;
    end

```

```
end
```

```
end
```

Listing 6.1: Generation of the pulse-width modulated signal

A simplified model for the micro-thrusters is implemented. The allocation problem is addressed in the same manner as for the RCS thrusters.

Unlike the RCS thrusters, the micro-thrusters can generate a quasi-continuous thrust with a certain resolution. This functionality is implemented using a **Quantizer** block in Simulink (see Fig.6.17).

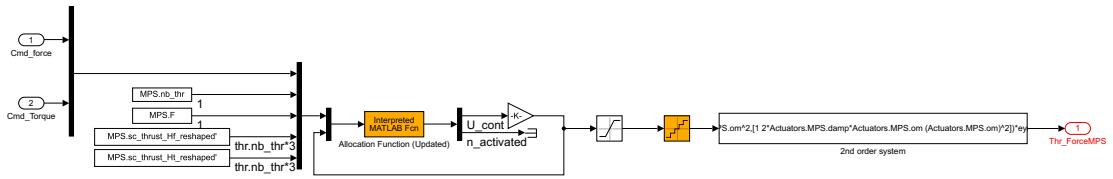


Figure 6.17: MPS thrusters model

6.2.1.3 Tank + Sloshing model

The sloshing model implemented in Simscape is reported in Fig. 6.18 and consists of 6 masses connected via cartesian joint (3 translational DoFs) with internal stiffness and damping along the 3 joint axis. The position of the masses with respect to the tank's center is provided in Eq. (4.1).

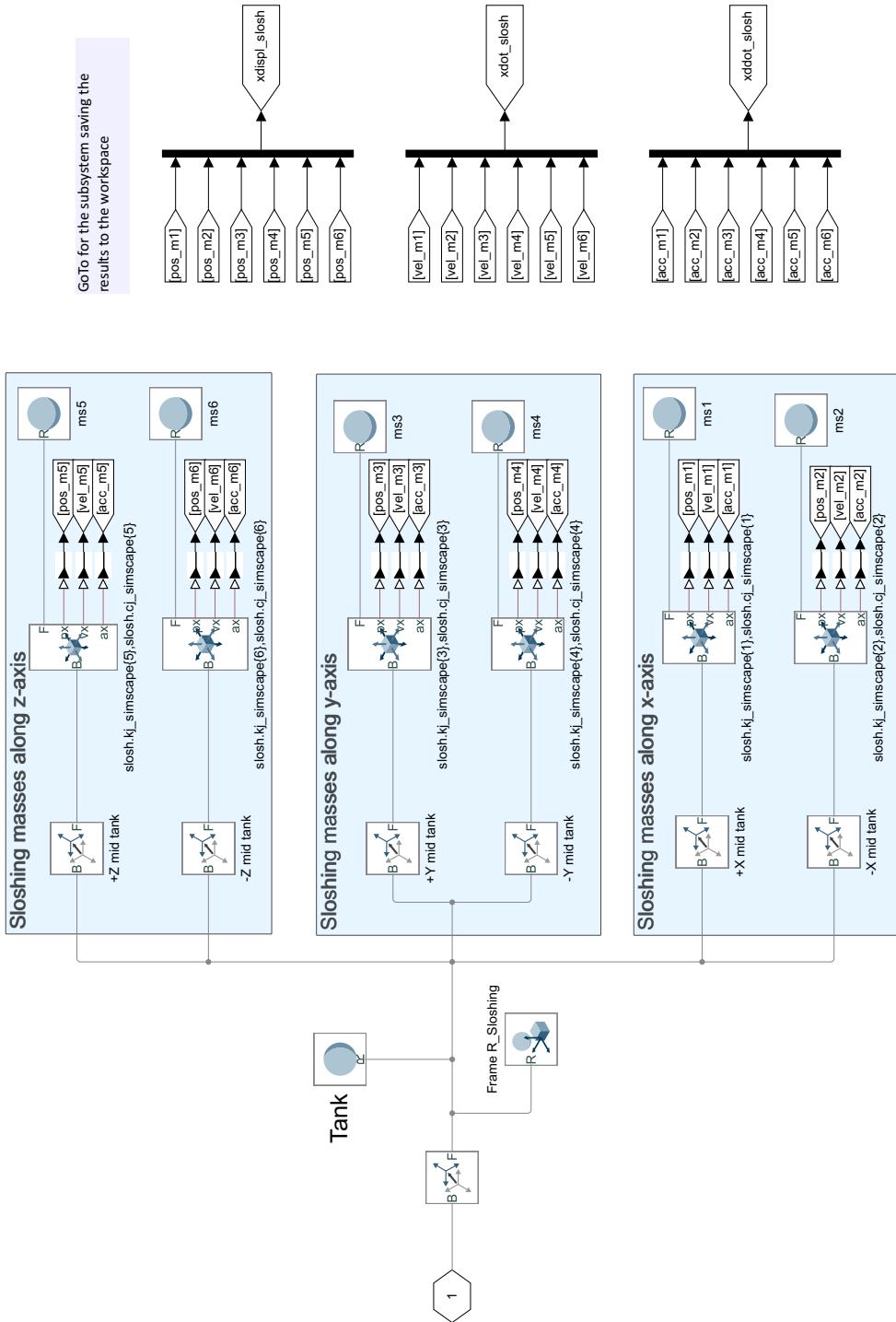


Figure 6.18: Tank and sloshing equivalent mechanical model

6.2.1.4 Payload System (Telescope structure, FSM, PMA system, Isolator)

Figure 6.19 shows the Payload System (Telescope structure, FSM, PMA system, Isolator) subsystem. The most significant blocks are highlighted with different colors in the figure:

- In **Orange blocks**, we have mechanical elements related to the payload isolator, the mounting plate and the telescope. Two blocks are used for the telescope structure:
 - The first one is the SIMSCAPE MULTIBODY block **Reduced Order Flexible Solid** created with a single boundary node (at the interface node 963 which is the interface with the isolator plate).
 - In Matlab version 2020b the modal reduction in the **Reduced Order Flexible Solid** block is not available, thus, in the case of the telescope model, it is not feasible to directly extract additional boundary points. This issue is solved by using the transfer function model (uncertain state-space to account for parameter variation for worst-case analysis and Monte Carlo simulation) extracted from SDTLIB with the additional outputs at the primary and secondary mirror.
- In the **Green blocks**, the actuators systems (FSM and PMA) are shown.
- In the **light blue blocks**, the optical sensitivity matrices from the two mirrors and the FSM contributing to the LOS are implemented.

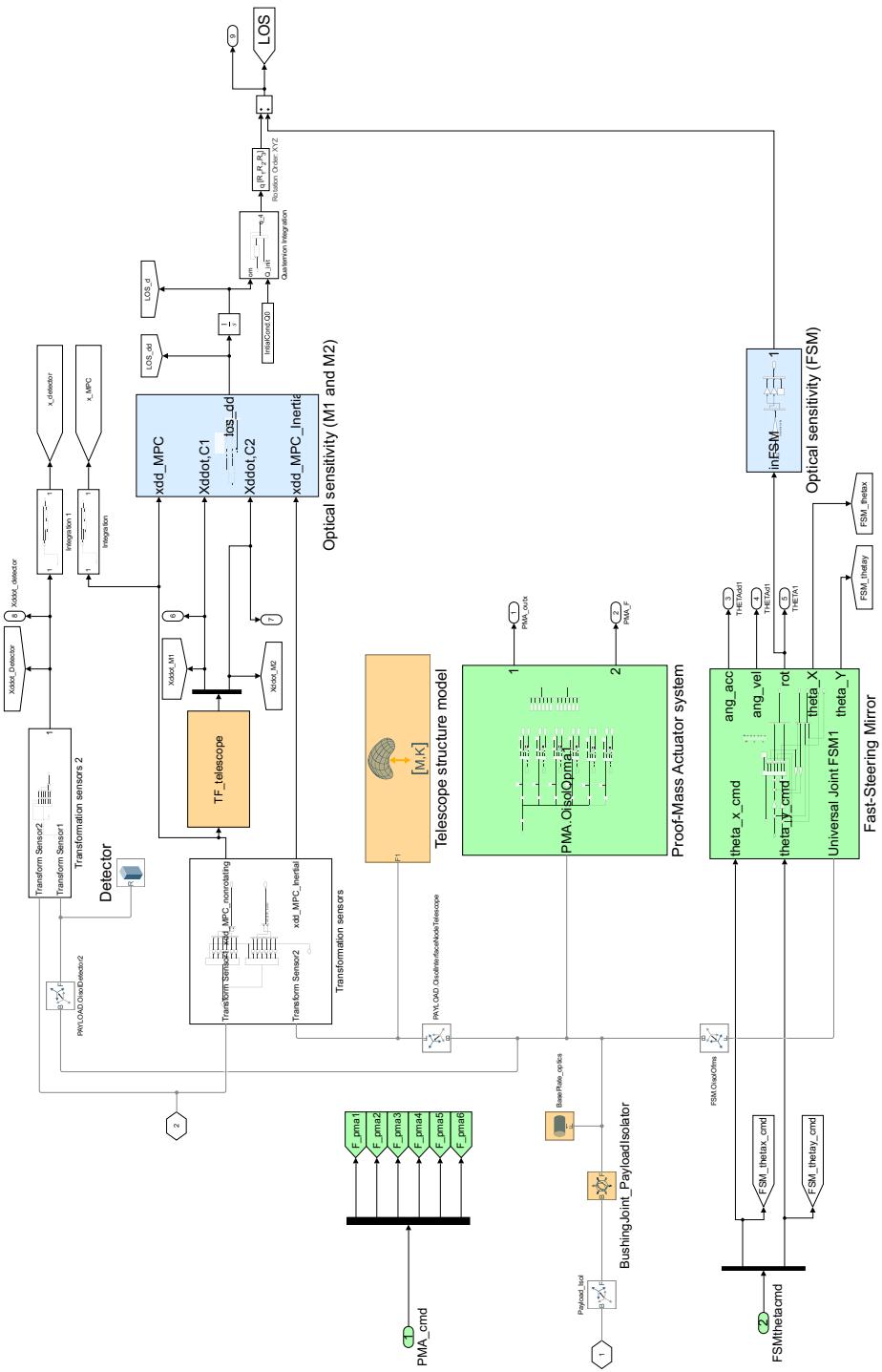


Figure 6.19: Payload System (Telescope structure, FSM, PMA system, Isolator)

6.2.2 Simulator subsystem: SENSORS

Figure 6.20 shows the content of the simulator SENSORS subsystem. The contribution of each noise on the model can be tested by activating/de-activating them as reported in Sect. 6.3.1. The noise data for each sensors and the acquisition frequency are reported in following list:

1. STAR TRACKER (STR)

- Noise:

Noise	Bias
Across boresight: $0.7 \text{ arcsec}/\sqrt{\text{Hz}}$ (3σ)	Across boresight: 0.4 arcsec (3σ)
Along boresight: $4.7 \text{ arcsec}/\sqrt{\text{Hz}}$ (3σ)	Along boresight: 2.7 arcsec (3σ)

- Acquisition frequency: 20 Hz

2. COARSE GYROSCOPE (GYR-C)

- Noise: Only ARW and bias instabilities are considered as noise contributions in the gyro model while quantization and RRW noise are omitted. The bias instability noise has a low-frequency nature and it shows up as the bias fluctuations in the data. The rate PSD associated with this noise is [36]:

$$S_\Omega(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right) \frac{1}{f} & \text{if } f \leq f_0 \\ 0 & \text{if } f > f_0 \end{cases} \quad (6.23)$$

where

- B is the bias instability coefficient;
- f_0 is the cutoff frequency.

To model such noise contribution a random signal with zero mean and unit variance is filtered through a filter matching the ASD correponding to the PSD in the Eq. 6.23. An approximation of the filter is generated by the function `designFlickerFilter2`.

ARW	Bias instability coefficient	Cut-off frequency
$< 0.0016^\circ/\sqrt{h}$ (1σ)	$< 0.01^\circ/h$ (1σ)	1e-4 Hz

- Acquisition frequency: 100 Hz (fixed at 20 Hz in the simulator)

3. FINE GYROSCOPE (GYR-F)

- Noise: Only ARW and bias instabilities are considered as noise contributions in the gyro model while quantization and RRW noise are omitted. The bias instability contribution is implemented in the same way as the coarse gyro.

ARW	Bias instability coefficient	Cut-off frequency
$< 0.0001^\circ/\sqrt{h} (1\sigma)$	$0.0005^\circ/h (1\sigma)$	1e-4 Hz

- Acquisition frequency: 100 Hz (fixed at 20 Hz in the simulator)

4. FINE GUIDANCE SENSOR (FGS)

- Noise: Across boresight: $\sigma_{fgs} = 10$ mas
- Acquisition frequency: 10 Hz

5. ACCELEROMETERS AT PAYLOAD LEVEL

- Noise:
 - the noise in the linear acceleration measurements is described by an ASD with the following template that shapes a random signal with zero mean and unit variance:
- $$\begin{cases} 3 \mu\text{g}/\sqrt{\text{Hz}} & f \leq 10 \text{ Hz} \\ 0.7 \mu\text{g}/\sqrt{\text{Hz}} & 10 \text{ Hz} \leq f \leq 200 \text{ Hz} \end{cases}$$
- the noise in the angular acceleration measurements is described by its standard deviation $\sigma_a = 0.00019 \text{ rad/s}^2$
 - Acquisition frequency: 200 Hz
 - The accelerometers cannot register static and quasi-static contribution and the minimum low cut-off frequency is 0.5 Hz. In the simulator an high-pass filter of 4th order is added, cutting the acceleration measurement at an higher frequency (20 Hz) in order to reduce the noise contribution in the output (see transfer function `Sensors.ACCm1.highpass`, `Sensors.ACCm1.highpass` and `Sensors.ACCdect.highpass`).

6. FSM ROTATION MEASUREMENT

- Noise: $\sigma_{fsm} = 10^{-8} \text{ rad}$
- Acquisition frequency: 200 Hz

7. RW TACHOMETER

- Noise: $\sigma_{tach} = 0.01 \text{ rad/s}$
- Acquisition frequency: 20 Hz

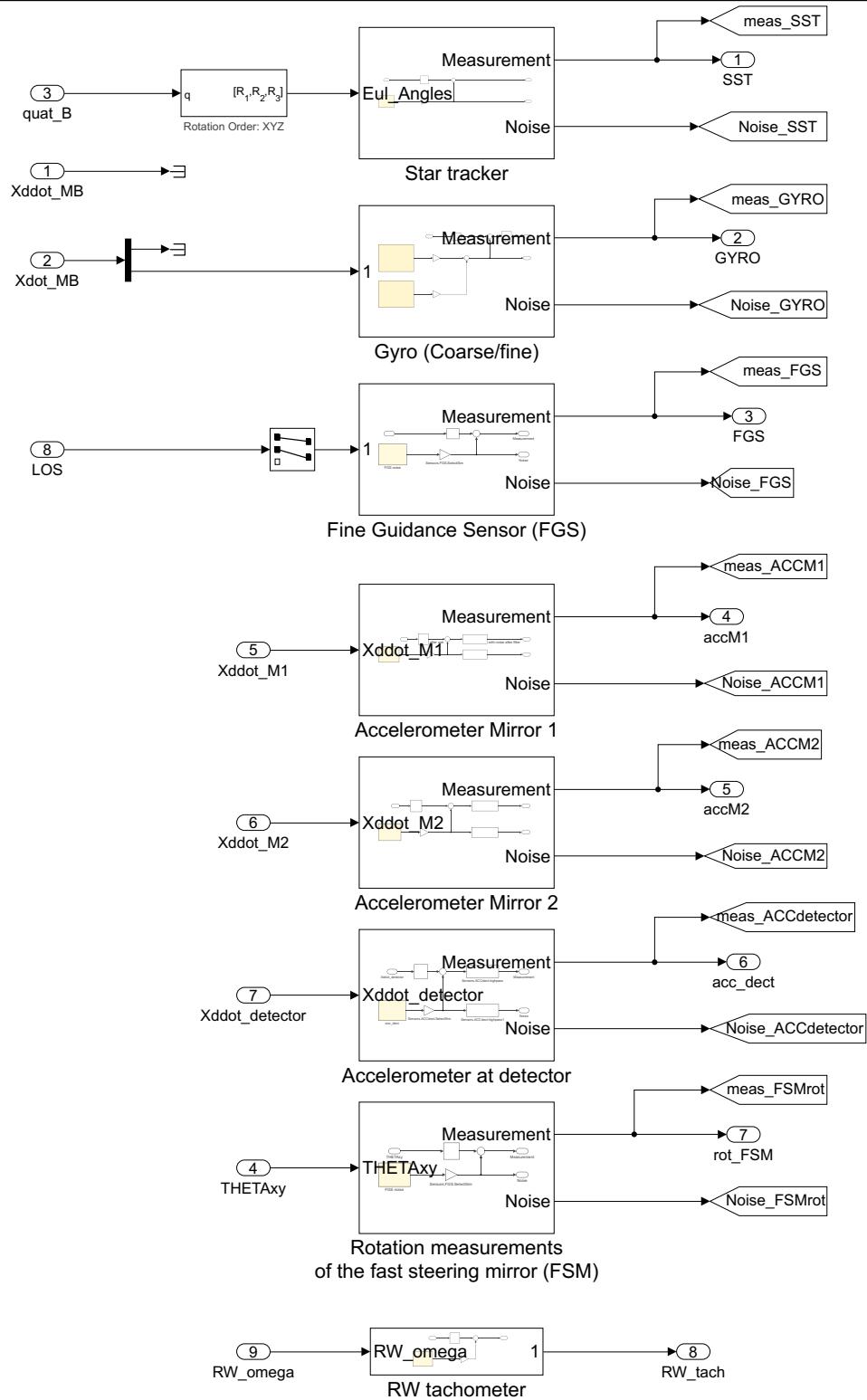


Figure 6.20: Simulator subsystem: SENSORS

6.2.3 Simulator subsystem: OUTPUT TO WORKSPACE

The OUTPUT TO WORKSPACE block is used to save the simulation data. A brief description of the data is reported in Table 6.3 and 6.4.

Table 6.3: List of the output of time-domain simulation (List split in two - Part 1)

Name	Brief Description
out.CoarseAPE_notMet	1 means that the coarse APE requirement has been met. See explanation on Sect. 6.2.4
out.FSM_thetax	FSM rotation about x-axis - [rad]
out.FSM_thetay	FSM rotation about y-axis - [rad]
out.LOS	Line-of-Sight - [rad]
out.slew_ref	3-axes reference trajectory for slew maneuver - [rad]
out.FineAPE_notMet	1 means that the fine APE requirement has been met. See explanation on Sect. 6.2.4
ut.PMA_fz1→6	PMA force - [N]
out.PMA_x1→6	PMA displacement - [m]
out.Phase_mission	Used to determine the phase of the mission the spacecraft is in at time t . 0, 1, 2, 3, 4 correspond to slew maneuver, slew to coarse transient, coarse pointing phase, coarse to fine transient and fine pointing phase respectively.
out.RCS_OutAllocation	Thruster forces as output of the allocation block - [N]
out.RCS_active_Allocation	Number of thrusters activated at time t
out.RCS_cmd2Thrusters	PWM command to thrusters - [N]
out.RW1→4_dist_at_revolute	Reaction wheel disturbances - [Nm]
out.RW1→4_speed	Reaction wheel speed - [rad/s]
out.RW_Friction	Reaction wheel friction - [Nm]
out.RW_FrictionEstimate	Reaction wheel friction estimate - [Nm]
out.RW_cmd	Torque command to reaction wheels - [Nm]
out.RW_torque_NoFrictionCompensation	Uncompensated reaction wheels torque - [Nm]
out.RW_torque_ideal	Ideal reaction wheel torque - [Nm]
out.RW_torque_spikes	Torque spikes - [Nm]

To reduce the size of the output file, whenever a type of actuator is not used throughout the entire simulation, its output is left empty (not a zero vector). Additionally, for Monte Carlo runs, the reaction wheel disturbances are not saved for the same reason.

Table 6.4: List of the output of time-domain simulation (List split in two - Part 2)

Name	Brief Description
out.SADM1_theta	Solar array 1 rotation - [rad]
out.SADM2_theta	Solar array 1 rotation - [rad]
out.Xddot_Detector	Acceleration at detector - [m^2/s]
out.Xddot_M1	Acceleration at primary mirror - [m^2/s]
out.Xddot_MB	Acceleration at Secondary mirror - [m^2/s]
out.cmd2FSM	Angle command to FSM (output of the controller) - [rad]
out.cmd2MPS	Torque command to MPS thrusters (Output of the controller) - [Nm]
out.cmd2PMA	Force command to PMA (Output of the controller) - [N]
out.cmd2RCS	Torque command to RCS thrusters (Output of the controller) - [Nm]
out.cmd2RW	Torque command to reaction wheels (Output of the controller) - [Nm]
out.meas_FGS	Measurement from fine guidance sensor
out.meas_GYRO	Measurement from gyro
out.meas_SST	Measurement from star tracker
out.quat_B	Quaternion vector $[q_0, q_1, q_2, q_3]$
out.time	Time sampled at 200 Hz
out.timeACS	Time sampled at 20 Hz
out.xdispl_slosh	Sloshing equivalent mechanical model masses displacement - [m]
out.SimulationMetadata	Simulation metadata

6.2.4 Simulator subsystem: CHECK ON APE REQUIREMENTS FOR PHASE TRANSITION

This section implements the conditions to transition among the mission phases (briefly discussed in Sect 6.1. Fig. 6.21 shows the blocks devoted to the transition between the slew maneuver and the slew to coarse transient phase. It is composed of 2 blocks:

- The first block outputs a signal equal to 1 when a) APE_1 requirement associated to the slew and b) the reference signal has reached 95 % of the steady state value.
- The second block checks the output of the first block is kept equal to 1 for a time window of duration `PHASE.Slew.TimeCheck` (100 s).

Once this condition is verified, the slew phase is completed.

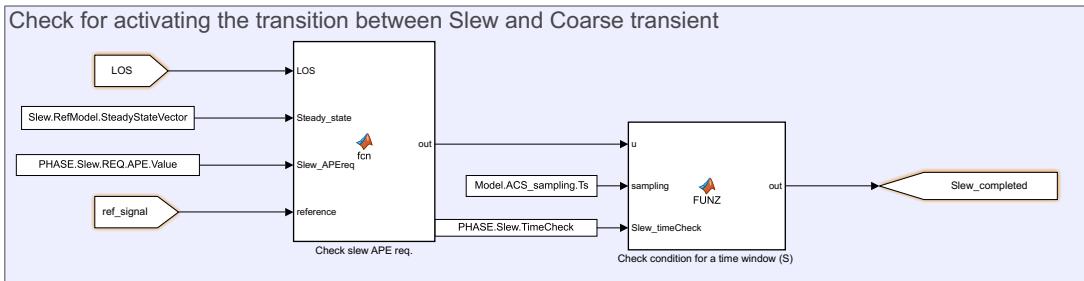


Figure 6.21: Check for completion of the slew phase

The second block is devoted to the transition between the coarse transient and the coarse pointing phase. The implementation is similar to the previous one:

- a first block outputs a signal equal to 1 when a) APE_2 requirement associated to the coarse pointing is met.
- a second block checks the output of the first block is kept equal to 1 for a time window of duration `PHASE.TransitionS2C.TimeCheck` (30 s).

$$APE(t) \leq APE_2 \quad \forall t \in [\bar{t}, \bar{t} + \Delta t_{s \rightarrow c}]$$

If the second condition is never met, the code forces the transition after a time interval of duration `PHASE.TransitionS2C.TimeCheck_reqNotMet` (360 s). If that is the case, the system will record this event and save this into the variable `CoarseAPE_notMet` (value equal to zero if the transition is forced).

The third block is devoted to the transition between the coarse pointing phase and coarse to fine pointing transition. The transition is initialized when the coarse pointing phase has lasted for the duration `PHASE.Coarse.Duration` (600 s)

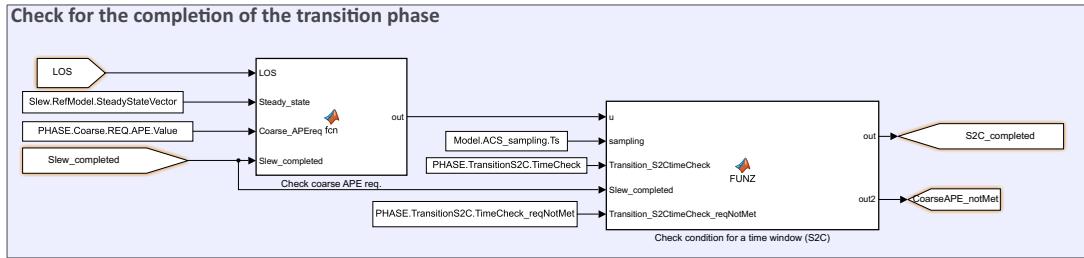


Figure 6.22: Check for initializing the transition between coarse transient and coarse pointing phase

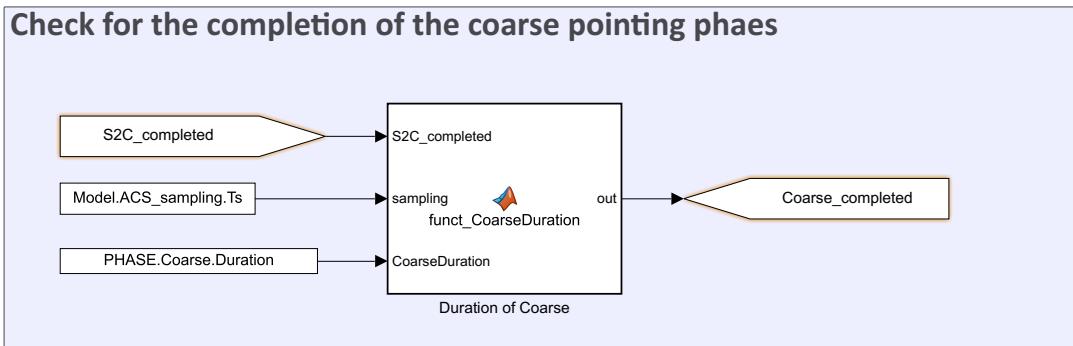


Figure 6.23: Check for coarse pointing completion

The second block is devoted to the transition between the fine transient and the fine pointing phase. The implementation is the following:

- a first block outputs a signal equal to 1 when a) APE_3 requirement associated to the fine pointing is met.
- a second block checks the output of the first block is kept equal to 1 for a time window of duration `PHASE.TransitionC2F.TimeCheck` (30 s).

$$APE(t) \leq APE_3 \quad \forall t \in [\bar{t}, \bar{t} + \Delta t_{c \rightarrow f}]$$

If the second condition is never met, the code forces the transition after a time interval of duration `PHASE.TransitionC2F.TimeCheck_reqNotMet` (360 s). If that's the case, the system will record this event and save this into the variable `FineAPE_notMet` (value equal to zero if the transition is forced).

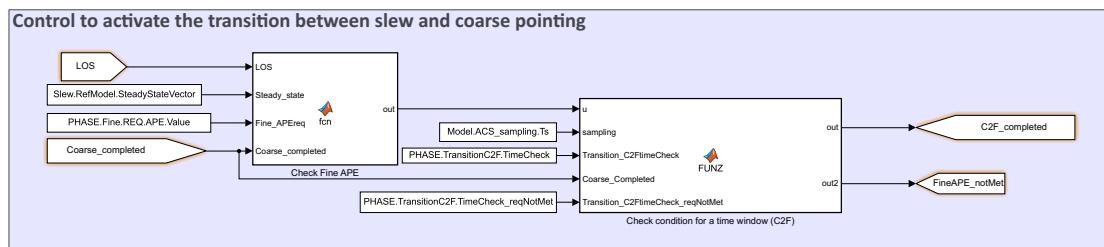


Figure 6.24: Check for initializing the transition between the fine transient and the fine pointing phase

6.3 Tutorial: Setting up a time-domain analysis

In order to start a time-domain analysis the user must select which type of simulation to run in the main file `Main_TimeDomain.m` (See Code 6.2).

Three types of simulations are available:

1. TIME-DOMAIN SIMULATION WITH NOMINAL CONFIGURATION of the uncertain parameters (refer to Table C.1 in Appendix C). While all the parameters are considered in nominal configuration, the varying parameters of the model (solar arrays angular position and RW speeds) are initialized with the following values and they corresponds to the initial condition provided in the simulator:

- Initial condition for the RW speeds:

$$\Omega_{RW} = [-1000 \quad 1000 \quad -1000 \quad 1000] \text{ RPM}$$

which corresponds to $0.25 \cdot \Omega_{max}$.

- Initial condition for the solar arrays' angle:

$$\theta_{SA}^{(1)} = \theta_{SA}^{(2)} = 0 \text{ deg}$$

2. TIME-DOMAIN SIMULATION WITH WORST-CASE (OR USER DEFINED) CONFIGURATION of the uncertain parameters provided by the user. The methodology on how to provide the worst-case configuration is documented in Sect. 6.3.2.
3. MONTE CARLO ANALYSIS: This type of analysis is run with the PARALLEL COMPUTING TOOLBOX as discussed in Sect. 6.3.3.

```
%%
% Select which type of analysis to run (Only one of the 3 can be selected
% at a time):
%
% ANALYSYS_TYPE = 1 if NOMINAL MODE ANALYSIS (all uncertain value are set
% to their nominal value
% as in file Config_Nominal.m)
% ANALYSYS_TYPE = 2 if WORST-CASE (OR USER DEFINED) CONFIGURATION ANALYSIS
% ANALYSYS_TYPE = 3 if MONTE CARLO ANALYSIS
%
%
ANALYSYS_TYPE = 1;
%
%
% IF SELECTION OF A WORST-CASE CONFIGURATION (OR USER DEFINED)
% use the code "WC_template" in folder Utilities\Functions\Analysis
% to generate a .mat (here the file is called WC_config)
Analysis.WCfileName='WC_config_RPEcoarse_TimeDomain_run';
```

```
% IF SELECTION OF MONTE CARLO ANALYSIS PLEASE DEFINE THE FOLLOWING
% PARAMETERS
MCsim.num_batches = 1; % Number of batches to run (each batch will be saved
                        % into a separate file)
MCsim.freecores = 0;   % Select the numbers of cores to not use for the
                        % parallel pool
MCsim.num_sub_batches = 1; % Number of sub-batches

% Name saved .mat file for Nominal and WC analysis
Name_Save_Nominal=      'Nominal';
Name_Save_NominalData=   'Nominal_Data';

Name_Save_WC=            'Benchmark_WC_stabilityCoarse';
Name_Save_WCData=        'Benchmark_WC_stabilityCoarse_Data';
```

Listing 6.2: Selection of the time domain analysis in code Main_TimeDomain.m

6.3.1 Time domain initialization

This section presents the code `Main_TimeDomain.m`, which is dedicated to initializing and setting up the time domain simulation. For brevity and clarity, only the essential parts of the code are discussed.

1. First, the code loads data for the sensor disturbances and requirements defined at the synthesis design stage thanks to the core routine `TimeDomain_Initialization` and the baseline controller solution proposed by ISAE- SUPAERO (Note: if the user wants to plug its own control solution, please refer to Section 6.4).
2. The user can specify which actuators are used during the benchmark. This step is necessary to remove unnecessary variables from the simulator's output file, thereby reducing its size.

```
% State the actuators used during the benchmark (are just used to
    % clean up
% the output file of the simulator from unnecessary variables) ->
    % reduce
% size

Actuators.RW.Select = 1;
Actuators.RCS.Select = 1;
Actuators.MPS.Select = 0;
Actuators.FSM.Select = 1;
Actuators.PMA.Select = 0;
```

3. The user can specify the parameters for the timeline duration and the checks on the transitions between controllers, as specified in the Table 6.1, are defined.

```
%% _____
% Definition of the timeline for the benchmark problem
%
PHASE.Ttotal = 2700; % Total duration of the benchmark problem

% The duration that are not fixed are the slew and the two transitions
PHASE.Slew.TimeCheck = 100; % Control to check if the APReq of
                             the slew (at steady state) is met

PHASE.Coarse.Duration = 600; % Duration of the coarse phase

% DeltaT in which the Coarse req. should be met to initialize coarse
% steady-state phase
PHASE.TransitionS2C.TimeCheck=30;
PHASE.TransitionS2C.TimeCheck_reqNotMet=360;
% DeltaT in which the Coarse req. should be met to initialize fine
% steady-state phase
PHASE.TransitionC2F.TimeCheck=60;
PHASE.TransitionC2F.TimeCheck_reqNotMet=360;
```

- The user can select which disturbances to include in the analysis. A value of 1 means the disturbance is active, while a value of 0 means it is inactive.

```
%% _____
% DISTURBANCES SELECTION
%
%
% RWS submodel
Benchmark.config.status.RWS_Harmdist =           1; % 1 for Harmonic
                                                 disturbances with static/dynamic mass imbalances
Benchmark.config.status.RWS_Sat =                 1; % 1 for saturation
                                                 block (No friction model and no friction spike)
Benchmark.config.status.RWS_Fric =                1; % 1 for Friction
                                                 model (Strikebeck-like model)
Benchmark.config.status.RWS_Frict_spikes =        1; % 1 for Friction
                                                 spikes generation

% RW broadband disturbance
Disturbance.RW.BroadBand.SelectSim = 1; % 0/1

% RW Torque noise (it is an additional term that enters directly the
% command of the RW)
Disturbance.RW.TorqueNoise.SelectSim = 1; % 0/1

% Orbital disturbance
Disturbance.External.SelectSim = 1; % 0/1

% SADM (sinusoidal + broadband disturbance)
Disturbance.SADM.SelectSim = 1; % 0/1
Disturbance.SADM.BroadBand.SelectSim = 1; % 0/1
```

5. The user can select which noises to include in the analysis. A value of 1 means the noise is active, while a value of 0 means it is inactive.

```
%%
% NOISE SELECTION
%
Sensors.SST.SelectSim      = 1;
Sensors.GYROc.SelectSim     = 1;
%Sensors.GYROc.RRW_SelectSim = 0;
Sensors.GYROc.BiasInstability_SelectSim = 1;

Sensors.GYROf.SelectSim     = 0;
%Sensors.GYROf.RRW_SelectSim = 0;
Sensors.GYROf.BiasInstability_SelectSim = 0;

Sensors.FGS.SelectSim       = 1;
Sensors.ACCh1.SelectSim     = 1;
Sensors.ACCh2.SelectSim     = 1;
Sensors.ACCh3.SelectSim     = 1;
Sensors.FSM.SelectSim       = 1;
Sensors.RW_tachometer.SelectSim= 1;
```

6. The user can select which model of RCS to include in the simulation. A value of 1 means the noise is active, while a value of 0 means it is inactive.

```
%%
% ACTUATION THRUSTERS
%
Actuators.RCS.SelectFullModel = 1; %Thruster allocation and PWM
Actuators.RCS.SelectNoPWM = 0; %Only thruster allocation but no PWM
Actuators.RCS.SelectIdeal = 0; %Direct application of the torque (No allocation)
```

6.3.2 Tutorial: Setting up worst-case analysis

We examine here how to provide specific parameters' configuration (such as worst-case configuration) to the non-linear simulator. The main code load the parameters' configuration from a .mat file with name given in the variable `Analysis.WCfileName` (see Sect 6.3.2.1 on how to generate this file).

Once the parameters are substituted into the non-linear simulator, the simulink file `Simscape_Benchmark_SIM1.slx` is run and the data are saved in two different files is saved in the folder `Solutions_TimeDomain`:

- `Benchmark_WC_NAME_yyyy_mm_dd_hh_min_sec.mat` only containing the output file of the simulation (out) and the duration of the simulation (`SimulationTime`);
- `Benchmark_WC_NAME_Data_yyyy_mm_dd_hh_min_sec.mat` containing all the other workspace variables (this is useful to keep track of the data provided in the code `TimeDomain_Initialization.m` to set up the simulation)

where "NAME" can be provided by the user in the `Main_TimeDomain.m`.

6.3.2.1 Worst-case analysis: how to provide the WC configuration

The .mat file (loaded with the variable `Analysis.WCfileName`) is structured specifically to be read by the code and substituted into the variables used in the Simulink model. To generate such file, the user can use the template code:

- `WC_template.m` contained in the folder `\Utilities\Functions\Analysis`

The user can substitute the values to the listed parameters in the code and generates the structure `WC_data`.

The list of parameters that can be substituted corresponds to the uncertainties in the LFT model of the spacecraft summarized in Table C.1.

To ease the use of the code, two example are also provided with the worst-case configuration for the coarse and fine pointing RPE metric found by ISAE-SUPAERO using μ -analysis on the coarse pointing closed loop system.

6.3.3 Tutorial: Setting up Monte Carlo analysis

The parsim function requires the PARALLEL COMPUTING TOOLBOX license to run the simulations in parallel. The user is asked at the start of the code 6.2 to define the characteristics of the parallel pool by providing the parameters listed in Table 6.5. The code automatically read how many cores are available in the hardware where the code (N_c) is running with the line of code:

- `MCsim.numcores = feature('numcores');`

Table 6.5: Parameters

Name	Brief Description
<code>MCsim.num_batches (N_b)</code>	Number of batches
<code>MCsim.freecores (N_{fr})</code>	Number of cores not allocated in the parallel pool (up to the user -Tests suggest that it is better to keep one core unallocated when using a laptop.)
<code>MCsim.num_sub_batches (N_{sb})</code>	Number of sub-batches

The size of a single batch (S_b) is given then by:

$$S_b = (N_c - N_{fr})N_{sb}$$

and the total number of runs S_{tot} by

$$S_{tot} = S_b N_b$$

Only the run in the batch are done in parallel. The size of the batch is limited by the available memory resources of the hardware running the Monte Carlo analysis. This solution is less optimal than running a single batch of size S_{tot} ; however this is unfeasible due to the size of the output file that is saved in the workspace.

Before starting the Monte Carlo simulation, the following message appears in the command window, displaying the parallel pool configuration.

```
Parallel pool using the 'local' profile is shutting down.
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 14).

ans =

ProcessPool with properties:

    Connected: true
```

```
NumWorkers: 14
    Cluster: local
AttachedFiles: {}
AutoAddClientPath: true
IdleTimeout: 30 minutes (30 minutes remaining)
SpmdEnabled: true

[13-Jul-2024 19:49:25] Checking for availability of parallel pool
...
[13-Jul-2024 19:49:25] Starting Simulink on parallel workers...
[13-Jul-2024 19:50:27] Configuring simulation cache folder on
parallel workers...
[13-Jul-2024 19:50:28] Loading model on parallel workers...
[13-Jul-2024 19:50:40] Transferring base workspace variables used
in the model to parallel workers...
[13-Jul-2024 19:50:48] Running simulations...
```

6.3.3.1 Saving Monte Carlo results

Differently from the two other types of simulations, for Monte Carlo runs first a folder with the name `MonteCarlo_yyyy_mm_dd_hh_min_sec.mat` is saved in the folder `Solutions_TimeDomain`.

Each batch is then saved into the folder as follows:

- `MC_i.mat` only containing the output results of the i-th batch (`out_MC`) and its duration (`SimulationTime`);
- `MC_Data_i.mat` containing all the other workspace variables (this is useful to keep track of the data provided in the code `TimeDomain_Initialization.m` to set up the simulation).

IMPORTANT REMARK: Note that if needed, the user can merge the results of two different folders, even if the number of simulations in each batch is different. This can be easily accomplished by renaming the files:

For instance, suppose we have two folders:

- **Folder 1:** contains `MC_1` and `MC_2` (with their respective data files)
- **Folder 2:** contains `MC_1`, `MC_2`, and `MC_3` (with their respective data files)

To merge these folders, rename the files in **Folder 2** as `MC_3`, `MC_4`, and `MC_5`, and then copy them into **Folder 1**.

```
% Save the data in the folder initially created with the name "MCsim.  
Name"  
file = strcat('MC_',num2str(j));  
save([strcat('./Solutions_TimeDomain\Complete_Timeline\',MCsim.Name,'\\')  
,file],"out_MC","SimulationTime",'-v7.3');  
  
file =strcat('MC_Data_',num2str(j));  
save([strcat('./Solutions_TimeDomain\Complete_Timeline\',MCsim.Name,'\\')  
,file],'-regexp','^?(?!out_MC|SimulationTime)$.'');
```

6.4 Tutorial: How to modify the controllers

Figure 6.25 shows what it is inside the block CONTROLLERS shown in the simulator (see Fig. 6.1). All the red colored blocks can be modified by the user to import their own controller and the guidance law to reach the slew steady state. The controllers and the guidance law that the user wants to use have to be added in the initialization code `Baseline_Controller.m` in folder `Utilities/BaselineInit/` as replacement of the baseline controller.

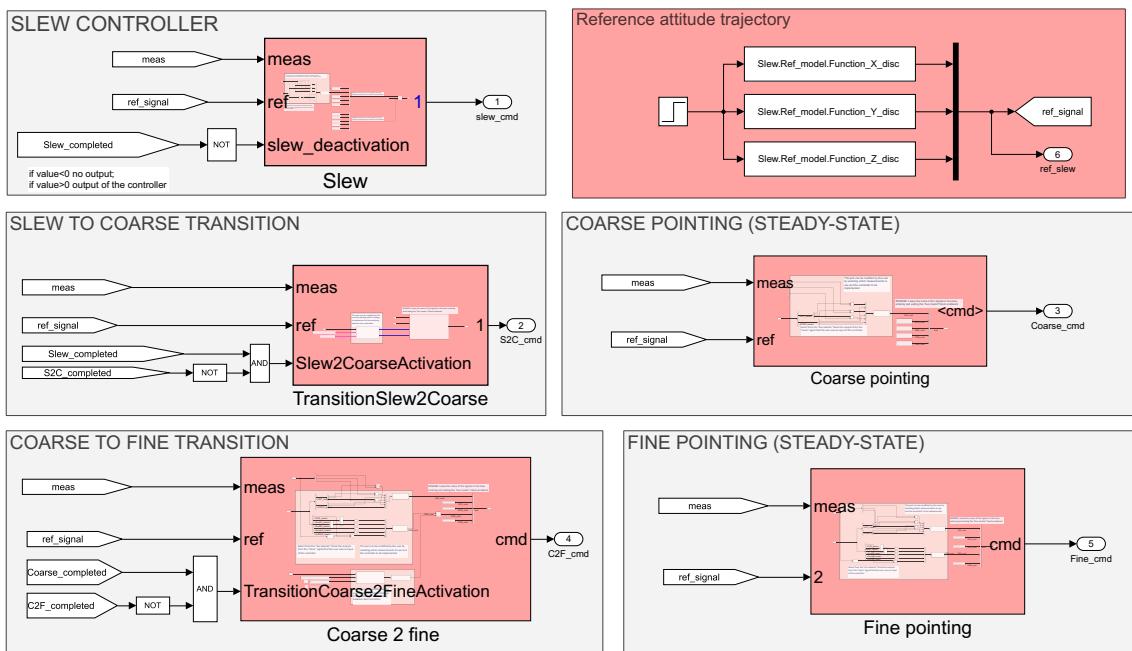


Figure 6.25: Non-linear simulator: CONTROLLERS subsystem

Figure 6.26 reports what it is inside the subsystem SLEW. The input of the model are:

- *ref*: the reference signal;
- *meas*: the measurements which are routed in the model as a `simulink bus`.
- *slew_deactivation*: parameters that activates the switch when the slew is completed and thus outputs a zero command from the subsystem SLEW.

Measurement are selected by means of a `bus selector` (see Fig. 6.27). The user can modify the highlighted section of Fig. 6.26 by providing their discrete-time slew controller. It is important to follow the instruction reported as annotations in the model and not modify the name in the channels. This potentially compromises the consistency of the

bus name between the various sub-blocks.

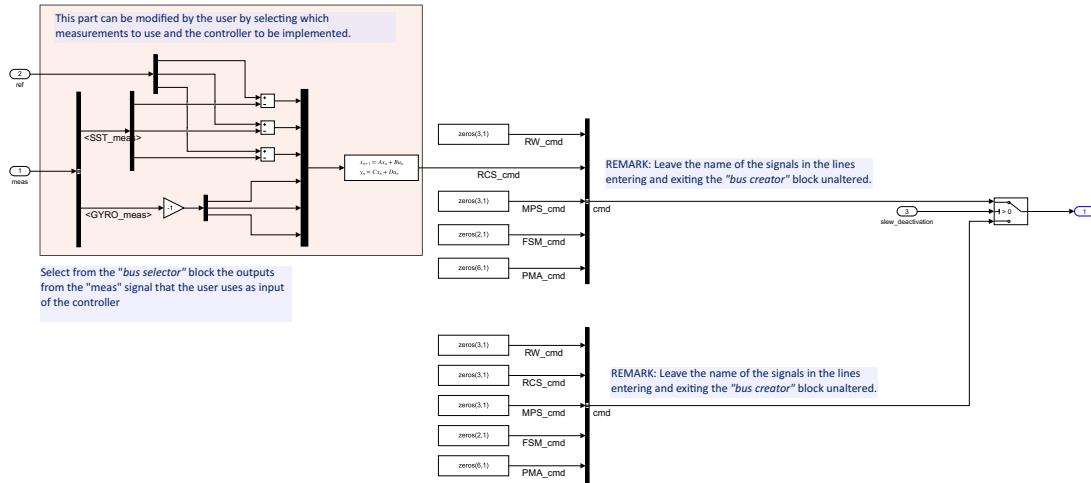


Figure 6.26: Subsystem SLEW

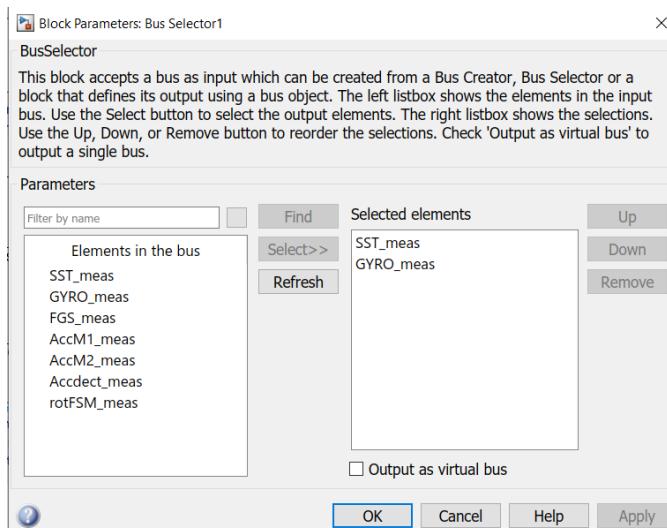


Figure 6.27: Simulink bus for outputting only specific measurements of the *meas* bus

Here we provide an example of the filters used in the initialization script file `Baseline_Controller.m` to generate the reference signal for the attitude. This signal is obtained as the step response of these second-order filters. The user can implement their own guidance law to complete a single-axis maneuver of 15° around the *x*-axis.

```
Slew . RefModel . SteadyStateValue _ X=15*pi /180;
Slew . RefModel . SteadyStateValue _ Y=0*pi /180;
Slew . RefModel . SteadyStateValue _ Z=0*pi /180;
Slew . RefModel . SteadyStateVector =[Slew . RefModel . SteadyStateValue _ X ; Slew .
RefModel . SteadyStateValue _ Y ; ...]
```

```

Slew . RefModel . SteadyStateValue _ Z ];

Slew . RefModel . freq = 0.0005;
Slew . RefModel . overshoot = 0;
Slew . RefModel . zeta = cos ( atan2 ( pi , - log ( Slew . RefModel . overshoot ) ) );

% For the controller
Slew . Ref_model . Function _ X = ( 2 * pi * Slew . RefModel . freq / Slew . RefModel .
    SteadyStateVector ( 1 ) ) ^ 2 / ...
( s ^ 2 + 2 * Slew . RefModel . zeta * ( 2 * pi * Slew . RefModel . freq / Slew . RefModel .
    SteadyStateVector ( 1 ) ) * s + ...
( 2 * pi * Slew . RefModel . freq / Slew . RefModel . SteadyStateVector ( 1 ) ) ^ 2 ) * Slew .
    RefModel . SteadyStateVector ( 1 );

Slew . Ref_model . Function _ Y = ( 2 * pi * Slew . RefModel . freq / Slew . RefModel .
    SteadyStateVector ( 2 ) ) ^ 2 / ...
( s ^ 2 + 2 * Slew . RefModel . zeta * ( 2 * pi * Slew . RefModel . freq / Slew . RefModel .
    SteadyStateVector ( 2 ) ) * s + ...
( 2 * pi * Slew . RefModel . freq / Slew . RefModel . SteadyStateVector ( 2 ) ) ^ 2 ) * Slew .
    RefModel . SteadyStateVector ( 2 );

Slew . Ref_model . Function _ Z = ( 2 * pi * Slew . RefModel . freq / Slew . RefModel .
    SteadyStateVector ( 3 ) ) ^ 2 / ...
( s ^ 2 + 2 * Slew . RefModel . zeta * ( 2 * pi * Slew . RefModel . freq / Slew . RefModel .
    SteadyStateVector ( 3 ) ) * s + ...
( 2 * pi * Slew . RefModel . freq / Slew . RefModel . SteadyStateVector ( 3 ) ) ^ 2 ) * Slew .
    RefModel . SteadyStateVector ( 3 );

```

For the transition phases (see Fig. 6.28), where the switching between two controllers occurs, the user can also import their own switching law/methodology. In the baseline architecture (for the slew) provided by ISAE-SUPAERO this is done by activating the coarse controller while de-activating at the same time the slew controller as follows:

$$\begin{cases} \bar{\mathbf{K}}_{slew} = \mathbf{K}_{slew}(1 - \alpha(t)) \\ \bar{\mathbf{K}}_{coarse} = \mathbf{K}_{coarse}\alpha(t) \end{cases}$$

with $\alpha(t) = k \cdot t$ ($k = 0.25$)

The transition between coarse to fine pointing controller is addressed in a similar way in the baseline architecture. In this case, as previously discussed, coarse and fine pointing share the same ACS, thus only the controller at payload level has to be activated and no de-activation is necessary.

Finally the user can modify the code to implement their own coarse and fine controllers in the script `Baseline_Controller.m`.

Finally when the code `Main_TimeDomain.m` runs, the following messages appears on the command window reminding the user that there are 4 baseline controllers/filters: the slew, coarse and fine controllers and the friction compensation filter.

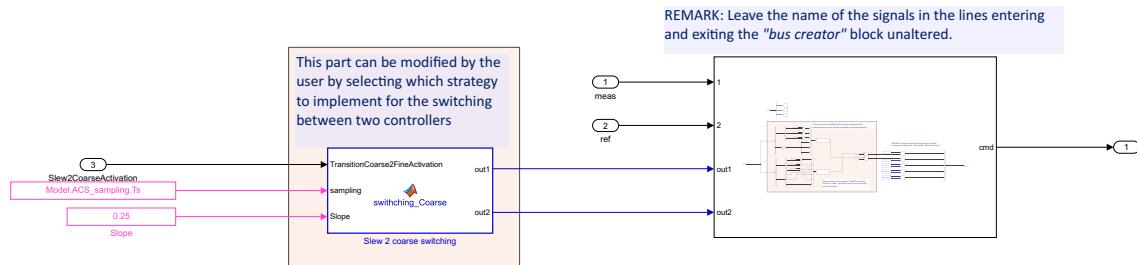


Figure 6.28: Subsystem TRANSITIONSLEW2COARSE

```

-----  

Loading baseline slew controller and filter for reference signal  

for attitude  

Comment this section if the user wants to provide a difference  

guidance and slew controller  

-----  

-----  

Loading baseline coarse controller  

Comment this section if the user wants to provide a different  

coarse controller  

-----  

-----  

Loading baseline fine controller  

Comment this section if the user wants to provide a different fine  

controller  

-----  

-----  

Loading baseline friction compensation scheme  

Comment this section if the user wants to test another algorithm  

-----  


```

6.5 Tutorial: How to post-process simulation data

This section discusses how to run the code for plotting the results of the time-domain simulations. The script `Main_PlotResults.m` addresses all three types of simulation: Nominal model, worst-case or generic parameter configuration, and Monte Carlo analysis.

According to the provided inputs the codes manages how the results are visualized. The user can provide as input the name of the .mat containing the results (up to 1 models

(with a single model config.) and 1 Monte Carlo run.

If an input is given as empty string " " then no results are visualized for that particular input. In this way the following case can be studied and compared:

1. Analysis of a single model (called *Model A*). The model A could be a simulation with nominal parameters or a generic configuration. In this case the other entries need to be left as empty string
 - InputFile_Model_B= ' ';
 - InputFile_Montecarlo = ' ';
2. Analysis of two models (*Model A + Model B*): This can be useful to compare a simulation with the nominal configuration with respect to the results of a run with worst-case parameters. In this case
 - InputFile_Montecarlo = ' ';
3. Analysis of a generic model compared to Monte Carlo run (*Model A + MonteCarlo*): In this case
 - InputFile_Model_B= ' ';

IMPORTANT REMARK: Providing only *Model B* without a *Model A* gives an error. If you have only a single result with a particular configuration use the input for *Model A*.

The analysis of a single model provides as output also a structure called **Metrics** that contains the following data:

1. **Phases**
 - **Time**: time instants when the transition between the each phase occurs;
 - **Label_phases**: label of the phase;
 - **DurationPhases**: duration of each phase.
2. **APE_slew**: mean and standard deviation of APE during slew maneuver
3. **APE_Coarse**: mean and standard deviation of APE during coarse pointing
4. **APE_Fine**: mean and standard deviation of APE during fine pointing
5. **RPE_Coarse**: mean and standard deviation of RPE during coarse pointing
6. **RPE_Fine**: mean and standard deviation of RPE during fine pointing
7. **PDE_Fine**: mean and standard deviation of PDE during fine pointing

Furthermore, if the user has the MATLAB package REPORT GENERATOR installed, the code saves a .pdf summary report in the folder "Reports" with the name:

- `Report_ModelA_yyyy_mm_dd_hh_min_sec.pdf` containing the statistics of the pointing metrics.

Similarly, a Monte Carlo analysis provides as output also a structure called `Metrics_MC` that contains the following data:

1. Phases

- **Time**: time instants when the transition between the each phase occurs for each MC run;
- **DurationPhases**: duration of each phase for each MC run. Each cell of `DurationPhases` contains a vector for numeric values (corresponding to the duration of the slew phase, slew to coarse transition, coarse phase, coarse to fine transition and finally fine phase). These data are stored separately into:
 - `Metrics_MC.Phases.DurationSlew`
 - `Metrics_MC.Phases.DurationSlew2Coarse`
 - `Metrics_MC.Phases.DurationCoarse`
 - `Metrics_MC.Phases.DurationCoarse2Fine`
 - `Metrics_MC.Phases.DurationFine`

2. **APE_slew** (mean and standard deviation) for each run.
3. **APE_Coarse** (mean and standard deviation) for each run.
4. **APE_Fine** (mean and standard deviation) for each run.
5. **RPE_Coarse** (mean and standard deviation) for each run.
6. **RPE_Fine** (mean and standard deviation) for each run.
7. **PDE_Fine** (mean and standard deviation) for each run.

If the user has the MATLAB package REPORT GENERATOR installed, the code saves a summary report in the folder "Reports" with the name:

- `Report_Model_MC_yyyy_mm_dd_hh_min_sec.pdf` containing the distribution of the means and standard deviations for each pointing metrics and also the distribution of the durations of the mission phases.

6.6 Computational aspects of the simulator

The computational cost of the simulation is briefly discussed here. The reported metric represents the ratio of the simulation time to the real mission duration (which is 2700 seconds for the full mission).

- For full mission this factor is around 2.8 (obtained as average of the simulation time obtained for the nominal and worst-case runs).

For the Monte Carlo analysis, as an example, four batches were run with difference size each (i.e. number of runs contained in a batch). Table 6.6 reports the simulation time for each batch and the average factor per single simulation in a single batch ($\frac{\text{SIMULATION TIME}}{n_{\text{batch}} * T_{\text{mission}}}$).

Batch	Number of runs	Simulation time	Factor
MC ₁	13	33940 s (9.4 h)	0.97
MC ₂	13	20705 s (5.75 h)	0.59
MC ₃	36	32284 s (8.96 h)	0.332
MC ₄	36	32710 s (9.1 h)	0.337
Total	98	-	-

Table 6.6: Batches of Monte Carlo and simulation time

The values reported here do not provide an accurate analysis of the computational costs of the simulations but just an estimation because they were:

1. run on two different hardware setups (first one: 12th Gen Intel(R) Core(TM) i7-12800H at 2.40 GHz with 32 GB at 4800 MHz and the second one: Intel(R) Xeon(R) W-2295 CPU at 3 GHz with 512 GB at 2934 MHz);
2. the hardware was occupied with other tasks while the simulations were running.

Appendix A

Overview of SDTlib and TITOP formulation

This section aims to briefly report the theoretical background used throughout this chapter and implemented in the Satellite Dynamics Toolbox library (SDTlib) [9, 37], a MATLAB/SIMULINK library used to derive LFT-LPV models of multi-body space systems.

The toolbox allows to model complex space systems by connecting together elementary dynamical blocks in a sub-structured way through their input/output ports. Indeed, several kind of substructures and mechanical systems are currently included in the library: Analytical models for elementary bodies [38] (beams, plates, multi-port rigid bodies, etc.), FE models of more complex structures [39] retrieved from PATRAN/NASTRAN models and analytical models of mechanisms (RWs [40], SADM [20], PMA [41], etc.)

The foundational theory of the library is based on the substructure modelling approach called Two-Input Two-Output Ports (TITOP) [42, 43]. The key peculiarity of the TITOP framework and of the SDTlib toolbox consists in providing a dynamical model of a dynamical system in the form of a LFT/LPV plant by capturing both the uncertain nature of the system and possible varying parameters (i.e. rotation of the solar arrays with respect to the hub, RWs speed, etc.). Therefore, the output of such modelling framework can be used for robust control synthesis and to assess the robustness of the design using robust analysis tools such as the structured singular value μ .

Let us consider the multi-body system in Fig. A.1-(a), which is composed of a flexible structure \mathcal{L}_i connected to a parent substructure \mathcal{L}_{i-1} at point P and to a child substructure \mathcal{L}_{i+1} at point C . The TITOP model $\mathcal{M}_{PC}^{\mathcal{L}_i}(s)$, whose block diagram representation is given in Fig. A.1-(b), is a linear state-space model with 12 inputs (shown in blue in Fig. A.1):

- the 6 components defined in $\mathcal{R}_{\mathcal{L}_i}$ of the acceleration vector $\ddot{\mathbf{x}}_P$ composed of the linear acceleration vector $\ddot{\mathbf{a}}_P$ and the angular acceleration $\dot{\omega}_P$ at the clamped node P ;
- the 6 components defined in $\mathcal{R}_{\mathcal{L}_i}$ of the wrench $\mathbf{W}_{\mathcal{L}_{i+1}/\mathcal{L}_i,C}$ defined in $\mathcal{R}_{\mathcal{L}_i}$ composed of the force vector $\mathbf{F}_{\mathcal{L}_{i+1}/\mathcal{L}_i,C}$ and the torque vector $\mathbf{T}_{\mathcal{L}_{i+1}/\mathcal{L}_i,C}$ applied by an adjacent body \mathcal{L}_{i+1} to the \mathcal{L}_i at the free node C ;

and 12 outputs (shown in red in Fig. A.1):

- the 6 components defined in $\mathcal{R}_{\mathcal{L}_i}$ of the acceleration vector $\ddot{\mathbf{x}}_C$ written in $\mathcal{R}_{\mathcal{L}_i}$ composed of the linear acceleration vector $\ddot{\mathbf{a}}_C$ and the angular acceleration $\dot{\omega}_C$ at the free node C ;
- The 6 components defined in $\mathcal{R}_{\mathcal{L}_i}$ of the wrench $\mathbf{W}_{\mathcal{L}_i/\mathcal{L}_{i-1},P}$ composed of the force vector $\mathbf{F}_{\mathcal{L}_i/\mathcal{L}_{i-1},P}$ and the torque vector $\mathbf{T}_{\mathcal{L}_i/\mathcal{L}_{i-1},P}$ applied by \mathcal{L}_i to the adjacent body \mathcal{L}_{i-1} at the clamped node P

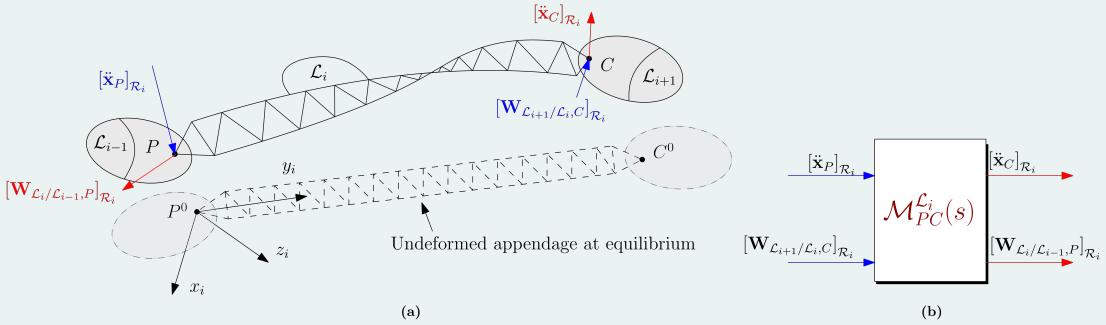


Figure A.1: (a) i -th flexible appendage of a complex sub-structured body; (b) Compact representation of plant $\mathcal{M}_{PC}^{L_i}$ (image credit: [44]).

It is important to point out that the approach is independent of the boundary conditions applied at the connecting points of the substructure \mathcal{L}_i . Indeed, the various input–output channels are invertible and different boundary condition for the body can be reproduced. For instance, inverting the second channel of $\mathcal{M}_{PC}^{L_i}$ provides the linear model of the substructure \mathcal{L}_i clamped on both point P and C (i.e. the acceleration of body \mathcal{L}_i at point P and C is prescribed by the adjacent bodies).

A.1 Rotation matrix of angle θ around z -axis

The Direct Cosine Matrix (DCM) $\mathbf{P}_{\mathcal{A}/\mathcal{B}}$ associated to the rotation of a given angle θ around the z -axis and transforming the frame $\mathcal{R}_b = (O_b, x_b, y_b, z_b)$ into the frame $\mathcal{R}_a = (O_a, x_a, y_a, z_a)$ is given by:

$$\mathbf{P}_{\mathcal{A}/\mathcal{B}}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In order to obtain a *minimal* LFT representation of the rotation when θ is assumed to be a varying or uncertain parameter particular parametrization proposed in [45] of the two trigonometric functions is used. The parametrization is function of the tangent of the quarter-angle $\tau = \tan(\theta/4)$:

$$\cos \theta = \frac{(1 + \tau^2)^2 - 8\tau^2}{(1 + \tau^2)^2}, \quad \sin \theta = \frac{4\tau(1 - \tau^2)}{(1 + \tau^2)^2}, \quad \text{for all } \tau \in]-1; 1]$$

Thanks to this parametrization, the whole rotation matrix is represented with an uncertain block of order 4 only. Thus, each solar array gives 16 repetitions of the parameter τ due to the fact that the matrix $\mathbf{P}_{\mathcal{A}/\mathcal{B}}^{\times 2} = \begin{bmatrix} \mathbf{P}_{\mathcal{A}/\mathcal{B}} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{P}_{\mathcal{A}/\mathcal{B}} \end{bmatrix}$ (and its transpose) are used to transform the acceleration vector (and wrench vector) from the parent to the child body frame.

Appendix B

Overview of pointing metrics in time and frequency domain

The pointing performance of a spacecraft are specified by a set of requirements consisting of absolute (or instantaneous), window- and stability-time errors [AD-03]. It can be distinguished between three time-dependencies of pointing errors as shown in Fig. B.1:

- Instantaneous time t : pointing error at any point in time t during system lifetime or in a defined observation period. Absolute Pointing Error (APE) and Absolute Knowledge Error (AKE) are the associated instantaneous error metrics.
- window time Δt : pointing error within a time window Δt , where the time window can be evaluated at any point in time t ;
- Stability time Δt_s : pointing error metrics assessing stability among pointing errors in time-windows of length Δt separated by a time difference of length Δt_s .

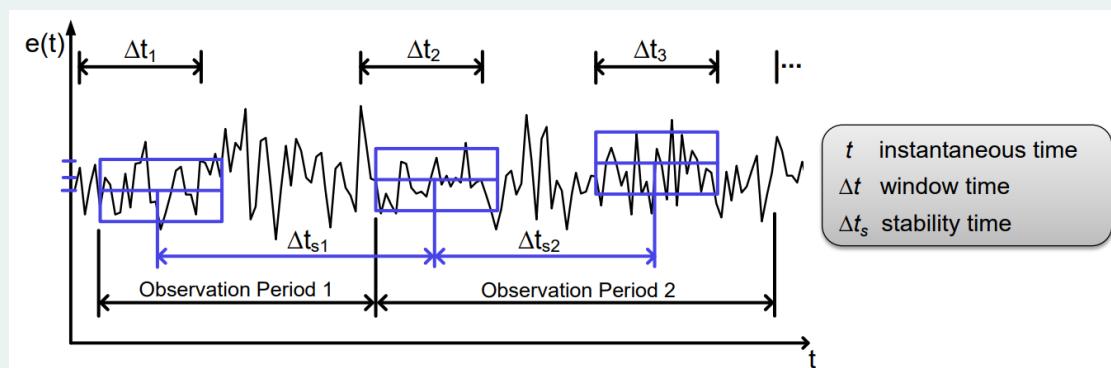


Figure B.1: Time dependency of pointing errors - Image credit: [AD-03]

The definition of the pointing performance indices is outlined in Table B.1. For more details on this topic (and on pointing knowledge indices, which are not mentioned in this section) the reader is referred to the ESA pointing error engineering handbook [AD-03] and to ref. [46].

Name	Definition
Absolute performance error (APE)	$e(t)$
Mean performance error (MPE)	$\bar{e}(t, \Delta t) = \langle e(t) \rangle_{\Delta t} = \frac{1}{\Delta t} \int_{t-\Delta t/2}^{t+\Delta t/2} e(t) dt$
Relative performance error (RPE)	$e(t) - \bar{e}(t, \Delta t)$
Performance drift error (PDE)	$\bar{e}(t, \Delta t) - \bar{e}(t + \Delta t_s, \Delta t)$
Performance reproducibility error (PRE)	
[*] PDE and PRE share the same definition, however the PDE metric is evaluated between two intervals in the same observation period while the PRE is defined within different observation periods	

Table B.1: Time-domain representation of pointing performance error indices

Frequency-domain expressions for the pointing metrics can be derived as in [47] and are particularly useful to attitude control design and evaluate pointing error index performance for closed-loop specification. The metrics are restricted to signals that are stationary random noise processes with zero mean value [46]. Table B.2 reports the expression of rational weighting function for the metrics of interest for the current benchmark problem.

Error Index	Rational Weighting filters
APE	$\mathcal{F}_{APE}(s) = 1$
MPE	$\mathcal{F}_{MPE}(s, \Delta t) = \frac{2(s\Delta t+6)}{(s\Delta t)^2+6(s\Delta t)+12}$
RPE	$\mathcal{F}_{RPE}(s, \Delta t) = \frac{\Delta t s (\Delta t s + \sqrt{12})}{(\Delta t s)^2 + 6(\Delta t s) + 12}$
PDE/PRE	$\mathcal{F}_{PDE}(s, \Delta t, \Delta t_s) = \mathcal{F}_{MPE}(s, \Delta t) \frac{2s\Delta t_s (s\Delta t_s + 6)}{(s\Delta t_s)^2 + 6(s\Delta t_s) + 12}$

Table B.2: Pointing error weighing filters

These filters are then each scaled by a constant factor (ϵ_{metric}) that overbounds the maximum target value for the specific pointing metric (see for example eq. B.1). Thus, in the next section, we will use the notation $\mathbf{W}_{\text{metric}}(s)$ to identify the filter used to impose the pointing specification in the closed-loop control design.

$$\mathbf{W}_{RPE}(s, \Delta t) = \epsilon_{RPE}^{-1} \mathcal{F}_{RPE}(s, \Delta t) \quad (B.1)$$

Appendix C

Parametric Uncertainties

The complete set of uncertainties considered in this study is listed in Table C.1.

System	Parameter	Description	Nominal Value	Uncertainty
Central Bus \mathcal{B}_\bullet	$m^{\mathcal{B}}$	Mass	1000 kg	$\pm 20\%$
	$\begin{bmatrix} J_{xx}^{\mathcal{B}} & J_{xy}^{\mathcal{B}} & J_{xz}^{\mathcal{B}} \\ J_{yx}^{\mathcal{B}} & J_{yy}^{\mathcal{B}} & J_{yz}^{\mathcal{B}} \\ sym & J_{zx}^{\mathcal{B}} \end{bmatrix}$	Inertia in \mathcal{R}_G frame ^[1]	$\begin{bmatrix} 75 & 1 & 2 \\ 80 & -1 & \\ sym & 40 \end{bmatrix} \text{kg m}^2$	$\begin{bmatrix} \pm 20\% & \pm 1 \text{kg m}^2 & \pm 2 \text{kg m}^2 \\ \pm 20\% & \pm 1 \text{kg m}^2 & \\ sym & & \pm 20\% \end{bmatrix}$
	$[\mathbf{x}_G^{\mathcal{B}}]_{\mathcal{R}_b}$	Center of Mass in \mathcal{R}_b frame	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \text{m}$	$\pm [0.01, 0.01, 0.02] \text{m}$
Solar Array \mathcal{A}_\bullet ^[2]	$\omega_1^{A_\bullet}$	1st Flexible mode frequency	10.2 rad/s	$\pm 10\%$
	$\omega_2^{A_\bullet}$	2nd Flexible mode frequency	17.5 rad/s	$\pm 10\%$
	$\omega_3^{A_\bullet}$	3rd Flexible mode frequency	39.6 rad/s	$\pm 10\%$
	$\omega_4^{A_\bullet}$	4th Flexible mode frequency	48.2 rad/s	$\pm 10\%$
	$\omega_5^{A_\bullet}$	5th Flexible mode frequency	91.8 rad/s	$\pm 10\%$
	$\omega_6^{A_\bullet}$	6th Flexible mode frequency	111 rad/s	$\pm 10\%$
	$\tau^{A_\bullet} = \tan(\theta_\bullet/4)$	SADM rotor angle (adimensional)	0	$[-1 : 1]$
	$\omega_1^{\mathcal{H}}$	1st Flexible mode frequency	2.31 rad/s	$\pm 10\%$
Antenna \mathcal{H}	$\omega_2^{\mathcal{H}}$	2nd Flexible mode frequency	2.31 rad/s	$\pm 10\%$
	$\omega_3^{\mathcal{H}}$	3rd Flexible mode frequency	14.5 rad/s	$\pm 10\%$
	$\omega_4^{\mathcal{H}}$	4th Flexible mode frequency	14.5 rad/s	$\pm 10\%$
	$\omega_5^{\mathcal{H}}$	5th Flexible mode frequency	40.5 rad/s	$\pm 10\%$
	$\omega_6^{\mathcal{H}}$	6th Flexible mode frequency	40.5 rad/s	$\pm 10\%$
Sloshing \mathcal{S}	$m^{\mathcal{S}_\bullet}$	Sloshing mass	15 kg	$\pm 15 \text{ kg}$
Payload \mathcal{P}	$\omega_1^{\mathcal{P}}$	1st Flexible mode frequency	490.52 rad/s	$\pm 5\%$
	$\omega_2^{\mathcal{P}}$	2nd Flexible mode frequency	490.53 rad/s	$\pm 5\%$
	$\omega_3^{\mathcal{P}}$	3rd Flexible mode frequency	1562.87 rad/s	$\pm 5\%$
	$\omega_4^{\mathcal{P}}$	4th Flexible mode frequency	1842.60 rad/s	$\pm 5\%$
	$\omega_5^{\mathcal{P}}$	5th Flexible mode frequency	2315.48 rad/s	$\pm 5\%$
Payload Isolator $\mathcal{I}\mathcal{A}$ and payload equipment ^[3]	$m^{\mathcal{I}\mathcal{A}}$	Mass	120 kg	$\pm 10\%$
	$\begin{bmatrix} J_{xx}^{\mathcal{I}\mathcal{A}} & J_{xy}^{\mathcal{I}\mathcal{A}} & J_{xz}^{\mathcal{I}\mathcal{A}} \\ J_{yx}^{\mathcal{I}\mathcal{A}} & J_{yy}^{\mathcal{I}\mathcal{A}} & J_{yz}^{\mathcal{I}\mathcal{A}} \\ sym & J_{zx}^{\mathcal{I}\mathcal{A}} \end{bmatrix}$	Inertia in \mathcal{R}_{ia} frame	$\begin{bmatrix} 20 & -2 & -1 \\ 30 & 2 & \\ sym & 40 \end{bmatrix} \text{kg m}^2$	$\begin{bmatrix} \pm 10\% & \pm 1 \text{kg m}^2 & \pm 1 \text{kg m}^2 \\ \pm 10\% & \pm 1 \text{kg m}^2 & \\ sym & & \pm 10\% \end{bmatrix}$
	$\Omega^{\mathcal{R}\mathcal{W}\bullet}$	Speed range	0 rpm	$\pm 4000 \text{ rpm}$
Reaction Wheel $\mathcal{R}\mathcal{W}_\bullet$	$J_x^{\mathcal{R}\mathcal{W}\bullet}$	RW Isolator axial stiffness	160000 N/m	$[136000 : 184000] \text{ N/m}$
	$J_r^{\mathcal{R}\mathcal{W}\bullet}$	RW Isolator radial stiffness	160000 N/m	$[136000 : 184000] \text{ N/m}$

[1] Reference frame \mathcal{R}_b has origin at the launcher interface (center of the bottom panel of the central body) with z-axis along LoS and the other two axis to form an orthogonal reference system. Reference frame \mathcal{R}_G has the same axis directions of \mathcal{R}_b but origin in the nominal CoM of the central body.

[2] The two solar arrays share the same nominal values and same uncertain range ($\pm 10\%$) for the natural frequencies, however distinct uncertain parameters for the two solar arrays are used to generate the LFT model (for instance, in the benchmark main code the first natural frequency of the 2 SA is defined with two uncertain parameters $w_{_SolarArray1_1}$ (for solar array 1) and $w_{_SolarArray1_2}$ (for solar array 2).)

[3] This element includes the mass and inertia of the payload isolator and other payload systems (instruments, harness, etc) not directly accounted in the model. It does not include the telescope structure, the fast steering mirror and the PMA system.

Table C.1: Spacecraft Uncertainties

European Satellite Benchmark

Copyright (c) ISAE-SUPAERO, All Rights Reserved

ESA 4000141060/23/NL/MGu

[ESA Public Licence - v2.4 - Permissive \(Type 3\)](#)

User Manual

Bibliography

- [1] C. Dennehy et al., “Verification and validation (v&v) of guidance & control systems: Results from the first inter-agency workshop on gnc v&v,” *Proceedings of the 46th Rocky Mountain AAS GN&C Conference*, 2022 (cit. on pp. 1, 5).
- [2] O. A.-S. C. Dennehy, “Spacecraft micro-vibration: A survey of problems, experiences, potential solutions, and some lessons learned,” Technical Report NASA/TM-2018-220075, NASA, Tech. Rep., 2018 (cit. on p. 1).
- [3] J. Young, H. Frisch, G. Jones, and W. Walker, “Interactive analysis program activity,” NASA, Scientific and Technical Information Office., Large Space Systems Technology 1st annual technical review, 1979, p. 383 (cit. on p. 1).
- [4] R. Grogan and R. Laskin, “On multidisciplinary modeling of the space interferometry mission,” in *Proceedings of the 1998 ACC (IEEE Cat. No.98CH36207)*, vol. 3, 1998, 1558–1562 vol.3 (cit. on p. 1).
- [5] M. Levine, G. Mosier, G. Walsh, C. Blaurock, and G. Hartig, “Integrated modeling of the James Webb Space Telescope: pre-launch predictions, flight performance, and lessons learned,” in *Optical Modeling and Performance Predictions XIII*, M. A. Kahan, Ed., vol. 12664, SPIE, 2023, p. 1 266 409 (cit. on p. 1).
- [6] E. Perez, M. Bailly, and J. M. Pariot, “Pointing Acquisition And Tracking System For Silex Inter Satellite Optical Link,” in *Acquisition, Tracking, and Pointing III*, S. Gowrinathan, Ed., vol. 1111, SPIE, 1989, pp. 277 –288 (cit. on p. 1).
- [7] G. D. Racca et al., “The Euclid mission design,” in *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave*, vol. 9904, SPIE, 2016, 99040O (cit. on p. 1).
- [8] F. Sanfedino, D. Alazard, A. Kiley, M. Watt, P. Simplicio, and F. Ankersen, “Robust monolithic versus distributed control/structure co-optimization of flexible space systems in presence of parametric uncertainties,” *J. SMO*, vol. 66, no. 12, p. 247, 2023 (cit. on pp. 2, 5).
- [9] D. Alazard and F. Sanfedino, “Satellite dynamics toolbox for preliminary design phase,” in *43rd Annual AAS Guidance and Control Conference, 30 January 2020 - 5 February 2020 (Breckenridge, United States)*, 2020 (cit. on pp. 2, 101).

-
- [10] F. Sanfedino, D. Alazard, E. Kassarian, and F. Somers, “Satellite dynamics toolbox library: A tool to model multi-body space systems for robust control synthesis and analysis,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 9153–9160, 2023, 22nd IFAC World Congress (cit. on p. 2).
 - [11] K. Zhou, J. Doyle, and K. Glover, *Robust and optimal control*. Prentice hall Upper Saddle River, NJ, 1995 (cit. on p. 2).
 - [12] D. Alazard and F. Sanfedino, “Satellite Dynamics Toolbox library (SDTlib) - User’s Guide,” Institut Supérieur de l’Aéronautique et de l’Espace, Tech. Rep., 2021 (cit. on pp. 3, 24).
 - [13] E. Kassarian, F. Sanfedino, D. Alazard, and H. Evain, “Parametric sensitivity analysis for the robust control of uncertain space systems,” in *CEAS EuroGNC 2024, Bristol, UK*, 2024 (cit. on p. 5).
 - [14] C. Roos, “Systems modeling, analysis and control (smac) toolbox: An insight into the robustness analysis library,” in *2013 IEEE CACSD Conf.*, IEEE, 2013, pp. 176–181 (cit. on p. 5).
 - [15] C. Roos, J.-M. Biannic, and H. Evain, “A new step towards the integration of probabilistic μ in the aerospace v&v process,” *CEAS Space J.*, vol. 16, no. 1, pp. 59–71, 2024 (cit. on p. 5).
 - [16] P. Simplício, S. Bennani, A. Marcos, C. Roux, and X. Lefort, “Structured singular-value analysis of the vega launcher in atmospheric flight,” *JGCD*, vol. 39, no. 6, pp. 1342–1355, 2016 (cit. on p. 5).
 - [17] J. Veenman, C. W. Scherer, C. Ardura, S. Bennani, V. Preda, and B. Girouart, “Iqclab: A new iqclab based toolbox for robustness analysis and control design,” *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 69–74, 2021 (cit. on p. 5).
 - [18] H. Pfifer and P. Seiler, “Integral quadratic constraints for delayed nonlinear and parameter-varying systems,” *Automatica*, vol. 56, pp. 36–43, 2015 (cit. on p. 5).
 - [19] A. Marcos, H. Garcia De Marina, V. Mantini, C. Roux, and S. Bennani, “Optimization-based worst-case analysis of a launcher during the atmospheric ascent phase,” in *AIAA GNC Conf.*, 2013, p. 4645 (cit. on p. 5).
 - [20] F. Sanfedino, D. Alazard, V. Preda, and D. Oddenino, “Integrated modeling of microvibrations induced by solar array drive mechanism for worst-case end-to-end analysis and robust disturbance estimation,” *Mechanical Systems and Signal Processing*, vol. 163, p. 108168, 2022 (cit. on pp. 5, 57, 101).
 - [21] J.-M. Biannic, C. Pittet, L. Lafourcade, and C. Roos, “Lpv analysis of switched controllers in satellite attitude control systems,” in *AIAA GNC Conf.*, 2010 (cit. on p. 5).

- [22] F. Sanfedino, G. Thiébaud, D. Alazard, N. Guercio, and N. Deslaef, “Advances in fine line-of-sight control for large space flexible structures,” *J. AST*, vol. 130, p. 107961, 2022 (cit. on p. 5).
- [23] J.-M. Biannic, C. Roos, and C. Cumér, *Lft modelling and μ -based robust performance analysis of hybrid multi-rate control systems*, 2024. arXiv: [2407.04463 \[eess.SY\]](https://arxiv.org/abs/2407.04463) (cit. on p. 5).
- [24] C. Dennehy et al., “Verification and validation (v&v) of guidance & control systems: Results from the first inter-agency workshop on gnc v&v,” *Proceedings of the 46th Rocky Mountain AAS GN&C Conference*, 2022 (cit. on p. 9).
- [25] F. Sanfedino, G. Thiébaud, D. Alazard, N. Guercio, and N. Deslaef, “Advances in fine line-of-sight control for large space flexible structures,” *Aerospace Science and Technology*, vol. 130, p. 107961, 2022 (cit. on pp. 14, 34).
- [26] P. Valentin, J. Cieslak, D. Henry, S. Bennani, and A. Falcoz, “A \mathcal{H}_∞/μ solution for microvibration mitigation in satellites: A case study,” *Journal of Sound and Vibration*, vol. 399, Apr. 2017 (cit. on p. 14).
- [27] M. N. Sweeney, G. A. Rynkowski, M. Ketabchi, and R. Crowley, “Design considerations for fast-steering mirrors (FSMs),” in *Optical Scanning 2002*, S. F. Sagan, G. F. Marshall, and L. Beiser, Eds., International Society for Optics and Photonics, vol. 4773, SPIE, 2002, pp. 63–73 (cit. on p. 16).
- [28] V. Preda, J. Cieslak, D. Henry, S. Bennani, and A. Falcoz, “A $\mathbf{h}\infty/\mu$ solution for microvibration mitigation in satellites: A case study,” *Journal of Sound and Vibration*, vol. 399, pp. 21–44, 2017 (cit. on p. 16).
- [29] R. Rodrigues, F. Sanfedino, P. Valentin, and J. Olucha, “Linear parameter-varying gain-scheduled attitude controller for an on-orbit servicing mission involving flexible large spacecraft and fuel sloshing,” in *ESA GNC and ICATT Conference*, Sopot, Poland, 2023 (cit. on p. 29).
- [30] J. Marie, F. Cordero, D. Milligan, E. Ecale, and P. Tatry, “In-orbit experience of the gaia and lisa pathfinder cold gas micro-propulsion systems,” in *Space Operations: Inspiring Humankind’s Future*, H. Pasquier, C. A. Cruzen, M. Schmidhuber, and Y. H. Lee, Eds. Cham: Springer International Publishing, 2019, pp. 551–574 (cit. on p. 31).
- [31] D. C. Redding and W. G. Breckenridge, “Optical modeling for dynamics and control analysis,” *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 5, pp. 1021–1032, 1991. eprint: <https://doi.org/10.2514/3.20745> (cit. on p. 38).
- [32] R. A. Masterson, “Development and validation of empirical and analytical reaction wheel disturbance models,” Ph.D. dissertation, Massachusetts Institute of Technology, 1999 (cit. on p. 61).

-
- [33] H. Olsson, K. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky, “Friction models and friction compensation,” *European Journal of Control*, vol. 4, no. 3, pp. 176–195, 1998 (cit. on p. 64).
 - [34] “Mitigation of wheel friction torque instabilities - final report,” Rockwell Collins - ESA TRP Program, Tech. Rep., 2015 (cit. on pp. 64, 65, 67).
 - [35] T. Ieko, Y. Ochi, and K. Kanai, “New design method for pulse-width modulation control systems via digital redesign,” *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 1, pp. 123–128, 1999. eprint: <https://doi.org/10.2514/2.4358> (cit. on p. 71).
 - [36] “Ieee standard specification format guide and test procedure for single-axis interferometric fiber optic gyros,” *IEEE Std 952-1997*, pp. 1–84, 1998 (cit. on p. 78).
 - [37] D. Alazard and F. Sanfedino. “Satellite dynamics toolbox,” Accessed: Dec. 13, 2022. [Online]. Available: <https://personnel.isae-supaero.fr/daniel-alazard/matlab-packages/satellite-dynamics-toolbox.html> (cit. on p. 101).
 - [38] J. Chebbi, V. Dubanchet, J. A. P. Gonzalez, and D. Alazard, “Linear dynamics of flexible multibody systems : A system-based approach,” *Multibody System Dynamics*, vol. 41, no. 1, pp. 75–100, 2017 (cit. on p. 101).
 - [39] F. Sanfedino, D. Alazard, V. Pommier-Budinger, A. Falcoz, and F. Boquet, “Finite element based n-port model for preliminary design of multibody systems,” *Journal of Sound and Vibration*, vol. 415, pp. 128–146, 2018 (cit. on p. 101).
 - [40] F. Sanfedino, D. Alazard, V. Pommier-Budinger, F. Boquet, and A. Falcoz, “A Novel Dynamic Model of a Reaction Wheel Assembly for High Accuracy Pointing Space Missions,” ser. Dynamic Systems and Control Conference, V003T40A002, vol. Vol.3, Sep. 2018 (cit. on p. 101).
 - [41] V. Preda, F. Sanfedino, S. Bennani, F. Boquet, and D. Alazard, “Robust and adaptable dynamic response reshaping of flexible structures,” *Journal of Sound and Vibration*, vol. 468, p. 115 086, 2020 (cit. on p. 101).
 - [42] D. Alazard, J. A. Perez, C. Cumér, and T. Loquen, “Two-input two-output port model for mechanical systems,” in *AIAA Guidance, Navigation, and Control Conference*. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2015-1778> (cit. on p. 101).
 - [43] J. A. Perez, D. Alazard, T. Loquen, C. Pittet, and C. Cumér, “Flexible Multi-body System Linear Modeling for Control Using Component Modes Synthesis and Double-Port Approach,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 12, Aug. 2016 (cit. on p. 101).

-
- [44] R. Rodrigues, V. Preda, F. Sanfedino, and D. Alazard, “Modeling, robust control synthesis and worst-case analysis for an on-orbit servicing mission with large flexible spacecraft,” *Aerospace Science and Technology*, vol. 129, p. 107865, 2022 (cit. on p. 102).
 - [45] N. Guy, D. Alazard, C. Cumér, and C. Charbonnel, “Dynamic modeling and analysis of spacecraft with variable tilt of flexible appendages,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, Jan. 2014, 021020 (cit. on p. 103).
 - [46] T. Ott, W. Fichter, S. Bennani, and S. Winkler, “Precision pointing H_∞ control design for absolute, window-, and stability-time errors,” *CEAS Space Journal*, vol. 4, no. 1–4, 13–30, Jan. 2013 (cit. on p. 106).
 - [47] M. Pittelkau, “Definitions, metrics and algorithms for displacement, jitter and stability,” in *Flight Mechanics Symposium, NASA Goddard Space Flight Center - NASA/CP-2003-212246*, 2003 (cit. on p. 106).