

POLITECNICO DI TORINO

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

Master's degree course in Aerospace Engineering

Master's Degree Thesis

**Design of the ESA PROBA-3
Alternative SPS algorithm for
formation flight applications**



**Politecnico
di Torino**



Supervisor
Prof. Elisa Capello

Candidate
GABRIEL JOSÉ GUTIERREZ
ID: 268940

Research Tutor
Dr Maurizio Pancrazzi
Dr Silvano Fineschi

ACADEMIC YEAR 2020-2021

This work is subject to the Creative Commons Licence

Summary

The ESA Project for On-Board Autonomy - 3 (PROBA-3) mission is a technological demonstration for precise formation flying of two small spacecraft in a Highly elliptical Earth Orbit. The mission's main objective is to demonstrate and validate formation flying techniques with high precision, performing formation control, collision avoidance, and rendezvous operations.

In addition to the technological goals, PROBA-3 accommodates a scientific payload that takes advantage of the formation flying capabilities derived from the millimetre relative position precision. The Association of Spacecraft for Polarimetric and Imaging Investigation of the Corona of the Sun (ASPIICS) is a coronagraph system distributed over two satellites: the Coronagraph SpaceCraft (CSC) carrying a telescope imaging the solar corona in visible light, and the Occulter SpaceCraft (OSC) carrying the Sun occulter disk that will produce an artificial eclipse. This configuration creates a 150-meter instrument that shall obtain fundamental scientific measurements to study the solar corona.

The onboard metrology system shall ensure precise reciprocal positioning of the two satellites, using a closed-loop arrangement to accomplish these tasks. Thus, a metrology chain of increased accuracy is employed, culminating with the Shadow Positioning Sensors (SPS) subsystem. These shall return the three-dimensional displacement of the penumbra centre with respect to the entrance pupil, taking as input the irradiance measured by four sensors. Then, an algorithm will be used to compute the necessary correction with respect to the nominal position.

In this thesis, developed in collaboration with the Istituto Nazionale di AstroFisica (INAF), the overall functioning of the SPS metrology subsystem will be described. Specifically, the main objective of the work is to develop and validate an alternative algorithm that shall validate the lateral positioning results coming from the satellite's algorithm. The flight algorithm is based on a different approach that allows a faster and straightforward calculation of the centre of the umbra position while maintaining an elevated accuracy. However, this method intrinsically leads to a non-axis-symmetric penumbra profile obtained from the interpolation of the expected irradiance with a pseudo-paraboloidal fit, which could lead to errors. For this reason, ESA proposed an alternative algorithm that uses Cardano's method to determine the radial position of each sensor concerning the instrument's pupil centre and,

using the known positions, reconstructs the desired umbra centre coordinates in the pupil plane reference system. This procedure is computationally costly with respect to the onboard algorithm, but it can achieve a comparable accuracy while guaranteeing an axis-symmetric model for the penumbra profile.

The opening chapter of this thesis provides an exhaustive introduction to the mission: spacecraft and orbit characteristics, the mission objectives, a high-level explanation of the scientific payload and, finally, an introduction to the metrology chain that enables the fulfilment of the mission. A dedicated chapter goes to the Shadow positioning Sensors. These are the hearth of the fine metrology subsystem that will allocate the developed algorithm. A thorough explanation of data's logical and electrical flow is needed to better understand the following considerations.

Consequently, a detailed description of the fine positioning algorithms is presented. First, an introduction to the expected penumbra profile interpolated with a third-order equation is given, followed by the needed corrections and assumptions leading to the algorithm's lateral and longitudinal solutions. The linear and differential algorithms are accurate yet straightforward procedures that will be useful for validating and integrating the more complex methods used in the pseudo-paraboloidal and alternative algorithms.

Once the existing approaches have been defined, the alternative algorithm is introduced. A detailed explanation of the variations and adaptations done by the author to this method is provided, where the penumbra profiles are compared, and the mathematical solutions of the system are analyzed. It is important to note that the developed algorithm needs to follow the same structure as the pseudo-paraboloidal implemented onboard the PROBA-3 spacecraft. Thus, a considerable effort was made to uniform the inputs and outputs of these parallel systems to provide a unified and seamless integration for the comparison of the results.

A first approach describes the development of the alternative algorithm Matlab Script, where a fine mesh of 1000x1000 points inside the "requirement" and "goal" box is used to validate that the precision requirements are fulfilled. The next step consists of integrating simulated irradiance measures (obtained from the flight algorithm simulator) to validate the algorithm's functioning and structure, and compare these results to the pseudo-paraboloidal. The analysis inside the requirement and goal box also saw the comparison of a combination of the algorithms in order to determine the best possible solution.

A series of tests were performed from these locally generated irradiance measurements to compare the different algorithm's performance. Inside the ± 10 mm requirement box, the alternative algorithm (using Cardano's method) achieves the best results, followed by the pseudo-paraboloidal + linear (PPL) algorithm. The error applying Cardano's is one order of magnitude lower than the PPL, effortlessly satisfying the $50 \mu\text{m}$ precision requirement. Furthermore, it achieves an almost uniform distribution of the error throughout the box, making it the most reliable algorithm for the validation purposes required by the PROBA-3 mission, but a high

computational cost is expected.

The analysis is significantly more complex regarding the goal box since the obtained errors change the shape and precision zones between algorithms. The alternative algorithm delivers an odd semi-circular shaped error that spikes at around 20 to 30 mm from the centre. The pseudo-paraboloidal, linear and PPL algorithm's errors follow an almost uniform distribution inside the goal box, with an error that rises as the points approach the edges (or corners).

For this reason, the preferred algorithm in this box shall be the combination of the pseudo-paraboloidal and the linear. This algorithm, developed by INAF, delivers outstanding results with a low computational cost. For calibration reasons, the alternative algorithm might be preferred in the area adjacent to the origin. This zone achieves exceptional accuracy, unmatched by the other algorithms. Hence, the main objective of this thesis work is proven to be plausible using a locally generated penumbra profile.

The SPS Simulator, also developed by INAF, is a powerful tool that allows the determination of the illumination profile, and consequently, the computation of the fitting parameters and the umbra centre coordinates in the Coronagraph Instrument (CI) pupil plane. An essential aspect of the simulator is the possibility of varying the fitting parameters to test the algorithm's robustness. This simulator was developed for the pseudo-paraboloidal algorithm (and PPL). Therefore, a modification has been introduced by the author to make it compatible with Cardano's method too.

Consequently, a file reading alternative algorithm script is necessary to integrate the results, performing a preliminary analysis using a set of random points. Next, the structure of the simulator's algorithms is described, with an emphasis on the required flags and checks that shall be performed for the alternative algorithm, using the same inputs and outputs as the flight algorithm. The modifications needed to adapt the alternative algorithm to the SPS simulator are highlighted, and the used Matlab scripts and functions are presented.

Finally, once the SPS simulator has been adapted for the new algorithm, the robustness of the alternative algorithm and the fitting parameters are tested. This procedure allows the determination of the percentage variation range allowed for each reconfigurable parameter of the algorithm to deliver a valid solution, and it provides an assessment of the accuracy needed in the determination of each single fitting parameter. This information shall be useful during the missions calibration procedures.

Acknowledgements

The author acknowledges the support provided by the Istituto Nazionale di Astrofisica - Astrophysical Observatory of Turin (INAF-OATo). Especially to the Research Tutor, Dr Maurizio Pancrazzi, that followed and supported the author during the thesis. To Dr Silvano Fineschi, director of INAF-OATo, that provided the opportunity for the author. To Francesco Amadori for the assistance and insight in the development of the algorithm and simulator, and to all the scientist and engineers from INAF and ESA involved in the PROBA-3 Mission. Furthermore, the author acknowledges the help supplied by Politecnico di Torino supervisor Prof. Elisa Capello.

Moreover, the author is grateful for the support and motivation provided by colleagues, friends and family during the past years and in the elaboration of this thesis.

Contents

List of Tables	x
List of Figures	xI
I General Introduction	1
1 ESA PROBA-3 Mission	3
1.1 Spacecraft characteristics	4
1.2 Mission objectives	5
1.3 Orbit characteristics	5
1.4 ASPIICS: Association of Spacecraft for Polarimetric and Imaging Investigation of the Corona of the Sun	6
1.5 Metrology chain	8
1.6 Formation flying	11
1.6.1 Previous formation flying missions	11
2 SPS: Shadow Positioning Sensors	13
2.1 SiPM sensors	14
2.2 SPS requirements	15
2.3 SPS design	16
3 Fine positioning algorithm for formation flight	19
3.1 Penumbra Profile	19
3.1.1 Corrections to the irradiance curve	20
3.1.2 Expected irradiance curve	22
3.2 Pseudo - Paraboloidal algorithm	24
3.3 Linear algorithm	27
3.4 Pseudo-paraboloid + Linear algorithm (PPL)	29
3.5 Differential algorithm	31

II	Alternative SPS Algorithm & Simulator	33
4	Alternative SPS Algorithm	35
4.1	An alternative model for irradiance	35
4.1.1	Boundary conditions	38
4.1.2	Inverting the system	39
4.1.3	Plot of the alternative penumbra profile	39
4.1.4	Cardano's Method - Deriving the radial distance	41
4.1.5	Deriving the position of the centre of the umbra	43
4.2	Matlab function for Cardano's Method	44
5	Local validation of the Alternative Algorithm	45
5.1	Variation of the coordinate system	46
5.2	Locally Generated profile for validation	47
5.2.1	Modification of the boundary conditions	48
5.2.2	Generation of local position	48
5.2.3	Solving the local system	49
6	Algorithm comparison: Results from locally generated data	51
6.1	Lateral Error	52
6.1.1	Requirement box - Alternative	52
6.1.2	Requirement box - Pseudo-paraboloid	54
6.1.3	Requirement box - Linear & Pseudo-paraboloid + Linear (PPL)	55
6.1.4	Requirement box - Alternative + Pseudo-paraboloid	57
6.1.5	Requirement box - Algorithm comparison	58
6.1.6	Goal box - Alternative	63
6.1.7	Goal box - Pseudo-paraboloid	65
6.1.8	Goal box - Linear	66
6.1.9	Goal box - Alternative + Linear	67
6.1.10	Goal box - Pseudo-paraboloidal + Linear	70
6.1.11	Goal box - Alternative + PP algorithm	72
6.1.12	Goal box - Algorithm comparison	73
6.2	Longitudinal Error	79
6.2.1	Algorithm comparison	80
6.3	Conclusions and remarks	82
7	SPS Simulator	85
7.1	File reading algorithm	86
7.1.1	Results from example input	87
7.2	SPS Simulator structure	89
7.2.1	Input Pre-Processing	90
7.2.2	Validity Flag 1	91
7.2.3	Algo Verification	91

7.2.4	Algorithm implementation	92
7.2.5	Validity Flag 2	94
7.3	Alternative SPS simulator	95
7.3.1	Sensitivity test	97
8	Conclusions and final remarks	103
	Bibliography	106
A	Goal box comparison with fine fitting parameters	109
A.1	Lateral error	109
A.2	Longitudinal error	115
B	Sensitivty test results for the alternative algorithm	119
B.1	Coarse parameters	120
B.2	Fine parameters	124
C	Algorithm comparison with locally generated points using redundant SPS set	129
C.1	Alternative Algorithm	130
C.2	Pseudo-paraboloidal Algorithm	132
D	Matlab Codes	135
D.1	Cardano's Method function	135
D.2	Alternative_SPS_Algorithm_Generate	137
D.2.1	Data acquisition	137
D.2.2	Irradiance model Parameters (fine or coarse)	139
D.2.3	Solving with Cardano's Method function	141
D.2.4	Longitudinal position calculation	142
D.2.5	Solution validation	143
D.3	Alternative Algorithm - Read	146
D.3.1	.JSON File loading	146
D.3.2	SPS irradiance reading	146
D.3.3	Cardano's solution	147
D.3.4	Linear solution	148
D.4	Reconfigurable_params.json	149
D.5	RAW_DATA.json	150
D.6	Test case SPS readings	150
D.7	Main_Alt	151
D.8	SPS_Algorithms_Alt_2	154
D.8.1	Alternative algorithm function	156
	Acronyms	161

List of Tables

1.1	PROBA-3 SC characteristics [16] [18]	4
1.2	Orbital parameters of PROBA-3 [13] [16]	5
1.3	Specifications of the ASPIICS coronagraph [16]	7
3.1	Error of the average of algorithms at nominal ISD [5]	31
6.1	Lateral error of the algorithms and calculation time for 100x100 grid inside the requirement box	62
6.2	Lateral error of the algorithms and calculation time for 100x100 grid inside the goal box	77
7.1	Lateral error for Alternative algorithm	87
7.2	Lateral error for Pseudo-paraboloidal algorithm	88
7.3	Lateral error for Alternative + Linear algorithm	88
7.4	Parameters variation range for valid coordiantes	98
7.5	Parameters uncertainty range for error lower than 50 μm	99
D.1	SPS High Gain (HG) measurements inside the requirement box used as “RAW_DATA” for the Read algorithm	150

List of Figures

1.1	Illustration of the CSC (left) and of the OSC (right) [16]	4
1.2	Operational phase actions during an orbit [13]	6
1.3	Coronagraph Optical Box [16]	7
1.4	PROBA-3 Metrology Chain in incremental precision order [16]	8
1.5	PROBA-3 equipment layout of CSC front face and OSC rear face [16]	10
2.1	Distribution of umbra/penumbra projected over the CSC pupil plane [13]	14
2.2	Requirement (yellow) and goal (red) box representation [2]	16
2.3	SPS logic measurements design [13]	17
3.1	SPS responsivity (solid line) and the variation as a function of wavelength of the limb darkening coefficients u_λ (dotted line) and v_λ (dashed line) [2]	21
3.2	Level of expected SPS response current in the penumbra on the entrance pupil plane as a function of the distance from the telescope optical axis around the SPS location radius at 55 mm (vertical solid line). Neglecting the limb darkening (solid line), including the limb darkening (dashed line) and adding to the limb darkening a sunspot located at the limb (dotted line). [2]	22
3.3	Digitized penumbra profile (DN) [11]	23
3.4	SPS position around the CI pupil	25
3.5	Pseudo-paraboloid lateral error inside the requirement box [7]	27
3.6	Linear algorithm lateral error inside the requirement box [7]	28
3.7	Histogram of statistical error of both algorithms [7]	29
3.8	Error associated to the lateral position computation of the combined algorithm [7]	30
3.9	Error associated to the longitudinal position computation of the combined algorithm [7]	30
4.1	Pseudo-paraboloid penumbra profile (Irradiance in mW) [22]	36
4.2	Penumbra profile: Third-order fine fitting curve	37
4.3	Alternative penumbra profile (Irradiance in DN)	40
4.4	Alternative 3D penumbra profile	40
5.1	Updated coordinate system [13]	46

5.2	Coarse and fine penumbra profiles	47
6.1	2D Lateral error plot in the requirement box using Cardano and 100x100	52
6.2	Lateral error surface in the requirement box using Cardano	53
6.3	2D Lateral error plot in the requirement box using pseudo-paraboloid	54
6.4	Lateral error surface in the requirement box using pseudo-paraboloid	55
6.5	Lateral error plot in the requirement box	55
6.6	Lateral error surface in the requirement box using linear algorithm	56
6.7	Lateral error surface in the requirement box using PPL	56
6.8	Lateral error plot in the requirement box for Alternative + PP	57
6.9	Statistic of both algorithms error for the lateral position inside the requirement box for a 1000x1000 grid	58
6.10	Lateral error surface in the requirement box using both algorithms	59
6.11	Statistic of linear and PPL algorithms error for the lateral position inside the requirement box	60
6.12	Comparison error plot in the requirement box for Alternative + PP	60
6.13	Lateral difference of the absolute error (Cardano - PP) in the requirement box	61
6.14	Comparison of the precision of the alternative and flight algorithms inside the requirement box	62
6.15	Lateral error in the goal box using Cardano	63
6.16	Lateral error surface in the goal box using Cardano	64
6.17	Lateral error in the goal box using Pseudo-paraboloid	65
6.18	Lateral error surface in the goal box using pseudo-paraboloid	66
6.19	Lateral error in the goal box using linear algorithm	67
6.20	Lateral error surface in the goal box using linear algorithm	67
6.21	Lateral error in the goal box using alternative + linear	68
6.22	Lateral error surface in the goal box using alternative + linear	68
6.23	Lateral error in the goal box using alternative and alternative + linear	69
6.24	Lateral error surface in the goal box using Pseudo-paraboloid + Linear	70
6.25	Lateral error in the goal box using PP and PPL	71
6.26	Lateral error in the goal box using Alternative + Pseudo-paraboloid	72
6.27	Lateral error in the goal box using Cardano and PP	73
6.28	Statistical error histogram in the goal box using Cardano and PP	74
6.29	Lateral difference of the absolute error in the goal box for Cardano vs PP	74
6.30	Lateral error in the goal box using Cardano + Linear and PPL	75
6.31	Lateral difference of the absolute error in the goal box for Cardano + Linear vs PPL	76
6.32	Comparison of the precision of the alternative and flight algorithms inside the goal box	77
6.33	Division of the algorithm's precision inside the goal box	78

6.34	Longitudinal error plot in the requirement box	80
6.35	Longitudinal error plot in the requirement box using both algorithms	80
6.36	Statistic of both algorithm's error for the longitudinal position with a 1000x1000 grid inside the requirement box	81
6.37	Statistic algorithm comparison for the longitudinal position with a 1000x1000 grid inside the requirement box	81
7.1	General architecture of the algorithm [8]	90
7.2	SPS alternative simulator: "[X, Y, Z] to [SPS]" at Y= 5 and Z= 1 mm	96
7.3	SPS alternative simulator: "[SPS] to [X, Y, Z]"	97
7.4	Sensivity test longitudinal error result	100
7.5	Sensivity test lateral error result	100
7.6	Sensivity test longitudinal error result	101
A.1	2D Lateral error plot in the goal box using Cardano and 100x100 grid	110
A.2	Lateral error surface in the goal box using Cardano and 100x100 grid	111
A.3	2D Lateral error plot in the goal box using pseudo-paraboloid	111
A.4	Lateral error surface in the goal box using pseudo-paraboloid	112
A.5	Statistic of both algorithm's error for the lateral position with a 1000x1000 grid inside the goal box	113
A.6	Lateral error surface in the goal box for both algorithms	114
A.7	Lateral difference of the absolute error surface in the goal box	114
A.8	Lateral difference of the absolute error in the goal box	115
A.9	Longitudinal error plot in the goal box using both algorithms	116
A.10	Longitudinal difference of the absolute error in the goal box	116
A.11	Statistic of both algorithm's error for the longitudinal position with a 1000x1000 grid inside the goal box	117
B.1	<i>a_coarse</i> sensivity test result plots	120
B.2	<i>b_coarse</i> sensivity test result plots	121
B.3	<i>c_coarse</i> sensivity test result plots	122
B.4	<i>RO_coarse</i> sensivity test result plots	123
B.5	<i>a_fine</i> sensivity test result plots	124
B.6	<i>b_fine</i> sensivity test result plots	125
B.7	<i>c_fine</i> sensivity test result plots	126
B.8	<i>RO_fine</i> sensivity test result plots	127
C.1	Alternative algorithm inside the goal box using redundant SPS set .	130
C.2	Alternative algorithm inside the ± 100 mm box	131
C.3	PP algorithm inside the goal box using redundant SPS set	133

Part I

General Introduction

Chapter 1

ESA PROBA-3 Mission

The Project for On-Board Autonomy - 3 (PROBA-3) mission is part of the General Support Technology Program (GSTP) of the European Space Agency (ESA). This mission is a technological demonstration for precise Formation Flying (FF) of two spacecraft in a Highly Elliptical Earth Orbit (HEO). PROBA-3 shall perform formation maintenance, resizing, pointing and reorientation manoeuvres that aim to change the relative position and attitude of the spacecrafts.

The main objective of this mission is to demonstrate and validate the FF with high precision to gather the knowledge necessary for future applications. The mission will contribute to developing, designing, and implementing the required tools for the FF verification, including relative dynamic experiments such as six degrees of freedom formation control, collision avoidance, and rendezvous operations.

Furthermore, the PROBA - 3 mission accommodates a scientific payload that takes advantage of the exceptional FF capabilities derived from the millimetre relative position precision. The Association of Spacecraft for Polarimetric and Imaging Investigation of the Corona of the Sun (ASPIICS) consists of a coronagraph system distributed over the two satellites, the Coronagraph SpaceCraft (CSC) carrying the detector and the Occulter SpaceCraft (OSC) carrying the Sun occulter disk. This configuration generates a 150-meter instrument that creates a compelling demonstration of FF, and a highly efficient never-built coronagraph that shall provide a crucial scientific return.

Since October 2006, an association of industrial companies in ESA member states has gathered the knowledge acquired from the previous FF LEO missions, TanDEM-X [23] and PRISMA [15], to derive the requirements for an order of magnitude enhancement in FF precision. Furthermore, the FF manoeuvres will be performed autonomously for numerous hours during the apogee phase of the HEO, followed by a formation break and reacquisition every single orbit. This is possible by virtue of the complex metrology and relative navigation systems onboard both SC that can perform calibration, collision avoidance and formation control [17].

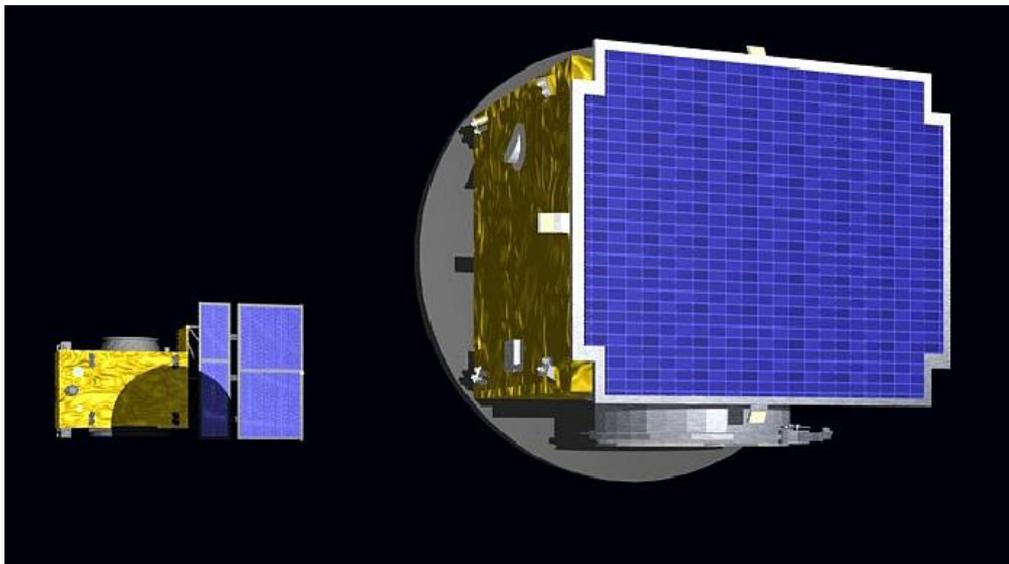


Figure 1.1: Illustration of the CSC (left) and of the OSC (right) [16]

1.1 Spacecraft characteristics

The virtual rigid structure of the coronagraph will be produced by two mini-satellites flying in a formation with relative position precision in the order of a millimetre. The principal characteristics of the SC are summarized in the following table:

Table 1.1: PROBA-3 SC characteristics [16] [18]

Parameter	Occulter Spacecraft	Coronagraph Spacecraft
Area	1.77 m^2	3.34 m^2
Volume	$0.9 \times 1.4 \times 0.9 \text{ m}^3$	$1.1 \times 1.8 \times 1.7 \text{ m}^3$
Wet mass	$\sim 221 \text{ kg}$	$\sim 339 \text{ kg}$
Dry mass	190 kg	327 kg
Thrust per thruster	10 mN	1000mN
SRP coefficient	1.9 (1.5)	1.29

The CSC Guidance, Navigation and Control (GNC) system is supplied with four reaction wheels, six Sun sensors, two GPS receivers, two three-axis gyroscopes and a three-headed star tracker. Furthermore, this SC hosts the ASPIICS scientific instrument and monopropellant thrusters for large manoeuvres.

The OSC is dominated by a 1400 mm diameter occulting disk and a solar array on the opposite side. The GNC system is analogous to the one onboard CSC, but this satellite hosts the optical metrology sensors and an additional three-axis gyroscope [18].

1.2 Mission objectives

After the completion of the PROBA-3 mission, the following results are expected:

- Validated formation flying control algorithms.
- Mature formation flying metrology for future readiness of “of the shelf” equipment.
- Scientific return of coronagraph measurements.
- Technology experiments operation and demonstration.
- AOCS software improvements for FF.

1.3 Orbit characteristics

ESA PROBA-3 two year mission is expected to be launched aboard a VEGA European launcher within 2023 [21]. A Highly elliptical orbit with an apogee of 60.530 km was chosen to minimize the propellant consumption during the FF manoeuvres, while a 59° inclination lessens the radiation dose from the Van Allen Radiation Belts.

Table 1.2: Orbital parameters of PROBA-3 [13] [16]

Mission duration	2 yr.
Perigee & Apogee	600 km & 60.530 km
Semi-major axis	36943 km
Eccentricity	0.8111
Inclination	59°
Orbital period	19 hr. 38 min.
FF period	~ 6 hr.

The formation flying demonstration and the scientific observation occurs throughout the apogee pass, where the gravity gradient is negligible and the relative velocity is minimum. The high precision FF is attained autonomously, optimizing power consumption while preserving stability and safety. In this FF phase, the PROBA-3 mission will validate the metrology technologies, complete FF acquisition and maintenance, resizing, retargeting and stationkeeping while using the Sun as a reference direction.

Furthermore, this nearly six hours long phase will be above the Global Navigation Satellite System (GNSS) constellation range. Therefore, the relative position of the SC is derived from the information gathered in the prior perigee phase (that lays below the GNSS) and the metrology subsystem chain. This high-velocity phase

obliges a formation break, where the two spacecraft are arranged in a secure orbit to avoid collision and formation evaporation. Using ground and space references, PROBA-3 combines the relative and absolute position data for orbit propagation while establishing housekeeping and communications down-link with mission control.

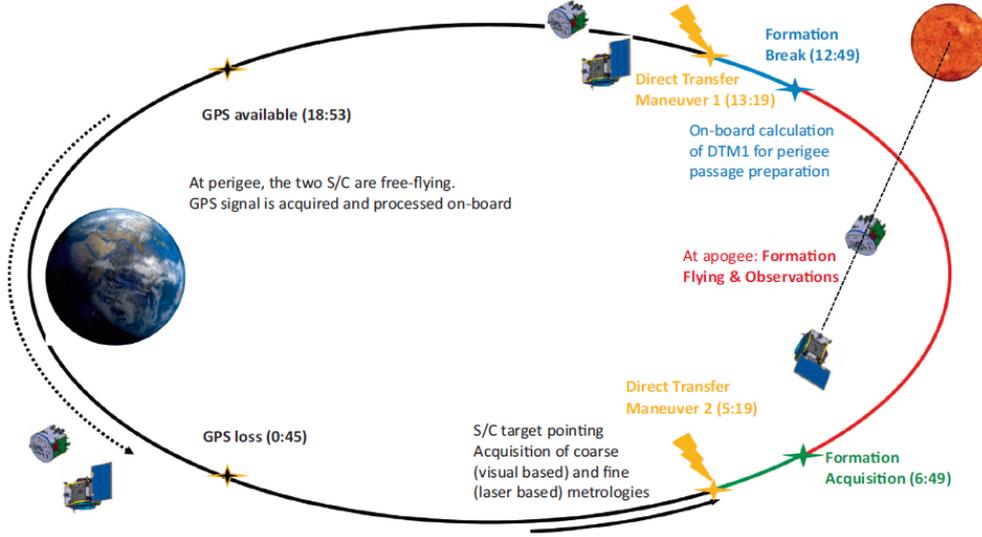


Figure 1.2: Operational phase actions during an orbit [13]

1.4 ASPIICS: Association of Spacecraft for Polarimetric and Imaging Investigation of the Corona of the Sun

The PROBA-3 coronagraph payload exerts an important scientific and technological role during the mission. ASPIICS is designed to perform imaging in narrow spectral bandpasses and in polarized wide-bandpasses of the solar corona, observing a FOV from $1.02 R_{Sun}$ to around $3 R_{Sun}$. This particular radius permits the study of the magnetic field, in a region which plays an essential role in the dynamics of the Sun corona. The high precision of the FF allows an uninterrupted observation (close to a total solar eclipse) without the undesirable effects of the atmosphere. To accomplish the minimum 1,000 hours of coronagraph observations of the mission, a total of 167 orbits are considered necessary (about 4.5 months).

These scientific objectives are attained by detecting the low-contrast structures in the corona, requiring a minimal level of instrumental stray light and a high spatial resolution. The instrument permits the observation as close to the solar limb as any other space coronagraph has never achieved, enabling the characterization of the local plasma characteristics.

The scientific payload is composed of the Coronagraph Instrument and four (eight for redundancy) SPS (Shadow Position Sensors) mounted on the CSC, and an Occulting Disk with six OPS (Occulter Position Sensors) installed on the OSC. The design is an adaptation to the formation flying configuration of a traditional externally occulted coronagraph, where the occulter hiding the bright solar disk is positioned on front to the entrance pupil of the telescope.

Table 1.3: Specifications of the ASPIICS coronagraph [16]

Optical specifications	
Entrance aperture radius	25 mm
Field of view	1.02 - 3 R_{Sun}
Spatial scale	2.8 arcsec/pixel
Telescope effective focal length	1150 mm
Stray light rejection	$5 \cdot 10^8$
Detector characteristics	
Operating temperature	$< 60^\circ$
Size	2048 x 2048 pixels (15 μm pixel)
Dynamic range	16 bits

The instrument can be divided into different main parts: the Coronagraph Optical Box (COB), which preserves the position steadiness of the external parts parts; the Tube, that supports the SPS; the Front Door Assembly (FDA) and the Internal Occulter (IO); the Equipment Box (EQB); the Filter Wheel Analyser (FWA), that has six different optical filters; and the SHutter Mechanism (SHM). A graphic representation of the external parts can be seen in Figure 1.3.

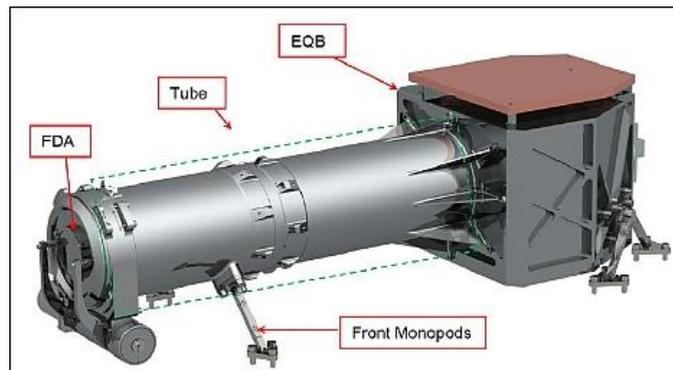


Figure 1.3: Coronagraph Optical Box [16]

The technological role of the ASPIICS payload depends on the metrology sensors: the Occulter Position Sensor Emitter (OPSE) and the Shadow Position Sensor (SPS). These subsystems are explained in detail in the following section.

1.5 Metrology chain

In order to accomplish the formation flying and relative navigation between the spacecrafts, the PROBA-3 guidance, navigation and control (GNC) function has been split into two software modules: the Formation Flying GNC (FF-GNC) and the Spacecraft GNC (SC-GNC). The FF-GNC software is responsible for the navigation and guidance, and the computation of the control commands for the relative position and attitude of both SC. The SC-GNC software monitors the absolute position and attitude of the individual SC. Furthermore, the SC-GNC software is responsible for handling the commands to the actuators, included those received from the FF-GNC software.

FF algorithms are designed to merge and propagate the available metrology sensors measurement. The estimation of the performance of the FF depends on the highest accuracy element of the metrology chain. This chain comprises GPS receivers, CLS (Coarse Lateral Sensor) and FLLS (Fine Lateral and Longitudinal Sensor) to finally acquire the CI (Coronagraph Instrument) subsystems.

The ASPIICS images of the corona and the SPS subsystem readings are combined with the orbital parameters and GNC sensors onboard. To compute a better estimate of the relative position of the spacecrafts, this data will be merged with the post-processed ground information [16] [20].

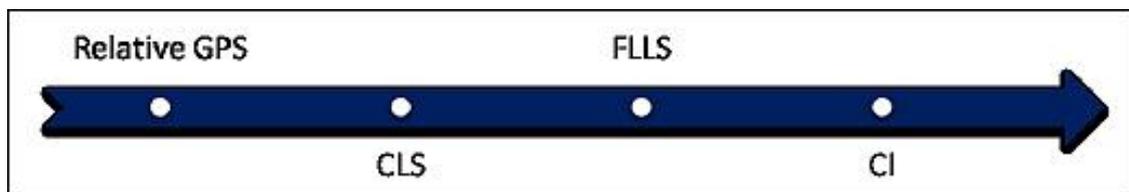


Figure 1.4: PROBA-3 Metrology Chain in incremental precision order [16]

Metrology system: Relative GPS

This is the first stage of the metrology chain, where the data gathered during this phase is essential for the relative longitude between the spacecraft. The Relative GPS is a method used to obtain relative position and velocity between two satellites in Low Earth Orbit; this necessitates GPS receivers in both SC and a communications link (between the satellites and ground), acquiring the relative dynamics of the SC. This process is performed both onboard the SC and on ground, where the execution of manoeuvres during the perigee pass shall be avoided in order to have the maximum propagation accuracy.

GPS constellation is in a Medium Earth Orbit (MEO); thus, the PROBA-3 mission will be able to collect the Relative GPS data only during the low altitude pass (with an accuracy of 7.5 cm (1σ)). When both SC get closer to the apogee

phase, the beforehand known data is propagated with an accuracy lesser than 10 meters, allowing the initiation of the CLS metrology system.

Metrology system: CLS

The Coarse Lateral Sensor (CLS) is the second stage in the chain of relative metrology, and in combination with the Relative GPS, allows the precision required to acquire the fine metrology that distinguishes PROBA-3 from the former FF missions. This laser-based system allows a 1 mm accuracy (1σ) from a 150 m lateral distance and a ± 10 arcsec attitude control. The operational range (only during the apogee phase) of 150 ± 13 m in lateral and 25 - 250 m for longitudinal, acknowledges a precise formation acquisition in the initial apogee phase, which allows the following mechanism in the chain to work.

This system consists of a laser beam (aboard the CSC) that is projected towards a corner cube (on the OSC) that reflects back the beam to a detector (on the CSC). The detector is autonomously able of calculating the lateral shift and the relative coordinates of the target.

Metrology system: FLLS

The third stage of the chain is the optical metrology Fine Lateral and Longitudinal Sensor (FLLS) that, as for the CLS, is distributed between the two satellites. On the OCS a laser pulse train is created and split into two parts. One part is transmitted to a corner cube on the CSC, getting reflected and received back, while the other is sent to an interferometer as a reference measure. The reflected beam is then split once again and sent to a lateral position sensor and to a distance detector, where the distance is computed from the interference between the bounced signal and the reference beams. A range of ± 20.5 mm in lateral and 25 - 250 mm for longitudinal allows a measurement accuracy (1σ) of 21 μm (lateral) 30 μm (longitudinal). The development and validation of the FF algorithms and systems of this stage of the chain are the key challenges of the PROBA-3 mission.

To test and validate this precise metrology system, the FLLS needs to be calibrated to an extreme level of accuracy using the Coronagraph Instrument (CI) data. Hence, the FLLS measurement is combined with the SPS and OPSE metrology subsystems. These are part of the CI, the last link in the mission metrology chain.

Metrology sub-system: OPSE

The Occulter Position Sensor Emitter (OPSE) is a mechanism that consists of three light emitters mounted on the external occulting disc of the OSC facing the CSC. The images delivered by the coronagraph have a pattern that uniquely determines the position along the transverse axes regarding the instrument coordinate system.

Additionally, the OPSE can provide an estimate of the inter-satellite distance and the external occulter orientation.

The three-light emitters are placed close to the centre of the disk. The images acquired by the coronagraph sensor are then transmitted to ground for analysis. Therefore, the information provided is not usable onboard in real-time and entails a fully operational coronagraph (doors opened and umbra centred in the pupil).

Metrology sub-system: SPS

The Shadow Positioning Sensor (SPS) is part of the scientific ASPIICS payload. This subsystem verifies the correct alignment of the shadow cone generated by the occulting disc into the entrance pupil of the coronagraph. An algorithm has been developed to compute the absolute location of the umbra with high accuracy with respect to the centre of the pupil of the coronagraph instrument (CI). This algorithm permits the detection of movement in the umbra profile from the nominal position, preventing the risk of receiving the full Sun glare inside the coronagraph instrument, and allowing the acquisition of the solar corona with high precision. Furthermore, this subsystem upgraded accuracy shall be used as a reference to calibrate the preceding links of the metrology chain. The SPS subsystem is further explained in the following chapter.

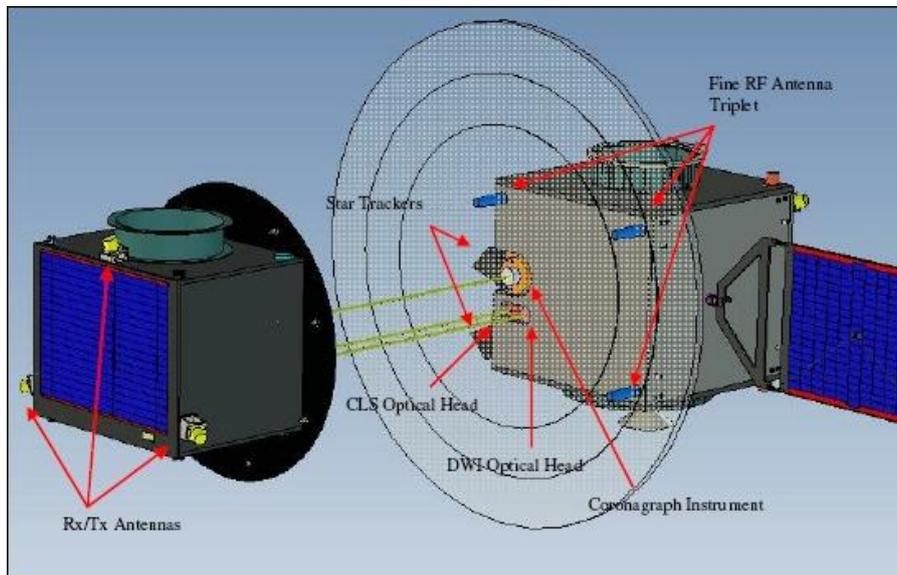


Figure 1.5: PROBA-3 equipment layout of CSC front face and OSC rear face [16]

1.6 Formation flying

The primary purpose of this mission is to demonstrate and develop the technologies required for a very high precision formation flying. This precision is attainable as a result of the metrology chain explained in the previous section. The technologies developed by ESA and the European aerospace industry for the previous missions are essential for the increase (by orders of magnitude) of the system’s precision.

During the 6 hours of apogee pass, the two SC will retain a very high precision FF for scientific remote sensing, whereas the perigee pass will maintain a safe relative position (about 250 meters apart). Moreover, PROBA-3 will validate the FF capabilities with:

- Formation keeping at 144 - 146 m.
- Resizing manoeuvres that adjust Inter Satellite Distance (ISD), ranging from 250 m to 25m.
- Retargeting manoeuvres of rigid rotation about the CSC centre of mass up to 30 °.
- A combination of both retargeting and resizing.

The formation keeping distance is a function of the relative distance of the Earth and the Sun. This distance derives from the diameter of the ASPIICS instrument and the penumbra profile generated by the OSC. The ISD is defined as:

$$ISD = \frac{(R_D - r_p)}{\tan(1.02 \theta_{Sun})}$$

where θ_{Sun} is the (apparent) radius of the Sun in radians, that changes according to the seasonal Earth-Sun distance variation; r_p is the pupil radius of the coronagraph instrument and R_D is the OSC disk radius [13].

The pupil of the ASPIICS instrument is $r_p = 2.5 \text{ cm}$ and the OSC has a $R_D = 71 \text{ cm}$. Since the Earth-Sun distance varies between $r_{ES} = 152.171522 \cdot 10^6 \text{ km}$ at aphelion and $r_{ES} = 147.166462 \cdot 10^6 \text{ km}$ at perihelion, and with an angle $\theta_{Sun} \sim 0.5^\circ$, it is possible to find the maximum value of $ISD_{max} = 146.935 \text{ m}$ and minimum of $ISD_{min} = 142.101 \text{ m}$.

Consequently, the PROBA-3 mission will maintain a FF ISD that depends on the relative position of the Earth-Sun system. Very accurate calibration of the instruments is required in order to fulfil the mission objectives.

1.6.1 Previous formation flying missions

Through this mission, ESA aims to gather the necessary knowledge for more complex and larger FF missions. PROBA-3 is the latest in a line of FF European missions that have straightened the path for this objective.

TAFF

TanDEM-X Autonomous Formation Flying Experiment (TAFF) was a German mission that used FF for X-band Synthetic Aperture Radar (SAR) interferometry with two spacecraft of around 1200 kg in Low Earth Orbit (514 km sun-synchronous).

Both spacecraft flew in a precise formation to form a SAR with baselines of 0.5 km. The generation of accurate digital elevation models (DEMs) was possible to grant a higher resolution data compared with previous missions. Likewise, a secondary objective of the TanDEM-X mission involves the demonstration of along-track SAR for measurements of the ocean currents [23].

An Autonomous Formation Flying (TAFF) system is equipped in the TanDEM-X SC. This system delivers a robust control algorithm for formation keeping and demonstrates a level of autonomy that facilitates the replacement of ground-based control. The relative positions and velocities are delivered by two synchronized GPS receivers onboard both SC, communicated through Inter-Satellite Link (ISL). The GNC uses an extended Kalman filter that allows the attainment of navigation accuracy at the meter level.

Consequently, the PROBA-3 mission shall accomplish a three to four degree of magnitude improvement in relative distance precision compared to the TAFF mission.

PRISMA

The Swedish Prototype Research Instruments and Space Mission technology Advancement (PRISMA) is a formation flying and rendezvous technology demonstration mission. Two small satellites fulfil, as primary objectives, GNC demonstrations and sensor technology experiments to gain the necessary knowledge for rendezvous and FF operations. Each SC is supplied with transmitter and receiver antennae as the coarse metrology measurement and optical system for more accurate data.

The primary sensor is the Formation Flying Radio Frequency FFRF that unites the GNC software and the ground segment communications. This system is responsible for the relative positioning of the SC, delivering a Line Of Sight (LOS) position and velocity vector as coarse metrology measurements. High accuracy is achieved by using a metrology chain composed of the FFRF sensor that uses ranging through the ISL in dual-phase; and the coarse and fine optical metrology subsystems. A precision in distance of one centimetre and velocity in the order of the millimetre per second is attainable with an accuracy of one degree in the LOS. However, external sensors are required for attitude guidance [15].

PRISMA mission was a significant step towards the development of the PROBA-3 mission, where the metrology chain was primarily tested into a precision lower of one order of magnitude.

Chapter 2

SPS: Shadow Positioning Sensors

The PROBA-3 metrology chain achieves its maximum accuracy with the Shadow Positioning Sensors (SPS). This subsystem is located in the CSC, where eight Silicon PhotoMultiplier (SiPM) sensors are placed in a circle of 55 mm radius centred around the coronagraph instrument entrance pupil. Every photodiode of the SPS subsystem shall measure the intensity of the penumbral light generated by the OSC projected onto the CSC pupil plane. The irradiance measured by each sensor is a function of the alignment of the SiPM with respect to the Sun's direction, where the amount of light depends on the effective cross-section of the SPS aperture.

Combining the raw digital measures of the SiPM, it is possible to measure the absolute position coordinates of the centre of the entrance pupil with respect to the centre of the umbra profile generated by the occulter disk [9].

From the formation flying analysis in Section 1.6, the inter-satellite distance was obtained as a function of the relative distance of the Sun and Earth (r_{ES}). In Figure 2.1, it is possible to observe the consequences of this ISD that projects an umbra/penumbra profile over the coronagraph pupil plane with a radius R_{shad} given by:

$$R_{shad} = R_D - ISD \cdot \tan(\theta_{Sun}) = 71 \text{ cm} - ISD \cdot \tan\left(\frac{R_{sun}}{r_{ES}}\right) = 38.43 \text{ mm} \sim 38.5 \text{ mm}$$

The SPS distance from the pupil plane centre r_{SPS} has been designed taking into account the ISD and the R_{shad} . Since these sensors work by acquiring the penumbra light, the SPS shall avoid the condition where two sensors are in complete shadow. Therefore, computing the interception between the shadow circumference (shifted concerning the pupil plane origin) and the SPS circumference, it is possible to establish this distance as:

$$d = \frac{1}{2} \left(r_{SPS} \pm \sqrt{2R_{shad}^2 - r_{SPS}^2} \right)$$

Additionally, this distance provides the minimum value for the r_{SPS} to avoid the unwanted situation that two SPS fall at the same time in the umbra, i.e:

$$2R_{shad}^2 - r_{SPS}^2 < 0 \rightarrow r_{SPS} > \sqrt{2}R_{shad} \simeq 54.4 \text{ mm}$$

Therefore, the proposed distance of $r_{SPS} = 55 \text{ mm}$ allows the fulfilment of this requirement, with the 8 SiPM spaced by an angular distance of 45° from each other.

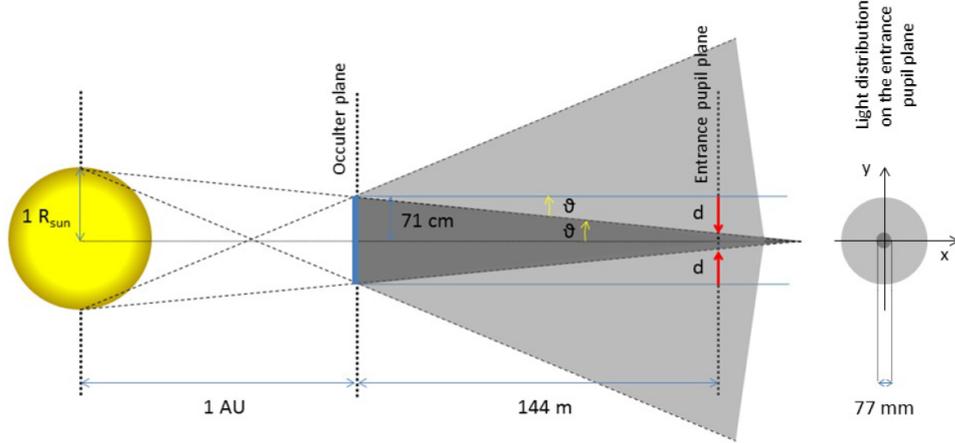


Figure 2.1: Distribution of umbra/penumbra projected over the CSC pupil plane [13]

2.1 SiPM sensors

The eight Silicon Photo Multiplier sensors that acquire the arriving solar irradiance are capable of quantifying low-light signals down to the single-photon level using a solid-state sensor. The chosen 3 by 3 mm square SiPM, made by SensL, provides outstanding uniformity of response in low voltage operations as well as mechanical robustness and magnetic field insensitivity.

Each detector is constituted by an array of 4774 (77×62) $35 \mu\text{m}$ resistor-coupled diodes providing improved robustness to radiation-induced damage with respect to a classical single larger area photodiode. The sensor is operated in “zero bias mode” with the anode directly tied to ground, and the diodes readings are averaged onboard, providing a single output value. In this configuration, the sensor has a lower responsivity, but its responsiveness is much more linear as a function of the incoming light. An optical band-pass filter coating has been deposited on the entrance window of each SiPM to reduce the sensor response to the wavelength. Finally, since the response of this device varies with the temperature, an active thermal monitor has been adopted onboard.

2.2 SPS requirements

The design of the SPS subsystem shall be compliant with the ESA defined requirements concerning the accuracy and pointing errors. Hence, the system shall deliver information of the centre of the umbra profile produced by the lateral and longitudinal position of the OSC relative to the entrance pupil origin plane with distinct accuracies given by [13]:

- **“SPS performance requirement:** The Shadow Position Sensors (SPS) shall be used to verify that the Coronagraph Instrument’s entrance pupil is centred within the umbra cone of the Occulter Disk. At the ISD and within ± 10 mm of the ideal position in lateral and ± 100 mm in range, the SPS shall have a lateral measurement accuracy of $50 \mu\text{m}$ (3σ) in each axis and longitudinal measurement accuracy of 1 mm (3σ). These accuracies are with respect to the axis connecting the centre of the occulter with the centre of the Sun.”
- **“SPS performance goal:** The SPS should be able to return a 3D relative position measurement at reduced performance within a range of ± 50 mm in lateral and ± 500 mm in longitudinal (i.e. the SPS should always return a 3D measurement within a box of 100 mm in width and height and 1000 mm in depth, centred on the ideal position).”

These requirements allow the definition of the **“requirement box”** as a region of 20 mm x 20 mm x 200 mm centred on the umbra profile, and the loosened **“goal box”** of 100 mm x 100 mm x 1000 mm. Since the requirements shall be satisfied in these volumes, it is possible to define the sensitivities and the dynamic range of the readout electronics of the SPS subsystem. The determination of the minimum and maximum current density change shall be considered in the worst cases (SiPM in umbra and full Sun, respectively) for a required displacement of $50 \mu\text{m}$ transversal or 1 mm longitudinal [9].

The SPS readings come from the irradiance measurements of the penumbra profile. Therefore, these depend on the light diffracted from the occulter edge, the limb darkening of the Sun, the presence of Sunspots and the ISD (these effects are briefly explained in the following chapter). Hence, from the expected penumbra profile at the nominal ISD, it is possible to define the SPS requirements of the irradiance range and sensitivity.

From Figure 2.2 it is possible to observe an illustration of the requirement and goal box (not to scale) alongside the expected umbra profile in the CI pupil plane. It is evident how the goal box is considerably larger, and the SPS shall gather light from an umbra that changes size as the ISD varies.

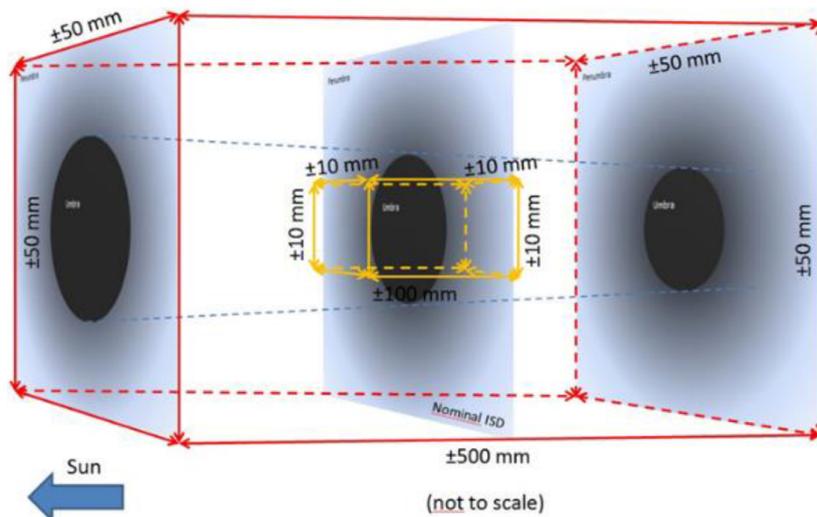


Figure 2.2: Requirement (yellow) and goal (red) box representation [2]

2.3 SPS design

The Shadow Positioning Sensor readout electronics are constrained by the sensitivity and dynamic range derived from the requirements, for this reason a double stage design has been adopted. The first stage allows covering the entire “goal box” using a transimpedance amplifier that transforms current into voltage. The second achieves a higher sensitivity and resolution for the lower signals in the “requirement box”. A pinhole of 2.5 mm is positioned in front of each SPS allowing a definite irradiance I to generate a proportional current C . Combined with the requirements, these values acknowledge the definition of the non-inverting amplification parameters of the electronic circuit.

The readings are transformed from current C to voltage V , and finally, through a 12-bit serial ADC (ADC128S102, working at 4.16 MHz) that samples the signals at a rate of 32.5 ksp/s, it is converted in digital reading. Then, an FPGA operates a running average (256 samples) on the digitized data and provides the appropriate value of the reading (Low Gain (LG) or High Gain (HG)) to be used in the position determination algorithm at a frequency of 2 Hz. Therefore, the algorithm shall be sufficiently fast to properly handle the data stream coming from the two chains (LG and HG) of each SPSs.

Figure 2.3 is a graphical representation of the logic of measurements in the SPS design and the calculated electrical parameters. The input comes from the solar irradiance L (mW) that is captured by the SPS. The responsivity of the sensor transforms this irradiance into current C (μA), which is then passed to

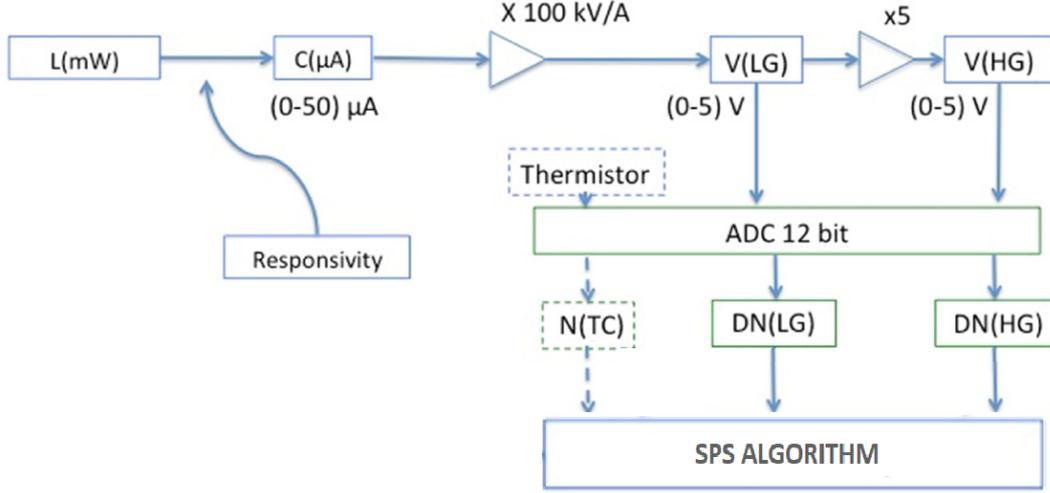


Figure 2.3: SPS logic measurements design [13]

the first amplification stage, providing the voltage $V(LG)$. Furthermore, a second amplification stage delivers the voltage $V(HG)$ which will be digitalized by the ADC collectively with the LG value. Ultimately, the SPS temperature readings and the digital “irradiance” value (DN) provided by the chain will serve as inputs to the SPS algorithm executed onboard the PROBA-3 missions.

It is possible to comprehend that the chain depends on the responsivity of the sensor R (mW/cm^2), where the factor R is reduced by both the filtering effect of the SPS radiation-hardened windows transmissivity and the coating applied on this window that cuts off the radiations above $\sim 600\text{ nm}^1$. The ADC converter consists of three inputs and outputs, where the LG and HG values become the final product of the chain and the input for the SPS algorithm. Furthermore, a thermistor monitors the sensor temperature in order to validate the SPS readings and for calibration purposes [13] [9].

The SPS subsystem has been tested using simulations and in the laboratory. An evaluation board equipped with a single sensor and an entire board demonstration model with a FPGA interface have been developed.

After the mission Critical Design Review (CDR), an advanced demonstration model was developed for engineering qualifications and flight model test campaigns. Lastly, an Engineering Qualification Model (SPS-EQM) allowed electrical and mechanical conformity verification, functional operations, and thermo-vacuum and vibration environment testing.

¹Above the 600 nm wavelength the responsivity of the SiPM is highly variable with temperature

Chapter 3

Fine positioning algorithm for formation flight

A fundamental component of the metrology chain is the fine positioning algorithm that uses the Shadow Positioning Sensors. These algorithms have been developed by INAF, where the position of the centre of the artificial eclipse with respect to the centre of the Coronagraph Instrument (CI) pupil is calculated. The primary algorithm is the pseudo-paraboloidal, where a third-order equation is inverted and analytically solved. A linear algorithm is also implemented to increase the precision, where irradiance values obtained from the SPS are deducted into the lateral position of the umbra using a linear equation.

These algorithms are based on the knowledge of the theoretical illumination profile specifically developed for this mission. The measured level of illumination by the single SPS inside the penumbra is related to the fraction of the solar disk perceived at a certain position, going from no illumination inside the umbra up to the full glare of the Sun outside the eclipse. In order to satisfy the requirements, this irradiance profile shall be known with very high accuracy (on the order of 0.5% to 0.05%) [2], including the effects of limb darkening, the transit of sunspots and the seasonal variations related to the Earth-Sun distance. This chapter presents an overview of the illumination profile determination and a description of the algorithms. However, for a more in-depth comprehension [2], [7], [11] and [13] shall be examined.

3.1 Penumbra Profile

The real irradiance profile has not been correctly measured, and this is one of the main objectives that shall be investigated in detail by the PROBA-3 mission. To develop the SPS algorithms, an analysis of the fraction of solar disk emerging from the occulter (source of penumbral light) is performed. This can be determined from the geometrical area of the fraction of the disk that is not covered by the occulter

and is seen by the SPS.

As described in the previous chapter, the output of the SPS is a current proportional to the incoming solar flux (or spectral irradiance) and is a function of the average disk intensity, the area of the solar disk and the total amount of radiation emitted by the Sun towards the collecting surface [2].

3.1.1 Corrections to the irradiance curve

Limb darkening

The incoming radiation is not uniform because of the limb darkening in the visible spectrum. This phenomenon can be encountered where the SPS, positioned near the edge of the umbra zone, “sees” the light of a thin crescent-shaped fraction of the solar disk emerging behind the OSC. This fraction of the disk is located at the solar limb, where the emission is reduced by a factor of about 2.7 [2] with respect to the emission coming from the centre of the solar disk. Therefore, the irradiance estimated in the penumbra, by assuming a uniform distribution, is significantly overestimated.

It is possible to analytically assess the variation of the intensity with the help of the limb darkening coefficients u_λ and v_λ . These are adimensional values that depend on the wavelength, and it is possible to demonstrate that the solar disk is more than a factor 2 fainter at the limb and a more than factor 1.2 brighter at the centre concerning the average disk intensity [2]. To evaluate the average solar radiation coming from the portion detected at the single SPS as the output current response R_{SPS} (A/m^2), it is necessary to numerically integrate over all the wavelengths present in the penumbra (200 to 1100 nm), taking into account the SPS responsivity $\epsilon_{SPS}(\lambda)$ and the solid angle Ω_{SPS} subtended by the fraction of the disk seen in the penumbra. Where:

$$R_{SPS} = \int_{\Omega_{SPS}} \int_0^{+\infty} I_\lambda(\theta) \epsilon_{SPS}(\lambda) d\lambda d\omega$$

$$R_{SPS} = \int_{\Omega_{SPS}} \int_0^{+\infty} I_{\lambda_0} (1 - u_\lambda - v_\lambda + u \cos \theta + v \cos^2 \theta) \epsilon_{SPS}(\lambda) d\lambda d\omega$$

With $I_\lambda(\theta)$ the disk intensity at wavelength λ at an angle θ from the disk and I_{λ_0} the disk intensity at wavelength λ at the disk centre.

This operation is simplified by the weak dependence on λ of the limb darkening coefficients over the SPS band-pass interval, as seen in Figure 3.1. Thus the integrals can be decoupled and the constant values u_0 and v_0 , obtained as the integral of u_λ , v_λ and ϵ_λ over λ , are used. Integrating the expression over the absorbed wavelengths, it was possible to determine that this approximation overestimates the irradiance by almost 2%; thus, the value must be corrected by the factor $K_i = 0.98$.

Earth-Sun distance

Another correction factor is used to adjust the irradiance continuous variation throughout the year, with $K_\gamma(t) = 1.0332$ at perihelium (early January) and $K_\gamma(t) = 0.9664$ at aphelion. Thus, it is necessary to continuously correct the expected irradiance curve to maintain the necessary accuracy.

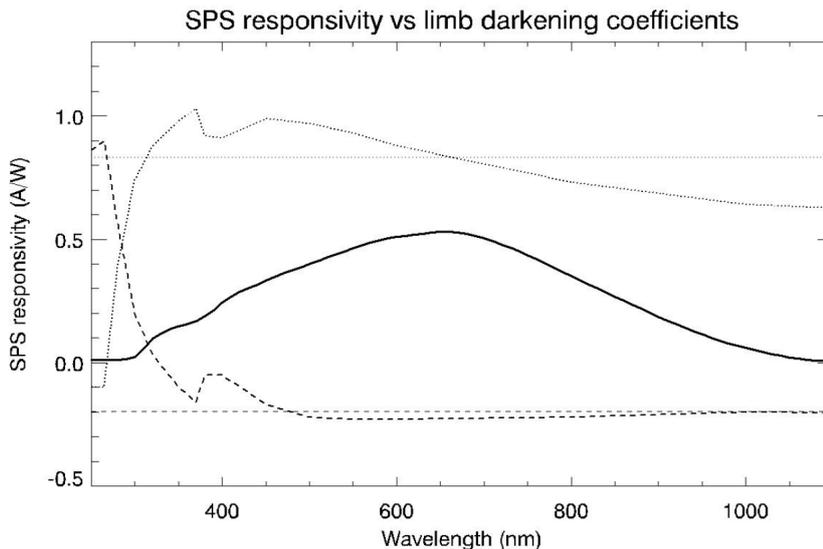


Figure 3.1: SPS responsivity (solid line) and the variation as a function of wavelength of the limb darkening coefficients u_λ (dotted line) and v_λ (dashed line) [2]

Physical dimensions of the sensor

Furthermore, an additional outcome that influences the output of the SPS is the smoothing effect due to the diode physical dimensions. The three millimetre side of the sensor leads to a variation of the illumination curve's slope, where the SPS close to the edge of the penumbra shall take this into account.

Effects of sunspots

A further reduction of the expected irradiance arrives when a sunspot group is positioned at the limb, where the photospheric emission coming from sunspots is lessened (down to 5-25% and 65-85% of the quiet Sun photospheric light in the sunspot umbra and penumbra respectively) [2]. Therefore, the expected SPS illumination shall depend on the sunspot size and location transiting the limb. The ASPIICS team has performed tests with real data to demonstrate that the variation of the expected irradiance curve (a decrease up to 3%) is more significant than the sensitivity demanded by the lateral displacement requirements.

This phenomenon is considered in the theoretical illumination profile development. Nevertheless, on flight measures will be needed to better assess the real contribution of sunspots.

3.1.2 Expected irradiance curve

Once the aforementioned factors have been considered in developing the theoretical illumination profile, the resulting SPS output response currents can be evaluated.

Assuming that both SC are perfectly aligned, the illumination in the penumbra is determined and displayed in Figure 3.2. The addition of the limb darkening effect corresponds to a considerable reduction (by almost a factor of 2) in the illumination level of the SPS. Whereas the presence of a sunspot further decreases, by roughly 10%, the expected current. Furthermore, it is possible to observe that the expected SPS current inside the umbra (lower than 40 mm) is null; this is an essential consideration for the achieved results in the following chapter.

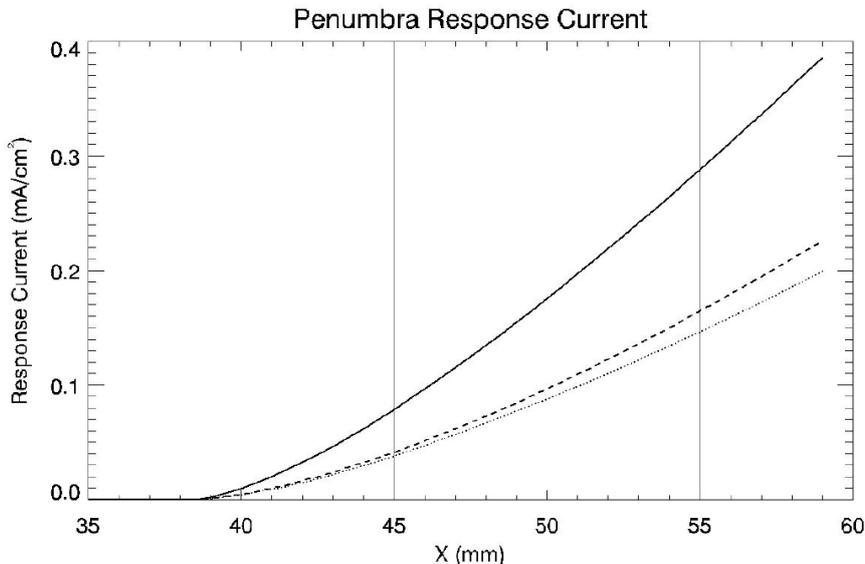


Figure 3.2: Level of expected SPS response current in the penumbra on the entrance pupil plane as a function of the distance from the telescope optical axis around the SPS location radius at 55 mm (vertical solid line). Neglecting the limb darkening (solid line), including the limb darkening (dashed line) and adding to the limb darkening a sunspot located at the limb (dotted line). [2]

In order to develop a successful algorithm, the required sensitivity to detect the minimum lateral and longitudinal displacements were identified based on numerical tests on the penumbra illumination curve. These have been computed as the relative difference in the response current curve once a minimum displacement by $50\ \mu\text{m}$ and by $1\ \text{mm}$ occurs along the lateral and longitudinal axis, corresponding to a sensitivity of 0.45% and a 0.048% respectively. Thus, the longitudinal requirement is more challenging; this leads to a thorough study of the variation of the irradiance with the ISD and the development of a model to determine the lateral displacement from the differences in the obtained current levels.

The resulting SPS response profile shown in Figure 3.3 has a shape similar to a rotational paraboloid, where an explicit function that can be inverted is needed to solve for the position given the irradiance. The determination of this function is a complex task that can be solved using a fitting for the irradiance surface. Thus, using a second-order polynomial fitting, the curve is reproduced with an error smaller than 3.5% . In contrast, a third-order fitting leads to a maximum error of 0.5% , that is therefore compatible to the sensitivity required to detect the $50\ \mu\text{m}$ lateral displacement.

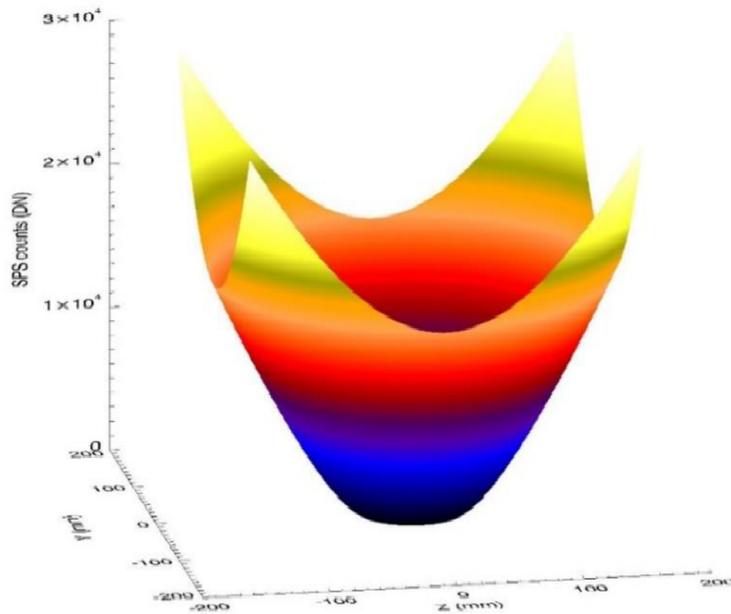


Figure 3.3: Digitized penumbra profile (DN) [11]

3.2 Pseudo - Paraboloidal algorithm

Once the penumbra profile has been accurately defined, an algorithm that uses the fitted irradiance (output current curve of the SPS) was developed to provide the occulter's absolute position with respect to the CI pupil centre. This fitting is done by a three-dimensional pseudo-paraboloidal function that relates the signal measured at the single SPS location (with coordinates x_i y_i z_i in the pupil plane) with the occulter centre, fulfilling the 50 μm minimum lateral displacement requirement and achieving an error lower than 0.5%. A system of four equations and four unknowns shall be solved for each measure, obtaining the location of each SPS and the desired centre of the penumbra, corresponding to the x_0 y_0 z_0 coordinates and the θ tilt angle of the occulter.

In general, an Nth-order pseudo-paraboloidal centred at the point (x_0, y_0) is described by the following equation:

$$R = R_0 + \sum_{n=1}^N \left(\frac{|x - x_0|^n}{a_n^n} + \frac{|y - y_0|^n}{b_n^n} \right)$$

Assuming no tilt angle between the SCs ($\theta = 0$), the oblateness coefficients a_n and b_n are equal. Thus, the third-degree pseudo-paraboloidal function is given by [13]:

$$R = R_0(z) + \frac{|x - x_0| + |y - y_0|}{a} + \frac{(x - x_0)^2 + (y - y_0)^2}{b^2} + \frac{|x - x_0|^3 + |y - y_0|^3}{c^3}$$

Where R (mW/cm^2 or DN) is the irradiance measured by the single SPS; x and y (cm) are equivalent to the x_i and y_i previously described; a, b and c are constant reconfigurable fitting parameters (from ground and calibration measurements), and R_0 is the irradiance at the pseudo-paraboloid vertex as a function of the ISD.

Hence, for the nominal and redundant SPS configuration, the positions of the sensors shown in Figure 3.4 together with the irradiance readings, it is possible to determine the umbra centre coordinates (x_0 and y_0).

Inverting the last equation and substituting known values for the nominal case:

$$\begin{cases} x_0^3 + c^3 \left(\frac{1}{a} + \frac{2 r_{SPS}}{b^2} + \frac{3 r_{SPS}^2}{c^3} \right) x_0 + (R_1 - R_5) \frac{c^3}{2} = 0 \\ y_0^3 + c^3 \left(\frac{1}{a} + \frac{2 r_{SPS}}{b^2} + \frac{3 r_{SPS}^2}{c^3} \right) y_0 + (R_3 - R_7) \frac{c^3}{2} = 0 \end{cases}$$

Where R_1, R_3, R_5 and R_7 are the readings of the nominal SPS set (see Figure 3.4). Substituting the constants with $A = c^3 \left(\frac{1}{a} + \frac{2 r_{SPS}}{b^2} + \frac{3 r_{SPS}^2}{c^3} \right)$, it is possible

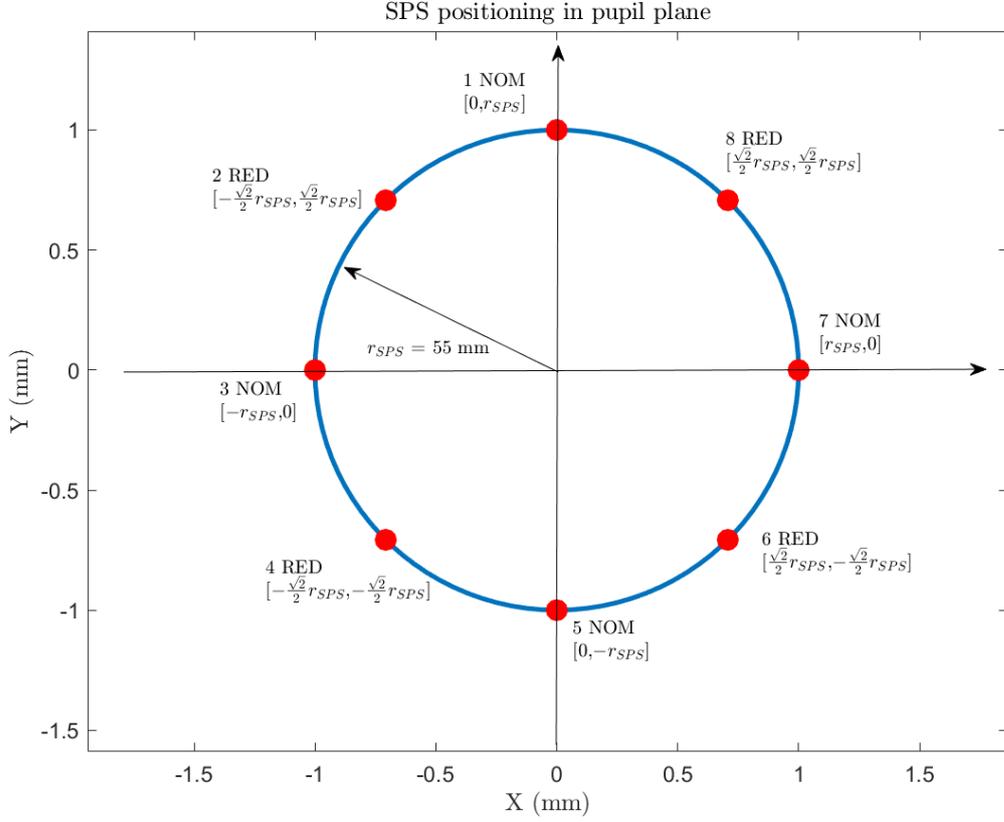


Figure 3.4: SPS position around the CI pupil

to obtain a simpler equation given as:

$$v^3 + Av + \Delta R = 0$$

$$\Delta R = \begin{cases} (R_1 - R_5) \frac{c^3}{2} & \text{for } v = x_0 \\ (R_3 - R_7) \frac{c^3}{2} & \text{for } v = y_0 \end{cases}$$

This third order equation has three roots, but only one of them allows a solution for $v = 0$ when the change in irradiance $\Delta R = 0$. Therefore this solution will be

considered as the only real solution for the pseudo-paraboloid centre, that is:

$$\begin{cases} x_0 = 2\sqrt{-\frac{A}{3}} \cos \left\{ \frac{1}{3} \left[\arccos \left(\frac{(R_5 - R_1) c^3}{4\sqrt{-\left(\frac{A}{3}\right)^3}} \right) + 4\pi \right] \right\} \\ y_0 = 2\sqrt{-\frac{A}{3}} \cos \left\{ \frac{1}{3} \left[\arccos \left(\frac{(R_7 - R_3) c^3}{4\sqrt{-\left(\frac{A}{3}\right)^3}} \right) + 4\pi \right] \right\} \end{cases}$$

Resolving these equations, the absolute measurement of the penumbra centre in the pupil plane is obtained. Looking at the pseudo-paraboloid equation, only the irradiance values $R_0(z)$ are needed to determine the remaining longitudinal position z_0 (normal to the pupil plane), where the average is computed as:

$$R_{0_i} = R_i - \frac{|x_i - x_0| + |y_i - y_0|}{a} - \frac{(x_i - x_0)^2 + (y_i - y_0)^2}{b^2} - \frac{|x_i - x_0|^3 + |y_i - y_0|^3}{c^3}$$

Neglecting possible tilt between the two SC, each SPS shall provide an independent measurement of $R(0)$; therefore, R_0 can be computed averaging the four measurements:

$$R_0 = \frac{1}{4} (R_{0_1} + R_{0_3} + R_{0_5} + R_{0_7})$$

The same procedure is valid for the redundant set, changing $i_{NOM} = [1,3,5,7]$ with $i_{RED} = [2,4,6,8]$. Once the irradiance at the vertex is computed, an approximation of the pseudo-paraboloid translated along the normal at the pupil plane (z-direction) is made with a linear fitting that allows the determination of the z_0 coordinate with the constant values H and K derived from the fitting:

$$z_0 = K + H \cdot R_0$$

Therefore, using constant parameters (the reconfigurable parameters of the fittings a , b , c , K and H are subject to calibration during the mission) and the SPS readings, it is possible to define the position of the umbra centre in the pupil-centred coordinate system that allows the precise determination of the absolute ISD and deviations between the two SC, closing the metrology chain.

This algorithm will be implemented onboard the satellite. The readings from the SPSs will be provided in Digital Number (DN) instead of mW/cm^2 by the readout electronic briefly described in Section 2.3. Hence, a positive integer from 0 to 4095 is expected from the Low Gain (LG) or High Gain (HG) digital readings. The advantage of this variation lies in the possibility of improving the accuracy in the requirement box using the HG readings and the fitting coefficients optimized for a fine fit. In contrast, the LG values are used together with the coarse fitting parameters in the goal box.

Expected errors

One of the fundamental parameters in the development of the algorithm is the lateral position error inside the requirement box. In Figure 3.5 the expected lateral error for the pseudo-paraboloidal algorithm is presented. This surface derives from the evaluation of the algorithm response using a mesh of equally distributed points inside the ± 10 mm requirement box. The graph shows the discrepancy between the reconstructed position and the real position of the umbra centre.

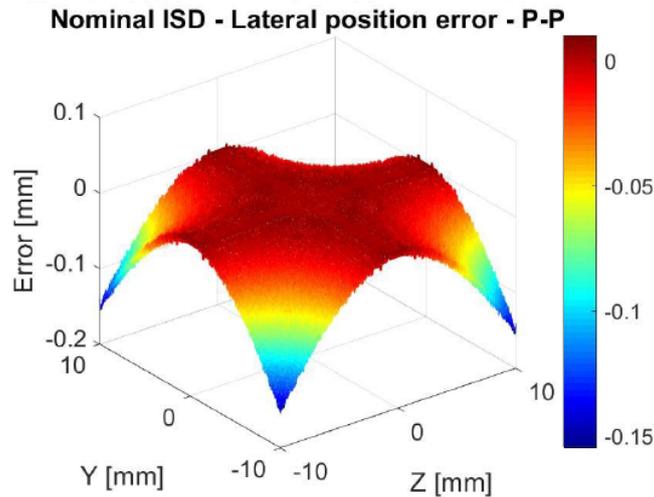


Figure 3.5: Pseudo-paraboloid lateral error inside the requirement box [7]

As can be seen, the error is not uniform, with a central cross that achieves a higher accuracy with respect to the outer edges of the box. The error goes from almost 0 mm down to -0.15 mm; therefore, the $50 \mu\text{m}$ requirement is not fully achieved solely with this algorithm. It is also possible to observe the paraboloid shape of these negative errors as they approach the outer edges. Further analysis of the lateral and longitudinal error of this algorithm are presented in Chapter 5.

3.3 Linear algorithm

A simpler yet accurate alternative to the pseudo-paraboloidal algorithm involves the determination of the absolute position of the vertex in the X-Y pupil plane using a linear correlation. The projected umbra centre is a function of the difference of irradiance measurements done by two opposite SPS sensors [13]. Differently from the pseudo-paraboloidal, this algorithm is perfectly axis symmetrical in the penumbra profile, therefore, the same behaviour is expected for a change in either

of the plane axis. The coordinates are given as:

$$\begin{cases} x_0 = \frac{R_5 - R_1}{d_x} \\ y_0 = \frac{R_7 - R_3}{d_y} \end{cases}$$

The coefficients d_x and d_y are determined on ground and are reconfigurable (for instance, after in-flight calibration activities) as the pseudo-paraboloidal parameters. Furthermore, these coefficients are equal when the occulter has no tilt angle (assumed perfectly circular). The determination of the longitudinal position z_0 is done with an analogous procedure to the previous algorithm.

This algorithm permits a lower cost computation and a similar accuracy compared to the pseudo-paraboloidal. Additionally, the umbra zones that are more prone to give more significant errors in one algorithm are the lowest in the other; thus, combining both methods increases the accuracy of the position in the pupil plane considerably.

Expected errors

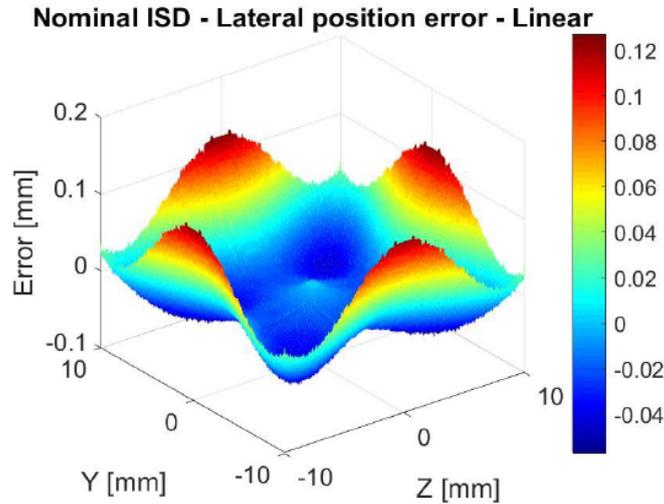


Figure 3.6: Linear algorithm lateral error inside the requirement box [7]

Similarly to the pseudo-paraboloidal lateral error, the linear algorithm achieves a maximum error in the order of 0.1 mm inside the requirement box. However, this error distribution determines an increased accuracy on the borders, whereas the outer edges remain at an acceptable 0.03 mm. Furthermore, the sign of the lateral position error of the linear algorithm is positive, while the pseudo paraboloidal is

negative. This is an introductory remark since the algorithm onboard the ESA PROBA-3 mission uses a combination of both results to determine the umbra centre position in the CI pupil plane.

3.4 Pseudo-paraboloid + Linear algorithm (PPL)

As previously affirmed, the magnitude and distribution of the lateral error are of great significance for the SPS subsystem since they will be used to calibrate the other systems in the metrology chain. Therefore, by merging the pseudo-paraboloidal and linear algorithm results, it is possible to satisfy the stringent requirements for lateral displacement, maintaining a reduced computational cost. This is performed applying a simple average of the results as:

$$\begin{cases} x_0 = \frac{x_{0_{PP}} + x_{0_{Lin}}}{2} \\ y_0 = \frac{y_{0_{PP}} + y_{0_{Lin}}}{2} \end{cases}$$

With the particularity that the most inaccurate zone of one algorithm is counter-balanced by the other, thanks to the opposite sign of the error, as can be seen in Figure 3.7.

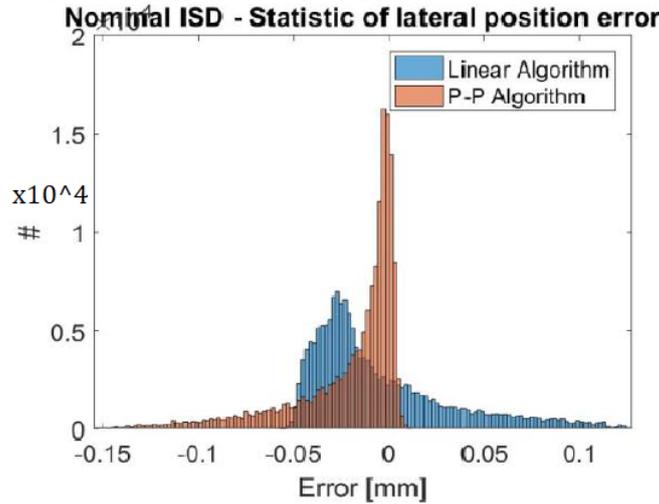


Figure 3.7: Histogram of statistical error of both algorithms [7]

Figure 3.8 exposes the combined algorithm error surface, with a shape that resembles the linear algorithm in the box sides and the pseudo-paraboloidal in the corners. The error remains in a range between $-60 \mu\text{m}$ and $+60 \mu\text{m}$; this is a

significant improvement, especially in the statistical distribution of the error, which is almost symmetrically distributed around the zero value. Without this union, neither of the algorithms is able to satisfy the $50 \mu\text{m}$ error requirement for lateral displacement.

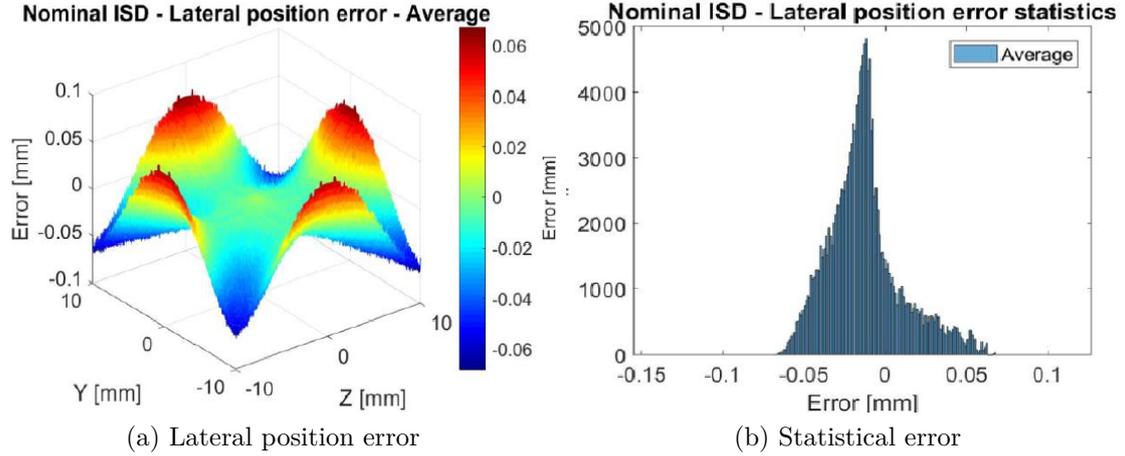


Figure 3.8: Error associated to the lateral position computation of the combined algorithm [7]

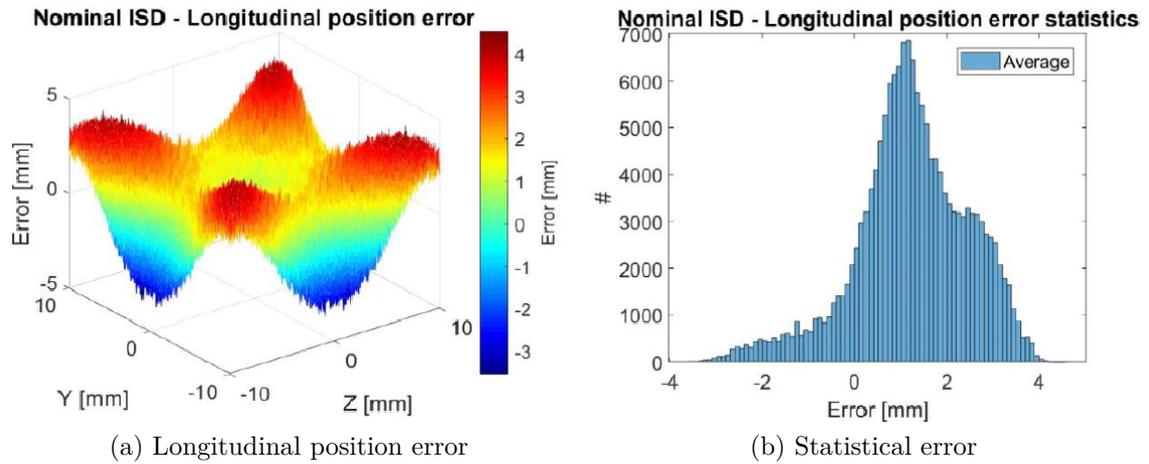


Figure 3.9: Error associated to the longitudinal position computation of the combined algorithm [7]

Finally, the longitudinal dispersion of the error, which shall be in the order of the millimetre from the requirement, is directly proportional to the lateral error; this comes as a consequence of the dependence of the longitudinal position from the lateral displacement. It is possible to observe from Figure 3.9 how the mean

statistical error remains in the 1 mm range, with peaks up to ± 4 mm at the box edges¹.

Table 3.1: Error of the average of algorithms at nominal ISD [5]

Position	Min [mm]	Mean & Std [mm]	Max [mm]
Requirement box			
Lateral	-0.066	5.626e-04 & 0.018	0.054
Longitudinal	-5.684	-0.393 & 1.704	5.990
Goal box			
Lateral	-1.013	2.595 & 2.187	7.007
Longitudinal	-834.366	-172.469 & 462.645	1465.995

3.5 Differential algorithm

An even simpler computation of the relative misalignment direction can be achieved using a differential algorithm. This procedure calculates the difference in irradiance of opposite SPS sensors in a given time for the nominal and redundant sets.

$$\text{Nominal set} \rightarrow \begin{cases} \delta_{51} = \frac{R_5 - R_1}{R_5 + R_1} \\ \delta_{37} = \frac{R_3 - R_7}{R_3 + R_7} \end{cases} \quad \text{Redundant set} \rightarrow \begin{cases} \delta_{62} = \frac{R_6 - R_2}{R_6 + R_2} \\ \delta_{48} = \frac{R_4 - R_8}{R_4 + R_8} \end{cases}$$

Where the δ indicates the distance between the pupil and the umbra centre, measured concerning the SPS position ($r_{SPS} = 55$ mm); furthermore, the sing of δ indicates the direction of the transversal misalignment of the plane. This method provides a qualitative estimation of the spacecraft's misalignment, but it can be useful in flight, especially in the first period of the mission when the calibration of the main algorithm needs to be optimised.

¹The shown graphs and results were obtained by M. Casti in [5] and [7] using an outdated set of fitting parameters. Thus, the results in the following chapters might differ from the ones exhibited.

Part II

**Alternative SPS Algorithm
& Simulator**

Chapter 4

Alternative SPS Algorithm

This thesis aims to develop an alternative algorithm that allows the determination of the umbra centre with respect to the coronagraph instrument (CI) entrance pupil in the SPS reference frame. The pseudo-paraboloidal algorithm previously analyzed fits the actual irradiance profile produced by the external occulter with a non-axis-symmetrical function; this is based on the analytical axis-symmetrical model of irradiance that is precomputed from a numerical simulation [2]. The following interpretation follows the document provided by Dr Raphaël Rougeot from ESA - ESTEC to INAF in April of 2016 [22], where the proposed Matlab codes will be modified and validated by the author.

The suggested alternative algorithm uses Cardano's Method to find the values of the centre of the penumbra, where the position is retrieved from the radius instead of the cartesian coordinates in the CI pupil plane. This approach is computationally expensive if compared with the pseudo-paraboloid; however, it is expected to achieve lower measurement errors and an axis-symmetrical profile. Consequently, this algorithm will not be implemented onboard the PROBA-3 SCs. Instead, it will be integrated into the processing pipeline used in the Scientific Operation Centre (SOC), located in the Liège Space Center (CSL), to validate the results coming from the flight (pseudo-paraboloidal + linear) algorithm. Therefore, the computational flow, flags, and inputs and outputs of this alternative algorithm shall maintain the same structure of the pseudo-paraboloid to seamlessly compute and compare the results.

4.1 An alternative model for irradiance

As it can be seen in Figure 4.1, the pseudo-paraboloid model has a non-cylindrical shape; consequently, it is not axis-symmetric. The shape of the penumbra is expected to be symmetric (if no critical tilt between the SC is assumed), so an approach that uses the distance r_i between the diode $i = A, B, C, D$ and the centre of the umbra,

located at the point $x_0 y_0$, is proposed by [22], where:

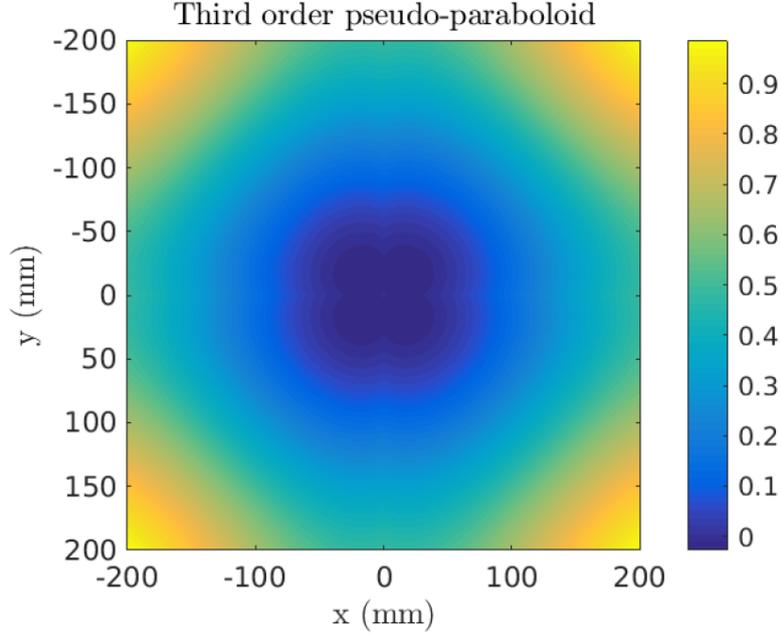


Figure 4.1: Pseudo-paraboloid penumbra profile (Irradiance in mW) [22]

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

$$R_i(x_i, y_i, z, x_0, y_0) = R_0(z) + \frac{r_i}{\alpha} + \frac{r_i^2}{\beta^2} + \frac{r_i^3}{\gamma^3}$$

R_i is the measured irradiance (in DN to keep the same structure as the pseudo-paraboloid algorithm) of a single SPS, and R_0 is the irradiance (DN) measured at a certain longitudinal position (the nominal ISD is considered). Together with the fitting constants α , β and γ , this value is derived from the refined penumbra profile. The same values obtained for the precomputed a , b and c for fine fitting of the pseudo-paraboloid penumbra are used for α , β and γ respectively. Using the following fine lateral parameters, the profile in Figure 4.2 is obtained:

- $\alpha = a = -3.6907368281401 \cdot 10^{-6} \text{ m}/DN$
- $\beta = b = 0.000437857008493415 \text{ m}^2/DN$
- $\gamma = c = -0.00350110848394894 \text{ m}^3/DN$
- $R_0 = 4034.634917033389 \text{ DN}$

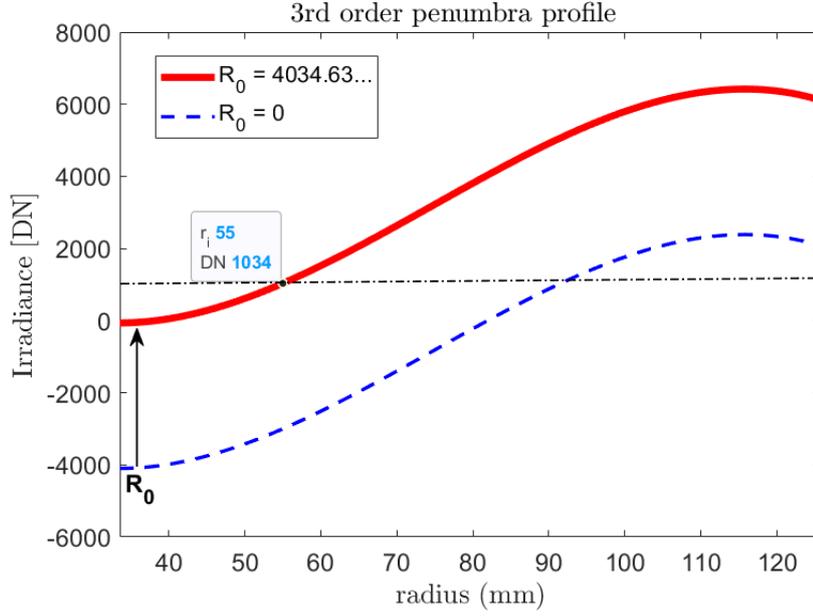


Figure 4.2: Penumbra profile: Third-order fine fitting curve

Value of R_0

The value of the constant R_0 is one of the main differences of the alternative algorithm with respect to the pseudo-paraboloidal. R_0 needs to be known to invert the equation that determines the lateral and longitudinal positions, whereas the value of R_0 used in the pseudo-paraboloid is determined from the longitudinal calculations as:

$$R_0 = \frac{1}{4} (R_{0A} + R_{0B} + R_{0C} + R_{0D})$$

Hence, R_0 is a known constant for the alternative algorithm, which is determined from the measured irradiance (in DN) at the nominal inter-satellite distance (around 145 m) with the centre of the penumbra aligned with the CI pupil ($z_0 = y_0 = 0$). In this ideal case, all the SPS shall measure the same irradiance of $R_{i_{ideal}} = 1034 \text{ DN}$ (longitudinal reconfigurable parameter L), determined as the “zero” position.

Evaluating the variation of the irradiance with the radius in Figure 4.2, it is possible to see that the blue dotted line that represents the equation with $R_0 = 0$ has a negative value for the zero position. Thus, finding the minimum of this equation and adding the ideal ISD value of irradiance, the value for R_0 is found:

$$R_{55} = R_0 + \frac{55\text{mm}}{\alpha} + \left(\frac{55\text{mm}}{\beta}\right)^2 + \left(\frac{55\text{mm}}{\gamma}\right)^3 = 1034 \text{ (Nominal ISD)}$$

$$R_0 = 1034 - \frac{55\text{mm}}{\alpha} + \left(\frac{55\text{mm}}{\beta}\right)^2 + \left(\frac{55\text{mm}}{\gamma}\right)^3 = \begin{cases} 4034.6349... \text{ DN (Fine)} \\ -758.2013... \text{ DN (Coarse)} \end{cases}$$

Therefore, the red curve is the fitted penumbra profile to be used, giving the same irradiance values as the PROBA-3 SPS simulator, verifying the outcomes. For the alternative algorithm simulator described in the following chapters, this system will be a function of the reconfigurable parameters, where the desired R_0 is a pre-calculated constant for the nominal ISD.

Further validation of the result comes from the precomputed parameters for the longitudinal measure, where the parameter “ $delta_R$ ” value subtracted to the calculated R_0 gives the exact value of irradiance expected at the nominal ISD. This variable is used by the onboard algorithm for the longitudinal calculation as:

$$R_{55} = R_0 - delta_R$$

Therefore, the onboard algorithm developed by INAF can determine the R_0 value needed for the alternative algorithm without the addition of new variables. However, the R_0 value calculated for the longitudinal is a function of the measured SPS irradiances. In contrast, the one implemented for the lateral position by the alternative algorithm employs the fitting reconfigurable parameters; thus, it will be constant, and it can be included as an input to avoid the determination of this value at each iteration.

The original version of the alternative algorithm employed only the fine parameters. Following the structure adopted for the pseudo-paraboloidal algorithm, the author extended the algorithm’s functionality, applying different fitting coefficients (coarse and fine) for different regions to improve the reliability of the computed solution.

4.1.1 Boundary conditions

In a preliminary approach, this algorithm will be used only for the four nominal SPS diodes, but a straightforward rotation of the coordinates allows the addition of the redundant set. The coordinates of the nominal set with respect to the centre of the pupil, with $r_{SPS} = 55 \text{ mm}$ are:

- Diode A: $x = +r_{SPS}, y = 0$ (SPS 7 in Fig 3.4)
- Diode B: $x = 0, y = +r_{SPS}$ (SPS 1 in Fig 3.4)
- Diode C: $x = -r_{SPS}, y = 0$ (SPS 3 in Fig 3.4)
- Diode D: $x = 0, y = -r_{SPS}$ (SPS 5 in Fig 3.4)

As stated in the SPS performance requirements (Section 2.2), a requirement box and a goal box are defined with different accuracies on the retrieving position. Only the lateral displacement is considered for this evaluation, with a goal box of $\pm 50 \text{ mm}$ and a requirement box of $\pm 10 \text{ mm}$. Thus, it is possible to establish that the

centre of the umbra resides within a box of 100 mm with respect to the centre of the CI entrance aperture. Considering the different locations of the four diodes, it is possible to determine the boundary conditions for the distance r_i , determined from the centre of each diode i :

$$\begin{aligned} (x_0, y_0) &\in [-50 \text{ mm} ; +50 \text{ mm}]^2 \\ r_{SPS} - 50 \text{ mm} &\leq r_i \leq r_{SPS} + 50 \text{ mm} \cdot \sqrt{2} \\ \implies r_i &\in [5 \text{ mm} ; 125.7 \text{ mm}] \end{aligned}$$

This value will help evaluate the results, where the calculated solution shall be inside this range to be acceptable. However, an adjustment is proposed in the next chapter to avoid the presence of more than one solution.

4.1.2 Inverting the system

From these conditions, it shall be possible to determine the location of the centre of the umbra with respect to the centre of the pupil while using this new axis-symmetric model to fit the real measurements of irradiance given by the four diodes (in DN).

$$R_A, R_B, R_C, R_D \rightarrow (x_0, y_0)$$

The mathematical approach is similar to the one used in the pseudo-paraboloid. First, it is necessary to know the irradiance at the four positions of the diodes A, B, C and D (SPS reading). Second, R_0 is assumed fixed¹ (thus the longitudinal position is fixed at each iteration) and that the pre-computed parameters α , β and γ are known [22].

4.1.3 Plot of the alternative penumbra profile

Consequently, once the system's boundary conditions are defined, it is possible to plot the penumbra profile for this alternative approach. In Figure 4.3 it is evident that the generated penumbra is axis-symmetrical (different from the one in Fig 4.1), satisfying the request for the development of an algorithm with a more accurate description of the penumbra profile. It is important to note that the central area of this profile shall have a higher irradiance count than the expected one; this is a result of the third-order interpolation of the penumbra profile that leads to an impossibility of interpreting the inner flat zone with null irradiance. However, in the proposed figure, this effect has been corrected in order to mimic the algorithm that

¹This assumption was tested with the different penumbra profiles generated with various inter-satellite distances to validate the algorithm thoroughly

uses the real irradiance and does not encounter this discrepancy since the measured value in this range shall be null.

The same profile can be plotted as a three-dimensional surface (Figure 4.4); this is possible as a consequence of the polar coordinates approach that is used for the axis-symmetrical solution of the system, using the irradiance as the revolution axis with the radius variation projected in the X and Y plane of the CI pupil entrance.

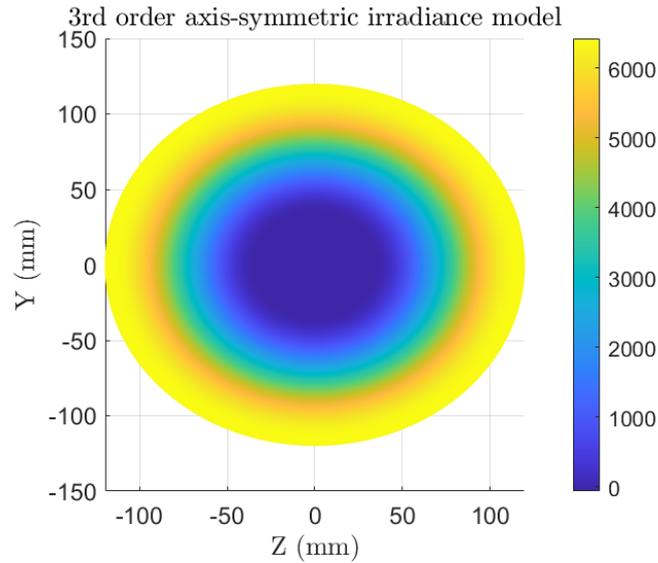


Figure 4.3: Alternative penumbra profile (Irradiance in DN)

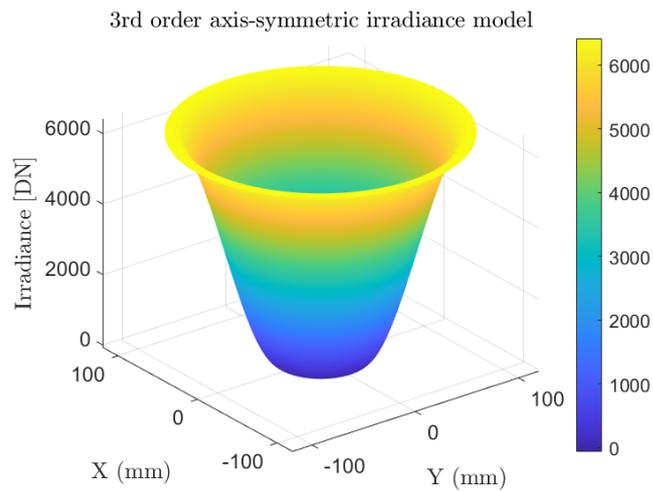


Figure 4.4: Alternative 3D penumbra profile

4.1.4 Cardano’s Method - Deriving the radial distance

Once the equation that models the irradiance is known, the third-order polynomial shall be inverted, deriving the r_i from the R_i . Thus, given the SPS irradiance R_i , only a real positive root for the third-order equation is expected (only one solution for r_i) due to the axial-symmetric profile of the penumbra. To achieve this, the dedicated Cardano’s Method is used to compute the solution of:

$$R_i = R_0 + \frac{r_i}{\alpha} + \frac{r_i^2}{\beta^2} + \frac{r_i^3}{\gamma^3}$$

Cardano’s Method

The Cardano’s Method (or formula) [4] is a mean of solving the real root of a cubic equation of the form (similar to the previous R_i expression):

$$ax^3 + bx^2 + cx + d = 0$$

This is possible using two steps: first it is necessary to “depress” the cubic equation changing the unknown to $x = y - \frac{b}{3a}$. Replacing in the main equation:

$$\begin{aligned} a \left(y - \frac{b}{3a} \right)^3 + b \left(y - \frac{b}{3a} \right)^2 + c \left(y - \frac{b}{3a} \right) + d &= 0 \\ ay^3 - by^2 + \frac{b^2}{3a}y - \frac{b^3}{27a^2} + by^2 - \frac{2b^2}{3a}y + \frac{b^3}{9a^2} + cy - \frac{bc}{3a} + d &= 0 \\ ay^3 + \left(c - \frac{b^2}{3a} \right) y + \left(d + \frac{2b^3}{27a^2} - \frac{bc}{3a} \right) &= 0 \end{aligned}$$

Once the equation has been handled into a form without the y^2 term, it is possible to solve the depressed system:

$$y^3 + Ay = B$$

Substituting:

$$\begin{aligned} 3st &= A \\ s^3 - t^3 &= B \end{aligned}$$

Finally, it is possible to solve for:

$$y = s - t$$

Using the Cardano's formula, the real and complex roots of the cubic polynomial can be found as:

$$\begin{cases} x_1 = S + T - \frac{b}{3a} \\ x_2 = -\frac{S+T}{2} - \frac{b}{3a} + \frac{i\sqrt{3}}{2}(S-T) \\ x_3 = -\frac{S+T}{2} - \frac{b}{3a} - \frac{i\sqrt{3}}{2}(S-T) \end{cases}$$

Where:

$$\begin{cases} Q = \frac{3ac - b^2}{9a^2} \\ R = \frac{9abc - 27a^2d - 2b^3}{54a^3} \end{cases} \rightarrow \begin{cases} S = \sqrt[3]{R + \sqrt{Q^3 + R^2}} \\ T = \sqrt[3]{R - \sqrt{Q^3 + R^2}} \end{cases}$$

Solving the SPS system

Therefore, in order to solve the SPS system for r_i with the Cardano's Method:

$$\begin{aligned} \left(r_i + \frac{\gamma^3}{3\beta^2}\right)^3 + \left(r_i + \frac{\gamma^3}{3\beta^2}\right) \cdot \left(-\frac{\gamma^6}{3\beta^4} + \frac{\gamma}{\alpha}\right) + \frac{\gamma^3}{27\beta^2} \left(\frac{2\gamma^6}{\beta^4} - \frac{9\gamma^3}{\alpha}\right) + \gamma^3 \cdot (R_0 - R_A) = 0 \\ \implies \rho_i^3 + \rho_i \cdot p + q = 0 \end{aligned}$$

Where $\rho_i = r_i + \frac{\gamma^3}{3\beta^2}$ becomes the unknown to be solved, with $p = \left(-\frac{\gamma^6}{3\beta^4} + \frac{\gamma}{\alpha}\right)$ and $q = \frac{\gamma^3}{27\beta^2} \left(\frac{2\gamma^6}{\beta^4} - \frac{9\gamma^3}{\alpha}\right) + \gamma^3 \cdot (R_0 - R_A)$. It is important to note that the q term cannot be precomputed since it depends on the measured irradiance R_i at each iteration, increasing the computational cost of the algorithm.

The next step is the computation of the term Δ :

$$\Delta = -\left(4 \cdot p^3 + 27 \cdot q^2\right)$$

The expression of the three solutions for ρ_i (three real/complex numbers) depends on the sign of Δ . The real solutions are:

- $\Delta > 0$:

$$\rho_i = \sqrt[3]{-\frac{q - \sqrt{\frac{\Delta}{27}}}{2}} + \sqrt[3]{-\frac{q + \sqrt{\frac{\Delta}{27}}}{2}}$$

- $\Delta = 0$: Only one solution is positive, depending on $\frac{\gamma^3}{3\beta^2}$

$$\rho_i = \pm \frac{3q}{p}$$

- $\Delta < 0$:

$$\left\{ \begin{array}{l} \rho_i = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(-\frac{q}{2}\sqrt{\frac{27}{-p^3}}\right)\right) \\ \rho_i = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(-\frac{q}{2}\sqrt{\frac{27}{-p^3}}\right) + \frac{2\pi}{3}\right) \\ \rho_i = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(-\frac{q}{2}\sqrt{\frac{27}{-p^3}}\right) + \frac{4\pi}{3}\right) \end{array} \right.$$

Once ρ_i is calculated, it is possible to derive the desired SPS radius as:

$$r_i = \rho_i - \frac{\gamma^3}{3\beta^2}$$

This process needs to be done for each of the SPS readings, finding the $r_{A,B,C,D}$ for every position measure.

The last step consists of selecting the unique real positive solution which matches the boundary conditions previously discussed. A remark is needed since it is not mathematically proven that a solution that meets these requirements can be found or is unique. However, based on a physical interpretation, only one solution shall exist [22].

4.1.5 Deriving the position of the centre of the umbra

To obtain the position of the centre of the umbra (x_0, y_0) with respect to the centre of the CI pupil, the distance r_i from each diode (i=A,B,C and D) to this centre needs to be known. Once the r_i has been calculated using Cardano's Method, the following system is defined:

- Diode A: $r_A = \sqrt{(r_{SPS} - x_0)^2 + y_0^2}$
- Diode B: $r_B = \sqrt{x_0^2 + (r_{SPS} - y_0)^2}$
- Diode C: $r_C = \sqrt{(-r_{SPS} - x_0)^2 + y_0^2}$
- Diode D: $r_D = \sqrt{x_0^2 + (-r_{SPS} - y_0)^2}$

Thus the centre can be solved as:

$$\begin{cases} x_0 = \frac{r_C^2 - r_A^2}{4r_{SPS}} \\ y_0 = \frac{r_D^2 - r_B^2}{4r_{SPS}} \end{cases}$$

4.2 Matlab function for Cardano's Method

To achieve the desired implementation of this procedure, a Matlab code was developed by Rouget and Casti [22]. The Cardano's Method function, reported in Appendix D.1, takes as input the polynomial coefficients $[p0, p1, p2, p3]$, where $p0 = R_0 - R_i$, $p1 = 1/\alpha$, $p2 = 1/\beta^2$ and $p3 = 1/\gamma^3$.

These must be real coefficients to solve the polynomial. Hence, an error is issued if this condition is not satisfied, stopping the calculation. This control can be avoided to speed the algorithm since the only parameter that can vary is $p0$, but the subtraction of $R_0 - R_i$ expects a real (positive or negative) solution of two positive natural numbers (in DN). The remaining controls concern the reconfigurable fine or coarse parameters (interpolation of the penumbra); these are inputs of the algorithm from ground, so a previous validity check is expected.

As stated before, the the Cardano's parameters q and Δ cannot be precomputed; thus, they are determined for each repetition of the function. Therefore, a reduction of the computational load of the algorithm might be helpful in the performance comparison with the pseudo-paraboloidal.

A series of controls are executed to select the method for computing the three roots of the polynomial (given as $[z1, z2, z3]$). Four different cases are possible; thus, four flags are implemented to keep track of the solution employed. As it can be seen, the computed roots can be positive or negative real or complex numbers.

The solution r_i (a 1x3 vector) is given by:

$$r_i = [z1, z2, z3] - \frac{p2}{3 p3}$$

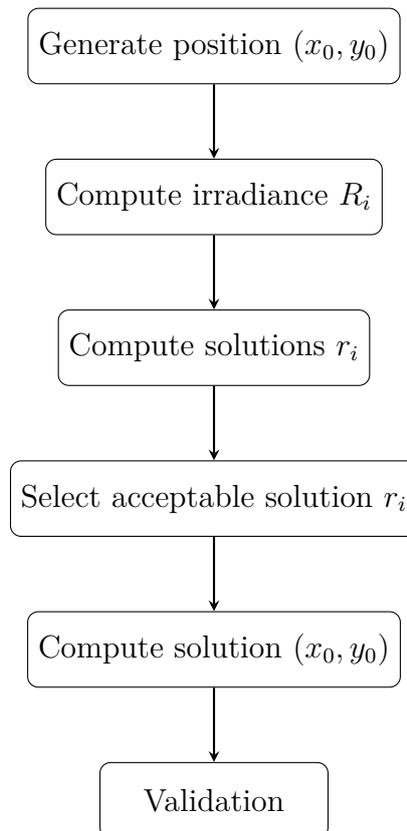
Solving for ρ_i as seen in the previous section. These vector needs to be accompanied by a boolean vector called "is_Real_Positive", where each root is evaluated. These vector elements are equal to one if and only if the imaginary part of the root is null and the real part is positive.

Finally, the output of this function is the solution 1x3 vector with the calculated roots, the boolean vector with the real positive control, the flag that indicates the used solution and the calculated Δ (this output can be omitted since it is not useful for the final algorithm).

Chapter 5

Local validation of the Alternative Algorithm

Once the Cardano's Method function has been defined, a preliminary numerical test needs to be performed in order to validate the analytical axis-symmetrical model for irradiance. This is possible using once again the proposed method in [22]:



This flow chart is performed with a Matlab script called “Alternative_SPS_Algorithm_Generate”, which will be explained in the following sections and can be found in Appendix D.2.

5.1 Variation of the coordinate system

In the prior paragraphs, the cartesian coordinate system centred in the coronagraph instrument pupil saw an X-Y reference plane for the lateral displacements and a Z-axis coming out of this plane for the longitudinal measure. However, this reference plane is used only to define the penumbra profile and the preliminary analysis.

An updated coordinate system that is used for the development of the algorithm and the simulator is presented. The system in Figure 5.1 diverges from the previous one in the definition of the Z-Y plane (with Z horizontal) as the plane containing the entrance pupil and the SPS sensors at the same centre, with a normal X-axis. Hence, the system is not right-handed, where the X and Z axis have been exchanged. This disagreement led to a change in the previously developed algorithms, and the author updated them to this new system. To summarise:

- Z-axis: Horizontal lateral measure in the CI pupil plane (Previously the X-axis).
- Y-axis: Vertical lateral measure in the CI pupil plane.
- X-axis: Longitudinal measure normal to the CI pupil plane (Previously the Z-axis).

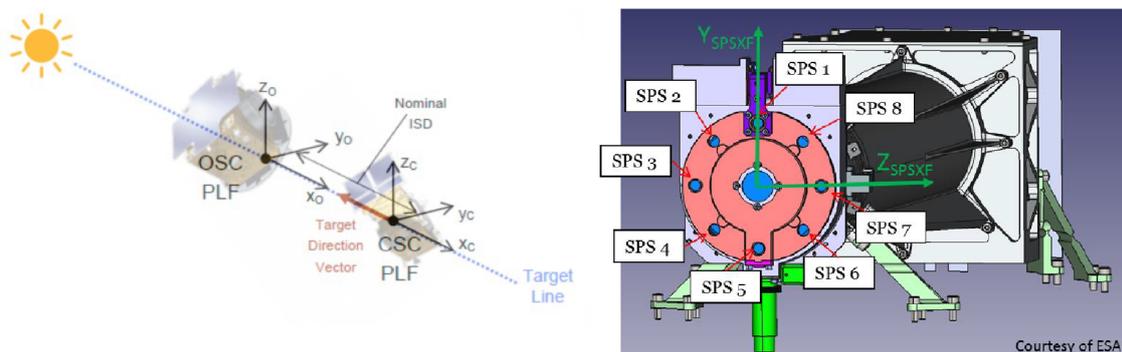


Figure 5.1: Updated coordinate system [13]

5.2 Locally Generated profile for validation

It is essential to note that the proposed method in [22] requires a model of irradiance that remains positive and monotone in the operational range. However, the profile proposed in [22] is not accurate since it uses a modified set of parameters (the absolute value of the outdated (2016) fitting parameters). Therefore, the proposed algorithm implements the same parameters as the pseudo-paraboloidal, where α and γ are negative, and R_0 is the one calculated in Section 4.1.

Plotting the fine and coarse penumbra profiles (Figure 5.2) in the operative range, it is possible to observe that there is a minimum and a maximum of these curves inside the initial boundary conditions given by the goal box of ± 50 mm (Figure 5.2a) and requirement box of ± 10 mm (Figure 5.2b).

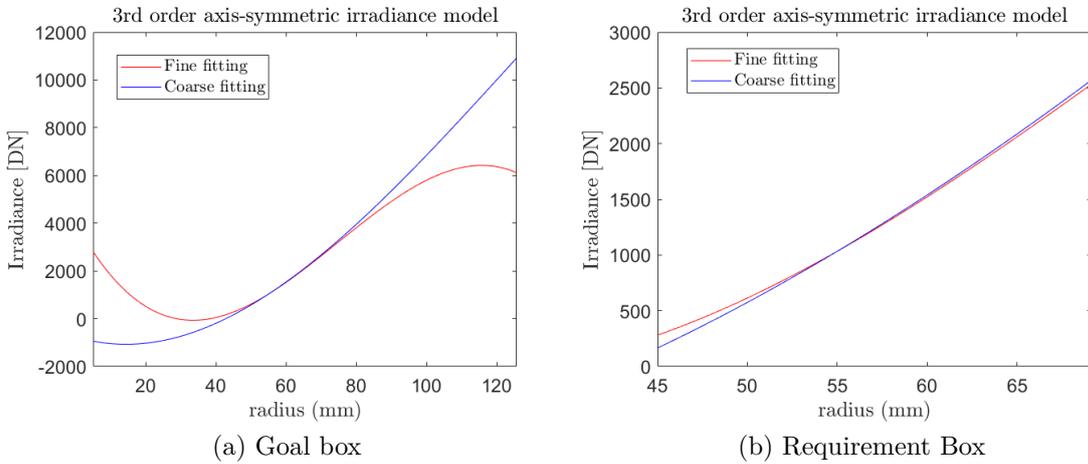


Figure 5.2: Coarse and fine penumbra profiles

The locally generated profile used for the validation of the algorithm uses both of these fitting parameters. The fine fitting reconfigurable parameters are used from the origin up to a decentring of the umbra profile of 13.8 mm with respect to the SC aligned position. Therefore, the requirement box is accurately approximated, whereas the coarse parameters cover the remaining of the goal box. An in-depth comparison of the alternative and pseudo-paraboloidal algorithms in the goal box using only the fine fitting parameters is described in Appendix A. If solely the fine parameters are employed, the illumination curve is correctly fitted; yet, the error approaching the edges increases rapidly, and only a range of ± 41 mm of the goal box can be covered without encountering more than one solution.

The functioning of the alternative algorithm with both the fine and coarse parameters is an addition made by the author to the script proposed by [22]. Using a method similar to the onboard algorithm, the coverage of the whole goal box is possible without encountering validity issues for the solution.

5.2.1 Modification of the boundary conditions

The presence of a minimum and maximum of the fine fitting curve inside the initial boundary conditions lead to a significant variation of the previously defined limits (applicable only to the “Generate algorithm” script), where the minimum and maximum boundaries shall be modified to avoid the presence of more than one solution using Cardano’s method. Consequently, if the only fine parameters are used for coordinates over the 13.8 mm threshold, the minimum limit for the r_i (employed to generate the irradiance measures) shall be increased from 5 mm to 33.5 mm (the minimum of Figure 5.2a curve, corresponding to the beginning of the umbra):

$$r_i \in [33.5 \text{ mm} ; 125.7 \text{ mm}]$$

This modification appears a significant adjustment, but it is only an approximation applicable to the algorithm that generates its own local SPS readings for the positions outside the requirement box (over 13.8 mm) using only the fine parameters (Matlab code in Appendix D.2.1). Instead, **if the fine + coarse parameters are used, the alternative algorithm maintains the default boundaries** of Section 4.1.1. This situation is not present in the “Read algorithm” script that uses the simulated values of the SPS to retrieve the position (the block “Compute the solution r_i ” of the flow diagram at the beginning of the chapter), since the readings expected in the 5 mm to 33.5 mm range shall be inside the umbra, with an irradiance value equal (or very close) to null. Additionally, a similar situation arises for the maximum boundary condition using the Cardano’s method with fine parameters, where the coordinates farther than 41 mm from the umbra centre generate more than one solution. However, the situation shall not be encountered since the coarse parameters are implemented in the goal box.

5.2.2 Generation of local position

The proposed algorithm in [22] generates random points inside the lateral range (using the “rand” Matlab function). An array of (z_0, y_0) points is generated in a ± 50 mm range in the “Umbra_centre” variable in the code; this allows a first evaluation of the algorithm inside the range. In addition, the author modified the code by implementing a mesh generation script to better sample the error. Initially, this algorithm will calculate only the points inside the requirement of ± 10 mm to avoid the multiple solution problems previously explained. Next, implementing the variation of the fitting parameters, it will be feasible to cover the entire ± 50 mm goal box with the alternative algorithm. In Appendix D.2.1 is possible to observe how this mesh is generated for $N = 10000$ points (100x100). The coordinate values are saved (from left to right and from down to up) in a 10000x2 vector that is used to calculate the distance r_i from each SPS to the CI pupil centre as:

$$r_i = \sqrt{(-y_0 - y_{iSPS})^2 + (-z_0 - z_{iSPS})^2}$$

The minus sign next to the y_0 and z_0 arrives from evaluating the generated irradiance value. Without this sign, the irradiance is incorrect since it is inversely proportional to the r_i value; therefore, this correction is fundamental for the algorithm's functioning to return the necessary correction for the GNC system.

Once the mesh has been created, and the r_i positions for the four SPS (nominal or redundant) are obtained, the penumbra profile is loaded (choosing the fine or coarse parameters). Appendix D.2.2 shows: the loading of the lateral and longitudinal parameters, the determination of R_0 , the 2D and 3D plots (Figures 5.2 and 4.4 respectively) and the approximation of the irradiance measurement (in DN) for each SPS in each r_i position. This step adopts the “fix” Matlab function that rounds the obtained number to the nearest integer, simulating the effective measure coming from the digital irradiance reading. The aforementioned step certainly increases the error in the subsequent inversion of the system, but it is closer to the real expected digital value.

5.2.3 Solving the local system

Finally, Appendix D.2.3 presents the solution of the system with the procedure illustrated in this chapter. Using the Cardano's function (Appendix D.1), the system is inverted and solved for r_i , where the positive and real solution is kept, and the boundary conditions are controlled. A check is implemented, where the solutions obtained for each r_i shall be unique. Thus, the sum of the number of solutions calculated for the four SPS shall amount to $4N = 40000$; if this check gives an error, the calculation shall be stopped since it is not physically possible to have more than one solution per reading.

The computation of the umbra centre's lateral coordinates is the algorithm's final goal, where the obtained solution is the correction needed in the Z-Y CI pupil plane. Additionally, the code implemented in the pseudo-paraboloidal algorithm is used to obtain an approximation of the longitudinal position (Appendix D.2.4), where the system is inverted once again to get the R_0 value as a mean of the 4 SPS readings in the nominal position. This step is an improvement made by the author to the [22] script in order to uniform the algorithm's outputs since the proposed strategy only considered the lateral computation.

The final and most crucial step of this procedure is the validation of the obtained lateral and longitudinal solutions (Appendix D.2.5), where the calculated umbra centre is compared to the generated initial value. The absolute value of the difference is then compared to a threshold ϵ to analyse the algorithm's precision graphically. The outcomes of the different investigated cases are extensively described in the next chapter.

Chapter 6

Algorithm comparison: Results from locally generated data

As previously described, in order to validate the algorithm, a set of locally generated data is used. From the generated mesh of 100x100 or 1000x1000 points and utilising the code available in Appendix D.2, the SPS measurements are estimated and inverted using the alternative, the pseudo-paraboloidal and the linear algorithms. This process leads to a simulated sequence of occulter displacements placed at 0.1 mm (or 0.01 mm if 1000x1000 grid is used) one from the other in the z and y directions to retrieve a diagram of the algorithm's error in computing the umbra centre position.

The following results cover the longitudinal and lateral errors inside the requirement box (± 10 mm) and the goal box (± 50 mm), using 2D and 3D error plots and a statistical error evaluation using histograms. The pseudo-paraboloidal and the linear algorithm implemented in this chapter were taken from the SPS flight algorithm script, where the author adapted them from a single input - single output configuration, allowing the determination of the mesh positions and the respective errors. Furthermore, the irradiance values are derived from a combination of the fine (for (z_0, y_0) from 0 to 13.8 mm) and coarse parameters available in Appendix D.4.

Moreover, the results using the outdated set of reconfigurable parameters inside both the requirement and goal were performed to validate the script with the results available in [5]. Where it was possible to conclude that the “Alternative_SPS_Algorithm_Generate” in Appendix D.2 successfully inverts the system and obtains an error similar to the one expected. Hence, the results obtained using the different algorithms (and their combinations) will determine valid outcomes.

6.1 Lateral Error

The lateral error is the most relevant parameter to be considered in the development of the algorithm. From the requirements in Section 2.2, measurement accuracy of $50\mu m$ for each axis is demanded inside the requirement box. An evaluation of this error for the different algorithms inside the requirement and goal box is introduced in this section. The three-dimensional plots show the absolute value of the error, whereas the histograms take the sign of the deviation into account.

6.1.1 Requirement box - Alternative

Figure 6.1 shows the alternative algorithm solution error inside the $\pm 10\text{ mm}$ box. It is possible to conclude from Figure 6.1b that **the algorithm satisfies the requirement of an error lower than $50\mu m$ inside the entire requirement box**. The adjacent plot in Figure 6.1a is useful for the identification of the accuracy patterns in the Z-Y plane. The value of $\epsilon = 10^{-5.6}$ is arbitrarily chosen to validate the pattern obtained from the top view of the 3D error surface plot in Figure 6.2a. The expected central symmetry of the alternative algorithm is evident in this plot.

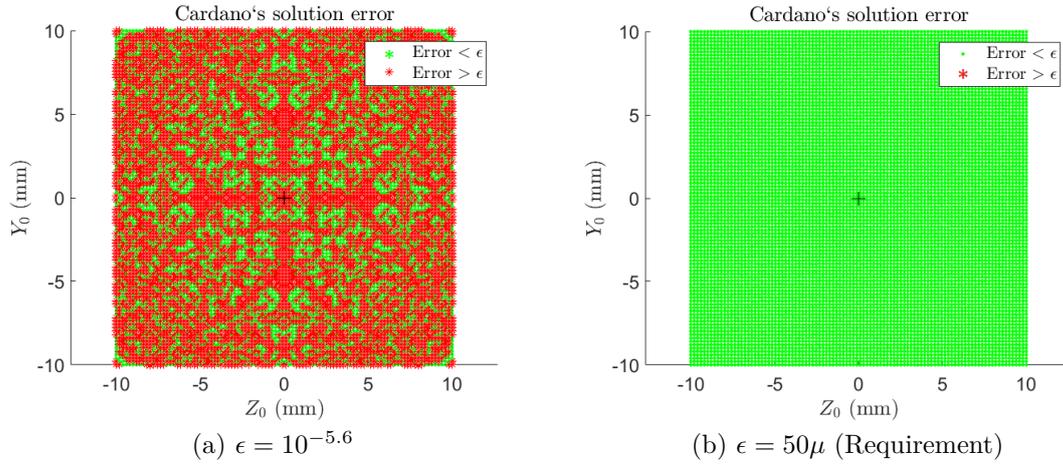


Figure 6.1: 2D Lateral error plot in the requirement box using Cardano and 100×100

From Figure 6.2b it is possible to examine the distribution of the error throughout the requirement box. A very uniform distribution of the error is an excellent outcome since the algorithm is expected to accurately work inside the entire interval. The expected peak error lies in the order of $6 \cdot 10^{-3}$ mm, which is an order of magnitude lower than the requirement, verifying once again that the alternative algorithm can be confidently adopted to determine the lateral position of the umbra centre.

A grid of 1000×1000 points is used to increase the precision of the fitting; this significantly increases the computational cost but allows the observation of the

accuracy patterns in Figure 6.2c. As expected, Cardano’s method for the alternative algorithm allows an axis-symmetric position determination with very high precision. From Figures 6.2b & 6.2d it is possible to observe that a circular area surrounding the centre (around $\pm 5mm$) has slightly higher accuracy, with a greener colour with respect to the yellow around the borders.

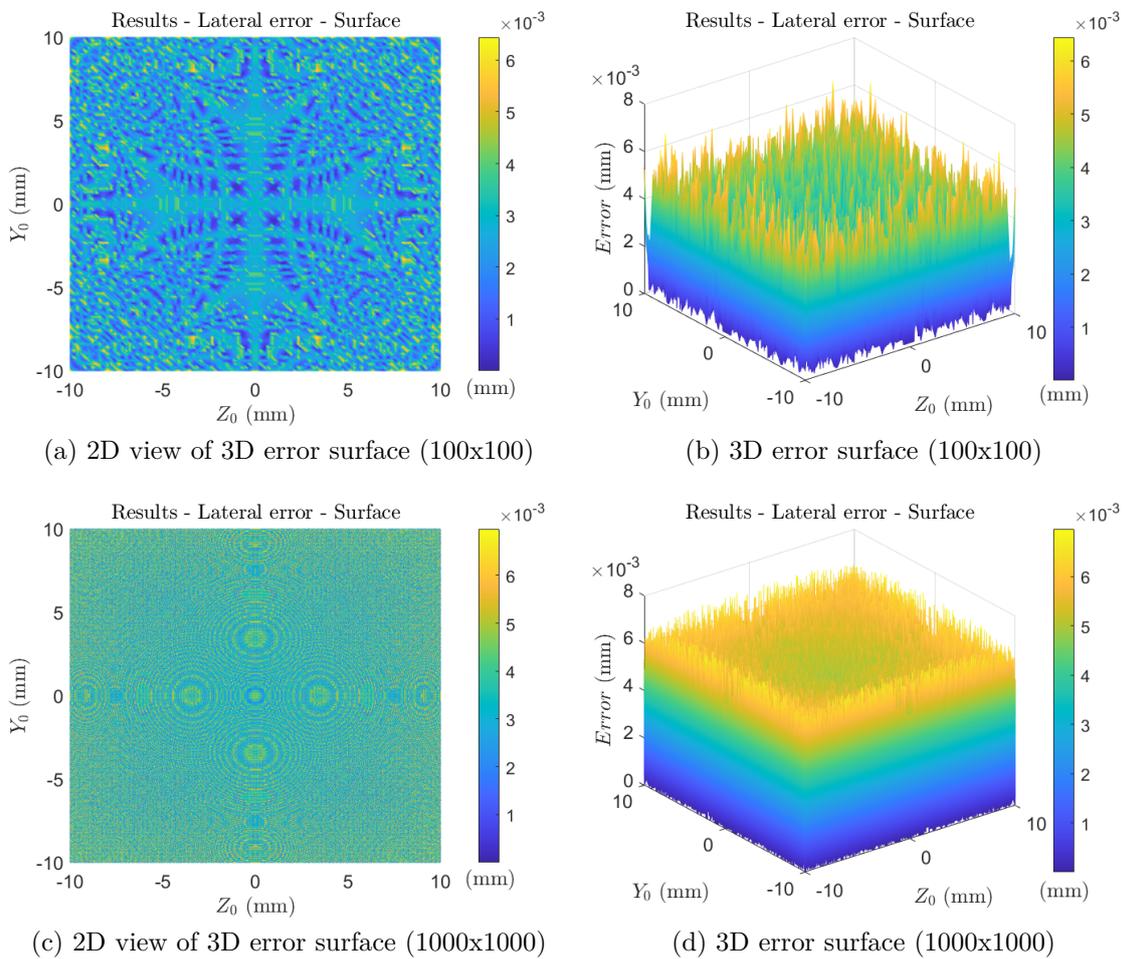


Figure 6.2: Lateral error surface in the requirement box using Cardano

6.1.2 Requirement box - Pseudo-paraboloid

The same procedure used to determine the alternative algorithm’s lateral error inside the requirement box is applied to the pseudo-paraboloidal (PP) algorithm. Only the part of the pseudo-paraboloidal algorithm code used to determine y_0 and z_0 was adapted into the "Generate" code. Consequently, the same locally generated SPS measures are inverted to directly compare the accuracy of both algorithms in the same mesh. Furthermore, the beneficial contribution derived from the combination with the linear algorithm is omitted at this step in order to contrast the third order equation solution accuracy of both algorithms.

As can be seen in Figure 6.3b the $50 \mu\text{m}$ requirement is not satisfied in the whole requirement box, only inside a 10 mm sided hexagon centred in the pupil centre; this might be a consequence of the variation of the fitting parameters used by the PP algorithm. The red zones in this figure correspond to the increase of error in the 3D surface given by Figure 6.4b. These results are consistent with the error evaluation proposed in [5], [6] and [7], where the corners of the requirement box see an error that goes up to $100 \mu\text{m}$.

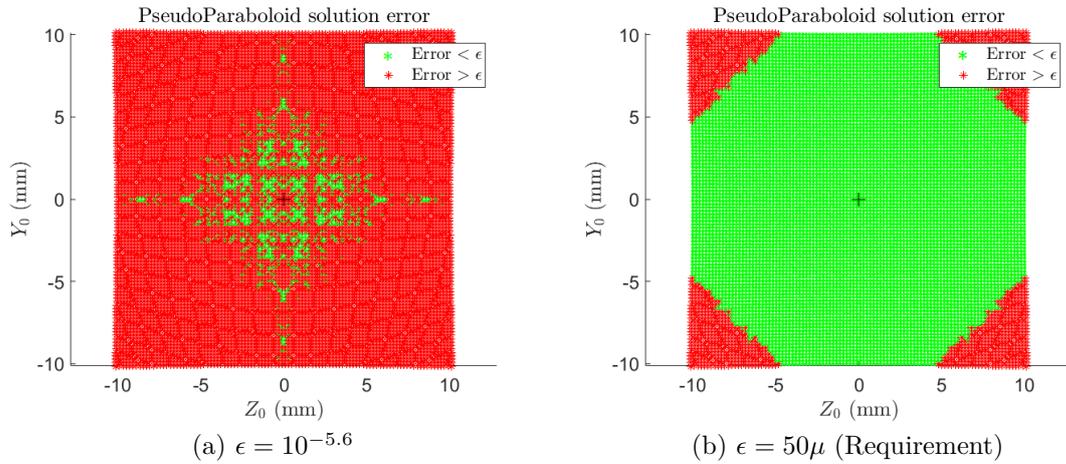


Figure 6.3: 2D Lateral error plot in the requirement box using pseudo-paraboloid

Figure 6.3a and 6.4a show how the precision of the pseudo-paraboloidal algorithm does not follow an ideal central symmetry. Furthermore, the increased accuracy of the points that lie in the coordinate axis ($y_0 = 0$ and $z_0 = 0$) leads to the darker "cross" in Figure 6.4a. This result further confirms the initial statement that lead ESA to the proposal of an alternative algorithm of increased precision and symmetry with a higher computational cost.

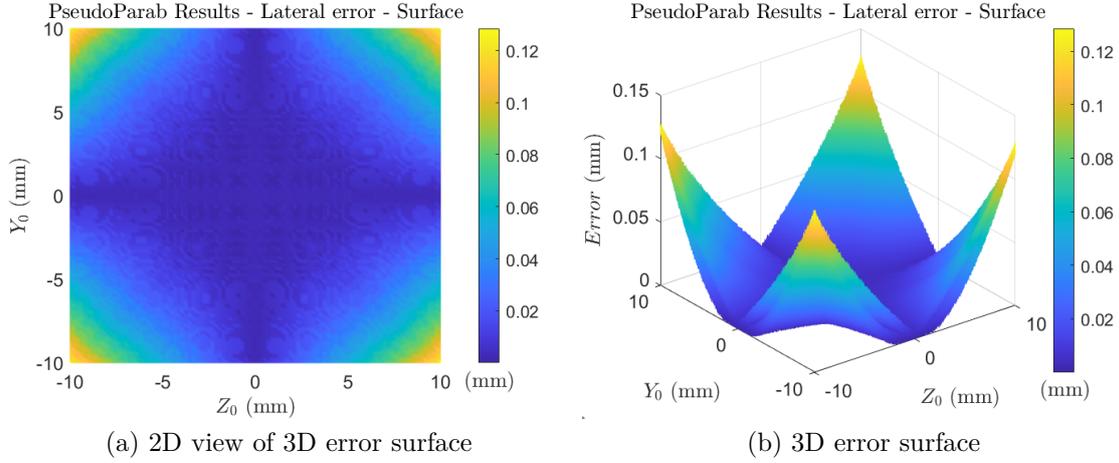


Figure 6.4: Lateral error surface in the requirement box using pseudo-paraboloid

6.1.3 Requirement box - Linear & Pseudo-paraboloid + Linear (PPL)

As introduced in Section 3.4, the combination of the pseudo-paraboloidal (PP) and linear algorithms shall theoretically lead to a reduction of the error inside the requirement and goal box. However, the results in [5] and [7] apply only the outdated set of reconfigurable parameters for the evaluation of the lateral positioning error. The following results shall lead to a different error distribution that sees a reduced performance of the linear algorithm since the “linear parameters” remained unchanged, whereas the lateral parameters were modified.

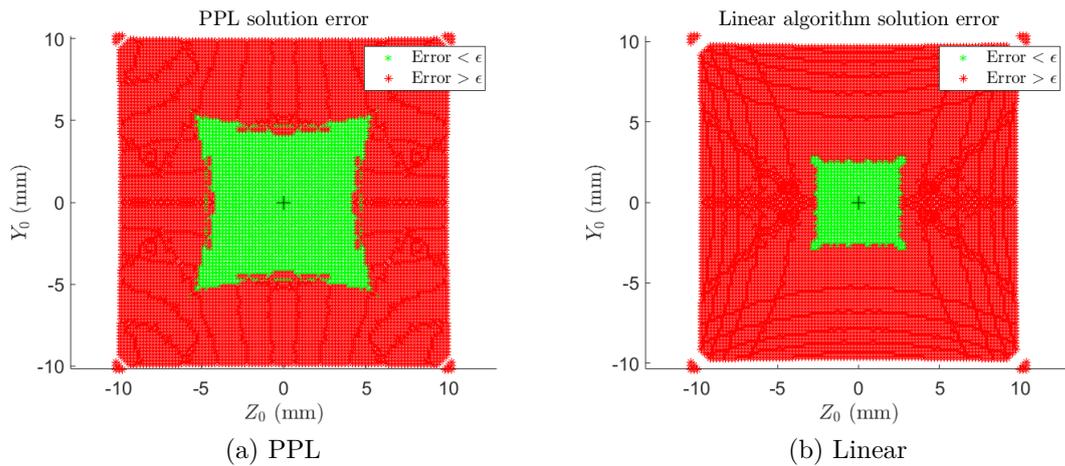


Figure 6.5: Lateral error plot in the requirement box

From Figure 6.5, it is evident how the requirement is fulfilled by neither the linear nor the PPL. Exclusively in the area surrounding the centre, the error is lower than $50 \mu\text{m}$; it is evident how this error increases rapidly, leading to a **reduction of the precision of the PPL algorithm with respect to the PP**.

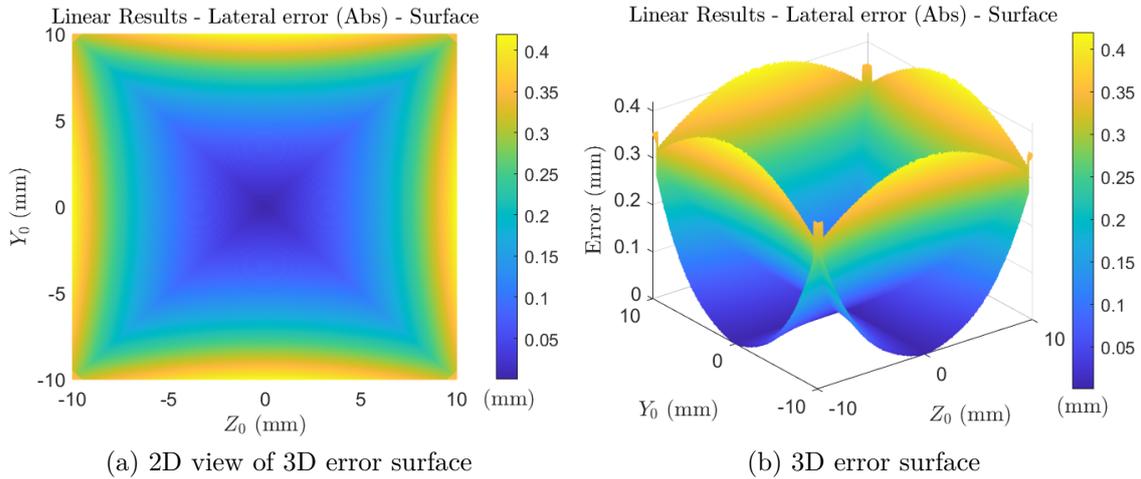


Figure 6.6: Lateral error surface in the requirement box using linear algorithm

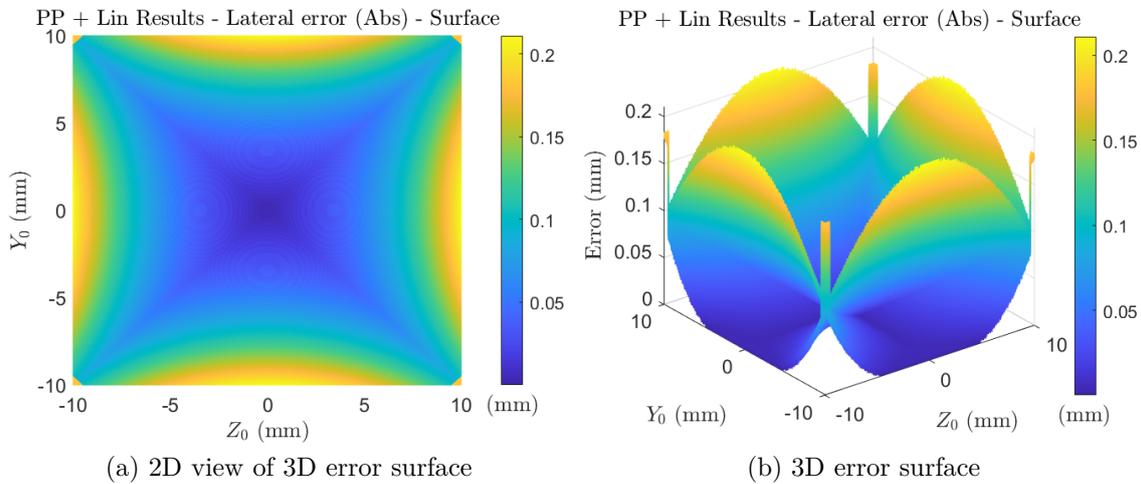


Figure 6.7: Lateral error surface in the requirement box using PPL

Examining Figure 6.6 is noticeable how the three-dimensional view of the lateral surface error is different from the expected one (Figure 3.6). The error is one order of magnitude higher despite the same linear parameters and algorithms were used.

Furthermore, the shape of this surface is considerably different from the one anticipated, with a rapid increase of the error as the points approach the box sides.

An important note is that the lower error zone creates an “X” shape (similar to the expected result) that leads to the decrease in error (expressed in absolute value) in the corners of Figure 6.7 that uses the pseudo-paraboloidal + linear algorithm. This algorithm delivers a higher performance with respect to the linear algorithm. It is visible how the maximum error is in the order of 0.2 mm, and the centre of this “X” pattern produces an even lower deviation.

The 3D plot in Figure 6.7b is not a perfect representation of the error because of a graphical issue. The matrix used for the determination of the error in Appendix D.2.5, combined with the Z-Y mesh, delivers a graph that is symmetrical to only one of the axes (despite the 2D representation of this error being accurate in both axes). Thus, a superposition of two three-dimensional surfaces is needed to provide an approximated illustration of the error. Consequently, in the following graphs, the lower error zone under the upper curve shall be ignored. In other words, only the higher error points shall be considered if two vertical points coincide. Furthermore, the results shown are the **absolute value of the lateral error**. Thus, the critical information regarding the sign of the error is lost.

6.1.4 Requirement box - Alternative + Pseudo-paraboloid

In order to achieve a better solution for the pseudo-paraboloid inside the requirement box, a combination with the alternative result is proposed.

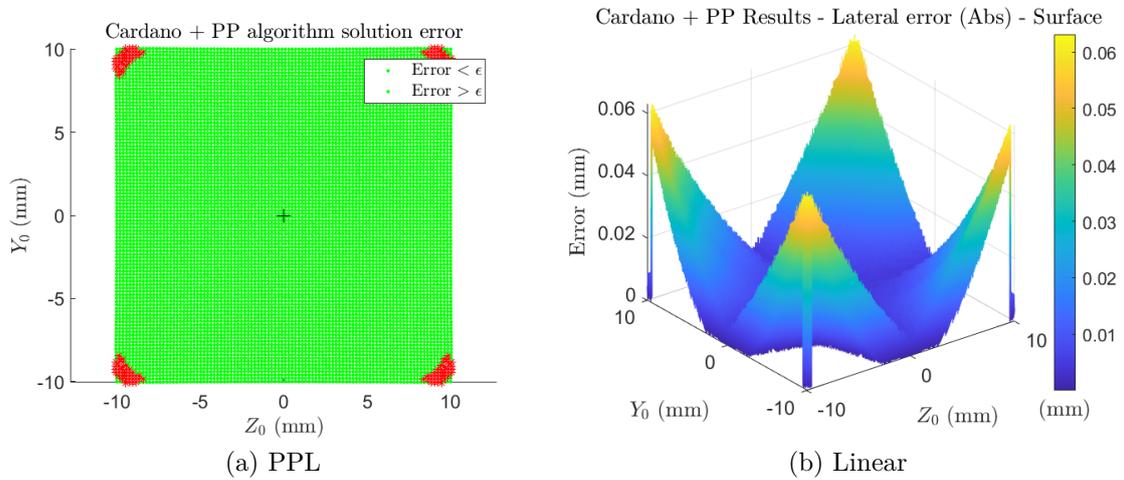


Figure 6.8: Lateral error plot in the requirement box for Alternative + PP

This new algorithm achieves outstanding results inside the requirement box, thanks to the PP’s increased precision and the central cross and the homogenous distribution of Cardano’s solution. This algorithm is the most computationally costly, but it reduces the maximum error of the pseudo-paraboloidal solution inside the requirement box.

6.1.5 Requirement box - Algorithm comparison

The previous results have highlighted how the alternative algorithm achieves a higher accuracy inside the requirement box. However, the evaluation of the statistical distribution of this error for the lateral displacements is a primary concern. From

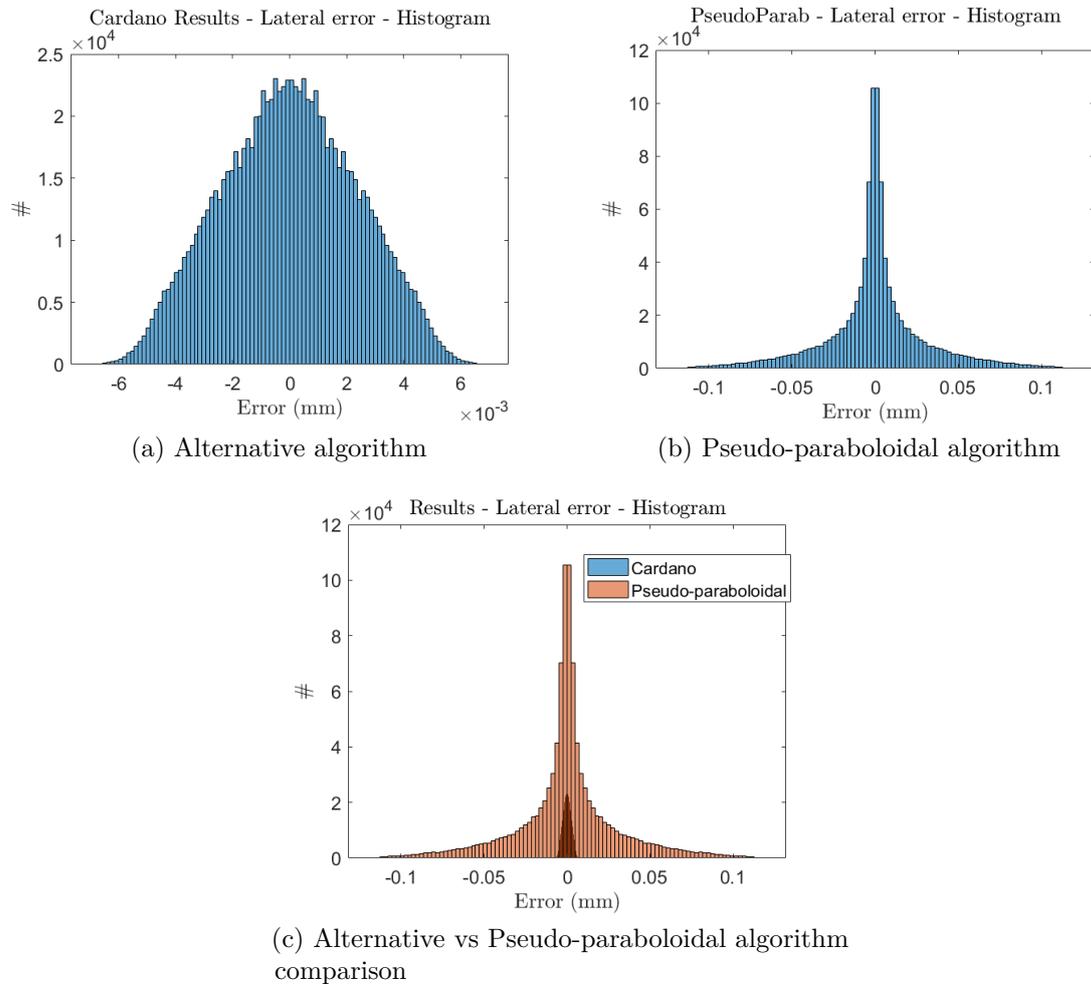


Figure 6.9: Statistic of both algorithms error for the lateral position inside the requirement box for a 1000x1000 grid

Figure 6.9 a direct comparison of both algorithms is possible, where it is evident that the alternative algorithm in Figure 6.9a achieves an error in the order of 10^{-3} mm; whereas the pseudo-paraboloidal algorithm achieves an order of magnitude higher error. The result in Figure 6.9b is similar to the one obtained in [5], but the achieved errors are distributed more uniformly (similar to a gaussian); this might be a consequence of the different fitting parameters used for the illumination profile

or the procedure used to calculate the mesh and the error.

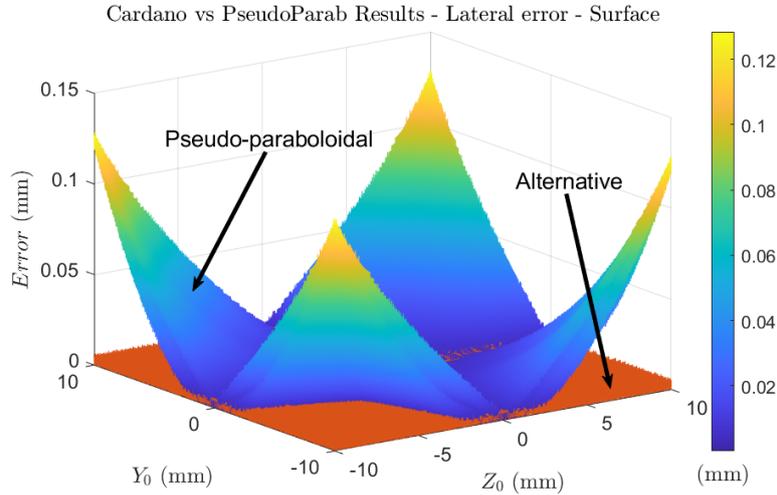


Figure 6.10: Lateral error surface in the requirement box using both algorithms

Combining the two histograms in Figure 6.9c is evident that the achieved accuracy of the alternative algorithm is superior to the pseudo-paraboloidal, with an error distribution that is contained inside the first four bars of the flight algorithm. Consequently, **the alternative algorithm achieves higher accuracy and more uniform distribution of the error compared to the pseudo-paraboloidal inside the requirement box.** These are both significant advantages that shall lead to a reliable validation algorithm to be used in the PROBA-3 mission SOC.

Furthermore, in Figure 6.10 the three-dimensional representation of the error decidedly exposes the increased accuracy (of one order of magnitude) of the alternative algorithm in the outer edges of the requirement box.

Regarding the implementation of the linear algorithm, in Figure 6.11 the difference in precision between the PP and the linear is obvious. The PP error remains between ± 0.1 mm, whereas the linear delivers an unacceptable ± 0.4 mm inside the requirement box.

From Figure 6.12, it is evident how the combination with Cardano's result augments the performance of the pseudo-paraboloidal algorithm. The maximum error is lowered from 0.1 mm to 50 μm , satisfying the requirement. Furthermore, the statistical evaluation exhibits how the error range and deviation is diminished.

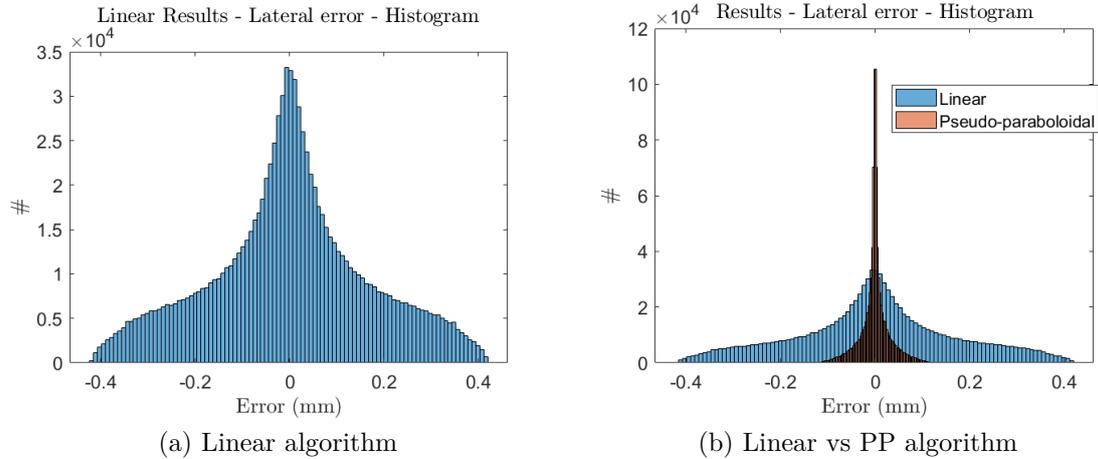


Figure 6.11: Statistic of linear and PPL algorithms error for the lateral position inside the requirement box

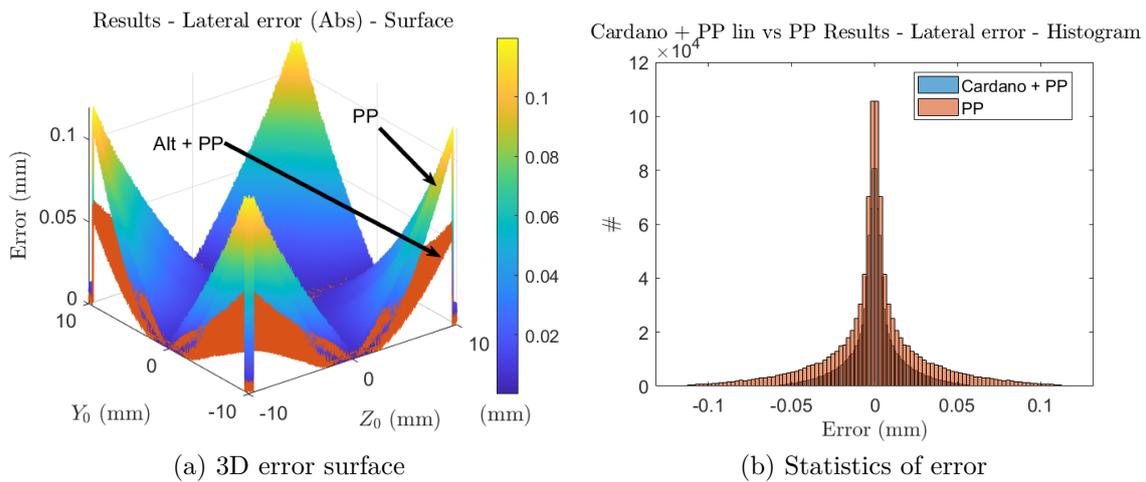


Figure 6.12: Comparison error plot in the requirement box for Alternative + PP

Therefore, from the locally generated data, it is possible to conclude that the alternative algorithm provides the best performance in the requirement box, followed by the pseudo-paraboloidal. Consequently, using **a combination of the linear with either the alternative or the PP algorithms shall determine a less accurate umbra centre position inside the requirement box**. These results are different from the ones obtained in [5], where the alternative obtains a less accurate performance than the PP. This is a consequence of the use of an outdated set of parameters for the umbra and the fact that the study was conducted using the algorithm developed in [22], that did not follow accurately the expected penumbra

profile curve, where the supposedly used R_0 value was wrongly equal to the PP parameter “A”.

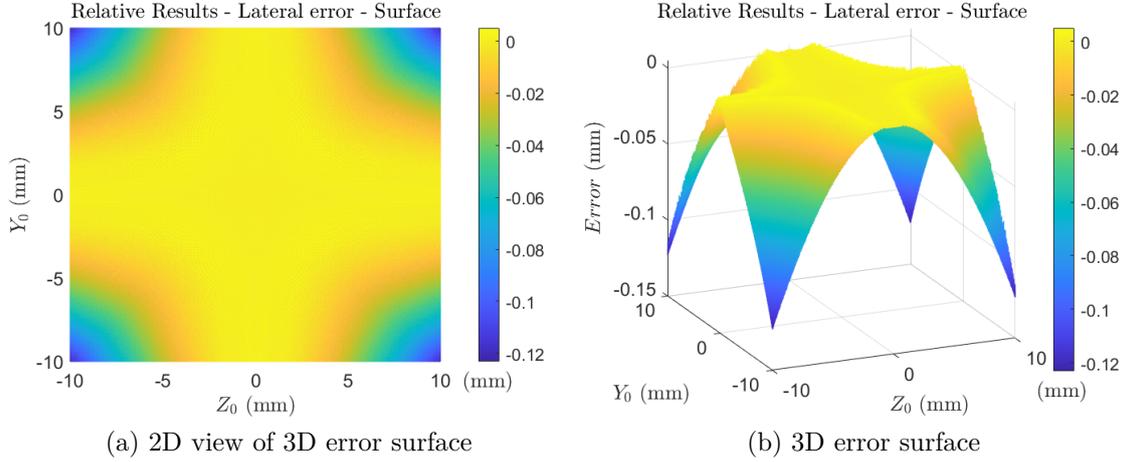


Figure 6.13: Lateral difference of the absolute error (Cardano - PP) in the requirement box

The next step in the analysis is the difference in the absolute error between both algorithms. In this case, the error of the alternative algorithm err_{Alt} gets subtracted by the pseudo-paraboloid err_{pp} to obtain the results in Figure 6.13. This graph allows the evaluation of which error is higher, where it is evident that Cardano’s achieves a better performance as the points approach the edges.

Both algorithms perform adequately inside the inner ± 5 mm, with a relative statistical error close to null. Nevertheless, the outer edges of the box rapidly climb to an error of -0.1 mm due to the parabolic nature of the flight algorithm.

Finally, the main results to be compared are arranged in decreasing order of accuracy in Table 6.1. The symmetry of the error is evident for all the algorithms, obtaining a mean value of zero. The alternative algorithm reaches a standard deviation (std) that is one order of magnitude lower than the rest, but the computational time for the 100×100 grid is significantly higher (around one second) than the pseudo-paraboloidal or the linear algorithms. The values obtained by the alternative algorithm are similar to the Table 3.1 results from [5].

These results confirm the choices made by INAF in the development of the algorithms, where the PP and PPL obtain a high precision with a low computational cost. On the ground, this cost is not relevant; thus, the implementation of the alternative algorithm for validation is an appropriate addition to the metrology chain.

Table 6.1: Lateral error of the algorithms and calculation time for 100x100 grid inside the requirement box

Algorithm	Min [mm]	Mean & Std [mm]	Max [mm]	Time [s]
Alternative	-0.0064	0.0000 & 0.0024	0.0064	1.0953
Alt + PP	-0.0579	0.0000 & 0.0157	0.0579	1.3381
Pseudo-paraboloidal	-0.1140	0.0000 & 0.0307	0.1140	0.0289
PPL	-0.2109	0.0000 & 0.0762	0.2109	0.0293
Alternative + Lin	-0.2118	0.0000 & 0.0851	0.2118	1.1130
Linear	-0.4198	0.0000 & 0.1708	0.4198	0.0016

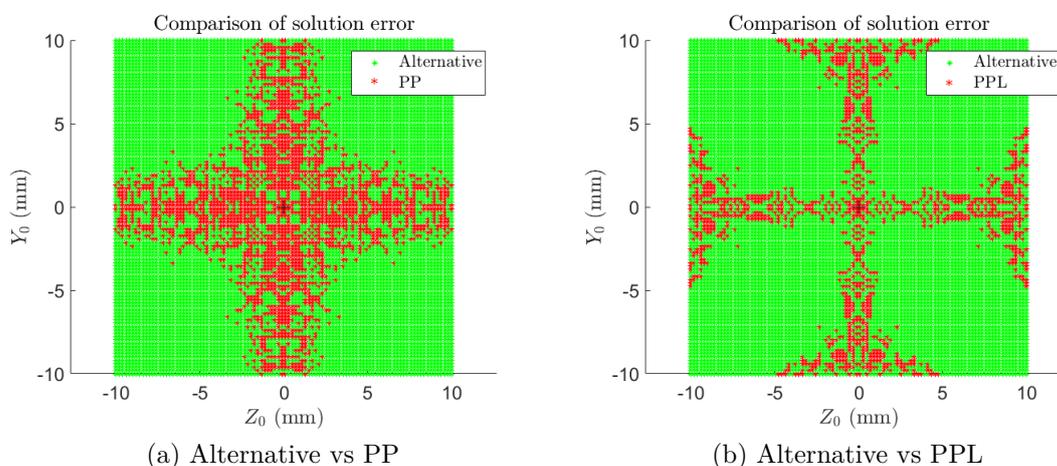


Figure 6.14: Comparison of the precision of the alternative and flight algorithms inside the requirement box

As described, the main objective of this investigation is to study and determine an SPS algorithm to be used for validation. For this reason, the results in Figure 6.14 are crucial for the ascertainment of which algorithm achieves a better precision inside the requirement box.

It is indisputable that the alternative algorithm reaches outstanding results inside the ± 10 mm box. However, it is possible to observe how the pseudo-paraboloidal algorithm reaches a similar accuracy in the central cross, and in some cases, it can be more precise than the Cardano’s calculated positions. On the other hand, comparing the alternative result with the PPL, the advantage of the computationally costly alternative algorithm is evident. These results might be helpful for the determination of whether to follow the results of one of the algorithms during the calibration, although the differences are minor in the requirement box and the overall precision of Cardano’s method is higher.

6.1.6 Goal box - Alternative

The same procedure applied for the requirement box is repeated in the goal box. The shall algorithm calculate the position at a reduced performance within a ± 50 mm box for the lateral displacement. It is important to remark how the algorithms work inside this box; the expected illumination profile is derived from a combination of the fine and coarse parameters. First, the alternative and flight algorithms calculate the umbra centre z_0 and y_0 position using the coarse fitting parameters. Afterwards, the computed position is compared with the threshold used to discriminate the fine or coarse fitting profile, i.e. 13.8 mm. If the found coarse solution lies inside the 0 to 13.8 mm range, the evaluated position is refined once again using the fine fitting parameters.

These variations of the reconfigurable parameters might lead to a solution that is not uniform in the goal box but delivers a reduced error. In Appendix A, the evaluation of a reduced goal box (± 41 mm) using only the fine fitting parameters for both the alternative and PP algorithms is presented.

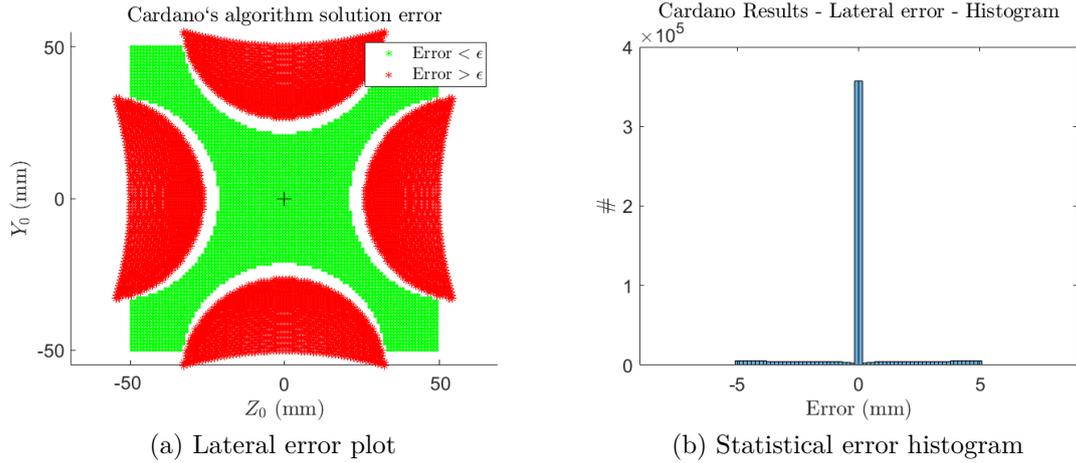


Figure 6.15: Lateral error in the goal box using Cardano

From Figure 6.15a, it is possible to identify the green area that fulfils the stringent precision requirement of $50 \mu\text{m}$ even outside the requirement box. This “X” configuration has an opposite trend compared to the pseudo-paraboloidal results presented in the previous section (inside the requirement box), where the precision decreased in the corners.

Analysing the three-dimensional representation of the error in Figure 6.16, it is evident how the centred “X” shape has a dark colour for a high accuracy sector (error lower than $50 \mu\text{m}$), whereas a symmetrically semi-circular shape (centred at ± 55 mm and the 0 mm coordinate) see a discontinuous absolute value of the error, that increases from almost null up to 5 mm.

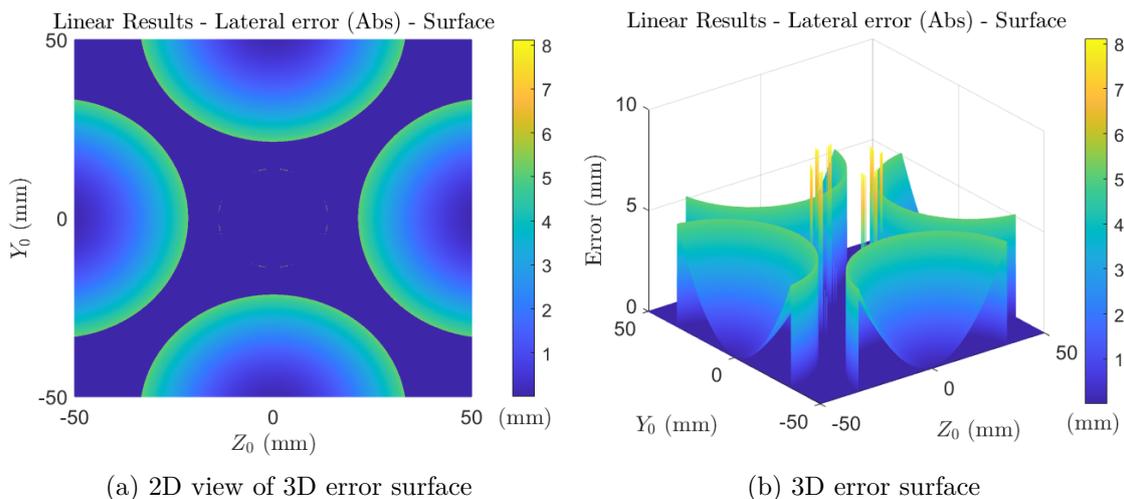


Figure 6.16: Lateral error surface in the goal box using Cardano

The plot in Figure 6.15a is obtained with the script developed by R. Rougeot in [22] and reported in Appendix D.2.5, that uses the calculated z_0 and y_0 coordinates to report the outcome of error compared with the required accuracy. Therefore, the white areas that are visible where the error peaks are a consequence of the poor accuracy of the calculated position. Hence, it is possible to observe how the error in these semi-circular lobes shifts the position away from the centre along the axis where it is focused.

From the statistical error evaluation, it is possible to observe that almost 80% of the one million points are in the two central bars, whereas an almost uniform distribution of the error extends up to the ± 5 mm deviation. The discontinuity generated inside the goal box has an error that spikes at around 20 mm from the centre and drops to zero at the edges. This difference does not follow the fitting parameter switch that is made in the 13.8 mm radius, whereas the central yellow peaks in Figure 6.16b are indeed a numerical consequence of this variation.

Furthermore, an evaluation of the flags that indicate the type of solution implemented by Cardano's method show how all the system inversions are performed with a positive Δ ($flag = 4$ in Appendix D.1). Thus, the reason that generates this spike in the absolute value of the error is not fully known to the author, and shall be tested and validated using the SPS simulator. In Appendix C, the evaluation of the SPS redundant set inside the goal box using the alternative algorithm is presented. The results are analysed, and a possible explanation of the error discontinuity is hypothesised, where the centre of the semi-circles coincides with the SPS radius of 55 mm, and the circle radius is equal to 33.5 mm (equivalent to the distance of the umbra to the pupil centre at the nominal ISD). Thus, **a single SPS inside the umbra might cause this error** (irradiance close to zero), returning a wrong

radius that causes a displacement of the output coordinates.

However, the results obtained using only the fine parameters in the ± 41 mm box in Appendix A does not show this discontinuity of the error, yet the error becomes higher (around 13 mm) as the SPS approaches the umbra at the edges.

6.1.7 Goal box - Pseudo-paraboloid

The analysis of the pseudo-paraboloidal algorithm inside the goal box is fundamental for understanding the correct algorithm to be followed in a specific region.

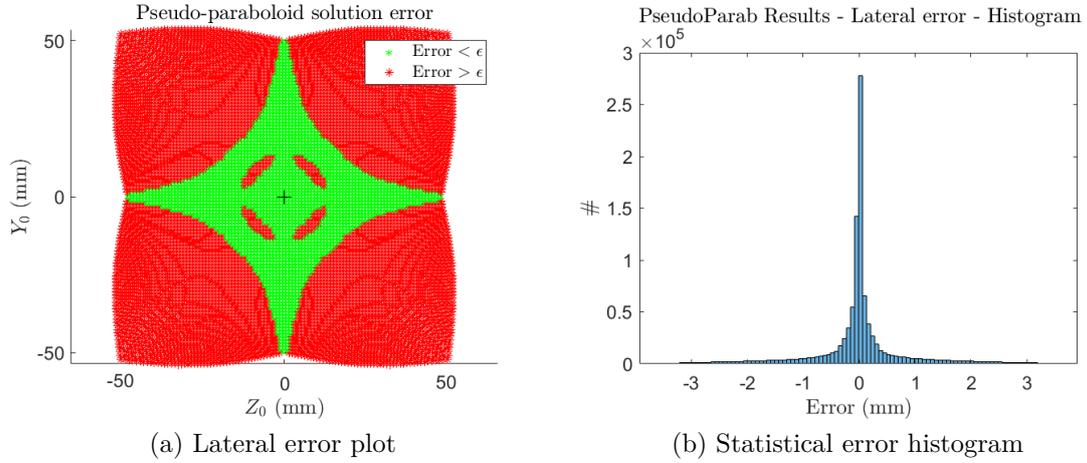


Figure 6.17: Lateral error in the goal box using Pseudo-paraboloid

From Figure 6.17a, it is possible to observe the increased precision reached by the PP algorithm along the Y and Z axis. As the points approach the edges of the box, the accuracy tends to diverge up to a 3 mm absolute error gradually. The statistical evaluation of this error shows how it is symmetrically distributed, with more than 85% of the points inside the ± 1 mm error range.

The three-dimensional representation of this deviation shows once again the central cross with increased accuracy, but the edges see a lobe-like formation that rapidly climbs up to the maximum absolute error. Unlike the previously analysed alternative algorithm result, the PP algorithm obtains a continuous distribution of the error even when the fitting parameters are changed (only a slight difference in the ± 13.8 mm centred circumference); this is a significant advantage point for this algorithm, which obtains an improved result with a lower computational cost. Once again, the numerical error spikes around the fitting parameter switch zone are visible in Figure 6.17a, and the green “hexagon” in the requirement box of Figure 6.3b is present as a green rhombus surrounded by a less accurate inner circle.

These results are similar, but not identical, to the ones in [5], where the maximum error (in absolute value) is obtained at the centre of the edges with a 6 mm deviation.

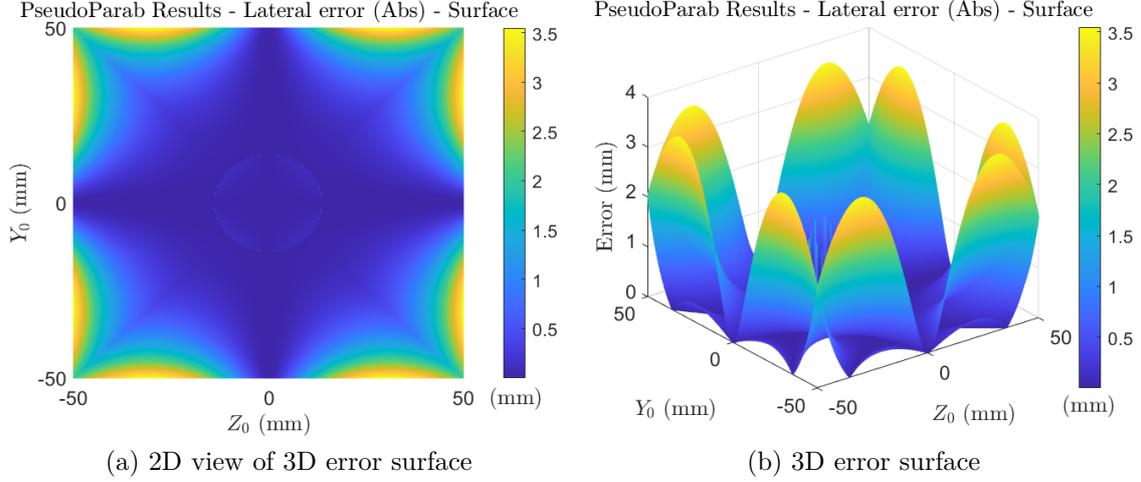


Figure 6.18: Lateral error surface in the goal box using pseudo-paraboloid

The locally generated error surface in Figure 6.18b shows high precision at the edges, but a “double-lobe” shape increases the error at the corners up to 4 mm (more accurate); this is a consequence of the different reconfigurable parameters used in the previous tests (October 2020), but the demonstration of this fact is omitted in this work. Furthermore, the evaluation of the PP algorithm using the redundant set of SPS is presented in Appendix C.

6.1.8 Goal box - Linear

The performance of the simpler linear algorithm inside the goal box in Figure 6.19a reveals how the 50 μm requirement is only satisfied by a small square in the centre (almost 3 mm side) and by two sets of 4 symmetrical points at around 30 mm and 35mm from the centre.

Indeed, the accuracy of this algorithm is lower closer to the centre, compared to the pseudo-paraboloidal and alternative, which use both sets of fine and coarse parameters. However, the three-dimensional lateral error surface reveals a maximum error of almost 4 mm at the edges of the box, whereas the inner area shows a non-uniform error that resides in the 0 to 1 mm range. The umbra points in the zone that utilise the fine fitting parameters (requirement box) is evident because of the increased error circle, which does not correspond to the results obtained by [5], as a consequence of the different parameters employed in the past studies. Nonetheless, the shape of the 3D surface in the outer goal box and the statistical error distribution are similar to the ones presented by M. Casti in [5], but the maximum error obtained in such analysis is two times higher, of almost 9 mm.

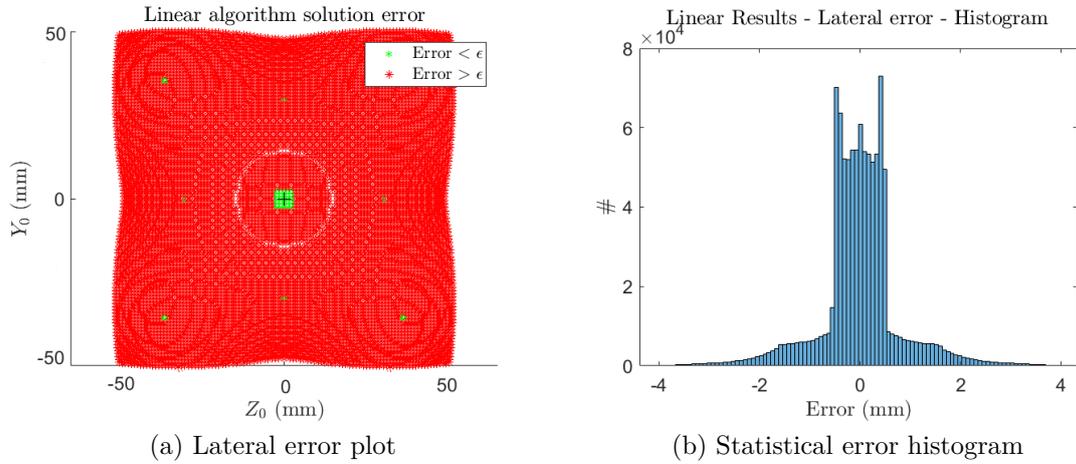


Figure 6.19: Lateral error in the goal box using linear algorithm

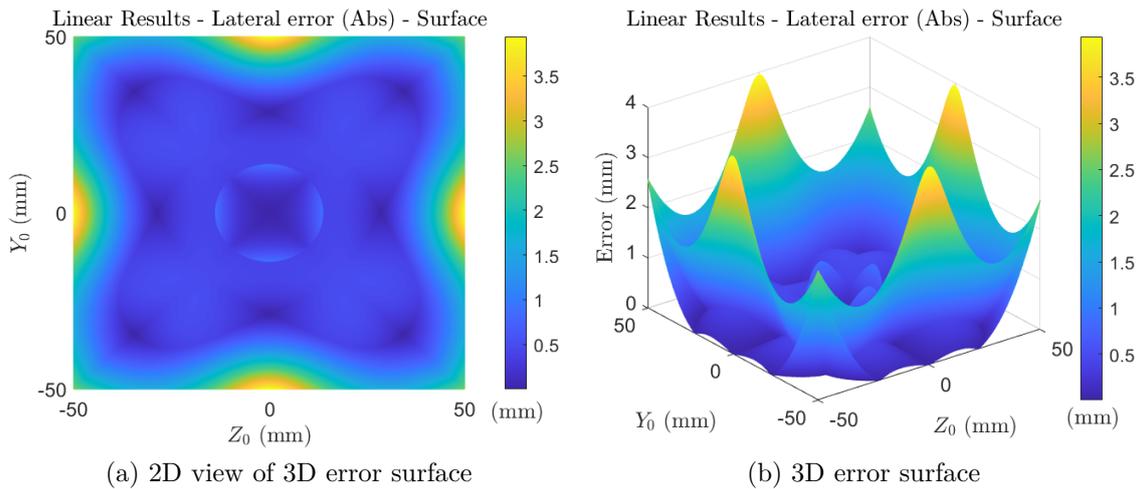


Figure 6.20: Lateral error surface in the goal box using linear algorithm

6.1.9 Goal box - Alternative + Linear

The combination of the alternative + linear algorithm was not studied inside the requirement box since the alternative algorithm's results already satisfied the accuracy requirements. Nonetheless, as observed in Figure 6.16, the alternative algorithm's error for points beyond ± 20 mm see an unexpected discontinuous error that rises and then returns to an acceptable value.

In order to contain this error, it is possible to evaluate the performance of the alternative + linear algorithm. As perceived from Figure 6.21a, the white zones surrounding the semi-circles get considerably reduced (hence the error peak is lower),

yet the accuracy inside the “X” gets diminished up to the point where the $50\ \mu\text{m}$ requirement is no longer satisfied (not even in the requirement box). The evaluation of the statistical error shows how the error range is reduced from $\pm 5\ \text{mm}$ to $\pm 3\ \text{mm}$, but the null error bar is considerably lessened.

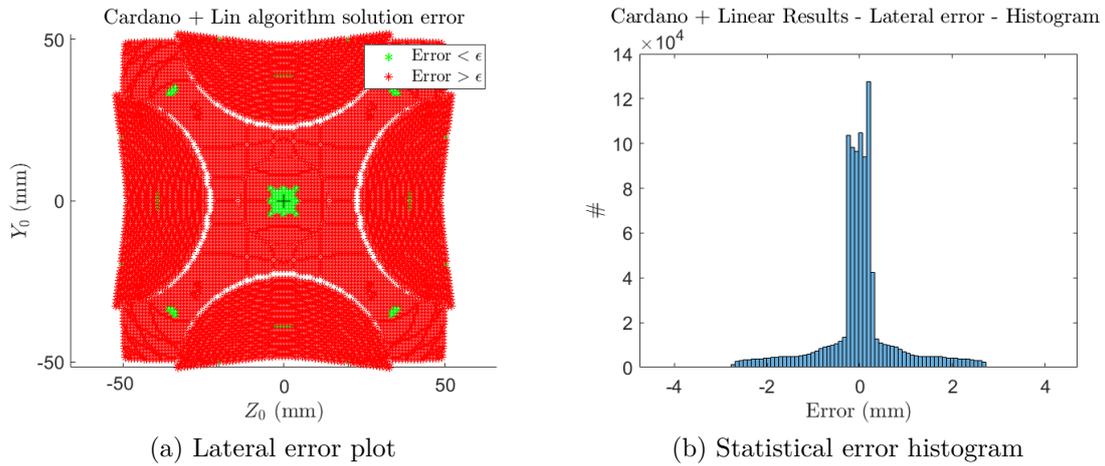


Figure 6.21: Lateral error in the goal box using alternative + linear

As noticeable from the 3D surface plot in Figure 6.22b, the numerical error spikes (in yellow) surrounding the requirement box are amplified by the increased error of the linear algorithm in this zone. The maximum error in the semi-circles is reduced by the union of the algorithms (almost halved), whereas the error approaching the edges marginally increase due to the linear algorithm’s shape.

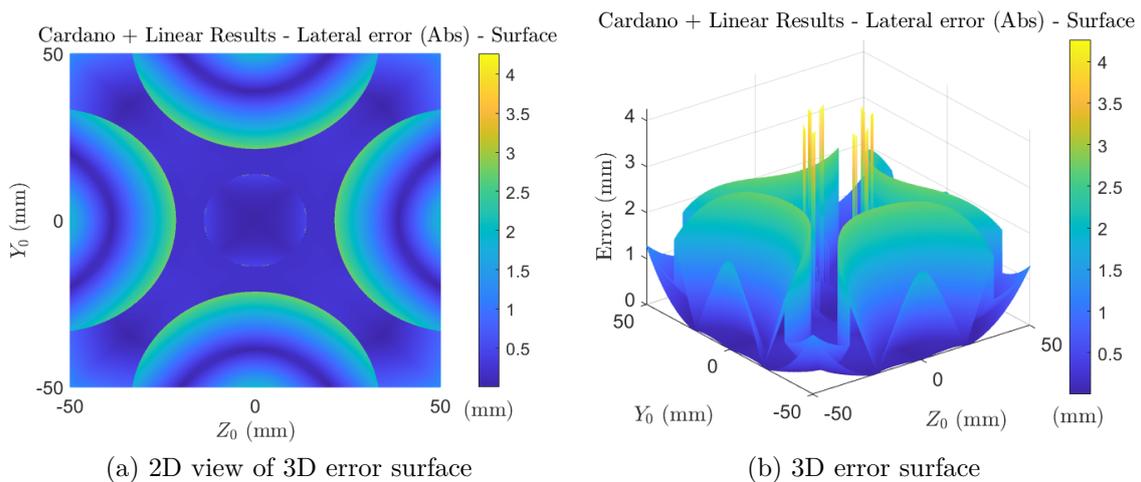
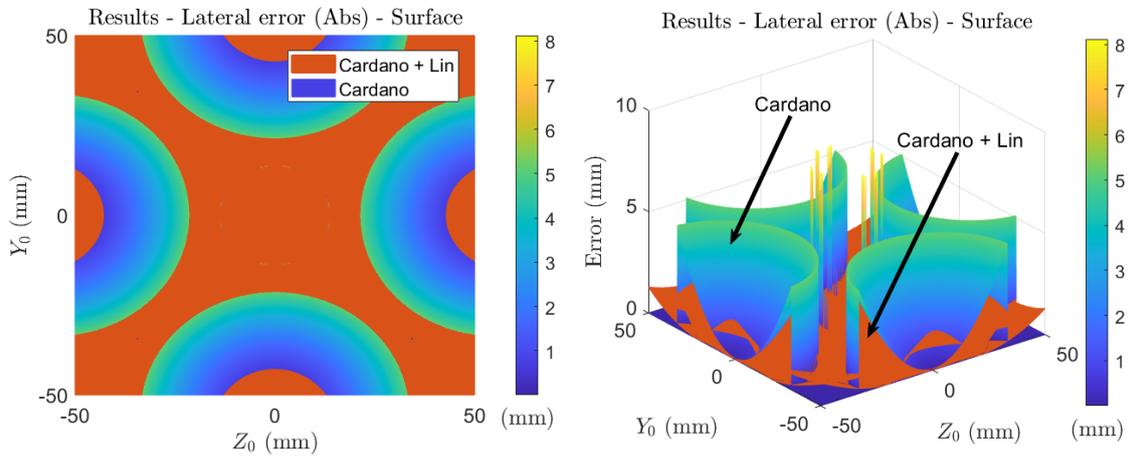


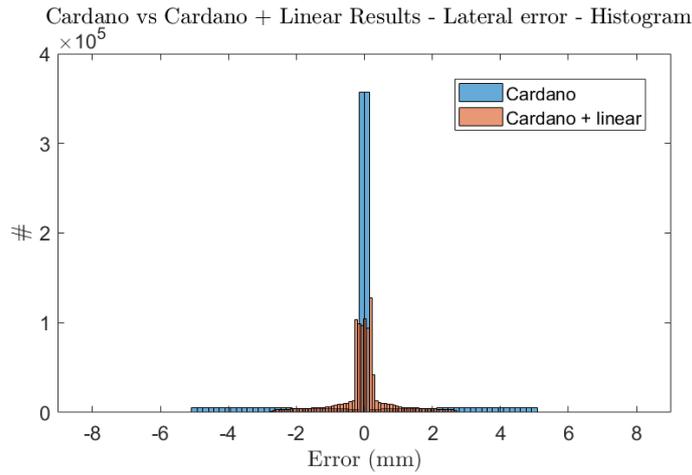
Figure 6.22: Lateral error surface in the goal box using alternative + linear

A helpful tool to understand the zones where the alternative + linear algorithm perform better than the alternative alone is by a direct comparison of the lateral error surface plot in Figure 6.23.



(a) 2D view of 3D error surface (the colour corresponds to the algorithm showing the maximum error for that point)

(b) 3D error surface



(c) Statistical error histogram

Figure 6.23: Lateral error in the goal box using alternative and alternative + linear

It is indisputable that the alternative algorithm performance inside the central “X” is far superior to any of the previously tested algorithms; therefore, if the points being considered remain in the orange area in Figure 6.23a (if orange is visible, then the error is higher), the alternative algorithm alone shall be used for the validation of the results. On the other hand, if the points lie inside the green/blue area, the

alternative + linear shall be used to ensure a reduction of the error.

The statistical error comparison of these results shows how the alternative + linear keeps the error closer to the zero zones, although the blue bars show that if only the Cardano is used, the error obtained has a higher probability of being closer to null.

6.1.10 Goal box - Pseudo-paraboloidal + Linear

The same analysis is now performed for the pseudo-paraboloidal + linear (PPL) algorithm. Once again, it is visible from Figure 6.24a how the combination with the linear algorithm damages the precision inside the requirement box, where the green central cross of the PP algorithm (Figure 6.17a) is no longer present.

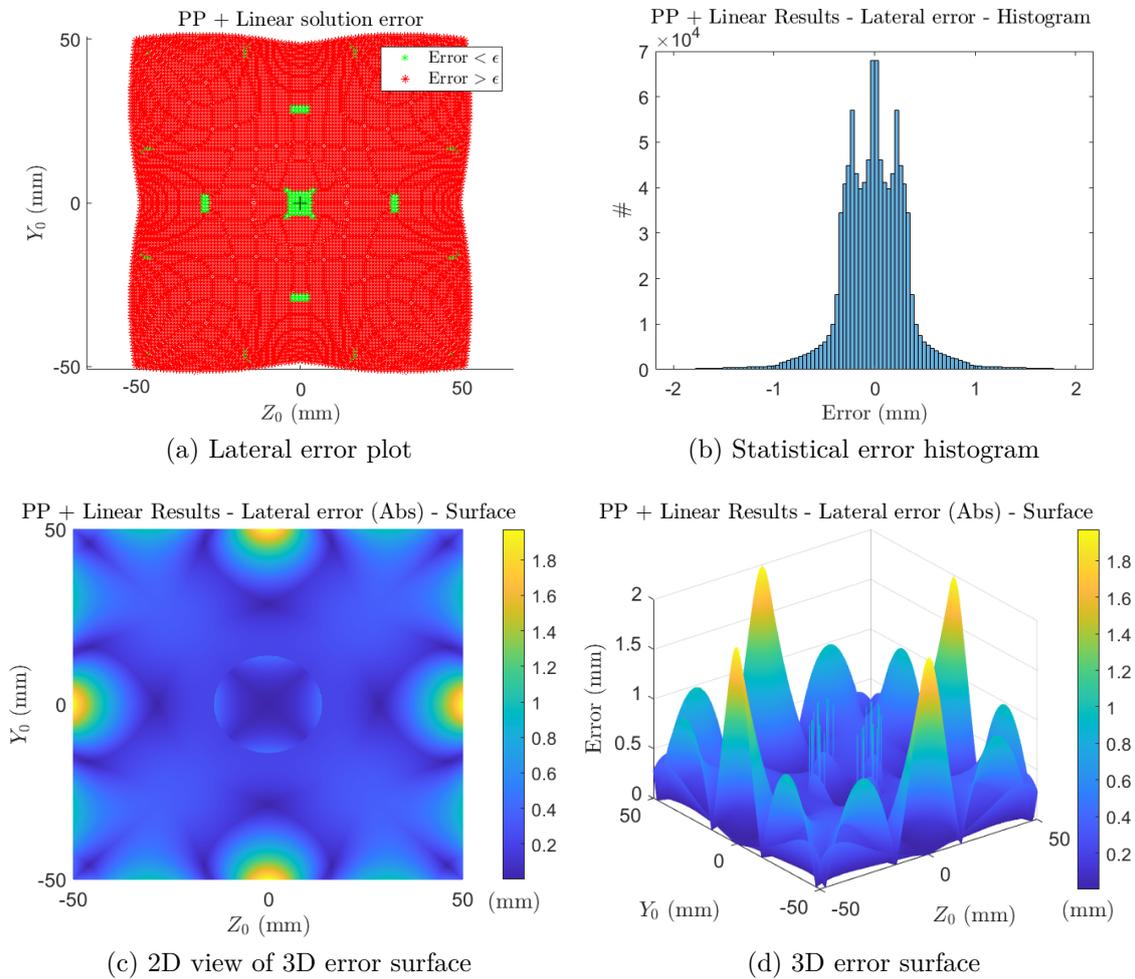


Figure 6.24: Lateral error surface in the goal box using Pseudo-paraboloid + Linear

Few points that satisfy the $50 \mu\text{m}$ are visible along the axis, where the error shape has a “dent” at the zero coordinate of the sides, showing how the error is higher and tends to shift the points towards the centre, opposite to the outward shift obtained when the linear is applied together to with the alternative algorithm. This result leads to the following algorithm combination where the PP (or PPL) results are merged with the alternative algorithm to counterbalance this error.

The statistical error distribution remains inside a $\pm 2 \text{ mm}$ range, with most points inside the three error peaks. The central peak with the zero error bars represents the most accurate places in the box, whereas the two symmetrical peaks, at around 0.3 mm , might be a consequence of the eight small lobes (with a maximum error of 1 mm) visible from Figure 6.24, which covers most of the surface as the points approach the edges. The achieved shape is very similar to the one obtained in [5], but the result in Figure 6.24d shows more hills around the sides; although, the maximum error is still a consequence of the linear algorithm, with an absolute value of 2 mm with respect to the 7 mm in [5].

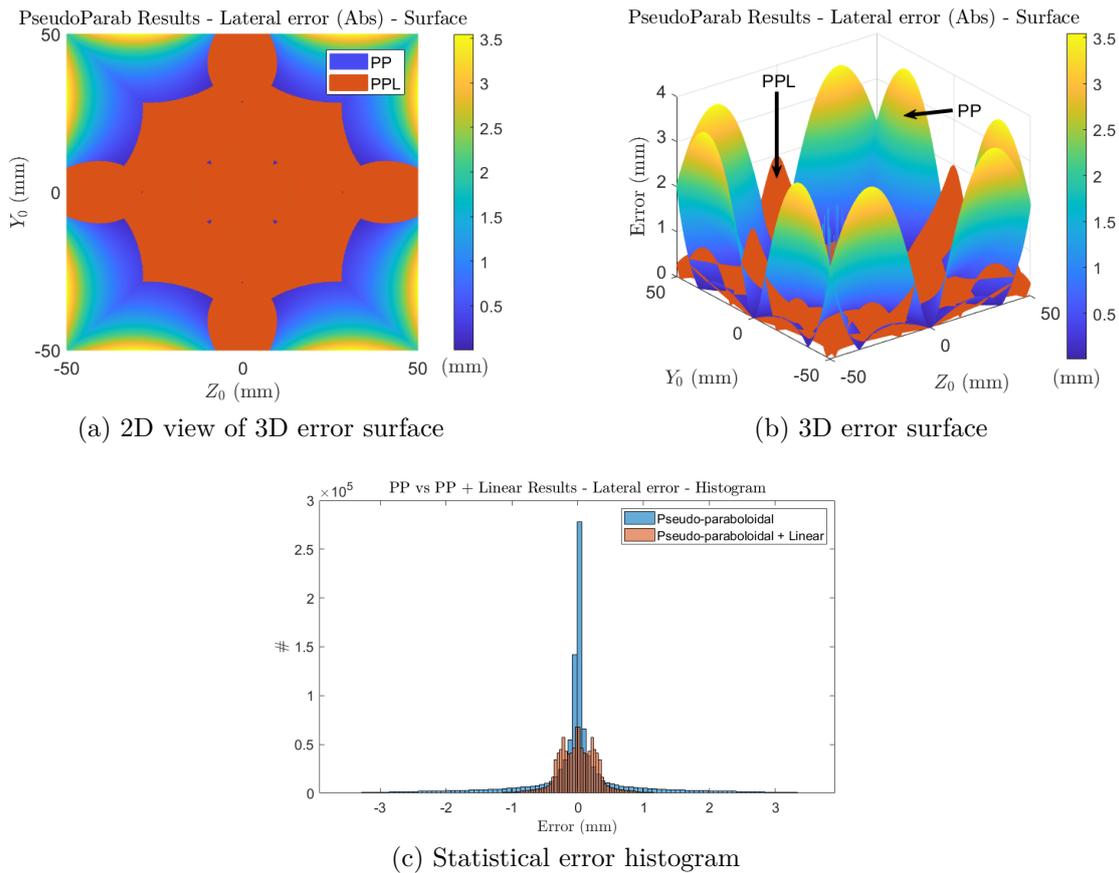


Figure 6.25: Lateral error in the goal box using PP and PPL

A direct comparison of the PP and PPL algorithms is presented in Figure 6.25. Similarly to the results of the alternative algorithm, the combination with the linear algorithm leads to reduced accuracy in the central area of the grid, creating the orange zone in the figure. However, from the 3D surface plot, it is evident how, as the point approaches the edge of the box, the linear algorithm contributes to reducing the height of the error lobes produced by the PP algorithm. This result is consistent with the analysis performed by INAF, where similar area preferences are indicated for the PP and PPL algorithms. Finally, the statistical evaluation shows how the PPL delivers a more uniform and bounded error with respect to the PP.

6.1.11 Goal box - Alternative + PP algorithm

A more accurate result using the alternative algorithm is expected if the pseudo-paraboloidal algorithm is combined with the expensive Cardano's method.

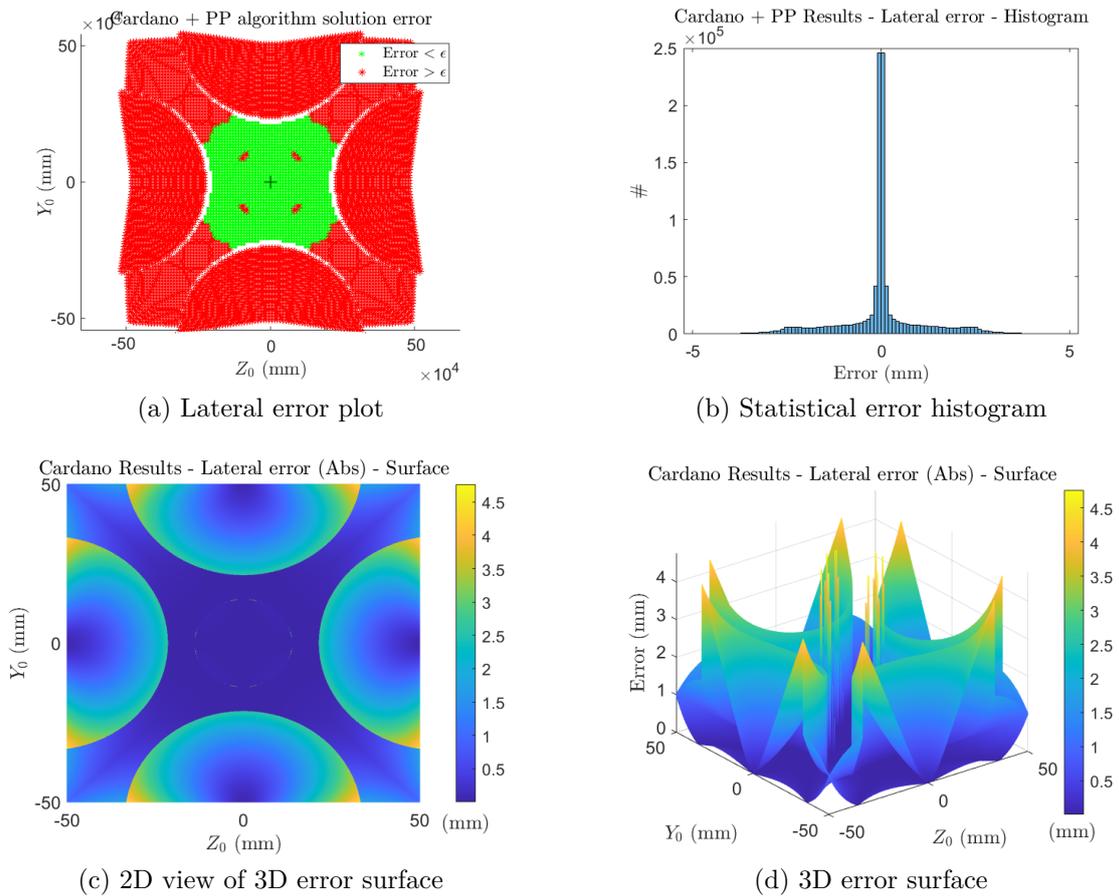


Figure 6.26: Lateral error in the goal box using Alternative + Pseudo-paraboloid

The inner requirement box still satisfies the $50\ \mu\text{m}$ accuracy, and the white zone generated by the alternative semi-circular error close to the edges of the box is reduced. From the statistical evaluation, the error range is reduced from $\pm 5\ \text{mm}$ to around $\pm 4\ \text{mm}$, and there is a single peak in the zero error zone on the histogram.

The 2D view of the three-dimensional error surface is very similar to the alternative, with the exception of the error that peaks close to the edges. Instead, the semi-circles see a more gradual increment of the shift. The corners of the box see a reduction of the error concerning the PP, due to the increased precision of the alternative in the previously identified “X”. The maximum absolute error is still in the 4 mm range, but a significant expansion of the accuracy is expected inside the inner area of the box.

6.1.12 Goal box - Algorithm comparison

To understand which algorithm achieves the best performance inside the goal box, a side by side comparison of the different combinations is presented, employing the three-dimensional surface plots and the statistical error histograms.

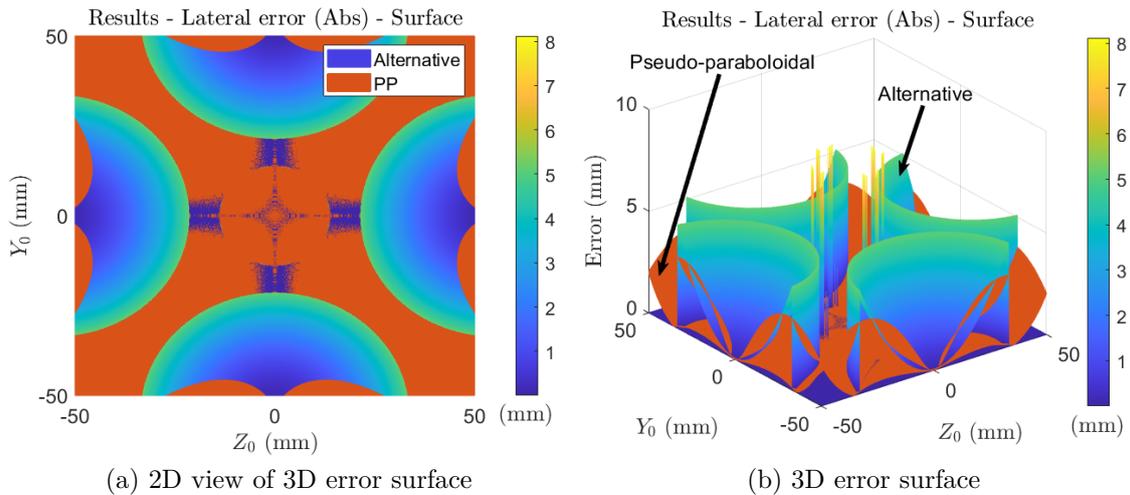


Figure 6.27: Lateral error in the goal box using Cardano and PP

First, in Figure 6.27 the alternative and pseudo-paraboloidal algorithms are compared. Although the Cardano-based algorithm shows a lower error in a wide zone of the goal box, the PP algorithm (in orange) achieves a lower maximum error, opposite to what happened inside the requirement box. Hence, **the validation of the penumbra profile shall prefer the pseudo-paraboloidal (or PPL) instead of the alternative algorithm inside the goal box (especially for coordinates over 20 mm).**

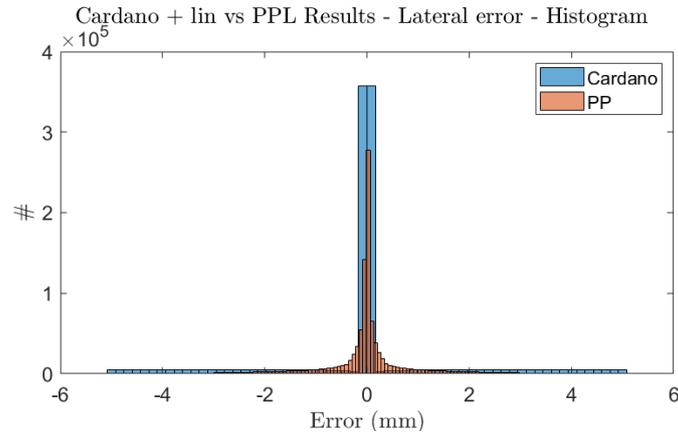


Figure 6.28: Statistical error histogram in the goal box using Cardano and PP

The statistical evaluation clearly shows the extended error range of the alternative algorithm due to the semi-circular zones with high error, but the central “X” achieves a superior accuracy.

The difference of the absolute errors ($|err_{Alt}| - |err_{PP}|$) is computed and presented in Figure 6.29, where it is evident how the error of almost 5 mm, generated by the alternative algorithm (positive value), dominates the surface plot.

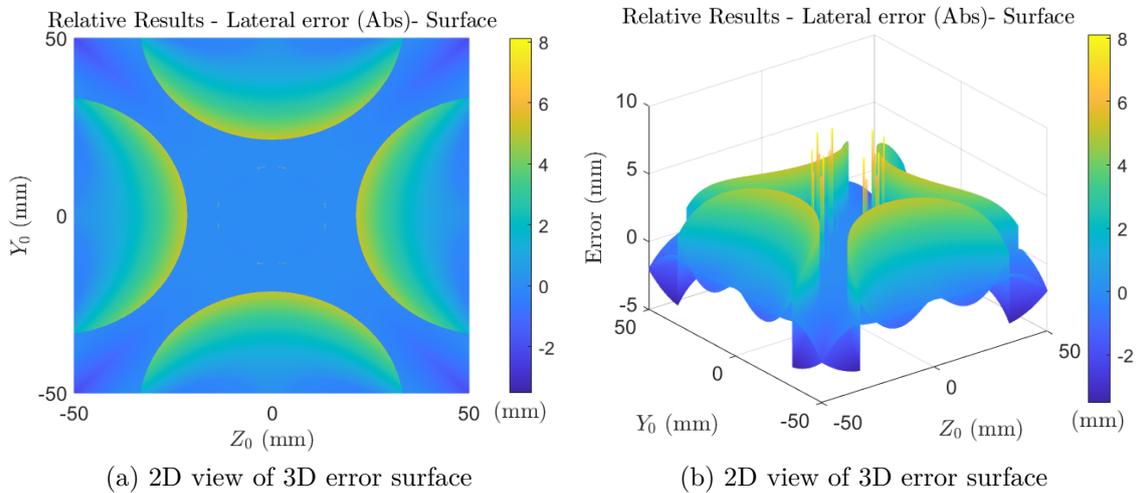


Figure 6.29: Lateral difference of the absolute error in the goal box for Cardano vs PP

Nevertheless, as the points approach the box’s corners, the pseudo-paraboloidal decreases the accuracy, leading to the negative relative error zone of around - 3 mm. From the 2D view, it is evident that the central high precision “X” generated by

the alternative algorithm dominates the inner area; still, the error of the PP in that zone is acceptable at around 1 mm.

Another essential comparison is made between the alternative and PP algorithms combined with the linear. Theoretically, the PPL algorithm shall deliver the best performance in the goal box, with the advantage of the compensating sign and shape of both algorithms errors.

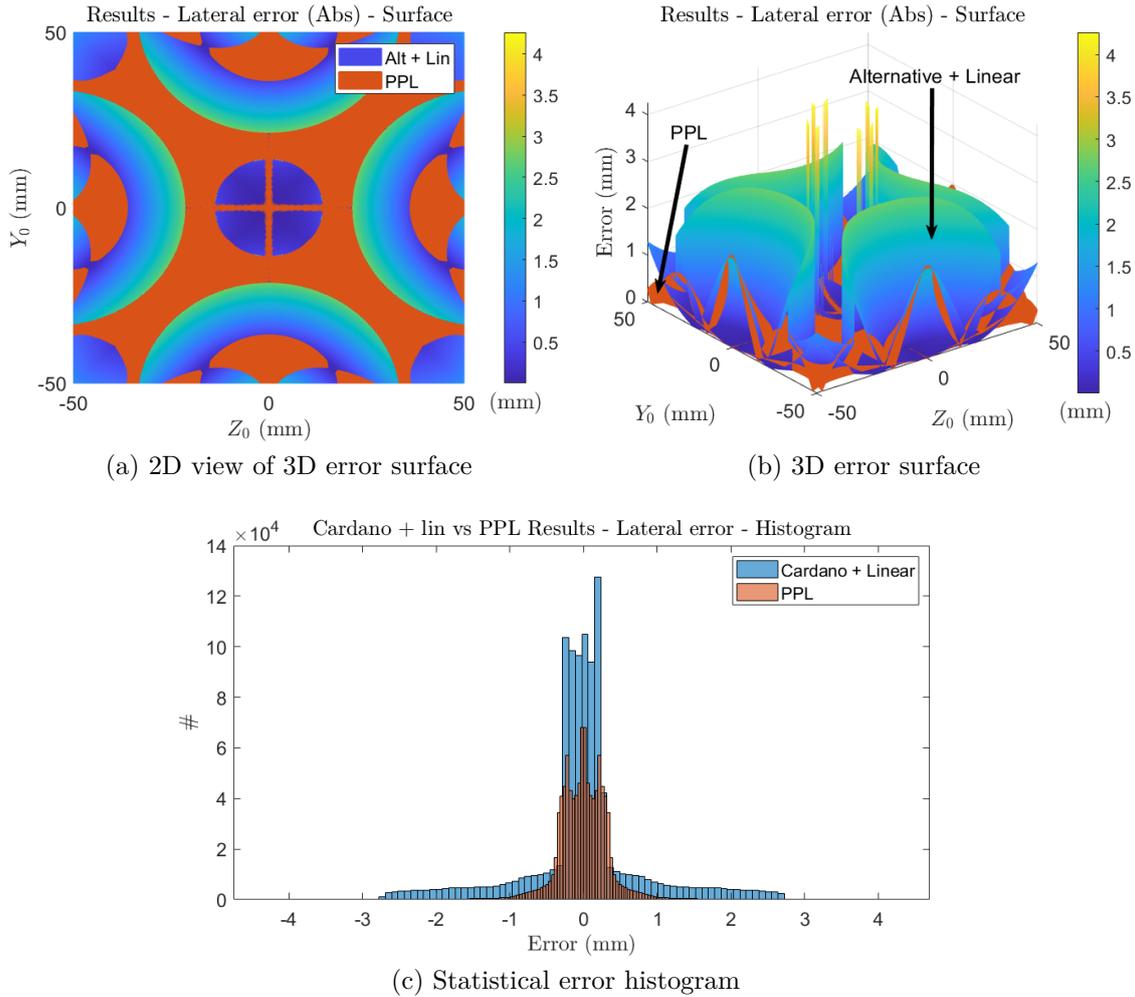


Figure 6.30: Lateral error in the goal box using Cardano + Linear and PPL

From Figure 6.30a, the blue/green zones cover most of the goal box area; **this suggests that the PPL algorithm is capable of providing a lower error with respect to the alternative + linear**. Even inside the requirement box, it is unmistakable how the error compensation by the algorithm developed by INAF allows an increased exactness. Figure 6.30b consents the observation of how the

central edges of the box are dominated by the linear algorithm's error, forming an identical peak, whereas the central cross of the alternative is still noticeable, but it is less accurate as a consequence of the combination with the linear.

The difference of the absolute error (Figure 6.31) exhibits additional proof of the decrease in precision of the PPL as it approaches the edges (compared to the alternative), with a green zone that gradually becomes a darker blue. Moreover, the statistical error evaluation shows how the linear algorithm increases the width of these error distribution bars, generating the three peaks evaluated before. However, it is evident that the **PPL shall be the main algorithm used for the calibration inside the goal box**, followed by the alternative + linear and finally the PP. Differently from the requirement box, the linear algorithm is beneficial for the accuracy if combined with the other algorithms inside the goal box.

As previously disclosed, the linear results obtained with this method leads to a discrepancy with the results obtained by INAF in [5], that used the outdated set of fitting parameters. However, the expected penumbra profile is precisely fitted with this method, and the used algorithm is identical; hence, the results presented are valid.

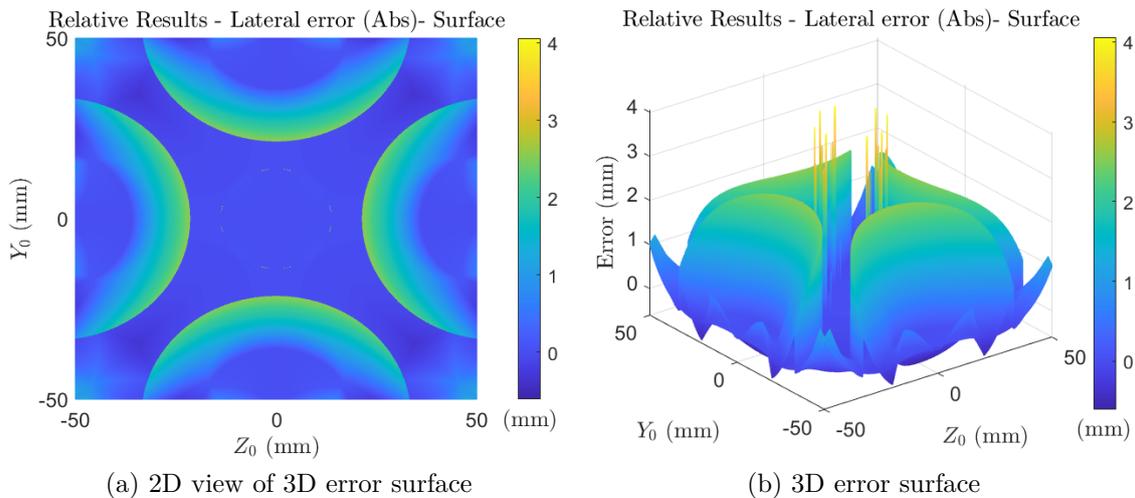


Figure 6.31: Lateral difference of the absolute error in the goal box for Cardano + Linear vs PPL

Using the same order applied for the requirement box, in Table 6.2 the lateral error calculated with the different algorithms is displayed. In this case, the ranking of accuracy is different, with the PPL as the most accurate and the alternative algorithm providing the worst precision in all the categories. In the second place, the alternative + linear algorithm provides a similar standard deviation to the PP, but the maximum and minimum errors are almost 1 mm lower (in absolute value).

The Alt + PP algorithm is an unnecessary addition inside the goal box since it provides an elevated error and the highest computational cost.

Table 6.2: Lateral error of the algorithms and calculation time for 100x100 grid inside the goal box

Algorithm	Min [mm]	Mean & Std [mm]	Max [mm]	Time [s]
Alternative	-5.0977	0.0000 & 1.6487	5.0977	0.3047
Alt + PP	-4.2598	0.0000 & 1.0909	4.2598	0.3952
Pseudo-paraboloidal	-3.5475	0.0000 & 0.8594	3.9343	0.0179
PPL	-1.9653	0.0000 & 0.3493	1.9653	0.0190
Alternative + Lin	-2.7250	0.0000 & 0.8159	2.7250	0.3198
Linear	-3.9343	0.0000 & 0.8389	3.9343	0.0017

It is important to remark that the computational time of the alternative algorithm is lower than the previous analysis because of the use of the default coarse parameters, avoiding the recalculation of the position with the fine.

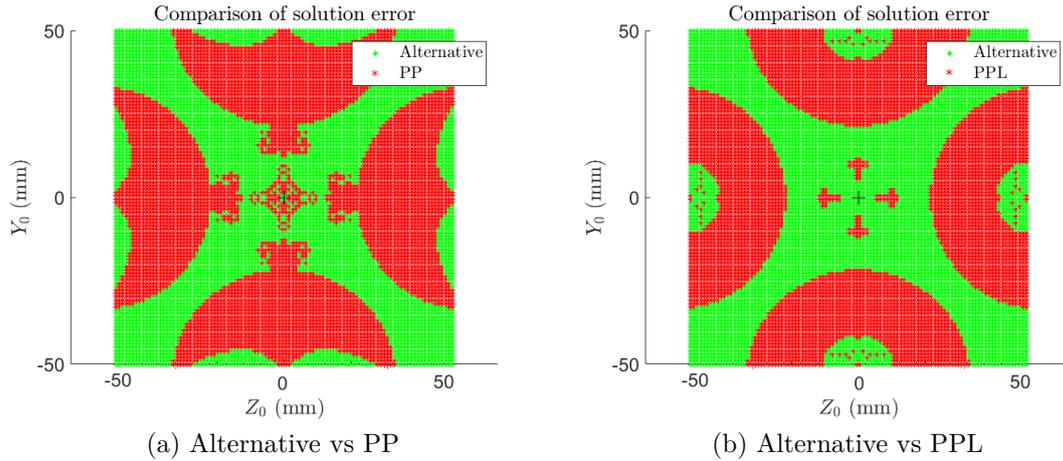
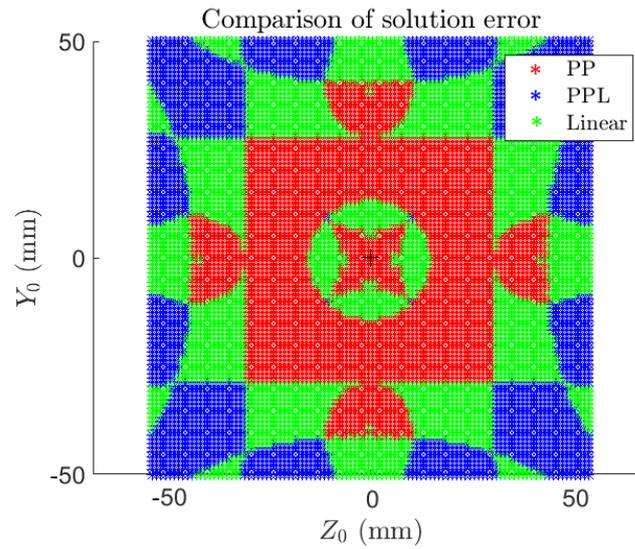


Figure 6.32: Comparison of the precision of the alternative and flight algorithms inside the goal box

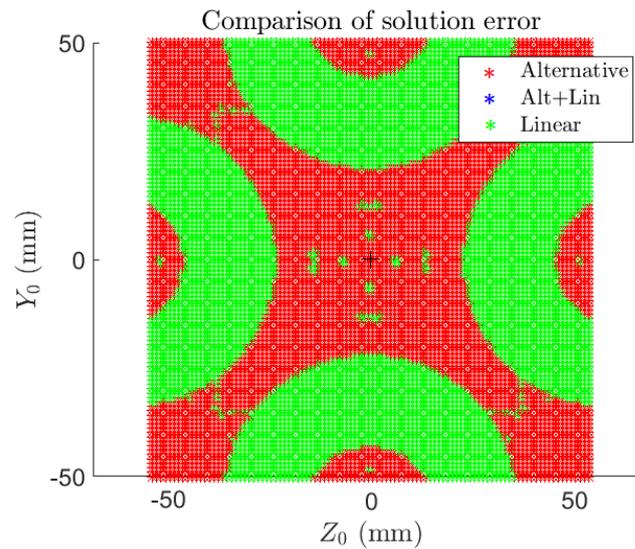
Finally, evaluating the area where one algorithm shall be preferred over the alternative, it is possible to identify the patterns of Figure 6.14 inside the requirement box; where the PPL provided a worst performance with respect to the PP.

From Figure 6.32 instead, as previously discussed, the PPL grants the best accuracy overall, but **the central “X” where the alternative algorithm gets an error lower than $50 \mu\text{m}$ shall be preferred when calibrating and validating the SPS readings.** A similar situation is visible with the pseudo-paraboloidal, but the precision near the centre is higher than the PPL.

An analysis similar to one proposed by INAF is presented in Figure 6.33, where the areas that provide the best solution inside the goal box were divided between the PP, PPL (Both) and linear algorithms. This is also reproduced for the alternative, alternative + linear and linear algorithms, to deliver a visual guide to selecting the most accurate method in a particular area.



(a) Pseudo-paraboloidal



(b) Cardano's method

Figure 6.33: Division of the algorithm's precision inside the goal box

6.2 Longitudinal Error

A secondary result to be analysed is the longitudinal error obtained from both algorithms. This error is not a primary concern for the development of the alternative algorithm because a dedicated script is used to assess the x_0 coordinate. The following results apply only for the nominal inter-satellite distance, with $x_0 = 0$ as the expected result.

Employing Appendix D.2.4 longitudinal computation script, utilising either the flight or alternative algorithm, the computed z_0 and y_0 positions of the umbra centre in the plane are the only necessary inputs. These are used to estimate each SPS irradiance, based on its position, to obtain an approximated R_0 (in DN). Subsequently, using a set of dedicated reconfigurable parameters, the longitudinal position is found as:

$$x_0 = \frac{-K - \sqrt{K^2 - 4H \cdot (L - R_{55})}}{(2H) + (J \cdot d_0^2)}$$

Therefore, the error found for x_0 is proportional to the error in the lateral displacement, with $R_{55}(y_0, z_0)$ and $d_0(y_0, z_0)$. A 1 mm accuracy shall be obtained inside the requirement box as a requirement for the SPS subsystem. However, this level of exactness is difficult to obtain, and other subsystems in the metrology chain shall deliver a more precise measure.

The results in Figure 6.34 show the error inside the requirement box for both algorithms. Comparing these results to the equivalent lateral error surfaces, it is possible to observe that the profile for each longitudinal error plot is equivalent to a “side view” of the three-dimensional lateral error surface. For example, the observed peaks and valleys of the 3D lateral error curve in Figure 6.2b can be recognised in the two-dimensional representation of the longitudinal error in Figure 6.34a.

Consequently, the obtained longitudinal error have an increased accuracy if the lateral error is reduced. From Figure 6.34a, it is possible to determine that **the alternative algorithm does not flawlessly accomplish the 1 mm accuracy demanded by the requirement throughout the box**. However, the maximum error is lower than 2 mm, and most of the results are in the 1 mm range inside the requirement box. A similar situation is visible in Figure 6.34b, where the points near the centre have an error close to 1 mm, although, as the points advance to the border, the error increases up to 10 mm.

The error inside the goal box is unquestionably above the accepted value, for both the alternative and flight algorithms, with a maximum deviation over 1 meter. Hence, the other metrology subsystems shall be used to obtain a high fidelity measure of the longitudinal position, and it will not be further analysed in this work.

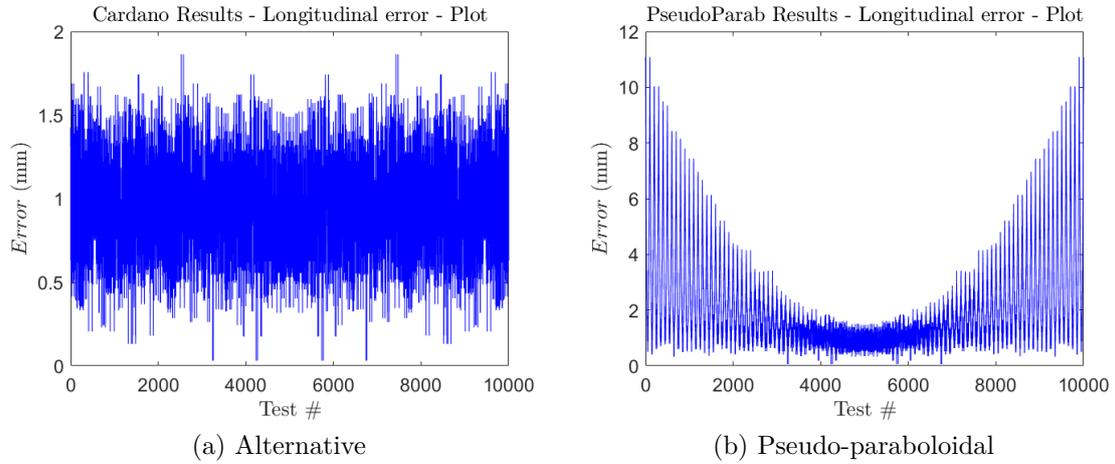


Figure 6.34: Longitudinal error plot in the requirement box

6.2.1 Algorithm comparison

Combining only the alternative and pseudo-paraboloidal algorithms performance inside the requirement box in Figure 6.35, a case analogous to Figure 6.10 is perceived. The alternative algorithm achieves a homogeneous distribution of the error, whereas the pseudo-paraboloid raises the error as it approaches the edges of the box.

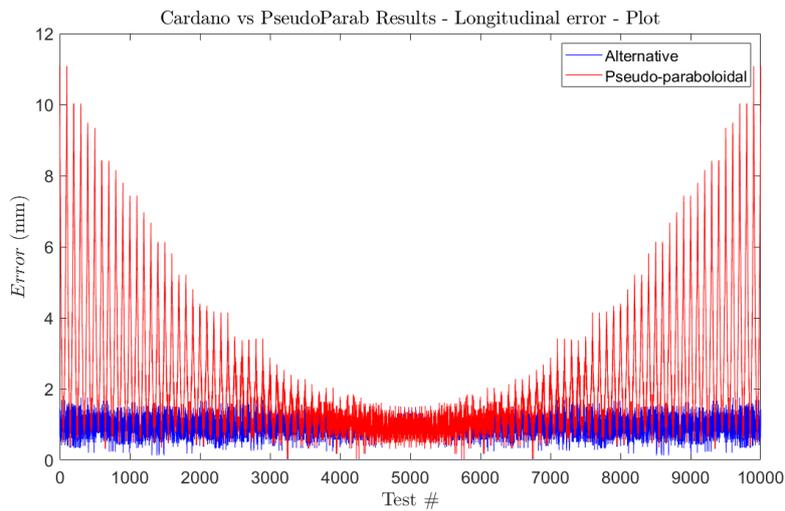


Figure 6.35: Longitudinal error plot in the requirement box using both algorithms

The statistics of the error analysis in Figure 6.36a displays that the alternative algorithm delivers a mean error that peaks in the 1 mm range, by an almost Gaussian distribution between zero and two millimetres. Still, the pseudo-paraboloidal algorithm has a similar peak in the 1 mm error range, but the deviation from the mean value is greater, with a considerable percentage in the 6 mm to 8 mm error range, which is unacceptable inside the requirement box.

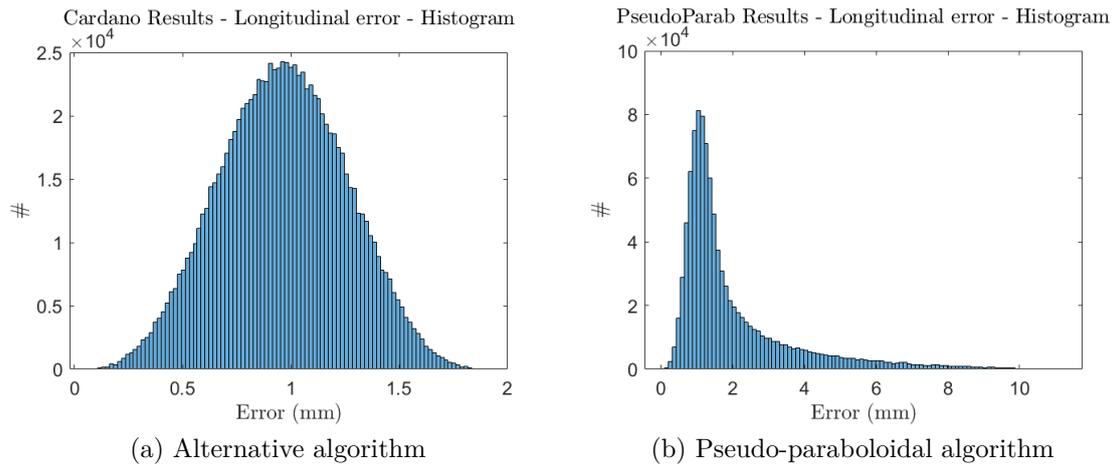


Figure 6.36: Statistic of both algorithm’s error for the longitudinal position with a 1000x1000 grid inside the requirement box

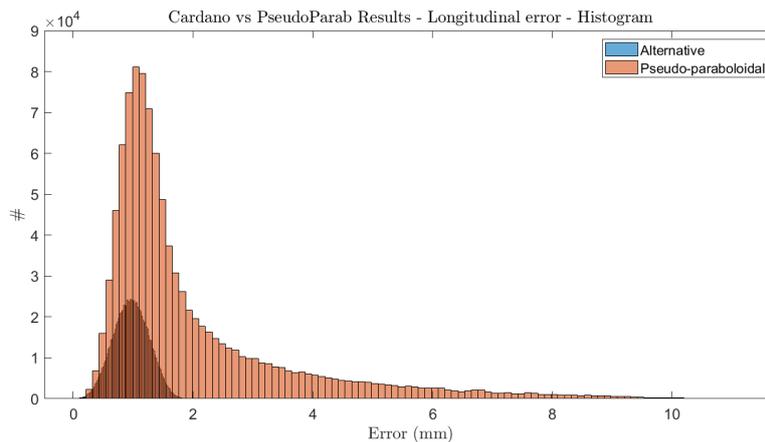


Figure 6.37: Statistic algorithm comparison for the longitudinal position with a 1000x1000 grid inside the requirement box

6.3 Conclusions and remarks

From the achieved results, it is indisputable that the alternative algorithm delivers an accurate and uniform solution for the umbra centre inside the requirement box, although the results in the goal box are not reliable enough. Hence, the main objective of this thesis work, which consisted of the development and testing of an alternative SPS validation algorithm, is proven to be plausible using a locally generated penumbra profile. The next step to be executed is the integration of this algorithm in the SPS simulator, which mimics the functioning of the software during the PROBA-3 mission.

As remarked before, the results obtained with the alternative algorithm are the most dependable among the tested algorithms inside the requirement box. The achieved accuracy allows an error one order of magnitude lower throughout the box. Furthermore, the symmetrical distribution of the error observed in the histograms enables the satisfaction of the requirements.

It is important to remark that the obtained results with the locally generated readings inside the requirement box are consistent with the test performed by INAF for the pseudo-paraboloidal but not for the linear algorithm; this is a consequence of the different penumbra fitting parameters used in the past test. For a conclusive validation of this result, the author performed similar tests using the outdated parameters, achieving similar results to [5], but the outcomes are omitted for simplicity.

The analysis is significantly more complex regarding the goal box since the obtained errors change the shape and precision zones between algorithms. The alternative algorithm delivers an odd semi-circular shaped error that spikes at around 20 mm from the centre, obtaining a maximum error (in absolute value) of almost 5 mm. The pseudo-paraboloidal, linear and PPL algorithm's error follows an almost uniform distribution inside the goal box, with an error that rises as the points approach the edges (or corners).

For this reason, the preferred algorithm in this box shall be the combination of the pseudo-paraboloidal and the linear. This algorithm, developed by INAF, delivers outstanding results with a low computational cost. At the moment this work was prepared, a series of validation tests for this algorithm are being executed using a simulated solar source inside the ALTEC facilities in Turin.

For validation reasons, the alternative algorithm might be preferred inside the central "X" previously identified. This zone achieves exceptional accuracy, unmatched by the other algorithms. An extra algorithm proposal might combine Cardano's method for the points in the 0 mm to 25 mm radius from the centre and the corners, with the PPL covering the remaining area of the goal box. This combination shall allow a more reliable umbra centre position, but it will further complicate the algorithm and increase the computational cost.

Furthermore, from the obtained results is evident how using the most recent

reconfigurable parameters, the linear algorithm struggles inside the requirement box. Instead, the linear algorithm performs adequately inside the goal box, reducing the alternative and PP's mean error. Therefore, the onboard algorithm's output can be improved if only the PP is used inside the requirement box and the PPL is preferred inside the goal box. This modification is relatively easy to implement since the algorithm already executes a position control to change between the fine and coarse parameters.

Chapter 7

SPS Simulator

The results obtained and analysed in the previous chapter arrived from the computation of the umbra centre position using a locally generated penumbra profile; this was conducted at the nominal inter-satellite distance (with $x_0 = 0$), and the penumbra profile employed a set of previously available reconfigurable fitting parameters. The SPS Simulator, developed by INAF, is a powerful tool that allows the determination of the illumination profile, and consequently, the computation of these fitting parameters.

Once the desired shape is determined, a set of input parameters is adopted, such as the SiPM responsivity and material transmittance. The simulator concedes the determination of the expected irradiance readings (in DN and mA) from a position (x,y and z) chosen by the user. This value, and the corresponding fitting parameters, are subsequently used for the inversion of the system and the determination of the umbra centre position in the CI pupil plane.

It is important to note that the previous results (Chapter 6) have higher accuracy with respect to the ones that shall be obtained from the file reading script in this chapter. This is a consequence of the higher number of points (10000 or 1000000) used to interpolate the irradiance curve, where the SPS simulator only uses around 380 points. Hence, a lateral position in the simulator might have the same SPS irradiance value as a neighbour point. In contrast, the locally generated and the expected real data shall achieve higher accuracy.

An essential aspect of the simulator is the possibility of varying the fine or coarse parameters to test the algorithms' robustness. The values can be changed singularly using a linear percentile variation or a Montecarlo method to combine all the variables simultaneously. This simulator was developed for the pseudo-paraboloidal algorithm (and PPL); hence, a function that evaluates single SPS readings with the same format as the algorithm onboard the satellite shall be developed and validated for the alternative algorithm.

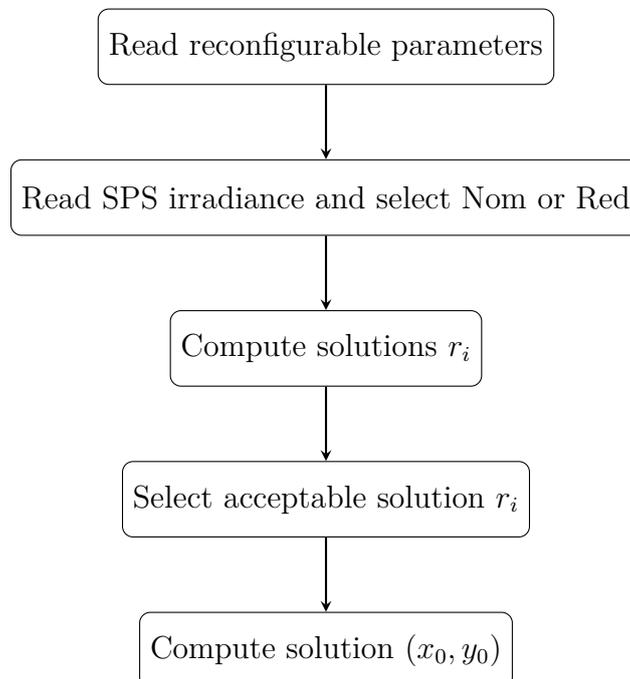
In this chapter, the development of a file reading algorithm is illustrated, alongside a preliminary analysis of the results for a set of random points. Next, the onboard

algorithm's structure is described, emphasising the required flags and checks that shall be performed for the alternative algorithm, using the same inputs and outputs as the flight algorithm. Finally, once the SPS simulator has been adapted for the new algorithm, the robustness is tested using both the fine and coarse parameters.

7.1 File reading algorithm

The first step towards developing the final algorithm to be implemented at the SOC of the PROBA-3 mission is constructing a “File reading algorithm”. The main difference of this Matlab script, concerning the one used in the previous chapter, is the interpretation of a single set of SPS irradiance readings instead of a mesh (An example is given in Appendix D.5). Furthermore, the reconfigurable parameters for the linear, alternative and pseudo-paraboloidal algorithms are imported from an external file (in the format of a “.json” file), following the structure of the onboard algorithm and simulator, present in Appendix D.4.

The uniformity of the inputs and outputs of the scripts used for the alternative algorithm and the pseudo-paraboloidal is a fundamental request in the development of this product. As a consequence, the following structure must be obeyed:



In Appendix D.3 the Matlab code used is described in detail. For the performed tests, twelve sets of SPS readings are analysed. The expected umbra centre positions are inside the requirement box, and the nominal (NOM) or redundant (RED) readings can be selected, setting the specific variable inside the script. The alternative algorithm proposed by ESA in [22] used only the nominal set. Nevertheless, in order to allow the use of the redundant set of SPS, the author modified the script, performing the computation and rotation of the umbra centre by selecting the desired SPS set, maintaining the same structure as the onboard algorithm (Appendix D.3.3).

The next step, given in Appendix D.3.2, selects the desired set of readings and extracts the high gain (HG) digital irradiance values. A critical remark is needed since the terminology previously used to classify the SPS saw a division by letters (from A to D), whereas the flight algorithm and the simulator use the numerical designation of Figure 3.4. Hence, if the nominal set is used, the A, B, C and D readings shall correspond to the 7th, 1st, 3rd and 5th from the “RAW_DATA.json” (excluding the low gain readings).

Finally, the lateral and longitudinal umbra centre positions (x_0 , y_0 and z_0) are obtained using the alternative algorithm described in Chapter 5, where four radii are computed and compared to the boundary conditions. A check is performed to examine if more than one solution is received, and if desired, the solutions can be combined with the linear algorithm.

7.1.1 Results from example input

Using twelve random points inside the requirement box, the performance of the alternative, alternative + linear and pseudo-paraboloidal algorithms are presented in Table 7.1, 7.3 and 7.2 respectively.

Table 7.1: Lateral error for Alternative algorithm

Test #	Expected Position [mm]		Calculated Position [mm]		Absolute error [mm]		Percentage error [%]	
	Y_{exp}	Z_{exp}	Y_0	Z_0	Y_{err}	Z_{err}	$Y_{\%}$	$Z_{\%}$
1	0	0	0	0	0	0	0	0
2	1.4	0.5	1.5392	0.51458	-0.13916	-0.014577	-9.94	-2.9154
3	4.52	1.9	4.6364	2.0552	-0.1164	-0.15516	-2.5753	-8.1663
4	-5.59	2.4	-5.6663	2.5897	0.076337	-0.18966	-1.3656	-7.9026
5	8.52	4.7	8.7881	4.669	-0.26813	0.031035	-3.1471	0.66033
6	0.18	0.9	0	1.0294	0.18	-0.12937	100	-14.3749
7	2.3	0.6	2.572	0.51458	-0.27199	0.085423	-11.8256	14.2372
8	8.9	2.5	9.3101	2.3371	-0.41006	0.1629	-4.6074	6.516
9	1.7	3.4	1.5392	3.5982	0.16084	-0.19817	9.4612	-5.8287
10	6.5	4.6	6.7013	4.669	-0.20127	-0.068965	-3.0965	-1.4992
1	3.3	-7.5	3.6316	-7.749	-0.33157	0.24895	-10.0477	-3.3193
12	1.1	-2.1	1.0294	-2.0552	0.070626	-0.04484	6.4205	2.1352

Table 7.2: Lateral error for Pseudo-paraboloidal algorithm

Test #	Expected Position [mm]		Calculated Position [mm]		Absolute error [mm]		Percentage error [%]	
	Y_{exp}	Z_{exp}	Y_0	Z_0	Y_{err}	Z_{err}	$Y\%$	$Z\%$
1	0	0	6.6405e-15	6.6405e-15	6.6405e-15	6.6405e-15	0	0
2	1.4	0.5	1.5257	0.51024	-0.12569	-0.010236	-8.9782	-2.0473
3	4.52	1.9	4.583	2.0366	-0.062953	-0.13655	-1.3928	-7.1871
4	-5.59	2.4	-5.5921	2.5805	0.0021403	-0.18049	-0.038287	-7.5203
5	8.52	4.7	8.6132	4.6432	-0.093237	0.05683	-1.0943	1.2092
6	0.18	0.9	6.6405e-15	1.0206	0.18	-0.12058	100	-13.3975
7	2.3	0.6	2.5478	0.51024	-0.24784	0.089764	-10.7755	14.9606
8	8.9	2.5	9.1125	2.3357	-0.21254	0.16435	-2.388	6.574
9	1.7	3.4	1.5257	3.5612	0.17431	-0.16119	10.2533	-4.7408
10	6.5	4.6	6.6005	4.6432	-0.10054	-0.04317	-1.5468	-0.93847
11	3.3	-7.5	3.6157	-7.6147	-0.31574	0.11466	-9.568	-1.5289
12	1.1	-2.1	1.0206	-2.0366	0.079423	-0.063445	7.2203	3.0212

Table 7.3: Lateral error for Alternative + Linear algorithm

Test #	Expected Position [mm]		Calculated Position [mm]		Absolute error [mm]		Percentage error [%]	
	Y_{exp}	Z_{exp}	Y_0	Z_0	Y_{err}	Z_{err}	$Y\%$	$Z\%$
1	0	0	0	0	0	0	0	0
2	1.4	0.5	1.5258	0.51025	-0.12577	-0.010249	-8.9835	-2.0497
3	4.52	1.9	4.5841	2.0367	-0.064078	-0.13673	-1.4177	-7.1963
4	-5.59	2.4	-5.5942	2.5731	0.004202	-0.17309	-0.07517	-7.212
5	8.52	4.7	8.6217	4.63	-0.10173	0.070039	-1.194	1.4902
6	0.18	0.9	0	1.0206	0.18	-0.12061	100	-13.4008
7	2.3	0.6	2.5481	0.51025	-0.2481	0.089751	-10.7871	14.9586
8	8.9	2.5	9.1222	2.3257	-0.2222	0.17429	-2.4966	6.9717
9	1.7	3.4	1.5258	3.5617	0.17423	-0.16174	10.2488	-4.7569
10	6.5	4.6	6.6041	4.63	-0.10413	-0.029961	-1.6021	-0.65133
11	3.3	-7.5	3.6053	-7.6204	-0.30535	0.12044	-9.2529	-1.6059
12	1.1	-2.1	1.0206	-2.0367	0.079392	-0.06327	7.2175	3.0129

It is important to emphasise that the SPS irradiance read from the “RAW_DATA” file is obtained from the simulator, where the high gain values used are reported in Appendix D.6. Hence, the digital readings used as input derive from a penumbra profile fitting that is not remarkably precise since it employs around 350 points. Consequently, the “expected positions” here presented are used to approximate the DN values by the simulator, but these values might not correspond to the precise position converted by the Cardano’s or pseudo-paraboloidal methods. In most cases, the absolute error obtained is higher than the 50 μm requirement; still, the result is just a preliminary test that shall be validated by comparing the outcomes to the pseudo-paraboloidal and adopting a new set of DN inputs from the simulator.

From these results, it is evident how the performance of both algorithms is very similar, leading to an error range from 1 to 10 per cent for all the positions analysed. The difference between the algorithms fluctuates in one percentile point; this slight discrepancy is well under the precision of the used penumbra fitting. For example, the test value #6, with a Y coordinate of 0.18 mm, gives an irradiance value of 1034; this is the “zero” point of the curve, and the algorithms correctly convert this value to a zero coordinate. Therefore, the number of points used to interpolate the

curve does not have the necessary resolution to see the almost 0,2 mm variation of the centre and a 100% error is obtained.

A crucial result not shown in the tables is the computational time required for each algorithm to compute a single position. The results obtained in the previous chapter, evaluating the entire grid, showed one order of magnitude difference between the alternative and pseudo paraboloidal. This is once again valid, with a **mean computational time of 0.001071 seconds for Cardano's and 0.0001478 seconds for the PP**. Hence, the computation is well under the half-second refresh rate of the SPS readings, but it is evident how the optimised onboard algorithm has a clear lead when computational costs are compared. These results were obtained running the Matlab code on the author's computer; thus, the expected computational time using the satellite's OBC and the embedded code might be different.

Furthermore, the alternative + linear results are more similar to the pseudo-paraboloidal results than the single alternative. This reduction of the error, of almost one per cent, does not coincide with the results obtained in the previous chapter, where the alternative + linear algorithm lead to lower accuracy. This might be a consequence of the low resolution of the readings, where Cardano's method follows the read value precisely, but this one does not correspond to the exact point. Another possibility of this discrepancy might be a slight difference in the fitting parameters used in the penumbra profile generation of the simulator and the locally generated data. Nevertheless, this test's objective was to develop a file reading algorithm following the flight algorithm structure. The evolved script successfully achieved this task, even if the results were not precise enough. Consequently, it is possible to advance into the next step of developing the alternative algorithm simulator.

7.2 SPS Simulator structure

To develop the alternative algorithm simulator, it is essential to understand the structure of the onboard software developed for the SPS subsystem. The following considerations are based on the latest (Settembre 2021) SPS Software Interface Document - Version 6.2 developed by INAF in [8].

The five blocks present in Figure 7.1 represent the various algorithms and functions that shall be executed for each set of irradiance readings to obtain the umbra centre position. The sequential process has been carefully divided, with specific inputs and outputs that depend on the SPS readings (Appendix D.5), the reconfigurable parameters (Appendix D.4) and the previous position measured by the algorithm. A detailed description of the functions, inputs and outputs are available in [8], and in Appendix D.7 the Matlab script used for the whole process is presented. A description of each block of Figure 7.1 is provided in the following sections.

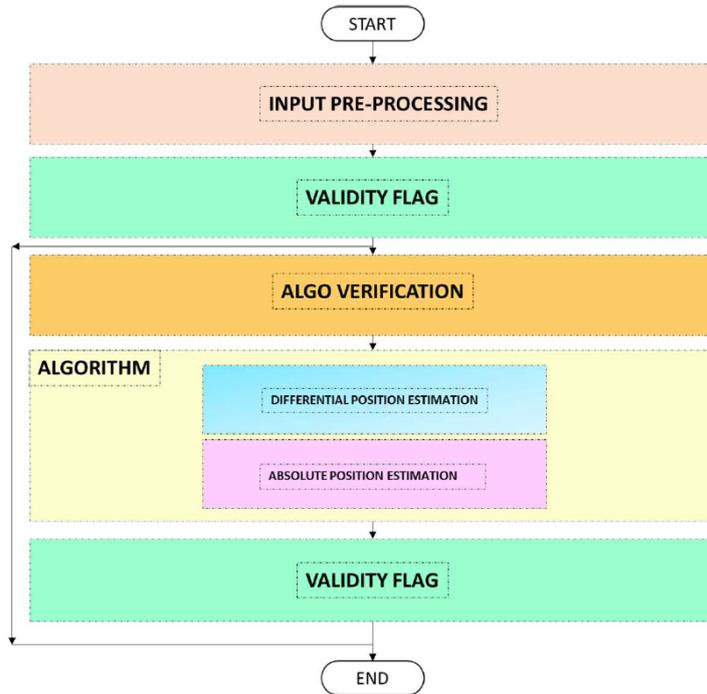


Figure 7.1: General architecture of the algorithm [8]

7.2.1 Input Pre-Processing

The first block evaluated by the algorithm receives the SPS measurements from the FPGA (a vector of 16 digital values) and the reconfigurable parameters stored onboard. Then, the algorithm verifies which set of SPS should be used for the position computation between nominal or redundant (or both) sets. For each reading, the 12-bits-depth DN measure in High Gain is compared to a threshold; if it is higher than 4000 (i.e. the HG reading is close to the saturation, 4095), the value is discarded and the Low Gain values are considered and multiplied by the g_i conversion gain to be used in continuity with the HG values. This step allows the system to maintain the desired accuracy for readings inside the requirement and goal box. The g_i coefficients are a vector of 8 values that consider the HG/LG variations of each SPS readout-chain during the mission ($g_i \approx 5$). Finally, this algorithm passes the SPS readings values to be used to the following block.

The implementation of this block does not resent the variation of the algorithm to be used for the position computation; thus, it will not be modified for the alternative SPS simulator.

7.2.2 Validity Flag 1

The second block involves a validity verification performed on the SPS readings, and it is preparatory for the relative position computation. Additionally, the block is responsible for the verification of the position of the door that covers the ASPIICS payload. For this reason, the “*door_tr*” parameter indicates with a boolean value whether the door is in transition (casting a shadow that invalidates the readings), and another parameter indicates if the door is open or closed. “Moreover, if the measurements have been acquired using the nominal set, the first step is to check the open/closed door status. In particular, if the door is closed, the values acquired by the nominal set are not valid for the GNC purposes (they are used on-ground for calibration). In fact, in this case, the SPSs of the nominal set will be covered by the neutral density filters. On the other hand, the SPSs of the redundant set, which is not covered, will be used for GNC tasks and for on-ground calibration (e.g., cross-calibration with the nominal set).” [8].

The algorithm checks if the SPS readings are valid and if the SiPM’s temperature is in the expected range before proceeding to the computational processes in the subsequent blocks. The design of the algorithm requires that the measurements of at least two diodes shall not be zero or be above a threshold of $4000 * g_i$ (for LG), where the SiPM responsivity is taken into account by the coefficients k_i .

If one of the aforementioned checks fails, a flag is set to zero, and the computation stops; otherwise, the data are passed to the Algo Verification block. Once again, this block will not be modified for the alternative algorithm.

7.2.3 Algo Verification

The third block is intended to evade using the pseudo-paraboloidal algorithm when the resulting computed position is undoubtedly affected by mathematical problems. As explained in Chapter 3, the pseudo-paraboloidal algorithm’s solution requires the computation of square roots and the *arc cosine* trigonometric function; therefore, a control of the inputs shall be performed to avoid complex numbers in the solution. This control is performed only if the PP or PPL algorithms are used; if exclusively the linear is implemented, this control is skipped.

The output of this algorithm is a vectorial flag called “*pp_check*”, containing the boolean outcome of the four computations of:

$$-1 < \arg(\arccos) < 1$$

Obtaining $pp_check(i) = 1$ for $i=0,1,2,3$ if the condition is satisfied. Oppositely, if complex numbers are expected, only the linear algorithm is used.

The alternative algorithm does not require this check since the inversion of the system inside the required boundary conditions leads to a positive Delta (Section 4.1.4), where only the square root of a positive real number is performed. Even if the

expected Delta has a negative sign, the controls performed within the alternative algorithm shall discard all complex solutions obtained. Consequently, the “pp_check” is performed only on the PP or PPL algorithms, and the inputs and outputs of this block will not be modified in order to maintain the same structure as the flight algorithm. In other words, the “pp_check” remains, but the alternative algorithm function will not employ it.

7.2.4 Algorithm implementation

The fourth block runs the metrology code, which is composed of two different algorithms executed in sequence:

- Absolute algorithm: Provides the desired umbra centre position.
- Differential algorithm: Provides a coarse calculation of the misalignment direction for a pair of SPS.

The lateral and longitudinal position coordinates of the umbra centre are calculated using the algorithm described in Chapter 3. A flag (the “algo” parameter), configurable via on-ground telecommand, indicates whether the PP, PPL or linear algorithms have to be used.

The outputs of the two codes are gathered in the same vector and then transferred to the validity flag routine to verify the computed absolute position.

Differential Position Estimation

The first code implemented by the block is the differential position estimation, which has been developed to determine the misalignments between the two spacecraft. “The output is qualitative and based on the difference of signal acquired by opposite SPSs. Since we do not have a perfect knowledge of the actual penumbra profile that the SPS will experience in space, this algorithm is helpful to retrieve qualitative information of satellite relative position and analyse the algorithm’s performance for the absolute position estimation.” [8].

This part of the code shall remain unaltered during the implementation of the alternative algorithm.

Absolute Position Estimation

The first duty of this algorithm is the control of the “pp_check” flag previously obtained for the coarse or fine fitting parameters. If “pp_check” for the coarse parameters is equal to zero (not satisfied), the procedure ends. Oppositely, if it is equal to one, the positions are calculated with the coarse parameters. As explained in the previous chapter, the fine fitting parameters allow increased precision in the inner zone of the umbra determined by the “delta_trans” value of 13.8 mm. Suppose

the umbra centre position computed using the coarse fitting lies within the inner zone. In that case, i.e. $\sqrt{z_0^2 + y_0^2} \leq \Delta_{trans}$ the “*pp_check*” is implemented once again in order to verify and calculate the position using the fine fitting parameters.

The “*algo*” flag is evaluated, and the desired algorithm is used for the lateral computation. This part of the code shall be modified for the alternative algorithm, substituting the pseudo-paraboloidal and PPL with the alternative and alternative + linear. It is very useful to maintain an identical structure of the functions inside the code, where:

```
[z_0_PP, y_0_PP, assign_value] = ...
paraboloid_calc(pp_coarse, pp_fine, pp_check, R1, R2, R3, R4);
```

Becomes:

```
[z_0_PP, y_0_PP, assign_value] = ...
alternative_calc(pp_coarse, pp_fine, pp_check, R1, R2, R3, R4);
```

It is possible to see that the alternative algorithm function maintains the same inputs and outputs, even though the “*pp_check*” vector is not used in the computation.

The Matlab script of the modified function for the SPS algorithm that uses both the fine and coarse parameters is reported in Appendix D.8. The principal difference in the aforementioned script lies in the substitution of the function previously noted.

The “*alternative_calc*” function is an adaptation of the “Read” algorithm developed in the first section of this chapter, where the fine and coarse parameters are loaded together with the four SPS readings (NOM or RED) to be inverted. The same procedure is followed, where the four radii of the SPS are determined, and the solutions are checked to remain inside the boundary conditions, and to be real numbers.

To inform the main algorithm that the found solutions are valid or not, the same flag employed by the flight algorithm (“*assign_value*”) is maintained as an output of the function. This flag determines if the obtained solution is valid; otherwise, the -999999 (fake) position shall be retrieved in the presence of a non-valid output. The flight algorithm changes this flag according to the “*pp_check*” values; hence the alternative algorithm shall implement a different approach for this control.

In the alternative algorithm, the checks of the solution’s validity are performed at the end of the computation, where the flags of each of Cardano’s method implementations and the number of solutions are controlled. If a non-valid situation is encountered, the “*sol_flag*” variable is set to one, and the algorithm shall change the “*assing_value*” flag accordingly.

An important remark is needed since a couple of “new” reconfigurable parameters shall be added to the original list. The values for R_0 using the fine and coarse parameters are fundamental for the functioning of the alternative algorithm, and they can be retrieved from the other available longitudinal and lateral parameters. However, the inputs of the function see only the lateral parameters (“*pp_fine*” and “*pp_coarse*”); hence, to maintain the structure, the R_0 values shall be added to each struct, avoiding a recalculation of the same parameter at each run and an alteration of the inputs. An example of how the R_0 values are obtained and organised in the reconfigurable parameter’s structure is available in Appendix D.7.

Furthermore, the original Cardano’s method script in Appendix D.1 performed a check of the real values of the Cardano’s parameters. This step can be avoided since the parameters used are the reconfigurable parameters (determined by ground) and the irradiance difference $R_i - R_0$ that shall be a real value. However, the control is maintained as a first iteration of the code, keeping the additional computational cost.

7.2.5 Validity Flag 2

The final block of the algorithm uses the retrieved positions (in the previous and current steps) and the reconfigurable parameters to perform a conclusive validity check. Every 0.5 seconds, a new SPS readout is obtained and processed; this block checks whether the computed lateral and longitudinal positions are valid based on the displacement and the lateral drift of the spacecrafts between the two time steps.

“Based on the maximum achievable velocity, if the lateral and the longitudinal displacement between two subsequent computed positions exceeds the maximum achievable due to the expected drift velocity, the evaluated position is considered not valid. Additionally, from the computed lateral position, if the just computed value indicates that the umbra centre is outside the circumference of a defined radius where the error associated with the longitudinal position is critical for the GNC corrections (i.e., the sign of the computed position is the opposite of the real one), the longitudinal position is considered not valid.” [8].

The alternative or pseudo-paraboloidal algorithm does not influence this final block of the algorithm; thus, it will not be altered.

7.3 Alternative SPS simulator

The SPS simulator is a great instrument that allows the analysis of the algorithm and its parameters. Furthermore, it provides the possibility of testing new illumination profiles and fitting parameters to calibrate and validate the results coming from the PROBA 3 satellites. For the alternative algorithm, the development of the SPS simulator is done by adapting the one created for the pseudo-paraboloidal, allowing a simplification of the workload as well as guaranteeing the uniformity of both simulators.

Furthermore, the alternative simulator implements both algorithms to compare and analyse in real-time the performance of the alternative, pseudo-paraboloidal and its combinations with the linear; this is possible using the available checkboxes that permit a simple transition between the algorithms. Moreover, the alternative algorithm is available in two configurations: the default one employs a combination of the fine and coarse parameters used in the previous chapter, but there is the option to use only the fine parameters, as in Appendix A.

The functions modified in the previous section are of fundamental importance for the alternative simulator. Once its functioning and structure have been tested, it is possible to adapt them to the alternative SPS simulator Matlab app.

The simulator operates by loading (or calculating) the desired penumbra profile; this allows the user to choose a specific umbra centre position in X, Y and Z and compute the expected SPS readings for the nominal and redundant sets. From the SPS irradiance readings (in DN), the simulator enables the inversion of the system and the resolution of the umbra centre position with the desired algorithm and reconfigurable parameters. [1] provides a deeper understanding of the structure and functioning of the SPS simulator.

Since the functions used by the simulator are identical to the ones described before, the simulator will allow the observation of the flags and checks of the algorithm to identify and solve possible issues. Furthermore, the simulator is equipped with the option of evaluating the variation of the solution when the profile fitting parameters are modified on purpose from their nominal value; this leads to the subsequent sensitivity tests that shall compare the robustness of the inversion of both algorithms solutions.

The initial tests performed with the alternative SPS simulator compared both algorithms performance inside the requirement and goal box (similar to the previous chapter) using random points. It is possible to conclude that both algorithms achieve a similar performance (same order of magnitude of the error), where certain zones see an advantage of the alternative over the pseudo-paraboloidal and vice-versa. It is essential to bear in mind that the simulator approximated fitting does not achieve a high enough resolution to observe smaller irradiance steps. Therefore, it is difficult to compare in detail the algorithm's performance; thus, the results in Chapter 6 become crucial to the objectives of this work.

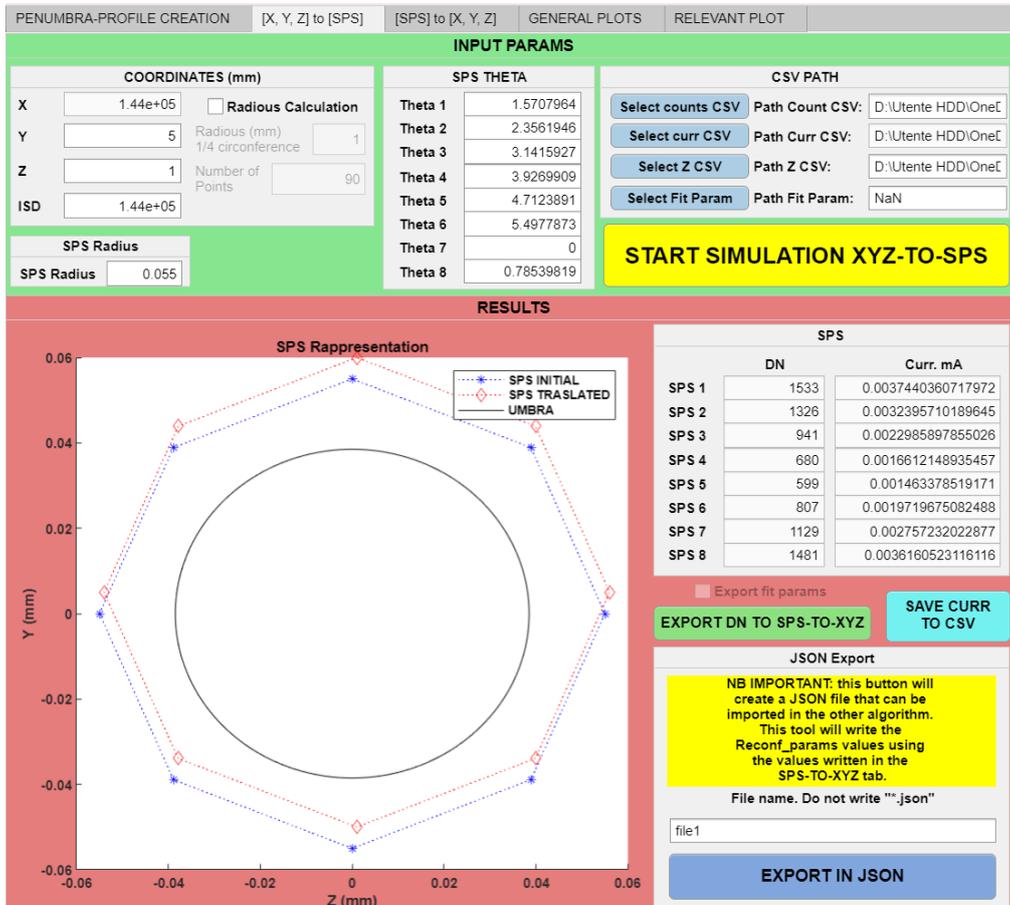


Figure 7.2: SPS alternative simulator: “[X, Y, Z] to [SPS]” at Y= 5 and Z= 1 mm

Due to the simulator’s characteristics, which allows only one position to be tested at each time, it is challenging to evaluate an entire grid as presented in the previous chapter. Therefore, a high-level evaluation of the positions assessed by the simulator makes it possible to validate the results obtained with the locally generated readings. However, a more extensive series of tests are needed to conclusively validate the alternative algorithm functioning inside the requirements and goal box.

From Figure 7.2 it is possible to identify the different parts of this section of the simulator, where the parameters are loaded (“CSV PATH” box), the desired coordinates are chosen (“COORDINATES” box), and the irradiance output is calculated (“SPS” box). Furthermore, the “SPS Rappresentation” box provides a visual interpretation of the position of the SPS sensors, allowing a simple understanding of the displacement.

7.3.1 Sensitivity test

The sensitivity test objective is to identify the range where the reconfigurable parameters can be modified, and a valid solution is still achieved. This is essential since the penumbra profile is not perfectly known. The fitting parameters used will be calibrated and changed throughout the mission to precisely follow the illumination encountered. Therefore, a more extensive range leads to enhanced flexibility of the algorithm at inverting the system. Furthermore, another evaluation is possible, where the parameters are changed and the achieved error is compared to the accuracy requirement of 50 μm . This provides an indication of the precision needed by the ASPIICS team for the determination of each reconfigurable parameter of the fitting.

The following results were obtained using the fitting parameters available in Appendix D.4, at the position $x_0 = 0$ mm and $y_0 = z_0 = 1$ mm. The detailed procedure followed and the error plots for each of the parameter’s variations are available in Appendix B, and a screenshot of the simulator is shown in Figure 7.3.

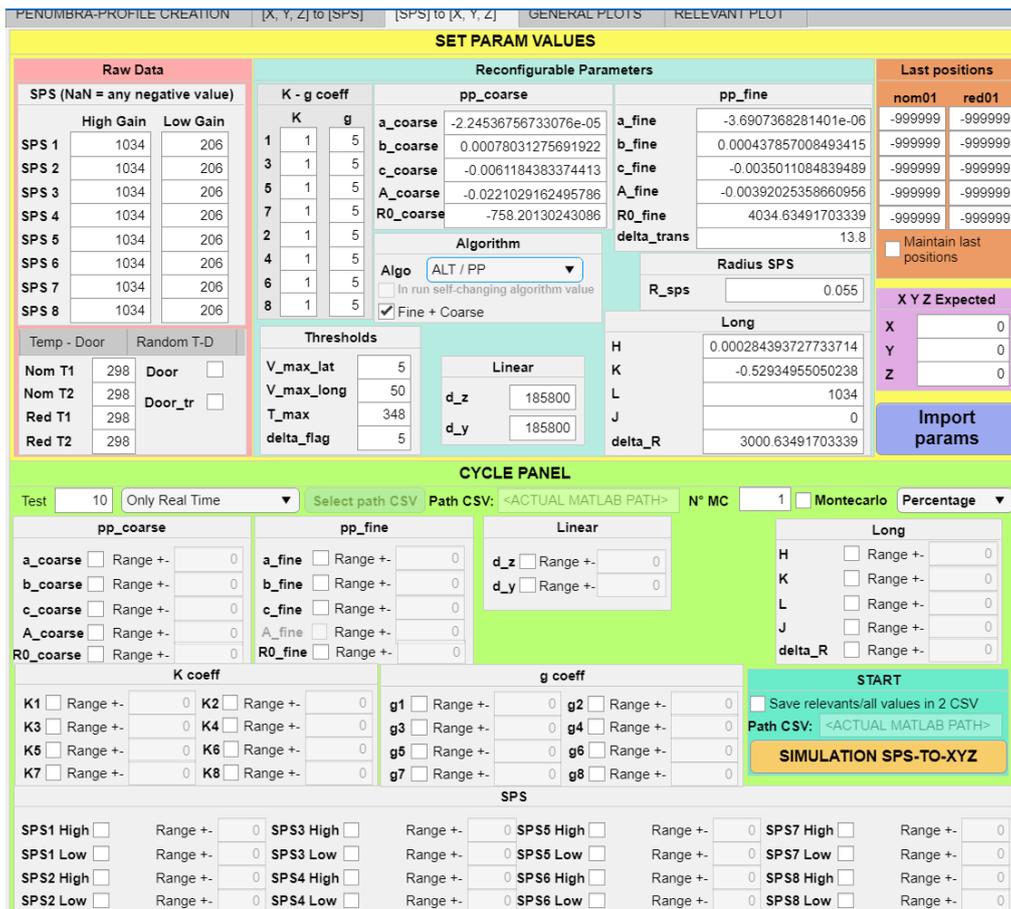


Figure 7.3: SPS alternative simulator: “[SPS] to [X, Y, Z]”

The simulator enables the fine and coarse parameters variation using a user-defined range in the “CYCLE PANEL” box. The inputs are the irradiance values in the “SPS” block in Figure 7.2, imported into the “Raw Data” box of Figure 7.3 and the reconfigurable parameters (from the “.json” file in “Reconfigurable Parameters” box).

Then, a specific number of tests is chosen. The simulator will iterate the parameter selected (enabling the checkboxes) for the sensitivity test, modifying it from the lowest to the highest value. The following results were acquired through 500 tests; thus, from the chosen percentage of variation ($\pm\%$), the parameter used will be increased at each iteration by the same amount, and the outcome will be saved and displayed in a table (or a “CSV” file). The outputs provided by the simulator are the coordinates and flags performed at each test, including the absolute and differential position estimation. Furthermore, the “GENERAL PLOTS” tab enables the evaluation of the lateral and longitudinal error and the composition of a plot using two user-chosen variables from the output list. Thus, the graphs available in Appendix B were obtained using the SPS simulator tools.

It is necessary to remark that the results for the validity range were calculated using the alternative algorithm without the linear, but the same results are expected for the combination. The main outcomes of these sensitivity tests are present in Table 7.4 and 7.5, and Appendix B.

Table 7.4: Parameters variation range for valid coordiantes

Parameter	Value	Max variation [$\pm\%$]	
		Alternative	Pseudo
a_coarse	-2.24536756733076e-5	60	43
b_coarse	0.00078031275691922	36	21
c_coarse	-0.0061184383374413	23	23
R0_coarse	-758.2013024308603	193	-
a_fine	-3.6907368281401e-6	14	10
b_fine	0.000437857008493415	3	4
c_fine	-0.00350110848394894	3	12
R0_fine	4034.634917033389	23	-

The initial parameters are identical to the ones used by the pseudo-paraboloidal, except for R_0 that is a function of the others. Consequently, the variation ranges present in the table directly compare the performance of the algorithms.

There are two crucial results shown in Table 7.4 and 7.5. The first is the maximum variation allowed for the reconfigurable parameters before encountering a not valid solution, corresponding to more than one solution (alternative) or a complex (PP) value. This indicates how flexible the algorithm can be if the initial parameters need to be changed after calibration.

Table 7.5: Parameters uncertainty range for error lower than 50 μm

Parameter	Max variation [$\pm\%$] for 50 μm error			
	Alternative	Pseudo	Alt+Lin	Pseudo + Lin
a_coarse	5.6	5.6	18.5	18.5
b_coarse	0.5	0.5	2.4	2.4
c_coarse	2	2	7.3	7.3
R0_coarse	171	-	193	-
a_fine	0.6	0.6	1.7	1.7
b_fine	0.1	0.1	0.4	0.4
c_fine	0.2	0.2	0.7	0.7
R0_fine	3.2	-	7.5	-

The second one is the maximum variation of a single fitting parameter required to get an error bigger than 50 μm ; this value shows the algorithm’s sensitivity to uncertainty in the parameters for a required solution.

Evaluating the range of validity, it is evident how the alternative algorithm has an advantage when the coarse parameters are used, leading to an increased range of +17 % and +15% for *a_coarse* and *b_coarse*, respectively. The *c_coarse* range remains identical for both algorithms, whereas the range of *R0_coarse* is very high since it does not modify the shape of the illumination curve; it only shifts the “zero” value. If a comparison is wanted, the equivalent parameter to *R0_coarse* for the pseudo-paraboloidal is *A_coarse*, with an expected variation range of 99%.

Nevertheless, the variation range using the fine fitting parameter follows a different trend, with numbers in the range of 10% for both algorithms. Evaluating the algorithms difference, a +4% is expected for the alternative *a_coarse* and a -1% and -9% for *b_coarse* and *c_coarse*, respectively. There is no information on the variation range of *A_fine*, but *R0_fine* once again allows one order of magnitude higher range than the other parameters.

The 50 μm error range shows how both algorithms achieve an almost identical performance when the parameters are changed; this arrives as a consequence of the capability of the algorithms to follow the curve accurately and the similar error that they achieve inside the requirement box. As previously remarked, the accuracy derived from the simulator’s approximation of the curve leads to difficulty comparing the algorithms.

Analysing the results, in Figure 7.4a and 7.4b it is possible to observe the outcome of the sensitivity test for the *R0_fine* and *R0_coarse* parameters, respectively (the remaining figures are available in Appendix B). Where the coarse variation leads to a displacement of one order of magnitude higher for the fine parameters at around -1 mm, compared to the 0.1 mm from *R0_coarse*.

Furthermore, the fine stays almost constant for the values lower than the nominal; instead, as $R0$ increases, the variation diverges to a negative error rapidly.

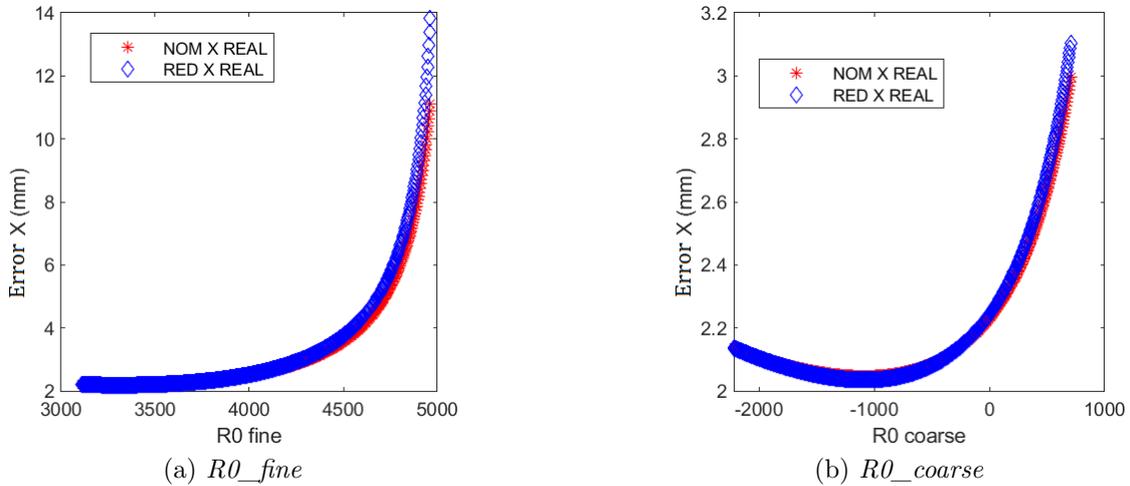


Figure 7.4: Sensitivity test longitudinal error result

A general evaluation of the remaining parameters (a , b and c) leads to the opposite outcome. **The error obtained from the coarse parameters is higher than the ones obtained by the fine**, with b_coarse achieving a maximum error of almost -7 mm, whereas a_fine reaches a maximum of around -1.1 mm. As an example of this behaviour, only the a_fine and a_coarse are shown in Figure 7.5.

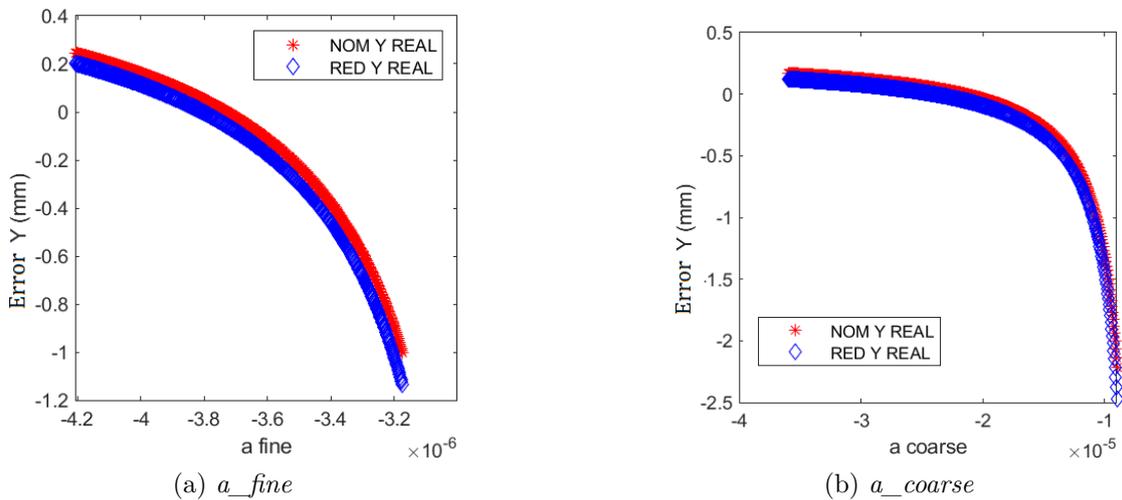


Figure 7.5: Sensitivity test lateral error result

Another main difference between the coarse and fine parameters is the slope of the lateral error curve. It is unmistakable that the fine parameters variation leads to

a more uniform interpretation of the coordinates. In contrast, the coarse parameters follow an almost horizontal trend for the lower values and a more vertical slope as the parameter increases. It might result from the higher variation obtained when modifying the fine parameters, leading to complex values in a smaller range.

The presence of complex numbers in the longitudinal position (Figure 7.6a) is ignored since the proposed analysis is focused on the lateral performance of the alternative and pseudo-paraboloidal algorithms. Moreover, the longitudinal displacement in Figure 7.6b diverges rapidly (almost exponentially) for the coarse parameters, with an error between 3 mm and 300 mm. Instead, the fine parameters negative variation delivers complex results, whereas the positive variation leads to a negative error in the -400 mm and -2000 mm range, certainly outside the accuracy requirement.

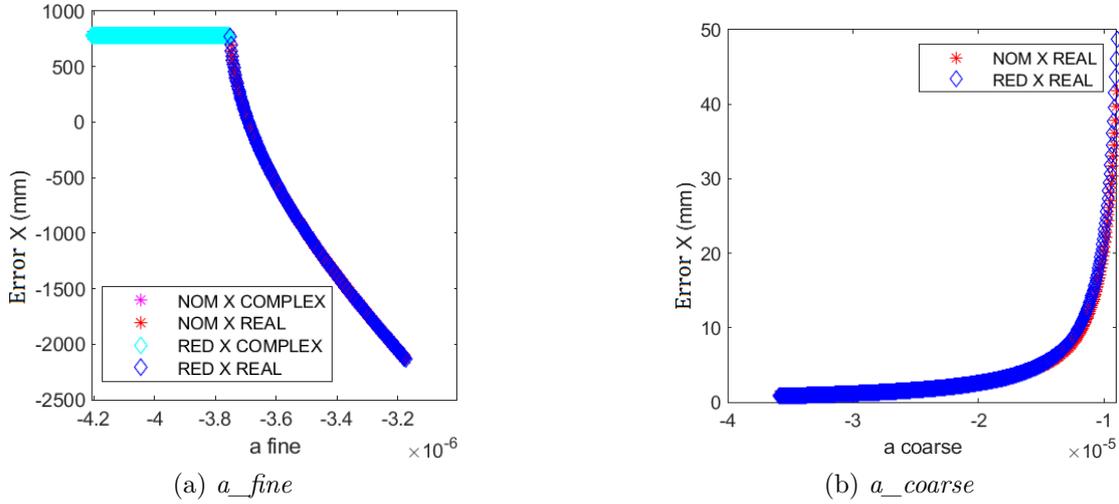


Figure 7.6: Sensitivity test longitudinal error result

It is possible to conclude that the alternative algorithm can perform in a slightly reduced range, compared to the PP, if the fine parameters are used, and in a significantly higher range if the coarse parameters are considered. Furthermore, the capability of both algorithms to reconstruct the curve with a certain degree of error is identical, and **the combination with the linear algorithm is advantageous, tripling the uncertainty range**. This is a consequence of the independence of the linear from the reconfigurable parameters, where the variation is reduced when the results are averaged. Furthermore, the results in Table 7.5 showed the accuracy required for the irradiance profile, where the fine parameters variation range corresponds to the $\sim 0.5\%$ obtained in [2].

Chapter 8

Conclusions and final remarks

The main objective of this thesis was the development, analysis and integration of the SPS alternative algorithm proposed by ESA into the main SPS simulator as a validation tool for the calibration of the flight algorithm outputs. The algorithm proposed by R. Rougeot accomplished the inversion of the third-order system using a particular mathematical method and a set of modified parameters. Nevertheless, the integration of this algorithm into the most recent version of the code needed significant updates. The set of penumbra profile fitting parameters adopted by the original version was the absolute value of an outdated set; this allowed to avoid the struggle encountered with the boundary conditions, assuring the existence of a (one and only one) solution. Still, the values determined in [22] were incorrect, and a significant amount of time was dedicated to solving this issue.

The pseudo-paraboloidal parameters were used to maintain uniformity between the two algorithms; one of the objectives of this work was in fact to update the ESA algorithm in such a way that the input and the output should have the same structure as the flight algorithm. This meant that the fitting parameters had to have the proper dimensions (and values) to accept the SPS readings (irradiance) in DN. Consequently, the same curve used by the pseudo-paraboloidal SPS simulator was successfully reconstructed for the alternative algorithm, later adopted by the local generation of the test points. However, to match the irradiance values obtained, the value R_0 was introduced for the developing algorithm; this “new” reconfigurable parameter allowed to shift the curve to reach the expected value of irradiance at the nominal inter-satellite distance ($R_i(0) = 1034 \text{ DN}$). Hence, a formula to find the optimised R_0 value was developed, and the obtained results were successfully correlated to those coming from the SPS simulator’s nominal and redundant readings.

Once a functioning alternative algorithm was achieved, the “Alternative_SPS_Algorithm_Generate” was developed. This complex Matlab script allowed to locally

generate the penumbra profile and the corresponding SPS readings for each sensor. An addition with respect to the original alternative algorithm is the novel possibility of choosing between the nominal or redundant set of SPS. Hereafter, the algorithm was tested step by step, determining the needed boundary conditions for each set of reconfigurable parameters and optimising the algorithm's performance. Furthermore, the alternative algorithm was modified from a single-input single-output to read and calculate the grid required to validate its functioning inside the requirement and goal box; this also applied to the pseudo-paraboloidal and linear algorithms taken directly from the onboard software developed by INAF.

The results from the locally generated profile using the alternative algorithm inside the requirement box are outstanding, achieving an error one order of magnitude lower than the pseudo-paraboloidal and the PPL. Additionally, the error follows an almost uniform distribution throughout the ± 10 mm box, and a nearly centrally symmetric pattern can be recognised in the two-dimensional view. Consequently, one of the main motives that led ESA to propose this algorithm as a validation tool was demonstrated by this result.

The obtained results with the alternative algorithm inside the requirement box permit the SOC to use this algorithm as a validation tool for the results coming from the PROBA-3 satellite's onboard SPS software.

The alternative algorithm inside the goal box delivers a discontinuous error distribution, with an almost 5 mm deviation at a semi-circle centred at the edges of the box, gradually decreasing to a null error as the points approach the centre of this peculiar displacement at 55 mm from the origin. An evaluation of the type of solution employed by Cardano's method was performed to understand why these error spikes are formed. The nominal SPS used in these tests coincides with the axis where the semi-circles are formed, so a correlation between the distance of the sensor to the centre might be the reason for this discontinuity. For this reason, a hypothesis of why the half-moons of increased error are formed (see Figure 6.16a) lies in the way the algorithm determines the solution, where the error arises when a single SPS enters the umbra zone (around 33.5 mm from the centre, equal to the half-moon radius).

Consequently, the script was modified by the author to repeat this test using the redundant set, where the results are shown in Appendix C. As expected, the error increased along the diagonals where the sensors are located, with a central cross that satisfies the $50 \mu\text{m}$ error requirement. Hence, further investigation is needed to fully understand this situation. Furthermore, a potential combination of the nominal and redundant solution using the alternative algorithm might lead to a more precise output. The error spikes of one set are counterbalanced by the other, and a more uniform solution throughout the goal box might be achieved. Still, this solution leads to a higher computational cost and was not developed for this thesis since a major overhaul of the script is needed.

Comparing the algorithms inside the goal box, it is possible to conclude that the

pseudo-paraboloidal + linear algorithm is the most accurate. This result confirms the INAF decision to implement this as the main algorithm onboard the PROBA-3 satellite. A relatively low computational cost is complemented by outstanding precision, retrieving the umbra centre coordinates with less than 2 mm of lateral error within the goal box. The alternative algorithm can be used as a validation tool, especially in the points inside the central “X” (or cross if the redundant set is used) that achieves the best performance overall. However, the points inside the aforementioned half-moons lead to an undesired accuracy. Hence, the SOC shall decide if the positions retrieved by the alternative algorithm are valid by comparing them to the zones in Figures 6.32 and 6.33.

It is important to remark that the performance obtained for the pseudo-paraboloidal and linear using the locally generated profile is not perfectly similar to that tested by INAF in October 2020. One reason for the discrepancy is the different reconfigurable parameters used, where even a slight variation of the penumbra profile can considerably modify the error surface. For this reason, a series of tests using the outdated set of reconfigurable parameters showed a stronger resemblance of the error surface shapes concerning the results in Chapter 6. Consequently, the comparison inside the goal box presented in this thesis is valid for the penumbra profile fitting reconfigurable parameters present in Appendix D.4. If a different set is used, a slightly different shape of the error is expected, and for this reason, the sensibility tests performed with the SPS simulator are of great value.

The proposed analysis showed only the results inside the requirement box regarding the longitudinal error since the obtained precision is below the requirement. The evaluation of the longitudinal error was not an objective of this work since Cardano’s method strived only to improve the lateral position estimation. Nevertheless, the longitudinal error is directly proportional to the lateral error; hence, the outcomes shown for both the requirement and goal box can be used for a later evaluation of the longitudinal displacement. As previously stated, other subsystems in the metrology chain obtain greater accuracy in this measure, and the SPS values shall be used only as a coarse reference.

The integration of the alternative algorithm script into the onboard software structure demanded meticulous attention since the inputs and outputs of the algorithm must be identical. Therefore, every function was evaluated, and if necessary, adapted to the alternative algorithm. This led to a unified development of the alternative file reading algorithm and the SPS alternative simulator. The aforementioned tests were repeated using a series of random points that allowed the validation of the algorithm’s functioning, where the obtained results were almost identical for the alternative, PP and PPL.

Finally, the algorithms were compared using the available tools of the simulator, where the sensitivity of the solution for a variation in the fitting parameters was evaluated. For each parameter, a range of uncertainty has been found as the variation of that parameter causing an error on the assessment of the position lower than 50

μm . The about $\pm 0.4\%$ variation range obtained using the fine parameters makes it possible to conclude that these values shall be approximated with an accuracy at least better of one order of magnitude (more significant figures). Moreover, the identical variation range obtained using both algorithms indicates that the output coordinates precision is of the same order of magnitude (at around one micron).

Recommendations for future developments

The following steps in developing the SPS alternative algorithm will be focused on updating the codes to the latest Software Interface Control Document (ICD) version, where a different set of flags are used for the SPS readings. Furthermore, additional tests shall be performed using the SPS simulator tools, evaluating the flags for false positive or false negatives throughout the entire volume and at different inter-satellite distances. Additionally, the alternative algorithm can be improved by selecting the desired solution when more than one is obtained from Cardano's method. Currently, the algorithm returns an error (as the false -999999 mm coordinate) when more than one solution is found. If the results can be compared to a certain threshold (similar to the fitting parameter variation), extending the algorithm's solution validity range might be possible.

Another essential step will be validating the alternative algorithm results using the SPS readings from the ASPIICS tests currently in development. Comparing the flight algorithm results to the alternative algorithm using the Solar Simulator Chamber facilities at ALTEC might serve as a rehearsal for the actual mission execution of the processing pipeline to be operated at the SOC.

A possible improvement of the current onboard algorithm might be the implementation suggested in Section 6.3, where the PP is used inside the requirement box, providing more accurate results with respect to the combination with the linear, and implementing the PPL in the remaining part of the goal box for improved performance.

Lastly, the cause of the semi-circular error disuniformities inside the goal box shall be further analysed. Finding a possible solution to this issue might lead to an alternative algorithm that reconstructs the penumbra profile with an error lower than $50 \mu\text{m}$ throughout the entire goal box.

Bibliography

- [1] F. Amadori et al. “End-to-end numerical simulator of the Shadow Position Sensor (SPS) metrology subsystem of the PROBA-3 ESA mission (P3-INF-TN-20033)”. In: (2020). DOI: <http://dx.doi.org/10.20371/INAF/TechRep/65>.
- [2] A. Bemporad et al. “The Shadow Positioning Sensors (SPS) for formation flying metrology on-board the ESA-PROBA3 mission”. In: *Solar Physics and Space Weather Instrumentation VI* (2015). DOI: [10.1117/12.2191829](https://doi.org/10.1117/12.2191829).
- [3] G. Capobianco et al. “The in-flight calibration procedures of the Shadow Position Sensors (SPS), a very accurate optical metrology system of the ESA/PROBA-3 formation flying mission”. In: *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)* (2019). DOI: [10.1109/metroaerospace.2019.8869623](https://doi.org/10.1109/metroaerospace.2019.8869623).
- [4] *Cardano’s Method*. URL: <https://brilliant.org/wiki/cardano-method/>.
- [5] M. Casti. “Implementation and Error Analysis of the Algorithms for PROBA-3/ASPIICS F2 Metrology (P3-INF-TN-17024)”. In: (2020).
- [6] M. Casti et al. “Fine positioning algorithms for the ESA/PROBA-3 formation flying mission”. In: *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)* (2019). DOI: [10.1109/metroaerospace.2019.8869551](https://doi.org/10.1109/metroaerospace.2019.8869551).
- [7] M. Casti et al. “PROBA-3 formation-flying metrology: algorithms for the shadow position sensor system”. In: *International Conference on Space Optics ICSSO 2018* (2019). DOI: [10.1117/12.2536209](https://doi.org/10.1117/12.2536209).
- [8] M. Casti et al. “SPS Software Interface Control Document V6.2 (P3-INF-IF-16000)”. In: (2021).
- [9] M. Focardi et al. “The shadow position sensors (SPS) formation flying metrology subsystem for the ESA PROBA-3 mission: present status and future developments”. In: *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave* (2016). DOI: [10.1117/12.2231699](https://doi.org/10.1117/12.2231699).
- [10] *Introduction to SiPM, Technical note*. URL: <https://www.sensl.com/downloads/ds/TN%20-%20Intro%20to%20SPM%20Tech.pdf>.

- [11] D. Loreggia et al. “Formation flying metrology system for the ESA-PROBA3 mission: the Shadow Positioning Sensors (SPS)”. In: *Optical Instrument Science, Technology, and Applications* (2018). DOI: [10.1117/12.2314814](https://doi.org/10.1117/12.2314814).
- [12] V. Noce et al. “An improved version of the Shadow Position Sensor readout electronics on-board the ESA PROBA-3 Mission”. In: *UV, X-Ray, and Gamma-Ray Space Instrumentation for Astronomy XX* (2017). DOI: [10.1117/12.2273694](https://doi.org/10.1117/12.2273694).
- [13] V. Noce et al. “Metrology on-board PROBA-3: The shadow position sensors subsystem”. In: *Advances in Space Research* (2020). DOI: [10.1016/j.asr.2020.08.004](https://doi.org/10.1016/j.asr.2020.08.004).
- [14] V. Noce et al. “The Shadow Position Sensors (SPS) metrology subsystem on-board PROBA-3 mission : Design and performance”. In: *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)* (2019). DOI: [10.1109/metroaerospace.2019.8869603](https://doi.org/10.1109/metroaerospace.2019.8869603).
- [15] *PRISMA (Prototype Research Instruments and Space Mission technology Advancement)*. URL: <https://earth.esa.int/web/eoportal/satellite-missions/p/prisma-prototype>.
- [16] *PROBA-3 (Project for On-Board Autonomy-3)*. 2021. URL: <https://earth.esa.int/web/eoportal/satellite-missions/p/proba-3>.
- [17] *Proba-3 Mission*. URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Proba_Missions/Proba-3_Mission3.
- [18] *Proba-3 Platforms*. URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Proba_Missions/Proba-3_Platforms.
- [19] *Proba-3 Science payloads*. URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Proba_Missions/Proba-3_Science_payloads.
- [20] *Proba-3 Technologies*. URL: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Proba_Missions/Proba-3_Technologies.
- [21] *Proba-3: Osservatorio Astrofisico di Torino*. Mar. 2021. URL: <https://www.oato.inaf.it/progetti/proba3/?portfolioCats=87,88>.
- [22] R. Rougeot and M. Casti. “Alternative SPS metrology algorithm (P3-INF-TN-20031)”. In: (2020).
- [23] *TanDEM-X Autonomous Formation Flying Experiment (TAFF)*. URL: https://www.dlr.de/rb/en/desktopdefault.aspx/tabid-10749/10535_read-23363/.

Appendix A

Goal box comparison with fine fitting parameters

A preliminary test of the locally generated penumbra profile for validating the alternative algorithm solution saw the implementation of the fine fitting parameters for the entire goal box (differently from the final algorithm that changed between fine and coarse).

Consequently, this led to an “unfair” comparison between the Cardano’s method solution and the pseudo-paraboloidal, since the first followed the fine parameters whereas the second used fine + coarse. The results presented in this appendix reveal a remarkable capacity of the alternative algorithm inside the goal box, yet the actual results are expected to be closer to the ones presented in Chapter 6. However, if the real penumbra profile follows the fine fitting parameters in the goal box, the following results might be useful to validate the algorithm’s outputs for both lateral and longitudinal position determination.

A.1 Lateral error

The lateral error is the most relevant parameter to be considered in the development of the algorithm. From the requirements in Section 2.2, an accuracy of 50 μm for each axis is demanded inside the requirement box.

Goal box - Alternative

The same procedure applied for the requirement box is repeated in the goal box. For the lateral displacement, the algorithm shall calculate the position at a reduced performance within a ± 50 mm box. As previously explained, the boundary condition in this box needs to be modified in order to calculate and invert the system with Cardano’s method successfully. Hence, Cardano’s method reduces the goal box

from ± 50 mm to ± 41 mm to obtain a single solution for each SPS radius r_i . The pseudo-paraboloidal algorithm does not encounter this issue; however, to maintain uniformity of the results, both algorithms are compared in the same modified goal box, using once again the “Generate” Matlab script.

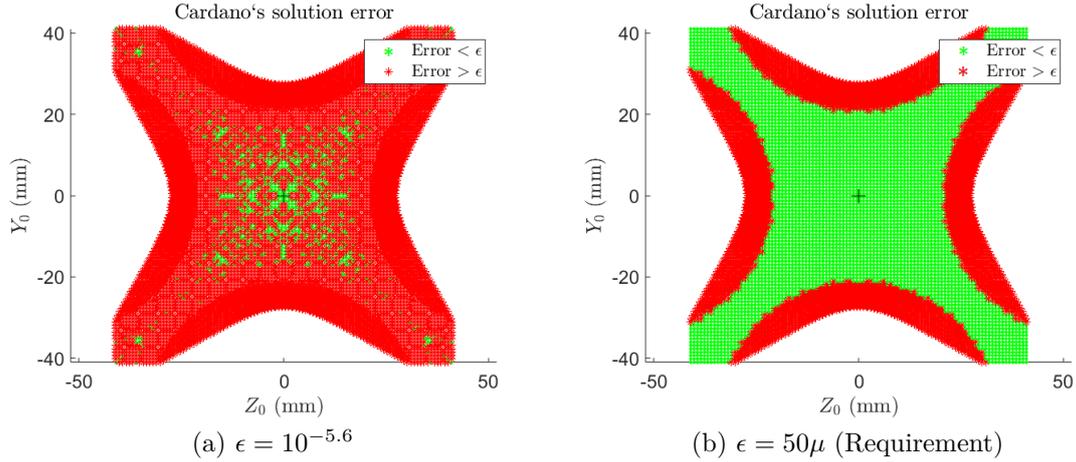


Figure A.1: 2D Lateral error plot in the goal box using Cardano and 100x100 grid

From Figure A.1, it is possible to identify the green area that fulfils the stringent precision requirement of $50 \mu\text{m}$ even outside the requirement box. This “X” configuration has an opposite trend compared to the pseudo-paraboloidal results presented in the previous section, where the precision decreased in the corners (inside the requirement box). Furthermore, Figure A.1a shows how the cylindrical symmetry is still present in the goal box.

Analyzing the three-dimensional representation of the error in Figure A.2, it is evident how the centred “X” shape has a dark colour for a high accuracy zone, whereas the outer edges of the plot see a steep increment of the error to almost 10 mm. It is possible to hypothesize a similar trend for the excluded area between ± 41 and ± 50 mm, with high precision at the corners and an error that increases quickly as the points approach the zero coordinate.

The 3D plot in Figure A.2b is not a perfect representation of the error because of a graphical issue. The matrix used for the determination of the error in Appendix D.2.5, combined with the Z-Y mesh, delivers a graph that is symmetrical to only one of the axes (despite the 2D representation of this error being accurate in both axes). Thus, a superposition of two three-dimensional surfaces is needed to provide an approximated illustration of the error. Consequently, in the following graphs, the lower error zone under the higher curve shall be ignored. In other words, only the higher error points shall be considered if two vertical points coincide.

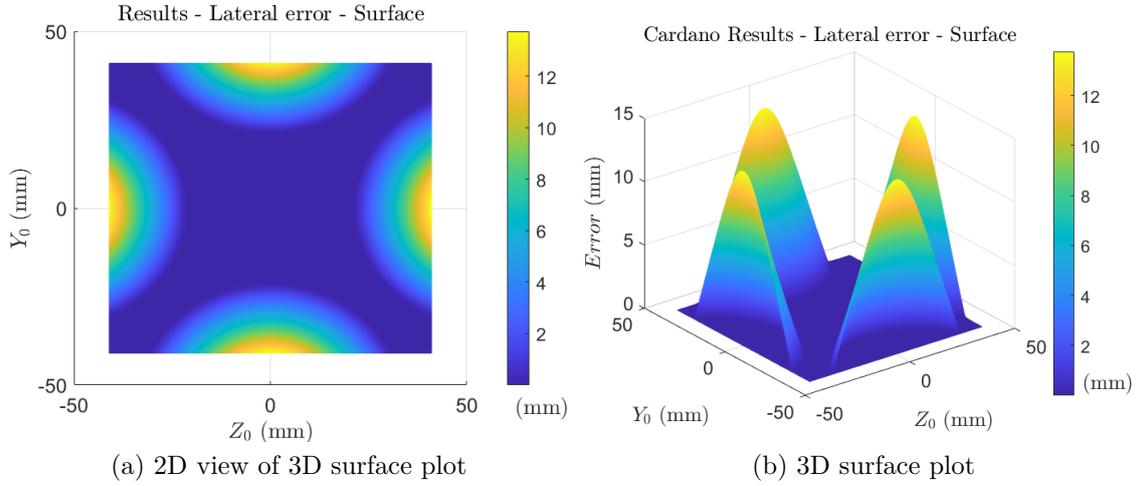


Figure A.2: Lateral error surface in the goal box using Cardano and 100x100 grid

Goal box - Pseudo-paraboloid

Using the fly algorithm, the green zone in Figure A.3b is the same as the one obtained in the requirement box in Figure 6.3b, where a cross with higher precision is present. As expected, the outer corners of the box see an increase in error, whereas Figure 6.3a shows how the solution maintains the same structure, with the points that lie in the central coordinate axis reach higher precision.

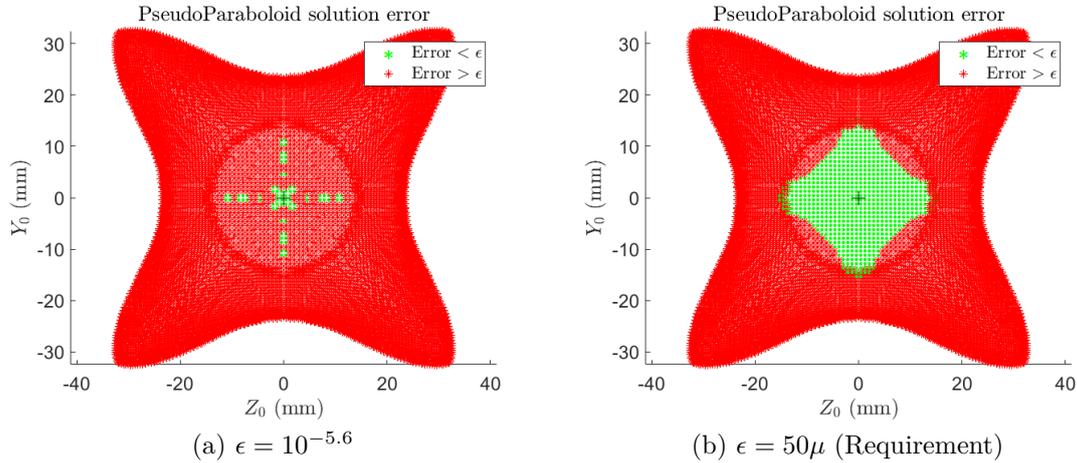


Figure A.3: 2D Lateral error plot in the goal box using pseudo-paraboloid

In Figure A.4b, the error surface plot for the goal box has a different shape compared to the requirement box. The maximum error (approximately 18 mm) is present at the sides of the boundary box, just like the alternative algorithm plot

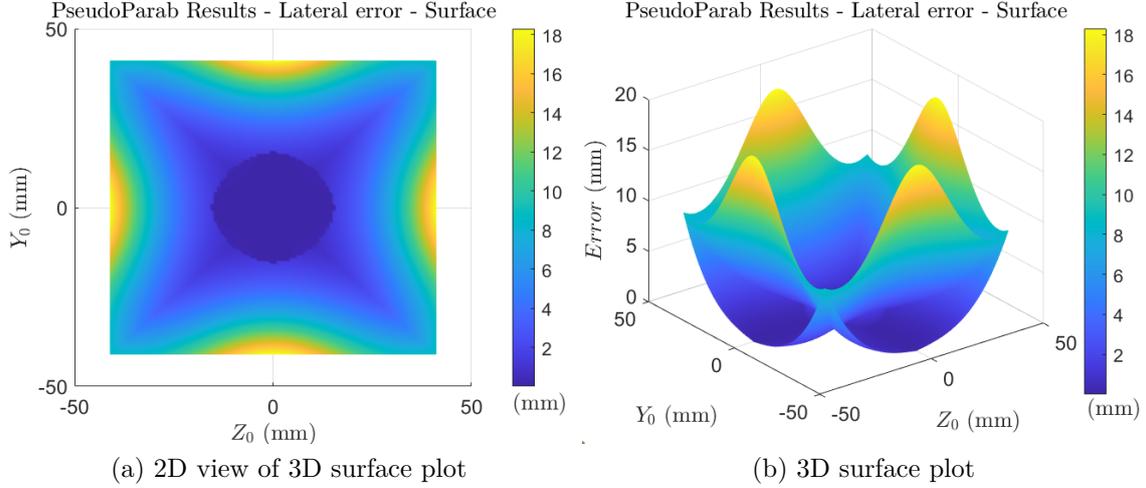


Figure A.4: Lateral error surface in the goal box using pseudo-paraboloid

in Figure A.2b. The inner circle of increased precision is a consequence of the fine fitting parameters, accompanied by an “X” of lower error in the corners, compared to the sides.

Goal box - Algorithm comparison

An important remark is needed since the alternative algorithm only uses the fine fitting parameters. In contrast, the pseudo-paraboloidal algorithm adopts by default the coarse parameters, and if a certain threshold (determined by the requirement box) is satisfied, the fine parameters are utilized. This leads to the high precision zone visible at the centre of Figure A.4a and A.8a, thanks to the fact that the generated profile uses the fine fitting parameters.

A side by side comparison of the statistics of the algorithm’s error reveals how the alternative algorithm reaches a greater accuracy with respect to the pseudo-paraboloidal inside the revised goal box, with approximately 80 % of the points leading to a deviation lower than 0.1 mm. In Figure A.5c the discrepancy between the algorithms is indisputable, with nearly 20% of the pseudo-paraboloid error under 0.1 mm (due to the variation in the fitting parameters used). Furthermore, the fly algorithm error in the goal box sees an error smeared through the zero to the sixteen-millimetre range.

Evaluating the 3D error surface throughout the mesh, it is possible to infer that the primary distinction of the algorithms inside the goal box is that the alternative achieves a higher precision in a more extensive domain.

The peaks reached by the pseudo-paraboloidal algorithm are slightly higher; this is visible in Figure A.6. However, analyzing the relative error in Figure A.7 and A.8,

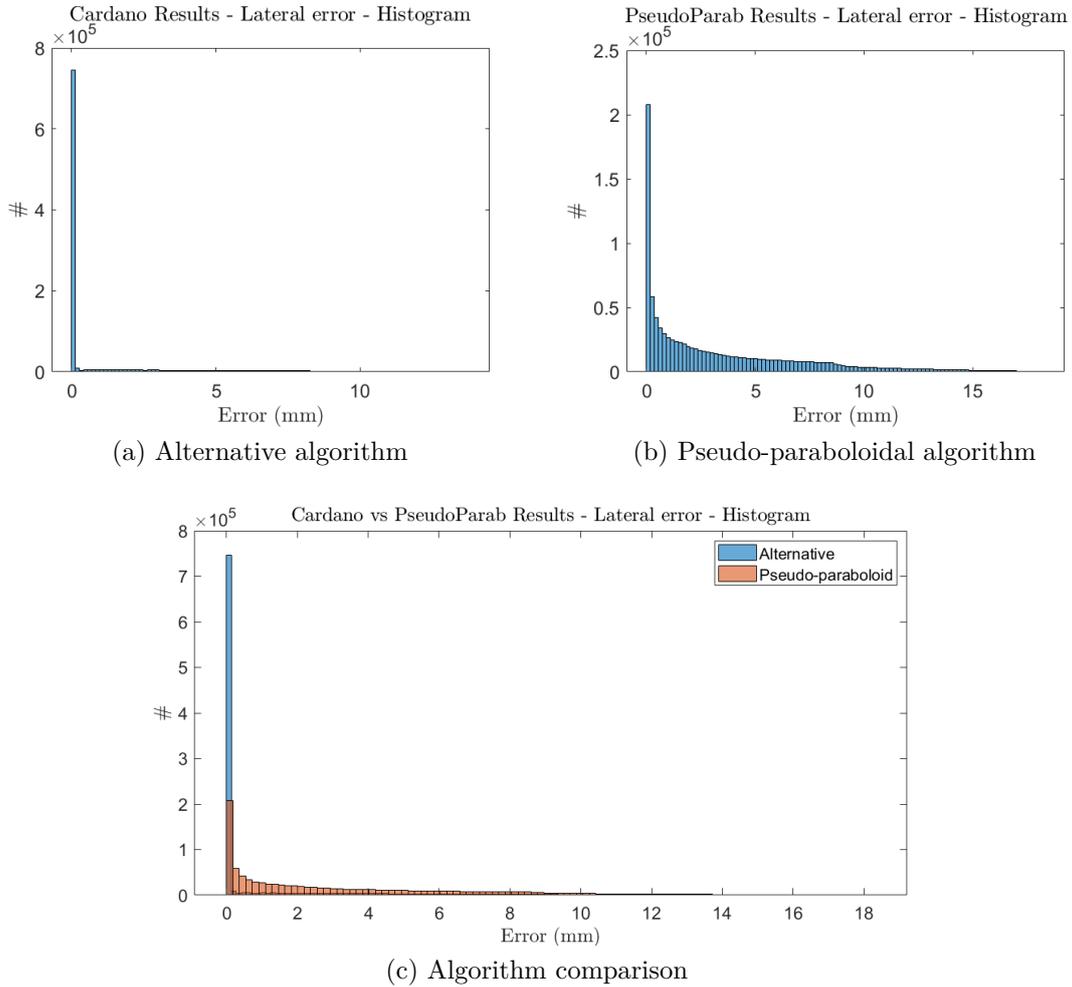


Figure A.5: Statistic of both algorithm’s error for the lateral position with a 1000x1000 grid inside the goal box

it is possible to notice that the most significant divergence among the algorithms is present at the corners of the box. This is a consequence of the elevated precision that the alternative algorithm reaches in this zone, whereas the sides of the box have a higher absolute error but a deficient relative error due to the similar shape of the surfaces.

An important observation for the application of the alternative algorithm as a validating tool for the fly algorithm comes from the shape of the relative surface. Figure A.8a exposes how greater is the precision of Cardano’s method in the outer edges of the box due to its cylindrically symmetric nature. This effect can be helpful for the determination and calibration of new possible sets of fitting parameters.

The statistical analysis of the error in Figure A.8b presents two peaks; this might

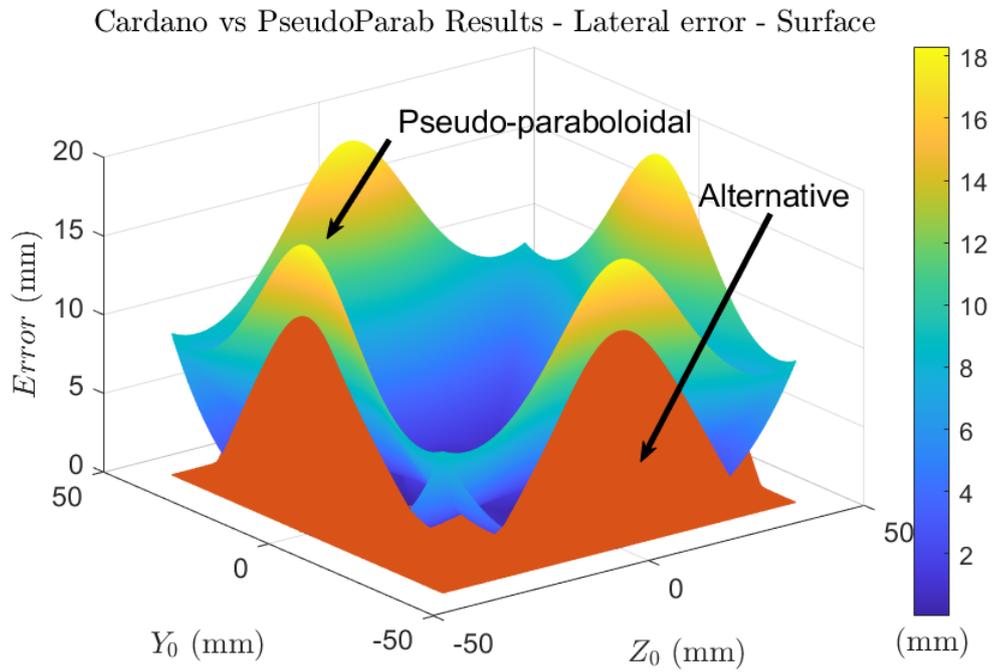


Figure A.6: Lateral error surface in the goal box for both algorithms

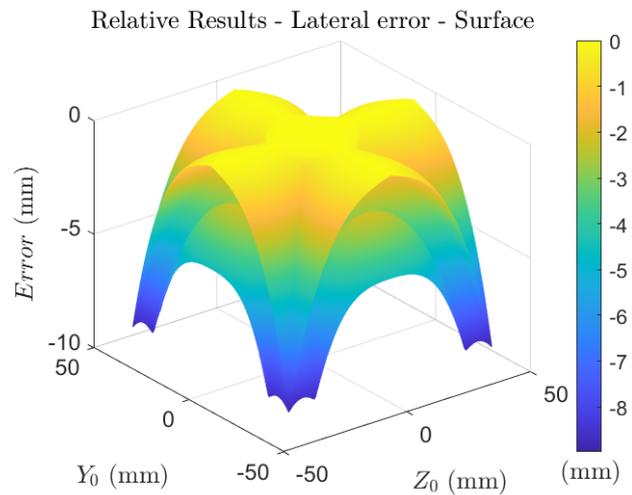


Figure A.7: Lateral difference of the absolute error surface in the goal box

be a consequence of the non-cylindrically symmetric configuration of the surface, with a petal-like shape that causes a secondary peak plateau concerning a relative error between -2 and -3 millilitres.

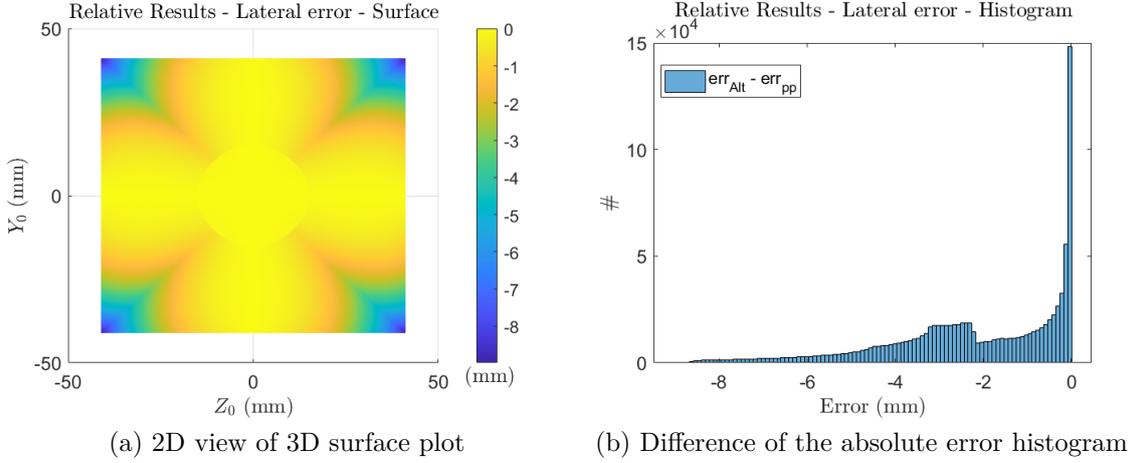


Figure A.8: Lateral difference of the absolute error in the goal box

A.2 Longitudinal error

A secondary result to be analyzed is the longitudinal error obtained from both algorithms. This error is not a primary concern for developing the alternative algorithm because a dedicated script is used to assess the x_0 coordinate. The following results apply only for the nominal inter-satellite distance, with $x_0 = 0$ as the expected result.

Goal box - Algorithm comparison

Alike the results inside the requirement box and involving a similar analysis to one applied for the lateral error, the algorithm’s longitudinal results inside the goal box reach a peak error outside the solicited accuracy.

As foreseen from the lateral error, the pseudo-paraboloidal algorithm produced a secondary peak in the statistical evaluation of the error. From Figure A.11b, the primary peak corresponds to the longitudinal displacement derived from the fine fitting parameters in the lateral error; whereas the secondary peak, with an almost mesa trend, corresponds to the zone adjacent to the umbra centre and the “X” pattern with increased precision.

From the direct comparison histogram in Figure A.11c, it is possible to conclude that the alternative algorithm reaches a higher precision and a more uniform distribution of the longitudinal error inside the goal box compared to the fly algorithm; with more than 50% of the outputs inside the “zero” histogram bar.

As previously discussed, the flower-like shape of the fly algorithm’s error inside the goal box produces a variation in the longitudinal error. The central part that utilises the fine parameters leads to the low relative error spike in Figure A.10b,

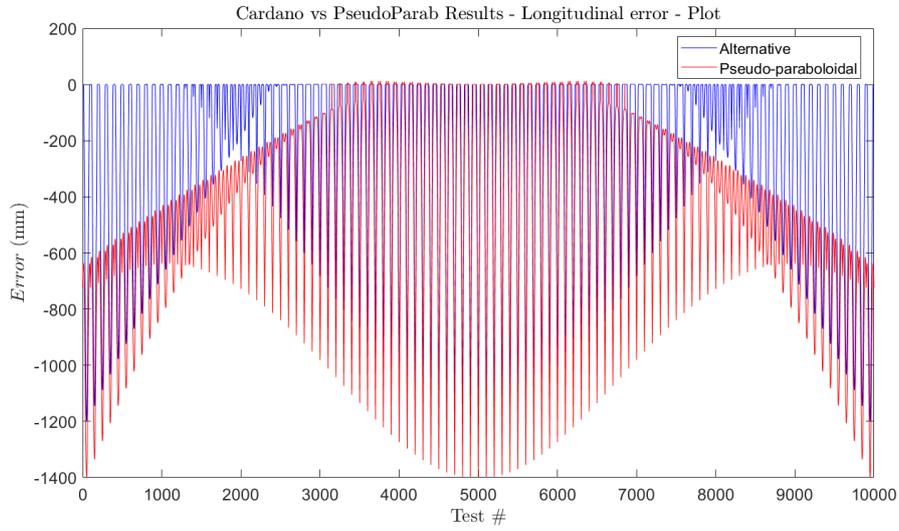


Figure A.9: Longitudinal error plot in the goal box using both algorithms

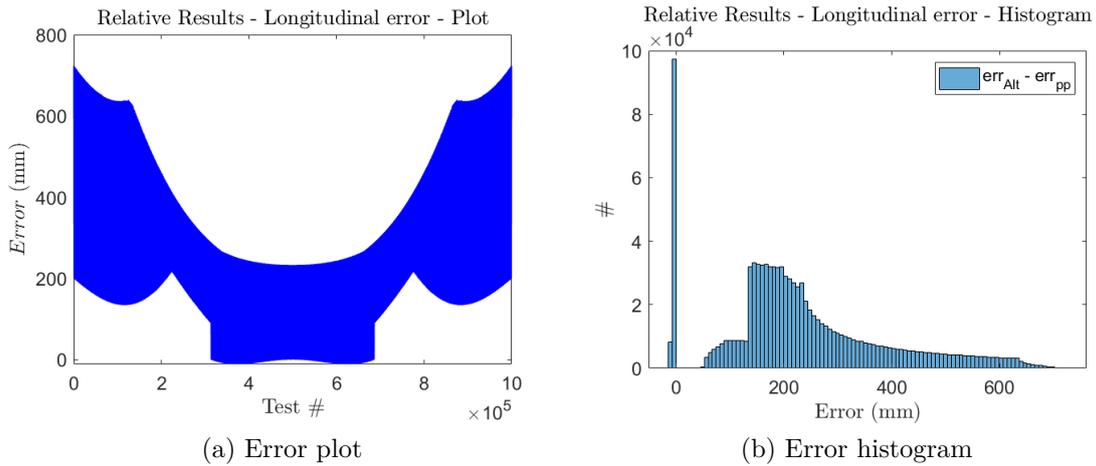
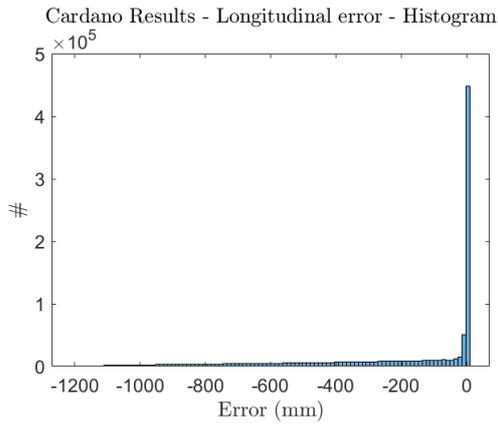
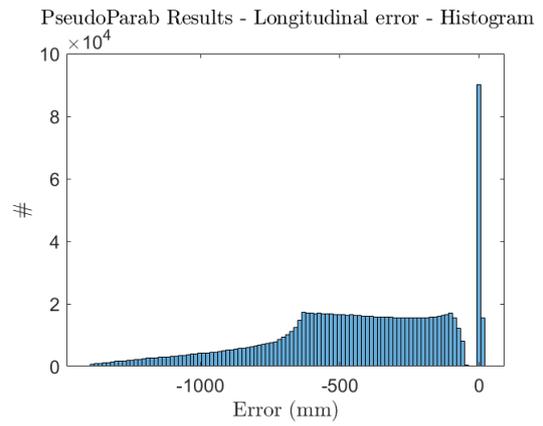


Figure A.10: Longitudinal difference of the absolute error in the goal box

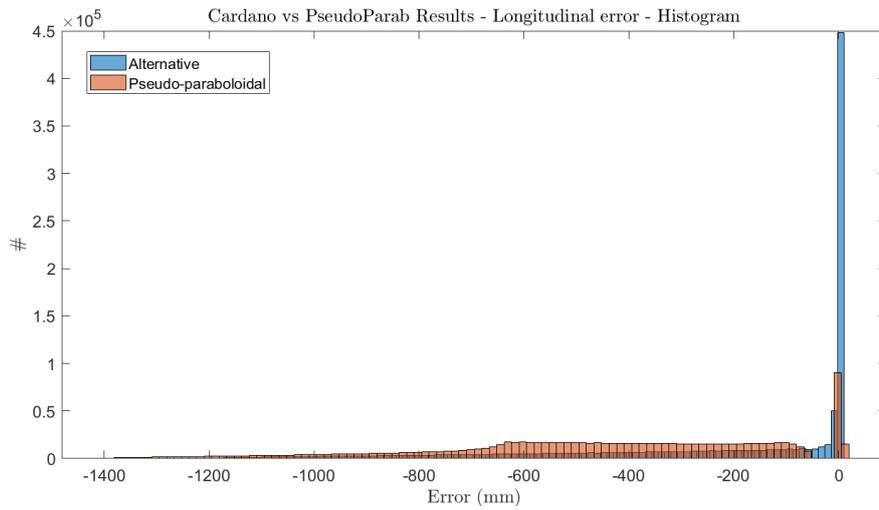
whereas the outer lobes, which are non-cylindrically symmetric, produce a more significant shift of the fly algorithm's results at the edges of the goal box. It is clear how the statistical distribution of the error follows the trend in the two-dimensional relative error plot.



(a) Alternative algorithm



(b) Pseudo-paraboloidal algorithm



(c) Algorithm comparisson

Figure A.11: Statistic of both algorithm's error for the longitudinal position with a 1000x1000 grid inside the goal box

Appendix B

Sensitivity test results for the alternative algorithm

The procedure followed for the evaluation of the reconfigurable parameters variation range is similar to the one performed for the pseudo-paraboloidal algorithm.

This first class of examination sees the individual variation of each parameter using the exact umbra centre coordinates. However, the nature of the alternative algorithm leads to a null variation of the result if the initial coordinate is at the CI pupil centre ($y_0 = z_0 = 0$, coordinates used for the PP tests). Therefore, the following alternative algorithm tests were performed at the $y_0 = z_0 = 1$ mm position, allowing the observation of the alteration of the parameters.

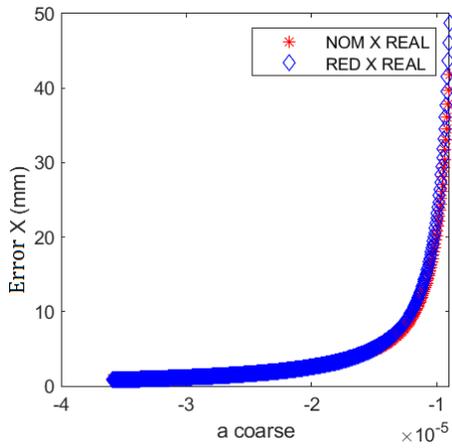
Another difference of this test with respect to the pseudo-paraboloidal is the simultaneous variation of the R_0 parameter when the others are changed (R_0 is a function of a , b , c and L). The PP algorithm tests performed a similar variation of the “A” parameters, but this one does not influence the alternative algorithm.

Once the SPS simulator has been adapted to the alternative algorithm and the desired position is chosen, 500 tests shall be performed for each parameter. First, a percentage variation of ± 300 % is applied and the complex solutions of the system are examined. Suppose a complex solution (or more than one) is present (a red light indicator will alleviate this step). In that case, the range is reduced, and the 500 tests are performed once again until no complex solutions are present. The same procedure is applied for the $50 \mu\text{m}$ error, where the error is controlled, and the percentage of variation is changed iteratively.

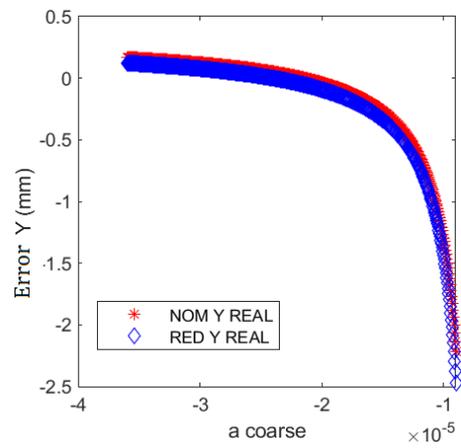
The following graphs show the results of these 500 tests, where the absolute error and the coordinate variation with the parameter are shown. It is important to note that the results exhibited are only for the Y coordinate. Still, an identical performance was obtained for the Z-axis, but it is omitted for simplicity.

B.1 Coarse parameters

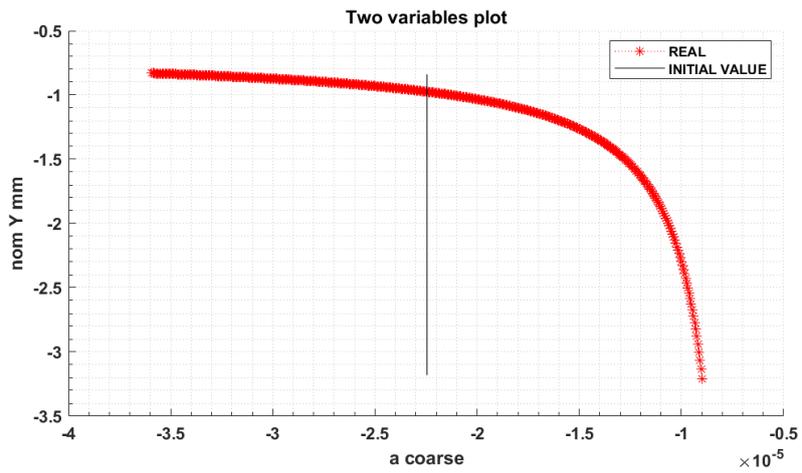
T1A01a_coarse



(a) Longitudinal error



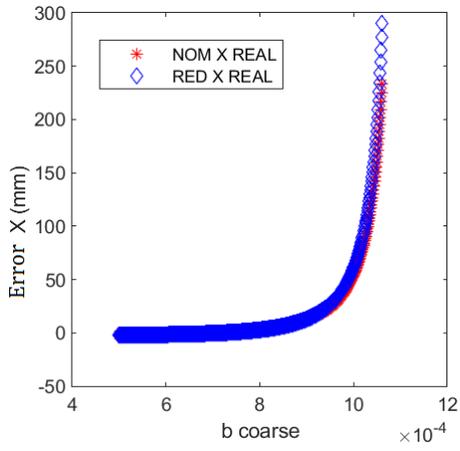
(b) Lateral error



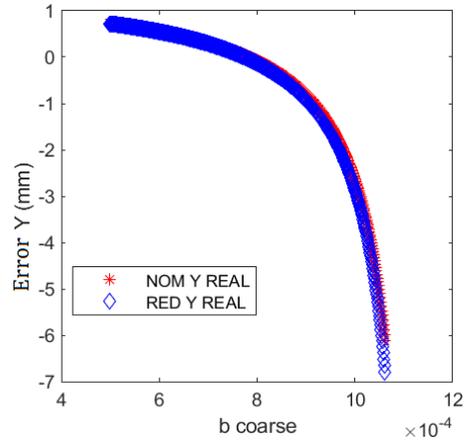
(c) Lateral coordinate variation with parameter

Figure B.1: a_coarse sensitivity test result plots

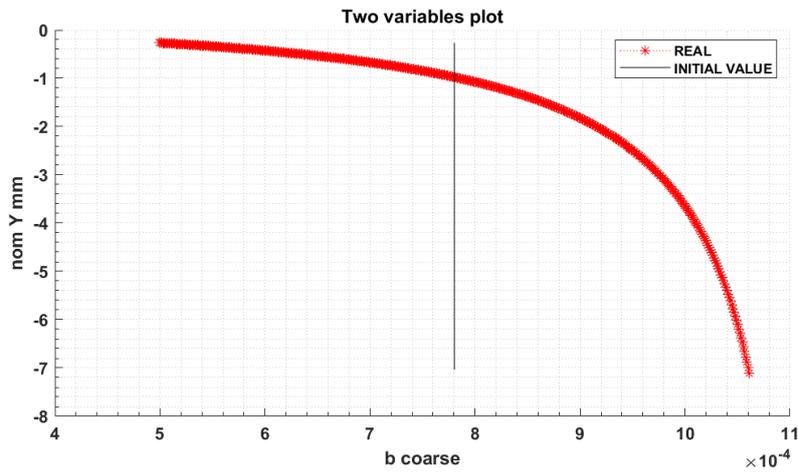
T1A02b_coarse



(a) Longitudinal error



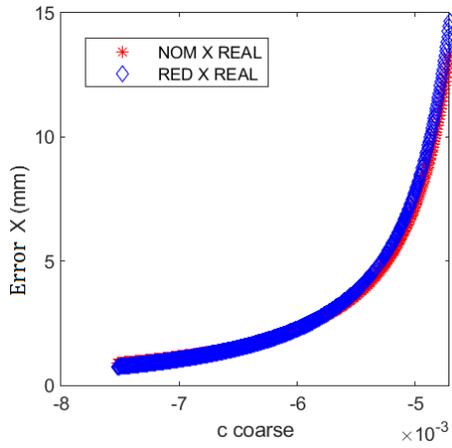
(b) Lateral error



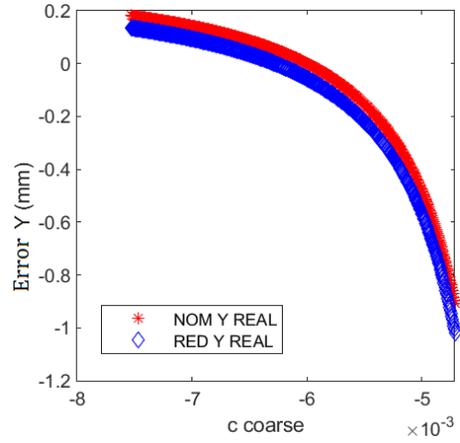
(c) Lateral coordinate variation with parameter

Figure B.2: b_coarse sensivity test result plots

T1A03c_coarse



(a) Longitudinal error



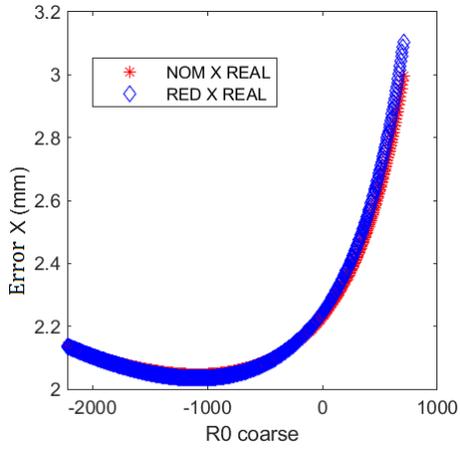
(b) Lateral error



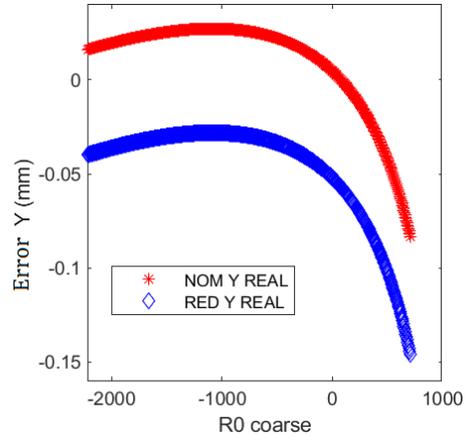
(c) Lateral coordinate variation with parameter

Figure B.3: c_coarse sensivity test result plots

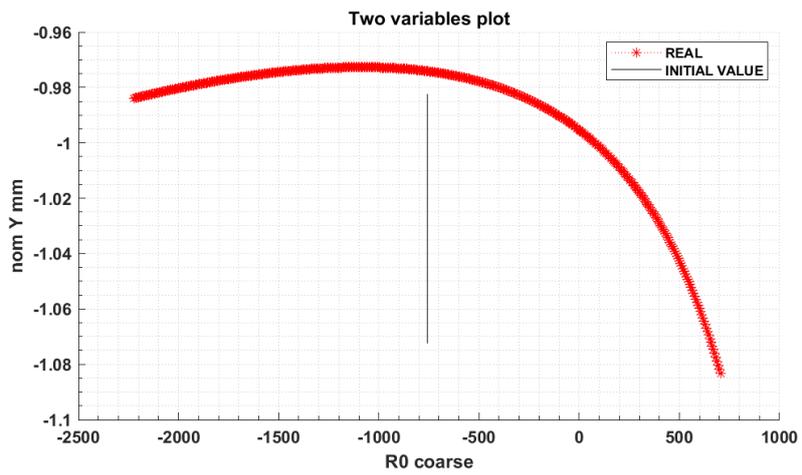
T1A04R0_coarse



(a) Longitudinal error



(b) Lateral error

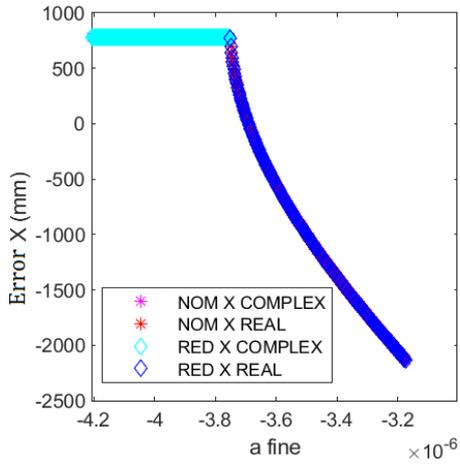


(c) Lateral coordinate variation with parameter

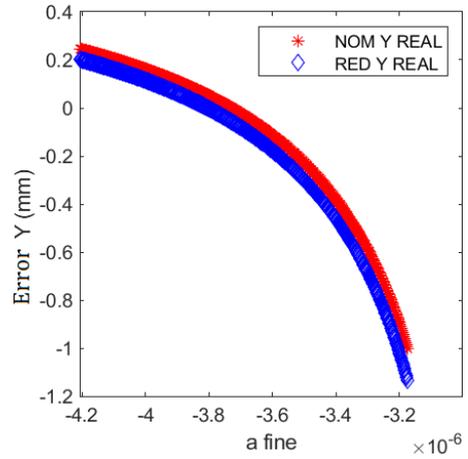
Figure B.4: $R0_coarse$ sensitivity test result plots

B.2 Fine parameters

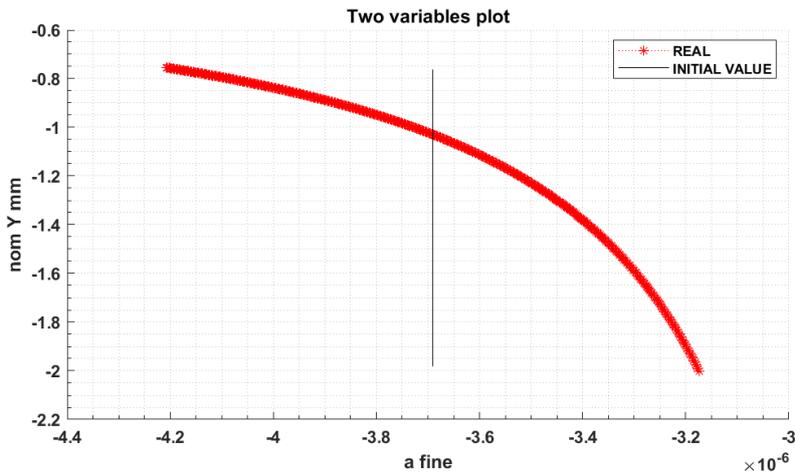
T1A05a_fine



(a) Longitudinal error



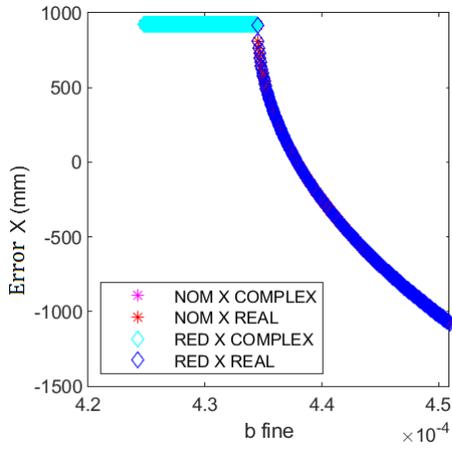
(b) Lateral error



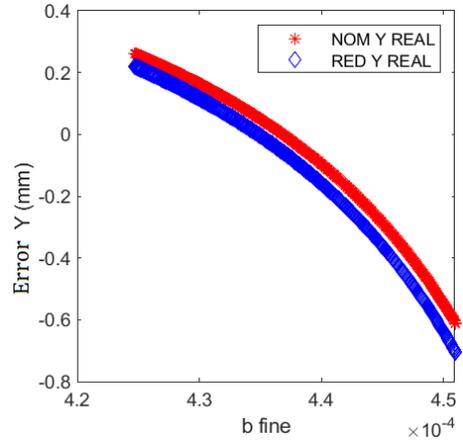
(c) Lateral coordinate variation with parameter

Figure B.5: a_{fine} sensitivity test result plots

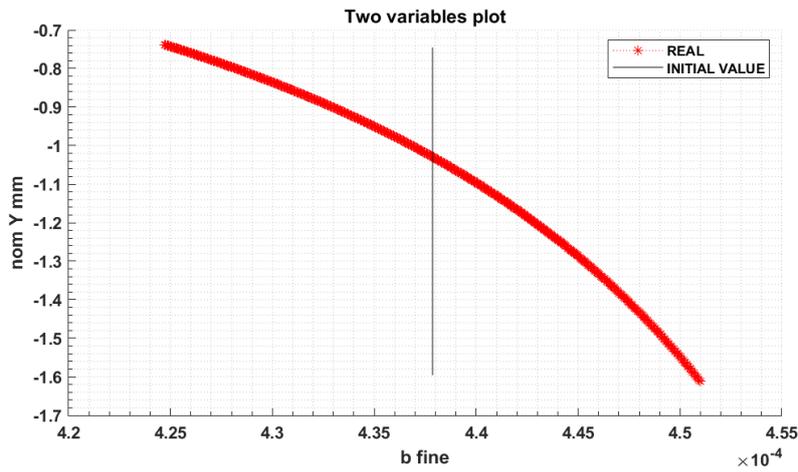
T1A06b_fine



(a) Longitudinal error



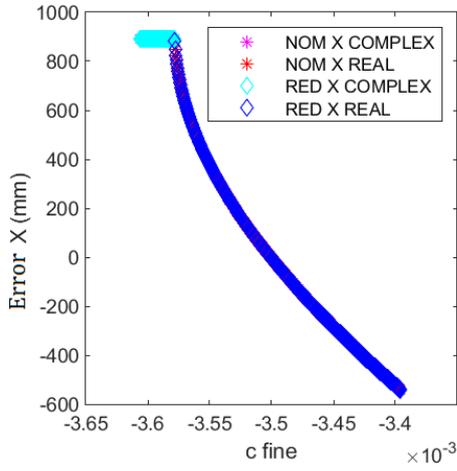
(b) Lateral error



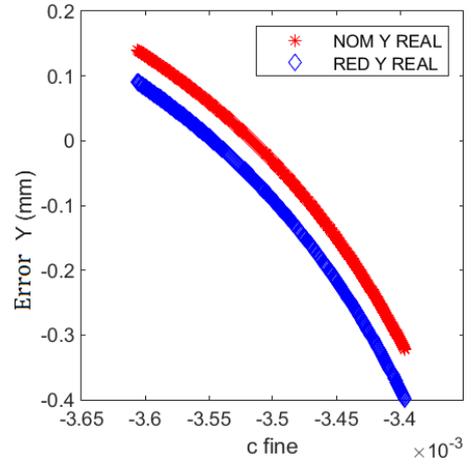
(c) Lateral coordinate variation with parameter

Figure B.6: b_fine sensitivy test result plots

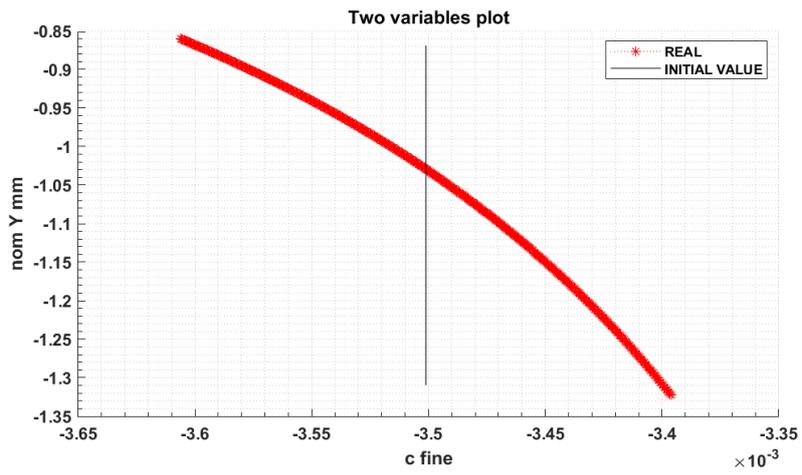
T1A07c_fine



(a) Longitudinal error



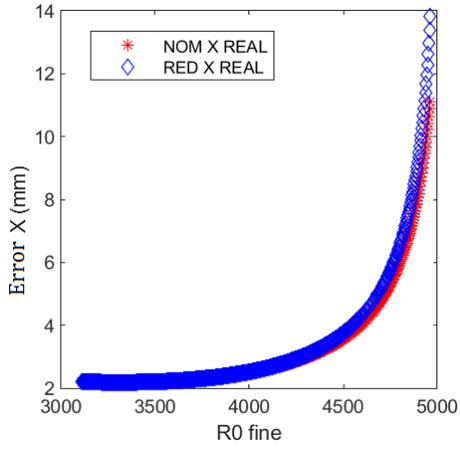
(b) Lateral error



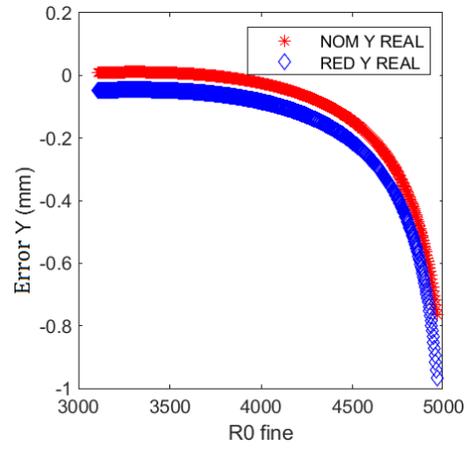
(c) Lateral coordinate variation with parameter

Figure B.7: c_fine sensitivity test result plots

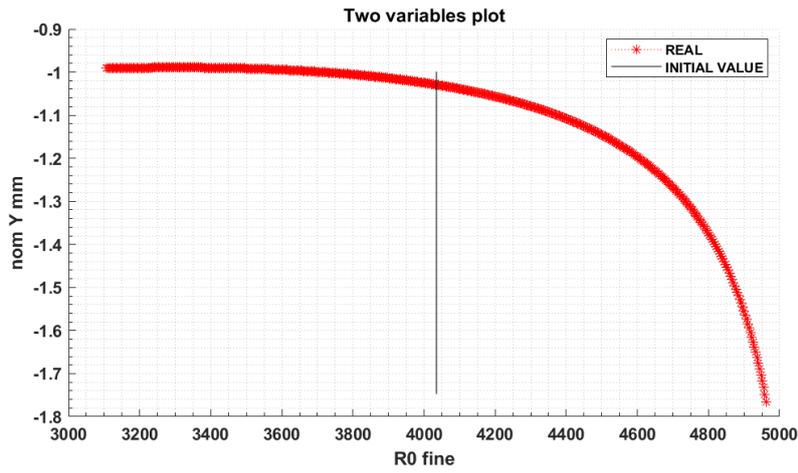
T1A08R0_fine



(a) Longitudinal error



(b) Lateral error



(c) Lateral coordinate variation with parameter

Figure B.8: $R0_fine$ sensivity test result plots

Appendix C

Algorithm comparison with locally generated points using redundant SPS set

The alternative SPS algorithm proposed by R. Rougeot and the requirements for the work developed in this thesis did not request the solution of the system using the SPS redundant set. Nevertheless, seamless integration of the alternative algorithm into the onboard software structure led the author to the addition of a script that performs this task.

Furthermore, the redundant set of SPS results have helped understand the discontinuity of the error encountered inside the goal box using the alternative algorithm.

To obtain these results, a modification of the script that generates the local profile is needed, where the redundant diode coordinates are the following:

```
Diode_A = [ r_SPS*(sqrt(2)/2), r_SPS*(sqrt(2)/2) ];           %[m] SPS 8
Diode_B = [-r_SPS*(sqrt(2)/2), r_SPS*(sqrt(2)/2) ];         %[m] SPS 2
Diode_C = [-r_SPS*(sqrt(2)/2), -r_SPS*(sqrt(2)/2) ];        %[m] SPS 4
Diode_D = [ r_SPS*(sqrt(2)/2), -r_SPS*(sqrt(2)/2) ];         %[m] SPS 6
```

And the umbra centre coordinates determined by each algorithm are replaced by:

```
y_0 = (sqrt(2)/2).*y_0_nom + (sqrt(2)/2).*z_0_nom;
z_0 = -(sqrt(2)/2).*y_0_nom + (sqrt(2)/2).*z_0_nom;
```

C.1 Alternative Algorithm

The performance of the alternative algorithm inside the goal box using the redundant set echoes the expected result, where the circular areas of increased error are located along the axis where the redundant SPS sensors are positioned, and they are centred at a 55 mm radius from the origin.

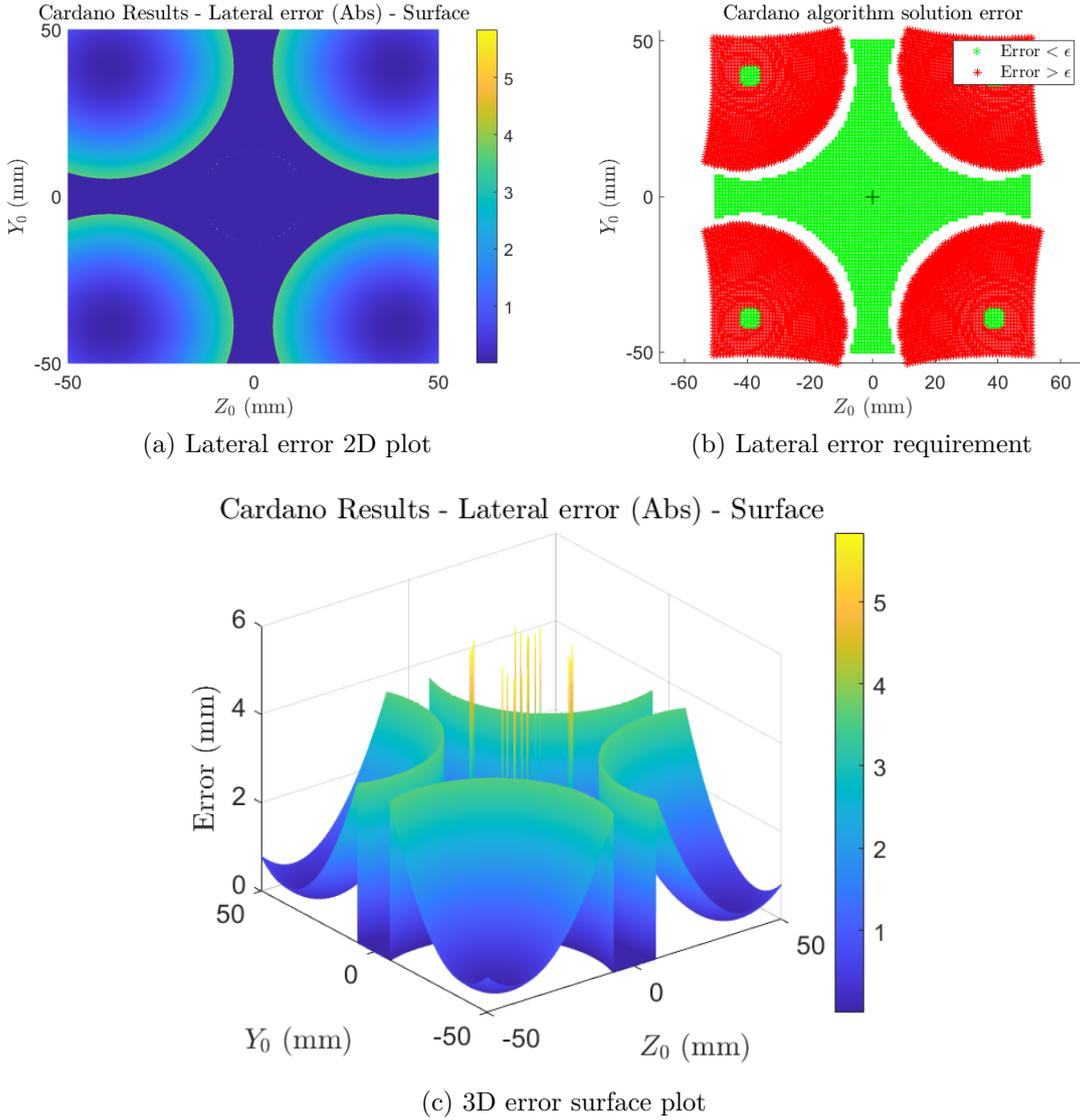


Figure C.1: Alternative algorithm inside the goal box using redundant SPS set

From Figure C.1b it is possible to identify the green centred cross that satisfies the 50 μm requirement; this is equivalent to the central "X" of Figure 6.15a rotated

by a 45 °angle due to the different set of SPS employed. Hence, it is reasonable to hypothesise that **the circle of increased error centred at a distance of 55 mm from the origin and with a radius of around 33.5 mm is a consequence of Cardano’s method that converts from polar to cartesian coordinates.**

This radius of 33.5 mm coincides with the distance from the origin to the beginning of the penumbra in the fitting curve of Figure 3.2. Thus it is possible to hypothesise that **when a single SPS is inside the umbra, the position estimation of Cardano’s method r_i delivers an inaccurate result, lowering the overall accuracy of the umbra coordinates.**

As a test, the employed volume was extended to ± 100 mm to observe how these saddle-like formations behave, and it is possible to observe in Figure C.2 that the error circles are present only in this range around the sensor. In contrast, the rest of the volume is covered with the outstanding precision seen in the requirement box. Consequently, the alternative algorithm delivers a reduced performance when the umbra centre approaches the SPS sensor.

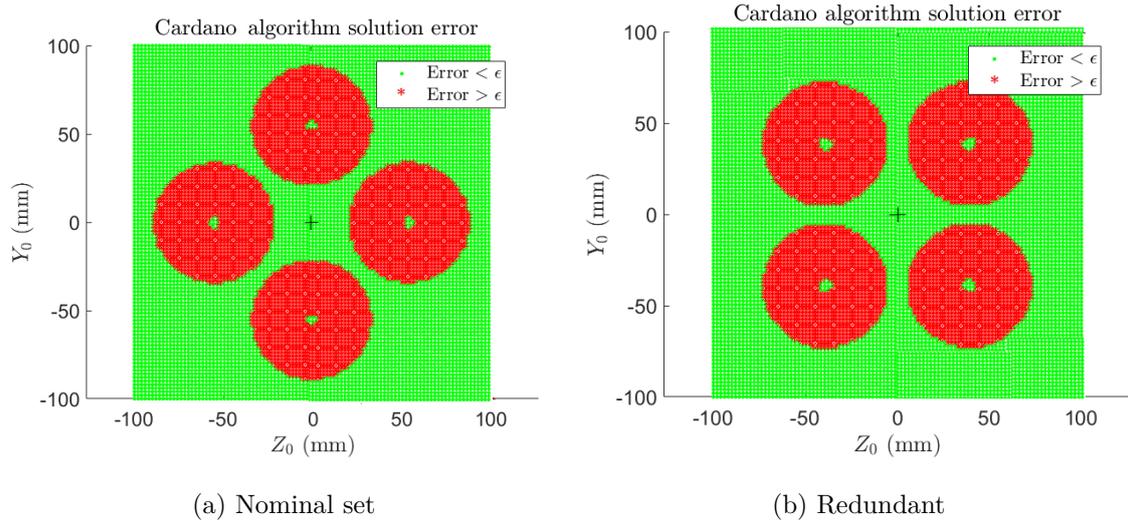


Figure C.2: Alternative algorithm inside the ± 100 mm box

The maximum error in Figure C.1c is lower (with 4 mm against the 5 mm using the nominal), and it is possible to observe how the error’s shape resembles the irradiance profile used for the inversion. A similar result is present in Figure A.1, where only the fine parameters are used, and the error increases considerably around a semi-circle centred at a 55 mm radius. Hence, this error disuniformity is present even when different reconfigurable parameters are used.

From these results, it is possible to imagine a viable explanation for this behaviour, but it goes beyond the objectives of this work. Nevertheless, a solution that might reduce the algorithm’s error inside the goal box is proposed, where the nominal

and redundant set coordinates are averaged as a single output. This might lessen the error spike in the central and diagonal semi-circles thanks to the increased precision of each set in these zones. However, an increase in the error where the spikes collide is expected. A significant overhaul of the script is needed to test these results, and a high computational cost is anticipated since the alternative algorithm shall be employed twice for every run (or four times if the points lie inside the requirement box). Hence, for future development of the work, the author encourages this application of the algorithm to demonstrate these premises.

C.2 Pseudo-paraboloidal Algorithm

To verify the correct functioning of the script, the same test was performed using the pseudo-paraboloidal algorithm. However, no previous data is available for the error shape of the PP algorithm using only the redundant SPS set.

It is evident how Figure [C.3b](#) and [6.17](#) show an almost identical distribution of the error in the central zone, with a rotation of 45 degrees as a consequence of the different SPS set used.

The maximum error obtained is unacceptable, with a peak at almost 18 mm; this might be a result of the rotation that makes the algorithm work in a not optimised region for its functioning (over the ± 50 mm allowed for each axis). Hence, these results are not a reflection of reality, yet the central zone allows to validate the functioning of the rotation script. Furthermore, if the edges are omitted, the performance is similar to the one expected.

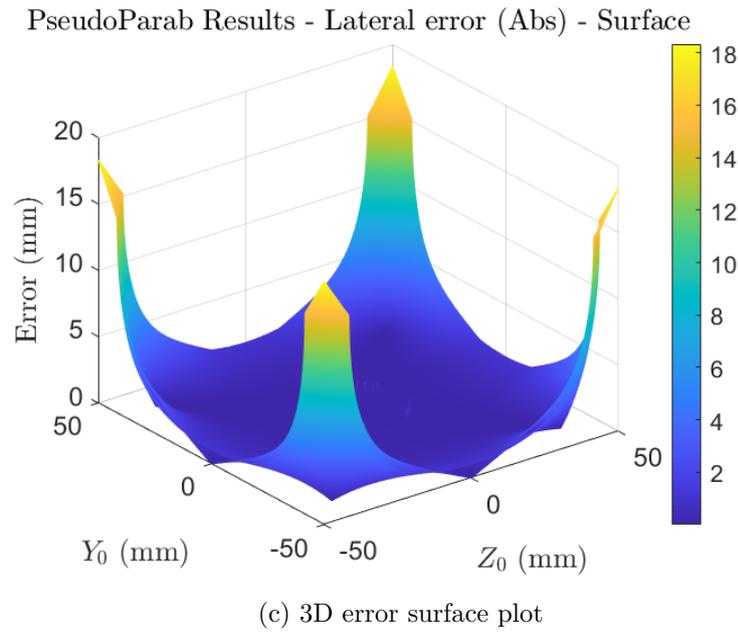
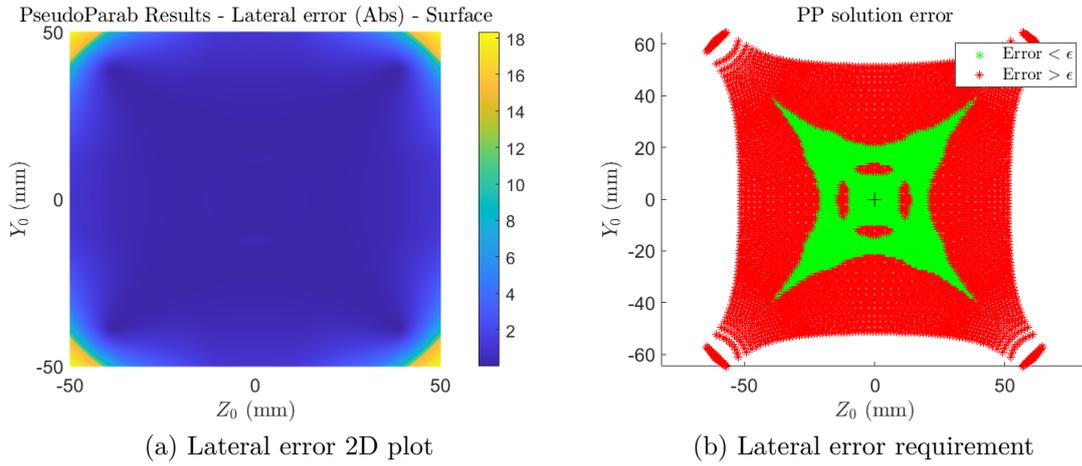


Figure C.3: PP algorithm inside the goal box using redundant SPS set

Appendix D

Matlab Codes

D.1 Cardano's Method function

```
function [ Solution, is_Real_Positive, flag, Δ] = ...
    Method_Cardan(p0, p1, p2, p3 )

% Polynomial must have real coefficients
flag = 0;
    if (~isreal([p0, p1, p2, p3]))
        error('Coefficient must be real');
    end

% Cardano's Parameter
p = -p2^2/(3*p3^2) + p1/p3;
q = p2/(27*p3) * (2*p2^2/p3^2 - 9*p1/p3) + p0/p3;
Δ = -(4*p^3 + 27*q^2);
j = -1/2+1i*sqrt(3)/2;

% Solution depends on the sign of Δ
    if(Δ<0)
        z1 = ((-q+sqrt(-Δ/27))/2)^(1/3) + ((-q-sqrt(-Δ/27))/2)^(1/3);
        z2 = j*((-q+sqrt(-Δ/27))/2)^(1/3) + ...
            conj(j)*((-q-sqrt(-Δ/27))/2)^(1/3);
        z3 = j^2*((-q+sqrt(-Δ/27))/2)^(1/3) + ...
            conj(j^2)*((-q-sqrt(-Δ/27))/2)^(1/3);
        flag = 1;
    elseif (Δ == 0)
        if (p==0 && q==0)
            z1=0;
            z2=0;
            z3=0;
            flag = 2;
        else
            z1 = 3*q/p;
```

```
        z2 = -3*q/(2*p);
        z3=z2;
        flag = 3;
    end
else
    u = ((-q+1i*sqrt(Δ/27))/2)^(1/3);
    z1 = u+conj(u);
    z2 = j*u + conj(j*u);
    z3 = j^2*u + conj(j^2*u);
    flag = 4;
end

Solution = [z1, z2, z3] - p2/(3*p3);
is_Real_Positive = [imag(Solution(1)) == 0 && real(Solution(1)) ≥ 0, ...
                    imag(Solution(2)) == 0 && real(Solution(2)) ≥ 0, ...
                    imag(Solution(3)) == 0 && real(Solution(3)) ≥ 0];

end
```

D.2 Alternative_SPS_Algorithm_Generate

D.2.1 Data acquisition

```
%% Data acquisition
% SPS framework
r_SPS = 0.055; % [m]
lateral_range = 0.041; %0.05 % [m]
% If higher than 0.041 there is more than one solution for Cardano's

% For mesh generation in the range +-10mm (Can be changed)
lateral_range = 0.01; % [m]
% [+10 mm for req box, +-50 mm for goal box]
%MAX for ONLY FINE = 0.041; MAX for ONLY COARSE = 0.02145

% Boundary conditions
BC_min = r_SPS - lateral_range; % [m]
BC_max = r_SPS + lateral_range*sqrt(2); % [m]

% MODIFIED BOUNDARY CONDITIONS
BC_min = r_SPS - 0.02145; % [m]
% The minimum radius shall be above the profile minimum (33.55 mm)
BC_max = r_SPS + 0.05*sqrt(2); % [m]
% The maximum radius stays unaltered

% Radius generation for umbra profile
r_ex=BC_min:0.00005:BC_max;

% Location of the four diodes
% FROM ESA MEMO & ICD V6.0, [Z,Y]
% (0,0) is the centre of CI entrance pupil [Z,Y]
Diode_A = [r_SPS,0]; % [m] SPS 7
Diode_B = [0,r_SPS]; % [m] SPS 1
Diode_C = [-r_SPS,0]; % [m] SPS 3
Diode_D = [0,-r_SPS]; % [m] SPS 5
```

Generate Local Data

```

%% Generate data (Local)
N=10000;      %Number of trials
epsilon = 10^-5.6; %10^-6           %[m], 1 micron of tolerance
% Exponent changed to see the precision patters in the green-red plot
epsilon = 50*10^-6;                %[m], 50 micron for requirement
% 50 micron req is satisfied

```

Random position generator

```

% Random position of the umbra centre Inside the goal box +/- ...
lateral_range
Umbra_centre = (2*lateral_range*rand(N,2)-lateral_range);      %[m]
Umbra_centre = [linspace(-lateral_range, lateral_range,N)',...
                linspace(-lateral_range, lateral_range,N)']*0;%[m]

```

Mesh generation

```

%% Generate mesh of N equi-spaced points Inside the goal box ...
+/- lateral_range
Z_centre = linspace(-lateral_range,lateral_range,sqrt(N))';
Umbra_centre = [linspace(-lateral_range, lateral_range,...
                        sqrt(N))', Z_centre(1)*ones(sqrt(N),1)];
for i = 2:sqrt(N)
Umbra_centre = [Umbra_centre;[linspace(-lateral_range, ...
                        lateral_range, sqrt(N))',...
                        Z_centre(i)*ones(sqrt(N),1)]];
end

% If an specific point is tested, modify the next Umbra Centre:
% Umbra_centre = [0,0];           %[m] [Z,Y]
% N=1;

% Distance diodes <--> Umbra centre
% (Changing order and sign of the centre to obtain the rigth SPS ...
readouts)
r_a = sqrt((-Umbra_centre(:,2) - Diode_A(2)).^2 + ...
            (-Umbra_centre(:,1) - Diode_A(1)).^2 ); %[m]
r_b = sqrt((-Umbra_centre(:,2) - Diode_B(2)).^2 + ...
            (-Umbra_centre(:,1) - Diode_B(1)).^2 ); %[m]
r_c = sqrt((-Umbra_centre(:,2) - Diode_C(2)).^2 + ...
            (-Umbra_centre(:,1) - Diode_C(1)).^2 ); %[m]
r_d = sqrt((-Umbra_centre(:,2) - Diode_D(2)).^2 + ...

```

```
(-Umbra_centre(:,1) - Diode_D(1)).^2 ); % [m]
```

D.2.2 Irradiance model Parameters (fine or coarse)

```
%% Irradiance model WITH DN(axis-symmetric) NEW
% RECONFIGURABLE PARAMETERS
% LATERAL FINE
alpha    = -3.6907368281401e-6;           % [m/DN]
beta     = 0.000437857008493415;        % [m^2/DN]
gamma    = -0.00350110848394894;        % [m^3/DN]
% R0     = 4034.634917033389;           % [DN] ...
    Irradiance

% LATERAL COARSE
% alpha   = -2.24536756733076e-5;       % [m/DN]
% beta    = 0.00078031275691922;        % [m^2/DN]
% gamma   = -0.0061184383374413;        % [m^3/DN]
% R0      = -758.2013024308603;          % [DN]

% LONGITUDINAL
H = 0.000284393727733714;               % [DN/mm^2]
K = -0.52934955050238;                  % [DN/mm]
L = 1034;                                % [DN]
J = 0;                                    % [1/mm]
Δ_R = 3000.634917033389;                 % [DN]
```

R_0 determination

```
R_55 = 55*10^-3/alpha + (55*10^-3)^2/beta^2 + (55*10^-3)^3/gamma^3;
%[DN] NEEDS TO BE = 1034 DN = L
R0 = L - R_55;                            % [DN]
```

Penumbra profile plotting

```
%% Plot of the Penumbra

y_fit = R0 + r_ex/alpha + r_ex.^2/beta^2 + r_ex.^3/gamma^3;
% Traslation of the profile to be > 0
y_fit_2 = r_ex/alpha + r_ex.^2/beta^2 + r_ex.^3/gamma^3;

figure('name','Model','visible','on')
plot(r_ex*1000, y_fit,'r-')
```

```
% hold on
% plot(r_ex*1000, y_fit_2,'b-')
title('3rd order axis-symmetric irradiance ...
      model','Interpreter','LaTeX');
xlim([BC_min*1000 BC_max*1000]);
xlabel('radius (mm)','Interpreter','LaTeX');
ylabel('Irradiance [DN]','Interpreter','LaTeX');
% legend('With R_0','Without R_0','Interpreter','LaTeX')
fig = gca;
fig.FontSize = 14;
```

Penumbra surface plotting

```
%% Penumbra 3D Surface
teta = linspace(0,2*pi,500);
r_surf = linspace(BC_min,BC_max-0.006,500);
[TETA,R] = ndgrid(teta,r_surf);
Y = R0 + R/alpha + R.^2/beta^2 + R.^3/gamma^3;
[X,Y,Z] = pol2cart(TETA,R,Y); % Set 'Z' As The Radius Matrix, 'Y' ...
      Matrix As 'Z'

figure('name','3D Penumbra','visible','on')
s = surf(X*10^3,Y*10^3,Z);
title('3rd order axis-symmetric irradiance ...
      model','Interpreter','LaTeX');
zlabel('Irradiance [DN]','Interpreter','LaTeX');
xlabel('Z (mm)','Interpreter','LaTeX');
ylabel('Y (mm)','Interpreter','LaTeX');
s.EdgeColor = 'interp';
s.FaceColor = 'interp';
colorbar
fig=gca;
fig.FontSize =14;
```

Aproximation of irradiance measurement

```
%% APROXIMATION OF SPS MEASUREMENT DATA
% Compute irradiance at the location of the diodes

% Very precise DN values (ideal, not integers...)
% R_A = R0 + r_a/alpha + r_a.^2/beta^2 + r_a.^3/gamma^3; % [DN]
% R_B = R0 + r_b/alpha + r_b.^2/beta^2 + r_b.^3/gamma^3; % [DN]
% R_C = R0 + r_c/alpha + r_c.^2/beta^2 + r_c.^3/gamma^3; % [DN]
% R_D = R0 + r_d/alpha + r_d.^2/beta^2 + r_d.^3/gamma^3; % [DN]
```

```
% Integer DN values (Real)
R_A = fix(R0 + r_a/alpha + (r_a./beta).^2 + (r_a./gamma).^3); % [DN]
R_B = fix(R0 + r_b/alpha + (r_b./beta).^2 + (r_b./gamma).^3); % [DN]
R_C = fix(R0 + r_c/alpha + (r_c./beta).^2 + (r_c./gamma).^3); % [DN]
R_D = fix(R0 + r_d/alpha + (r_d./beta).^2 + (r_d./gamma).^3); % [DN]
```

Vector initialization

```
%% Vectors initialization
% Solutions from Cardan Method
r_a_sol = zeros(N,3);
r_b_sol = zeros(N,3);
r_c_sol = zeros(N,3);
r_d_sol = zeros(N,3);
% Real+Positive boolean for the solution
bool_a_sol = zeros(N,3);
bool_b_sol = zeros(N,3);
bool_c_sol = zeros(N,3);
bool_d_sol = zeros(N,3);
```

D.2.3 Solving with Cardano’s Method function

```
%% SOLVE FROM GENERATED DATA
for i=1:N
[r_a_sol(i,:), bool_a_sol(i,:), a_flag(i,)] = ...
    Method_Cardano(R0-R_A(i), 1/alpha, 1/beta^2, 1/gamma^3);
[r_b_sol(i,:), bool_b_sol(i,:), b_flag(i,)] = ...
    Method_Cardano(R0-R_B(i), 1/alpha, 1/beta^2, 1/gamma^3);
[r_c_sol(i,:), bool_c_sol(i,:), c_flag(i,)] = ...
    Method_Cardano(R0-R_C(i), 1/alpha, 1/beta^2, 1/gamma^3);
[r_d_sol(i,:), bool_d_sol(i,:), d_flag(i,)] = ...
    Method_Cardano(R0-R_D(i), 1/alpha, 1/beta^2, 1/gamma^3);
end
```

Positive and real solutions

```
%% Keep only the positive solution
r_a_sol = r_a_sol.*bool_a_sol; % [m]
r_b_sol = r_b_sol.*bool_b_sol; % [m]
r_c_sol = r_c_sol.*bool_c_sol; % [m]
r_d_sol = r_d_sol.*bool_d_sol; % [m]
```

Controlling the Boundary conditions

```

%% Validation with respect to the boundaries conditions
BC_a_sol = (r_a_sol >= BC_min) .* (r_a_sol <= BC_max);
BC_b_sol = (r_b_sol >= BC_min) .* (r_a_sol <= BC_max);
BC_c_sol = (r_c_sol >= BC_min) .* (r_a_sol <= BC_max);
BC_d_sol = (r_d_sol >= BC_min) .* (r_a_sol <= BC_max);

Check_1 = sum(sum(BC_a_sol))+ sum(sum(BC_b_sol)) + ...
          sum(sum(BC_c_sol))+ sum(sum(BC_d_sol));
if (Check_1 <= 4*N)
error('More than 1 sol')
end

r_a_sol = sum(r_a_sol .* BC_a_sol,2);           % [m]
r_b_sol = sum(r_b_sol .* BC_b_sol,2);           % [m]
r_c_sol = sum(r_c_sol .* BC_c_sol,2);           % [m]
r_d_sol = sum(r_d_sol .* BC_d_sol,2);           % [m]

```

Computation of the lateral coordinates of the centre

```

%% Compute the position of the umbra centre [m]
Umbra_centre_sol = [-(r_c_sol.^2 - r_a_sol.^2)/(4*r_SPS), ...
                   -(r_d_sol.^2 - r_b_sol.^2)/(4*r_SPS)]; ...
                   % [m] [-Z,-Y] From the positions
% Cardano's algo gives the solution with opposite sign...
% (correction), this is different from the READ script that...
% uses the corrections (so no - sign is needed)

y_0 = Umbra_centre_sol(:,2);                     % [m] (VERTICAL)
z_0 = Umbra_centre_sol(:,1);                     % [m] (HORIZONTAL)

```

D.2.4 Longitudinal position calculation

```

%% Evaluation of the longitudinal position
d_0 = sqrt((z_0.*10^3).^2+(y_0.*10^3).^2);        % [mm]
% Compute the distance of the diodes from the evaluated umbra centre
rho_1_d_2 = sqrt(z_0.^2 + (r_SPS-y_0).^2);        % [m]
rho_3_d_4 = sqrt((r_SPS+z_0).^2 + y_0.^2);        % [m]
rho_5_d_6 = sqrt(z_0.^2 + (r_SPS+y_0).^2);        % [m]
rho_7_d_8 = sqrt((r_SPS-z_0).^2 + y_0.^2);        % [m]
% Compute digital number value stating from the measured values
R_01_02 = R_A - ...
          (rho_1_d_2./alpha)-(rho_1_d_2./beta).^2-(rho_1_d_2./gamma).^3;

```

```

R_03_04 = R_B - ...
    (rho_3_d_4./alpha)-(rho_3_d_4./beta).^2-(rho_3_d_4./gamma).^3;
R_05_06 = R_C - ...
    (rho_5_d_6./alpha)-(rho_5_d_6./beta).^2-(rho_5_d_6./gamma).^3;
R_07_08 = R_D - ...
    (rho_7_d_8./alpha)-(rho_7_d_8./beta).^2-(rho_7_d_8./gamma).^3;
R_0 = (1/4).*(R_01_02 + R_03_04 + R_05_06 + R_07_08);
% Digital Numbers read at 55 mm
R_55 = R_0 - Δ_R; % [DN]
% Compute longitudinal position
for i=1:N
if (K^2-4*H*(L-R_55(i)))<0
    x_0(i) = -9999999;
else
    x_0(i) = ((-K-sqrt(K^2-4*H*(L-R_55(i))))/(2*H)+(J*d_0(i)^2)); ...
        % [mm]
end
end
y_0 = y_0.*10^3; % [mm] (VERTICAL)
z_0 = z_0.*10^3; % [mm] (HORIZONTAL)

```

D.2.5 Solution validation

```

%% Verify the solution
Difference = abs(Umbra_centre - Umbra_centre_sol);
is_Correct = (Difference ≤ epsilon);
for j=1:N
mean_Diff(j) = abs(mean(Umbra_centre(j) - Umbra_centre_sol(j)));
end

% If error without absolute value uncomment this
% for j=1:N
% mean_Diff(j) = mean(Difference(j));
% end

error = mean_Diff(1:sqrt(N))*10^3;
for i=2:sqrt(N)
error = [error,mean_Diff(1+sqrt(N)*(i-1):sqrt(N)*i)*10^3];
end

```

Error 2D threshold plot

```

%% Plot a map showing the results
% This is the slowest part, write 'off' to speed up the Run
figure('name','Cardano`s solution error','visible','on')

```

```

for i=1:N
    if (is_Correct(i,1) ==1 && is_Correct(i,2)==1)
        hold on
        plot(z_0(i),y_0(i),'go','MarkerFaceColor','g','MarkerSize',2)
    else
        hold on
        plot(z_0(i),y_0(i),'r*')
        %hold on
        %plot(z_0(i),y_0(i),'r+')
    end
end
end
hold on
plot(0,0,'k+','MarkerSize',10)
title('Cardano`s solution error','Interpreter','LaTeX');
axis equal
xlabel('$Z_0$ (mm)','Interpreter','LaTeX');
ylabel('$Y_0$ (mm)','Interpreter','LaTeX');
legend('Error $< \epsilon$', 'Error $> ...
\epsilon$', 'Interpreter','LaTeX')
fig=gca;
fig.FontSize =14;

```

Lateral error 3D plot

```

%% Lateral Error plot
figure('name','Results - Lateral error - Plot','visible','off')
plot3(Umbra_centre(:,2)*10^3,Umbra_centre(:,1)*10^3,mean_Diff*10^3,'*')
hold on
title('Results - Lateral error - Plot','Interpreter','LaTeX');
xlabel('$Z_0$ (mm)','Interpreter','LaTeX');
ylabel('$Y_0$ (mm)','Interpreter','LaTeX');
zlabel('$Error$ (mm)','Interpreter','LaTeX');
fig=gca;
fig.FontSize =14;

```

Lateral error 3D surface

```

%% Lateral Error Surface
[Z_grid,Y_grid] = ...
    meshgrid(linspace(-lateral_range*10^3,lateral_range*10^3,sqrt(N)));
figure('name','Results - Lateral error - Surface','visible','on')
s1 = surf(Z_grid,Y_grid,error);
hold on
s2 = surf(Y_grid,Z_grid,error);
title('Results - Lateral error - Surface','Interpreter','LaTeX');

```

```
xlabel('$Z_0$ (mm)', 'Interpreter', 'LaTeX');  
ylabel('$Y_0$ (mm)', 'Interpreter', 'LaTeX');  
zlabel('$Error$ (mm)', 'Interpreter', 'LaTeX');  
s1.EdgeColor = 'interp';  
s1.FaceColor = 'interp';  
s2.EdgeColor = 'interp';  
s2.FaceColor = 'interp';  
colorbar  
fig=gca;  
fig.FontSize =14;
```

Longitudinal error plot

```
% Longitudinal Error plot  
figure('name', 'Results - Longitudinal error - Plot', 'visible', 'on')  
plot([1:N], x_0)  
hold on  
title('Results - Longitudinal error - Plot', 'Interpreter', 'LaTeX');  
xlabel('Test \#', 'Interpreter', 'LaTeX');  
ylabel('$Error$ (mm)', 'Interpreter', 'LaTeX');  
fig=gca;  
fig.FontSize =14;
```

D.3 Alternative Algorithm - Read

D.3.1 .JSON File loading

```

%% Load data from SPS
% Load Reconfigurable parameter (from 1 to 3, minor variations of ...
  the R0, increasing order of precision)
Reconf_params = ...
  jsondecode(fileread('Reconfigurable_params_Alt_3.json'));

% Acquire data from SPS (from 1 to 12 different random positions ...
  in +/-10mm)
Raw_data = jsondecode(fileread('RAW_DATA\raw_data_Alt_10.json'));

%Choose the set to be used
set = ['NOM'];
% set = ['RED'];

```

D.3.2 SPS irradiance reading

```

%% Irradiance reading
%% Nominal set
if strcmp(set, 'NOM')==1
  % Read DN in HG
  R_A = Raw_data.Read(7*2-1);      % [DN] SPS 7
  R_B = Raw_data.Read(1*2-1);      % [DN] SPS 1
  R_C = Raw_data.Read(3*2-1);      % [DN] SPS 3
  R_D = Raw_data.Read(5*2-1);      % [DN] SPS 5
end
%% Redundant set
if strcmp(set, 'RED')==1
  % Read DN in HG
  R_A = Raw_data.Read(8*2-1);      % [DN] SPS 8
  R_B = Raw_data.Read(2*2-1);      % [DN] SPS 2
  R_C = Raw_data.Read(4*2-1);      % [DN] SPS 4
  R_D = Raw_data.Read(6*2-1);      % [DN] SPS 6
end

```

D.3.3 Cardano's solution

```

% Solve the system to compute the umbra centre position
%% FROM FILE READING
[r_a_sol(:), bool_a_sol(:), a_flag]= Method_Cardano(...
    Reconf_params.cardan.R0-R_A, 1/Reconf_params.cardan.alpha,...
    1/Reconf_params.cardan.beta^2, 1/Reconf_params.cardan.gamma^3);
[r_b_sol(:), bool_b_sol(:), b_flag]= Method_Cardano(...
    Reconf_params.cardan.R0- R_B, 1/Reconf_params.cardan.alpha,...
    1/Reconf_params.cardan.beta^2, 1/Reconf_params.cardan.gamma^3);
[r_c_sol(:), bool_c_sol(:), c_flag]= Method_Cardano(...
    Reconf_params.cardan.R0-R_C, 1/Reconf_params.cardan.alpha,...
    1/Reconf_params.cardan.beta^2, 1/Reconf_params.cardan.gamma^3);
[r_d_sol(:), bool_d_sol(:), d_flag]= Method_Cardano(...
    Reconf_params.cardan.R0-R_D, 1/Reconf_params.cardan.alpha,...
    1/Reconf_params.cardan.beta^2, 1/Reconf_params.cardan.gamma^3);

%% Keep only the positive solution
r_a_sol = r_a_sol.*bool_a_sol;           % [m]
r_b_sol = r_b_sol.*bool_b_sol;           % [m]
r_c_sol = r_c_sol.*bool_c_sol;           % [m]
r_d_sol = r_d_sol.*bool_d_sol;           % [m]

%% Validation with respect to the boundaries conditions
BC_a_sol = (r_a_sol >= BC_min) .* (r_a_sol <= BC_max);
BC_b_sol = (r_b_sol >= BC_min) .* (r_b_sol <= BC_max);
BC_c_sol = (r_c_sol >= BC_min) .* (r_c_sol <= BC_max);
BC_d_sol = (r_d_sol >= BC_min) .* (r_d_sol <= BC_max);

Check_1 = sum(sum(BC_a_sol))+ sum(sum(BC_b_sol)) + ...
    sum(sum(BC_c_sol))+ sum(sum(BC_d_sol));
if (Check_1 <= 4)
    error('More than 1 sol')
end

r_a_sol = sum(r_a_sol .* BC_a_sol,2);     % [m]
r_b_sol = sum(r_b_sol .* BC_b_sol,2);     % [m]
r_c_sol = sum(r_c_sol .* BC_c_sol,2);     % [m]
r_d_sol = sum(r_d_sol .* BC_d_sol,2);     % [m]
%% Compute the lateral position of the umbra centre [m]
Umbra_centre_sol = [(r_c_sol.^2 - r_a_sol.^2)/(4*r_SPS),...
    (r_d_sol.^2 - r_b_sol.^2)/(4*r_SPS)];% [m] [Z, Y]

y_0 = Umbra_centre_sol(:,2);             % [m] (VERTICAL)
z_0 = Umbra_centre_sol(:,1);             % [m] (HORIZONTAL)

%% The rotation is applied only in the redundant set
if strcmp(set, 'RED')==1
    % Position in the SPSMRF reference frame : Rotation

```

```

z_0_nom = z_0;
y_0_nom = y_0;
y_0 = (sqrt(2)/2)*y_0_nom + (sqrt(2)/2)*z_0_nom;
z_0 = -(sqrt(2)/2)*y_0_nom + (sqrt(2)/2)*z_0_nom;
end

```

D.3.4 Linear solution

```

%% Lateral position computed using the linear & the alternative ...
   algorithm
% Calculate the centre of the penumbra within the lateral plane...
% using the linear algorithm
y_0_L = (R_D - R_B) / Reconf_params.linear.d_y*10^3;      % [mm]
z_0_L = (R_C - R_A) / Reconf_params.linear.d_z*10^3;      % [mm]

% The rotation is applied only in the redundant set
if strcmp(set, 'RED')==1
    % Position in the SPSMRF reference frame : Rotation
    z_0_nom_2 = z_0_L;
    y_0_nom_2 = y_0_L;
    y_0_L = (sqrt(2)/2)*y_0_nom_2 + (sqrt(2)/2)*z_0_nom_2;
    z_0_L = -(sqrt(2)/2)*y_0_nom_2 + (sqrt(2)/2)*z_0_nom_2;
end

y_0_2 = ((y_0_L + y_0) / 2)*10^-3;      % [m]
z_0_2 = ((z_0_L + z_0) / 2)*10^-3;      % [m]

```

D.4 Reconfigurable_params.json

```
"K" : [ 1, 1, 1, 1, 1, 1, 1, 1 ],
"g" : [ 5, 5, 5, 5, 5, 5, 5, 5 ],
"algo" : 0,
"pp_coarse" :
  {
    "a_coarse" : -2.24536756733076E-5,
    "b_coarse" : 0.00078031275691922,
    "c_coarse" : -0.0061184383374413,
    "A_coarse" : -0.02210291624957864,
    "R0_coarse" : -758.2013024308603;
  },
"pp_fine":
  {
    "a_fine" : -3.6907368281401E-6,
    "b_fine" : 0.000437857008493415,
    "c_fine" : -0.00350110848394894,
    "A_fine" : -0.0039202535866095616,
    "R0_fine" : 4034.634917033389;
    "Δ_transv" : 13.8
  },
"linear":
  {
    "d_z" : 1.858E+05,
    "d_y" : 1.858E+05
  },
"long" :
  {
    "H" : 0.000284393727733714,
    "K" : -0.52934955050238,
    "L" : 1034,
    "J" : 0,
    "Δ_R" : 3000.634917033389
  },
"thresholds":
  {
    "V_max_lat" : 5,
    "V_max_long" : 50,
    "T_max" : 348,
    "Δ_flag" : 5
  }
}
```

D.5 RAW_DATA.json

For the nominal $x_0 = y_0 = z_0 = 0$ position:

```
"Read": [1034, 206, 1034, 206, 1034, 206, 1034, 206, 1034, ...
         206, 1034, 206, 1034, 206, 1034, 206],
"Temp": [298, 298, 298, 298],
"door" : 0,
"door_tr": 0,
"pos_re": [0,0]
```

D.6 Test case SPS readings

Table D.1: SPS High Gain (HG) measurements inside the requirement box used as “RAW_DATA” for the Read algorithm

Test Case #	Input Position			SPS Measurements							
	X	Y	Z	1_HG	2_HG	3_HG	4_HG	5_HG	6_HG	7_HG	8_HG
1	0	0	0	1034	1034	1034	1034	1034	1034	1034	1034
2	0	1.4	0.5	1177	1129	987	896	896	987	1081	1177
3	0	4.52	1.9	1481	1226	851	639	639	896	1226	1533
4	0	-5.59	2.4	560	560	851	1276	1586	1639	1326	896
5	0	8.52	4.7	1914	1377	680	279	343	851	1533	2027
6	0	0.18	0.9	1034	987	941	941	1034	1081	1129	1129
7	0	2.3	0.6	1276	1129	987	851	807	941	1081	1226
8	0	8.9	2.5	1971	1586	896	377	311	680	1326	1914
9	0	1.7	3.4	1177	941	722	722	896	1177	1377	1377
10	0	6.5	4.6	1693	1226	680	411	484	987	1533	1858
11	0	3.3	-7.5	1429	1803	1803	1377	764	411	411	807
12	0	1.1	-2.1	1129	1276	1226	1129	941	851	851	987

D.7 Main_Alt

Initialization

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Formation Flying Metrology: SPS Alternative Algorithm for ...
%   Proba-3/ASPIICS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Previous Loop Data & Vector initialization

% Nominal Set: Position vector related to the measurements @ t = t_n-1
nom_01 = [-999999, -999999, -999999, -999999, -999999];
% Redundant Set: Position vector related to the measurements @ t = ...
%   t_n-1
red_01 = [-999999, -999999, -999999, -999999, -999999];
% NOTE that the nom_01 and red_01 are here initialized with their ...
%   starting values.
% This is acceptable when running the algorithm stand alone for ...
%   debugging and verification.
% During on-board operations, we expect that the nom_01 and red_01
% will be iteratively updated with new values picked up from a ...
%   storage memory
% where the n-1 position estimation is saved.
% Otherwise, the SPS algo will loop over a meaningless comparison ...
%   between
% the current position and previous one when taking into account ...
%   the drift velocity
% between the two spacecrafts (validity_flag_2).

% Nominal set: output vector initialization
nom = [0, -999999, -999999, -999999, -999999, -999999, 0, 0, 1];
% Redundant set: output vector initialization
red = [0, -999999, -999999, -999999, -999999, -999999, 0, 0, 1];

```

Data loading

```

% Load Reconfigurable parameters
Reconf_params = jsondecode(fileread('Reconfigurable_params.json'));

% Acquire data from SPSs
% Acquire data from SPS (from 1 to 12 different random positions ...
%   in +/-10mm)
% Raw_data = jsondecode(fileread('raw_data.json'));
Raw_data = jsondecode(fileread('RAW_DATA\raw_data_Alt_12.json'));
%THERE IS A DIFFERENCE IN THE ORDER OF THE SPS READINGS

```

```
Reconf_params.algo = 0; %0 for BOTH, 1 for PSEUDO/ALT, 2 for LINEAR

% Including the R0 as a reconf parameters
R_55_coarse = 55*10^-3/Reconf_params.pp_coarse.a_coarse +...
              (55*10^-3)^2/Reconf_params.pp_coarse.b_coarse^2 +...
              (55*10^-3)^3/Reconf_params.pp_coarse.c_coarse^3;
Reconf_params.pp_coarse.R0_coarse = Reconf_params.long.L - ...
R_55_coarse;
R_55_fine = 55*10^-3/Reconf_params.pp_fine.a_fine +...
            (55*10^-3)^2/Reconf_params.pp_fine.b_fine^2 +...
            (55*10^-3)^3/Reconf_params.pp_fine.c_fine^3;
Reconf_params.pp_fine.R0_fine = Reconf_params.long.L - R_55_fine;
```

Input Pre-Processing

```
[R, set, temps] = Input_Pre_Processing(Raw_data.Read, ...
Raw_data.Temp, Reconf_params.g);

%% if no set is available, the calculation will be skipped
if strcmp(set, 'NONE') == 0

    %% if the door is in transition, the calculation will be skipped
    if ~Raw_data.door_tr
```

First Validity Check

```
[R, nom(1), red(1)] = Validity_Flag_1(R, set, temps, ...
Raw_data.door, Reconf_params.thresholds.T_max, ...
Reconf_params.K, Reconf_params.g);
```

Algo Verification

```
[nom(9), red(9)] = Algo_Verification(R, set, ...
Reconf_params.algo, Reconf_params.pp_fine, ...
Reconf_params.pp_coarse);
```

Position computation

```
% Relative position Computation (Alt)
%Only Fine parameters
[nom(2:6), red(2:6)] = SPS_Algorithms_Alt(R, set, ...
    Reconf_params.algo, Reconf_params.pp_coarse, ...
    Reconf_params.pp_fine, Reconf_params.linear, ...
    Reconf_params.long, nom(9), red(9));

%Fine and coarse parameters
tic
[nom(2:6), red(2:6)] = SPS_Algorithms_Alt_2(R, set, ...
    Reconf_params.algo, Reconf_params.pp_coarse, ...
    Reconf_params.pp_fine, Reconf_params.linear, ...
    Reconf_params.long, nom(9), red(9));
time_alt = toc;
```

Second Validity check

```
[nom(7), red(7), nom(8), red(8)] = Validity_Flag_2( set, ...
    nom_01, nom(2:6), red_01, red(2:6), ...
    Reconf_params.thresholds );

    end
end
```

Output

```
nom_01 = nom(2:6);
red_01 = red(2:6);
```

D.8 SPS_Algorithms_Alt_2

Initialization

```

%% SPS_Algorithms Alt 2
function [nom, red] = SPS_Algorithms_Alt_2 (R, set, algo, ...
    pp_coarse, pp_fine, linear, long, pp_check_nom, pp_check_red)

    %% variables initialization
    nom = [-999999, -999999, -999999, -999999, -999999];
    red = [-999999, -999999, -999999, -999999, -999999];

    %% Nominal set
    if (strcmp(set, 'NOM')==1 || strcmp(set, 'BOTH')==1) && ...
        bitget(pp_check_nom, 2) == 0 && bitget(pp_check_nom, 3) == ...
        0 && bitget(pp_check_nom, 4) == 0
        nom = calculate_positions(R(1), R(2), R(3), R(4), linear, ...
            pp_coarse, pp_fine, long, algo, 'NOM', pp_check_nom);
    end
    %% Redundant set
    if (strcmp(set, 'RED')==1 || strcmp(set, 'BOTH')==1) && ...
        bitget(pp_check_red, 2) == 0 && bitget(pp_check_red, 3) == ...
        0 && bitget(pp_check_red, 4) == 0
        red = calculate_positions(R(5), R(6), R(7), R(8), linear, ...
            pp_coarse, pp_fine, long, algo, 'RED', pp_check_red);
    end
end
end

```

Position calculation

```

%% calculate_positions
function [position] = calculate_positions(R1, R2, R3, R4, linear, ...
    pp_coarse, pp_fine, long, algo, set_temp, pp_check)

    %% variables initialization
    r_SPS = 0.055;
    assign_value = 0;
    position = [-999999, -999999, -999999, -999999, -999999];

    % Absolute relative position computation using the absolute ...
    algorithm
    switch algo
        case 0
            %% Lateral position computed using the linear & the ...
            alternative algorithm

```

```

% Calculate the centre of the penumbra within the ...
% lateral plane using the linear
% algorithm
z_0_L = (R2 - R4) / linear.d_z;           %[m]
y_0_L = (R3 - R1) / linear.d_y;           %[m]
% Calculate the centre of the penumbra within the ...
% lateral plane
% using the paraboloid algorithm
[z_0_PP, y_0_PP, assign_value] = alternative_calc(...
    pp_coarse, pp_fine, pp_check, R1, R2, R3, R4);
% Compute the M point coordinates within the lateral plane
z_0 = (z_0_L + z_0_PP) / 2;               %[m]
y_0 = (y_0_L + y_0_PP) / 2;               %[m]

case 1
    %% Lateral position computed using the alternative ...
    % algorithm
    [z_0, y_0, assign_value] = alternative_calc(pp_coarse, ...
        pp_fine, pp_check, R1, R2, R3, R4);

case 2
    %% Lateral position computed using the linear algorithm
    % Calculate the centre of the penumbra within the ...
    % lateral plane using the linear
    % algorithm
    z_0 = (R2 - R4) / linear.d_z;           %[m]
    y_0 = (R3 - R1) / linear.d_y;           %[m]
    assign_value = 1;

end

%% The algorithm continues only if the positions calculated ...
% are valid
if assign_value
    %% The rotation is applied only in the redundant set
    if strcmp(set_temp, 'RED')==1
        % Position in the SPSMRF reference frame : Rotation
        z_0_nom = z_0;
        y_0_nom = y_0;
        y_0 = (sqrt(2)/2)*y_0_nom + (sqrt(2)/2)*z_0_nom;
        z_0 = -(sqrt(2)/2)*y_0_nom + (sqrt(2)/2)*z_0_nom;
    end
    %% Evaluation of the longitudinal position
    d_0 = sqrt((z_0*10^3)^2+(y_0*10^3)^2); %[mm]
    % Compute the distance of the diodes from the evaluated ...
    % umbra centre
    rho_1_d_2 = sqrt(z_0^2 + (r_SPS-y_0)^2); %[m]
    rho_3_d_4 = sqrt((r_SPS+z_0)^2 + y_0^2); %[m]
    rho_5_d_6 = sqrt(z_0^2 + (r_SPS+y_0)^2); %[m]
    rho_7_d_8 = sqrt((r_SPS-z_0)^2 + y_0^2); %[m]

```

```

% Compute digital number value stating from the measured ...
    values
R_01_02 = R1-(rho_1_d_2/pp_fine.a_fine)-...
(rho_1_d_2/pp_fine.b_fine)^2-(rho_1_d_2/pp_fine.c_fine)^3;
R_03_04 = R2-(rho_3_d_4/pp_fine.a_fine)-...
(rho_3_d_4/pp_fine.b_fine)^2-(rho_3_d_4/pp_fine.c_fine)^3;
R_05_06 = R3-(rho_5_d_6/pp_fine.a_fine)-...
(rho_5_d_6/pp_fine.b_fine)^2-(rho_5_d_6/pp_fine.c_fine)^3;
R_07_08 = R4-(rho_7_d_8/pp_fine.a_fine)-...
(rho_7_d_8/pp_fine.b_fine)^2-(rho_7_d_8/pp_fine.c_fine)^3;
R_0 = (1 / 4) * (R_01_02 + R_03_04 + R_05_06 + R_07_08);
% Digital Numbers read at 55 mm
R_55 = R_0 - long.d_R; % [DN]

% Compute longitudinal position
if (long.K^2-4*long.H*(long.L-R_55)) < 0
    x_0 = -9999999;
else
    x_0 = ((-long.K-sqrt(long.K^2-4*long.H*(long.L-R_55)))/...
(2*long.H)+(long.J*d_0^2)); % [mm]
end

y_0 = y_0*10^3;
z_0 = z_0*10^3;

%% Save results
% Qualitative relative position computed using the ...
    differential algorithm
position(1) = (R3-R1)/(R1+R3);
position(2) = (R2-R4)/(R2+R4);
% Position calculated
position(3) = x_0;
position(4) = y_0;
position(5) = z_0;
end
end

```

D.8.1 Alternative algorithm function

```

%% alternative_calc
function [z_0, y_0, assign_value] = alternative_calc(pp_coarse,...
    pp_fine, tau, R1, R2, R3, R4)

%% variable initialization
sol_flag = 0;
assign_value = 0;
%R0 = -758.2013024308603; % [DN] COARSE
% SPS framework

```

```

r_SPS = 0.055; ...
                                                    % [m]

% MODIFIED BOUNDARY CONDITIONS
BC_min = r_SPS - 0.02145; ...
                                                    % [m]

% The minimum radius shall be above the profile minimum (33.55 mm)
BC_max = r_SPS + 0.05*sqrt(2); ...
                                                    % [m]

% The maximum radius stays unaltered
RA = R4; %SPS 7 / 8
RB = R1; %SPS 1 / 2
RC = R2; %SPS 3 / 4
RD = R3; %SPS 5 / 6

%% Alternative calculation (coarse)

[r_a_sol(:), bool_a_sol(:), a_flag]= ...
    Method_Cardano(pp_coarse.R0_coarse-RA, ...
1/pp_coarse.a_coarse, 1/pp_coarse.b_coarse^2, 1/pp_coarse.c_coarse^3);
[r_b_sol(:), bool_b_sol(:), b_flag]= ...
    Method_Cardano(pp_coarse.R0_coarse-RB, ...
1/pp_coarse.a_coarse, 1/pp_coarse.b_coarse^2, 1/pp_coarse.c_coarse^3);
[r_c_sol(:), bool_c_sol(:), c_flag]= ...
    Method_Cardano(pp_coarse.R0_coarse-RC, ...
1/pp_coarse.a_coarse, 1/pp_coarse.b_coarse^2, 1/pp_coarse.c_coarse^3);
[r_d_sol(:), bool_d_sol(:), d_flag]= ...
    Method_Cardano(pp_coarse.R0_coarse-RD, ...
1/pp_coarse.a_coarse, 1/pp_coarse.b_coarse^2, 1/pp_coarse.c_coarse^3);

%% Keep only the positive solution
r_a_sol = r_a_sol.*bool_a_sol;           % [m]
r_b_sol = r_b_sol.*bool_b_sol;           % [m]
r_c_sol = r_c_sol.*bool_c_sol;           % [m]
r_d_sol = r_d_sol.*bool_d_sol;           % [m]

%% Validation with respect to the boundaries conditions
BC_a_sol = (r_a_sol >= BC_min) .* (r_a_sol <= BC_max);
BC_b_sol = (r_b_sol >= BC_min) .* (r_b_sol <= BC_max);
BC_c_sol = (r_c_sol >= BC_min) .* (r_c_sol <= BC_max);
BC_d_sol = (r_d_sol >= BC_min) .* (r_d_sol <= BC_max);
Check_1 = sum(sum(BC_a_sol))+ sum(sum(BC_b_sol)) + ...
    sum(sum(BC_c_sol))+ sum(sum(BC_d_sol));
if (Check_1 <= 4) || a_flag == 0 || b_flag == 0 || ...
    c_flag == 0 || d_flag == 0
disp('More than 1 sol');
sol_flag = 1;
end
r_a_sol = sum(r_a_sol .* BC_a_sol, 2);    % [m]
r_b_sol = sum(r_b_sol .* BC_b_sol, 2);    % [m]
r_c_sol = sum(r_c_sol .* BC_c_sol, 2);    % [m]

```

```

r_d_sol = sum(r_d_sol .* BC_d_sol,2);      %[m]

%% Compute the lateral position of the umbra centre [m]
Umbra_centre_sol = [(r_c_sol.^2 - r_a_sol.^2)/(4*r_SPS),...
                    (r_d_sol.^2 - r_b_sol.^2)/(4*r_SPS)];%[m] [-Z,-Y]
y_0 = Umbra_centre_sol(:,2);              %[m] (VERTICAL)
z_0 = Umbra_centre_sol(:,1);              %[m] (HORIZONTAL)

%% Alternative calculation (fine)
if (sqrt(z_0^2 + y_0^2) * 10^3 < pp_fine.Δ_transv)% && ...
(sol_flag ==0)
    %R0 = 4034.634917033389; %[DN] FINE

clear r_a_sol r_b_sol r_c_sol r_d_sol bool_a_sol ...
      bool_b_sol bool_c_sol bool_d_sol
[r_a_sol(:), bool_a_sol(:),a_flag]= Method_Cardano( ...
    pp_fine.R0_fine-RA, 1/pp_fine.a_fine, ...
    1/pp_fine.b_fine^2,1/pp_fine.c_fine^3);
[r_b_sol(:), bool_b_sol(:),b_flag]= Method_Cardano(...
    pp_fine.R0_fine-RB, 1/pp_fine.a_fine, ...
    1/pp_fine.b_fine^2,1/pp_fine.c_fine^3);
[r_c_sol(:), bool_c_sol(:),c_flag]= Method_Cardano(...
    pp_fine.R0_fine-RC, 1/pp_fine.a_fine, ...
    1/pp_fine.b_fine^2,1/pp_fine.c_fine^3);
[r_d_sol(:), bool_d_sol(:),d_flag]= Method_Cardano(...
    pp_fine.R0_fine-RD, 1/pp_fine.a_fine, ...
    1/pp_fine.b_fine^2,1/pp_fine.c_fine^3);

%% Keep only the positive solution
r_a_sol = r_a_sol.*bool_a_sol;            %[m]
r_b_sol = r_b_sol.*bool_b_sol;            %[m]
r_c_sol = r_c_sol.*bool_c_sol;            %[m]
r_d_sol = r_d_sol.*bool_d_sol;            %[m]

%% Validation with respect to the boundaries conditions
BC_a_sol = (r_a_sol ≥ BC_min) .* (r_a_sol ≤ BC_max);
BC_b_sol = (r_b_sol ≥ BC_min) .* (r_a_sol ≤ BC_max);
BC_c_sol = (r_c_sol ≥ BC_min) .* (r_a_sol ≤ BC_max);
BC_d_sol = (r_d_sol ≥ BC_min) .* (r_a_sol ≤ BC_max);
Check_1 = sum(sum(BC_a_sol))+ sum(sum(BC_b_sol)) + ...
          sum(sum(BC_c_sol))+ sum(sum(BC_d_sol));
if (Check_1 ≠ 4) || a_flag == 0 || b_flag == 0 || ...
    c_flag == 0 || d_flag == 0
disp('More than 1 sol')
sol_flag = 1;
end
r_a_sol = sum(r_a_sol .* BC_a_sol,2);      %[m]
r_b_sol = sum(r_b_sol .* BC_b_sol,2);      %[m]
r_c_sol = sum(r_c_sol .* BC_c_sol,2);      %[m]
r_d_sol = sum(r_d_sol .* BC_d_sol,2);      %[m]

```

```
    %% Compute the lateral position of the umbra centre [m]
    Umbra_centre_sol = [(r_c_sol.^2 - ...
        r_a_sol.^2)/(4*r_SPS), ...
        (r_d_sol.^2 - r_b_sol.^2)/(4*r_SPS)];    %[m] [-Z,-Y]
    % CHANGED SIGN WITH RESPECT TO GENERATE ALGO!
    y_0 = Umbra_centre_sol(:,2);                %[m] (VERTICAL)
    z_0 = Umbra_centre_sol(:,1);                %[m] (HORIZONTAL)
end
if sol_flag == 0
    % Assing value only if there is only one solution
    assign_value = 1;
end
end
```


Acronyms

AOCS Attitude and Orbit Control Subsystem

ASPIICS Association of Spacecraft for Polarimetric and Imaging Investigation of the Corona of the Sun

CI Coronagraph Instrument

CLS Coarse Lateral Sensor

CSC Coronagraph Spacecraft

CSL Centre Spatial de Liège

DN Digital Number

DOF Degree Of Freedom

ESA European Space Agency

FF Formation Flying

FLLS Fine Lateral and Longitudinal Sensor

FOV Field Of View

GNC Guidance, Navigation and Control

GNSS Global Navigation Satellite System

GSTM General Support Technology Program

HEO Highly Elliptical Orbit

ISD Inter Satellite Distance

ISL Inter-Satellite Link

LEO Low Earth Orbit

OSC Occulter SpaceCraft

PROBA Project for On-Board Autonomy

PP Pseudo-Paraboloid

PPL Pseudo-Paraboloid + Linear

RAAN Right Ascension of the Ascending Node

SC SpaceCraft

SiPM Silicon PhotoMultiplier

SOC Scientific Operation Centre

SPS Shadow Position Sensors

SRP Solar Radiation Pressure

STD STandard Deviation

TRL Technology Readiness Level