



Rapporti Tecnici INAF INAF Technical Reports

Number	65
Publication Year	2020
Acceptance in OA@INAF	2020-12-14T09:18:02Z
Title	End-to-end numerical simulator of the Shadow Position Sensor (SPS) metrology subsystem of the PROBA-3 ESA mission
Authors	AMADORI, FRANCESCO, CAPOBIANCO, Gerardo, LOREGGIA, Davide, CASTI, MARTA, BEMPORAD, Alessandro, FINESCHI, Silvano, NOCE, VLADIMIRO, ZANGRILLI, Luca, BELLUSO, Massimiliano, PANCRAZZI, Maurizio
Affiliation of first author	O.A. Torino
Handle	http://hdl.handle.net/20.500.12386/28811 , http://dx.doi.org/10.20371/INAF/TechRep/65

End-to-end numerical simulator of the Shadow Position Sensor (SPS) metrology subsystem of the PROBA-3 ESA mission

F. Amadori^{a1}, G. Capobianco¹, D. Loreggia¹, M. Casti², A. Bemporad¹, S. Fineschi¹, V. Noce³, L. Zangrilli¹, M. Belluso⁴, and M. Pancrazzi¹

¹INAF-Turin Astrophysical Observatory, via Osservatorio 20, 10025, Pino Torinese (TO), Italy.

²ALTEC SpA, Corso Marche, 79, 10146 Torino TO, Italy

³INAF-Arcetri Astrophysical Observatory, Largo Enrico Fermi, 5, 50125 Firenze FI

⁴INAF-Catania Astrophysical Observatory, Via S. Sofia, 78, 95123 Catania CT

^ae-mail: francesco.amadori@inaf.it

Abstract

PROBA-3 - *PR*oject for *OnBoard* Autonomy is an ESA mission to be launched in 2022 where a spacecraft is used as an external occulter (OSC-Occulter Spacecraft), to create an artificial solar eclipse as observed by a second spacecraft, the coronagraph (CSC-Coronagraph Spacecraft). The two spacecrafts (SCs) will orbit around the Earth, with an highly elliptic orbit (HEO), with the perigee at 600 Km, the apogee at about 60530 Km and an eccentricity of 0.81. The orbital period is of 19.7 hours and the precise formation flight (within 1 mm) will be maintained for about 6 hours over the apogee, in order to guarantee the observation of the solar corona with the required spatial resolution. The relative alignment of the two spacecrafts is obtained by combining information from several subsystems. One of the most accurate subsystem (with accuracy >0.5 mm) is the Shadow Position Sensors (SPS), composed by eight photomultipliers installed around the entrance pupil of the CSC. The SPS will monitor the penumbra generated by the occulter spacecraft, whose intensity will change according to the relative position of the two satellites. A dedicated algorithm has been developed to retrieve the displacement of the spacecrafts from the measurements of the SPS. Several tests are required in order to evaluate the robustness of the algorithm and its performances/results for different possible configurations. A software simulator has been developed for this purpose. The simulator includes the possibility to generate synthetic 2-D penumbra profile maps or analyze measured profiles and run different versions of the retrieving algorithms, including the “on-board” version. In order to import the “as built” algorithms, the software is coded using Matlab.

Keywords: PROBA-3 – Space Mission – Spacecraft Formation Flying – Numerical Simulator – Solar Corona

1 Introduction

This document describes a new simulation tool developed to operate in a single calculation flow the module for the generation of the penumbra profile [RD01] [RD02], and the SPS metrology algorithm [RD03] [RD04]. The simulation chain has been developed in a common language (MATLAB) to make uniform the structure of the algorithm and to optimize the calculation process. The simulator consists in three main modules:

- The first module is a MATLAB version of the code that generates the penumbra profile and returns the coefficients of the pseudo-paraboloid fit [RD01];
- The second module is a MATLAB version of the code that generates the solar irradiance reaching the eight SPS, estimated from penumbra profile as returned by the first module;
- The third module is the actual MATLAB code for the metrology algorithm that is used to estimate the correction to the X, Y, Z coordinates in order to maintain the formation flying.

In the following sections, we discuss in detail the different parts showing how the GUI interface can be used to realize an e2e simulation: from the penumbra to the XYZ formation coordinates.

2 Scope

The main purpose of this work is to provide the user with a proper tool for testing the complete SPS measurement chain, testing the metrology algorithm robustness and performances, and to verify how the error on the XYZ coordinate returned by the metrology algorithm varies while changing the penumbra fitting parameters. This permits to test the sensitivity of the calculation to these coefficients, and to identify the acceptable variation range where these coefficients can vary in order to have the SPS accuracy requirement satisfied. Moreover, this simulator represents an important verification tool for:

- establishing the acceptable tolerances applicable to the intrinsic uncertainty in knowledge of the real in-flight penumbra profile;
- correlating the results of the SPS calibration test done in lab, with the numerical analysis;
- performing a detailed verification of all the contributors that can affect the SPS measurements (tests).

3 Applicable Documents

The following documents are Project configured deliveries and provide under authorized request.
[AD01] P3-CSL-RS-14001 SPS Requirements Specification v3_1

4 Reference Documents

The following documents are Project configured deliveries and provide under authorized request.
[RD01] P3-INF-TN-14001_Expected penumbra illumination profile_v3.1
[RD02] P3-INF-TN-17022_Algorithms for PROBA 3 ASPIICS F2 Metrology v1.1
[RD03] P3-INF-SW-18001_Encoded SPS algorithm and test_v3.0 Draft_0
[RD04] P3-INF-IF-16000_SPS software ICD_v4.1
[RD05] P3-INF-IF-16000_SPS software ICD_v5.2
[RD06] P3-INF-TN-17024_Implementation and error analysis of the SPS algorithms_v1.1

5 Reference Code

[RC01] P3-INF-SW-18001_Encoded SPS algorithm and test_v3.0 Draft_0 (Algorithm script)

6 Simulator Initialization

When simulator is launched, the GUI interface appears in the following Figure 1.

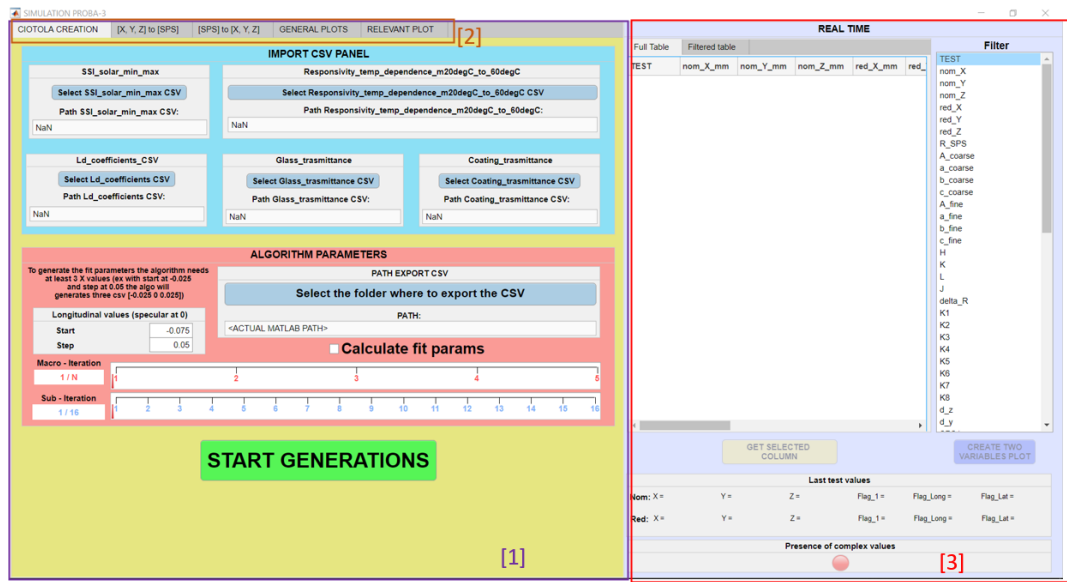


Figure 1. First view of the application when opened.

Section [1] contains the first tab displayed when the application starts. This first tab contains all the data that are required to generate the SPS configuration and that are obtained by an initial position X, Y, Z.

Section [2] shows where is possible to go through the three tabs: PENUMBRA PROFILE CREATION, [XYZ]-TO[SPS],

Section [3] presents the real time control part; in this section all the results obtained by the SPS-TO-XYZ algorithm (§8) will be displayed for real time control and comparison.

7 Penumbra profile generation tab

7.1 Interface description

The interface is divided into several panels:

- IMPORT CSV PANEL [1]:** Contains five sections for selecting CSV files:
 - SSI_solar_min_max:** Select SSI_solar_min_max CSV, Path SSI_solar_min_max CSV: NaN
 - Responsivity_temp_dependence_m20degC_to_60degC:** Select Responsivity_temp_dependence_m20degC_to_60degC CSV, Path Responsivity_temp_dependence_m20degC_to_60degC: NaN
 - Ld_coefficients_CSV:** Select Ld_coefficients CSV, Path Ld_coefficients CSV: NaN
 - Glass_transmittance:** Select Glass_transmittance CSV, Path Glass_transmittance CSV: NaN
 - Coating_transmittance:** Select Coating_transmittance CSV, Path Coating_transmittance CSV: NaN
- ALGORITHM PARAMETERS [2]:**
 - Choose X array range:** Radio buttons for "Specular at 0" (selected) and "Start-Step-End".
 - Specular at 0:** Text explaining the need for at least 3 X values. Input fields for Start: -0.075 and Step: 0.05.
 - Start - Step - End (auto-put 0 in the middle):** Text explaining the steps. Input fields for Start: -0.075, Step: 0.05, and End: 0.075.
- PATH EXPORT CSV [3]:** Select the folder where to export the CSV. PATH: <ACTUAL MATLAB PATH>
- Macro - Iteration [4]:** A progress bar showing 1 / N, with markers 1, 2, 3, 4, 5.
- Sub - Iteration [5]:** A progress bar showing 1 / 16, with markers 1 through 16.
- Calculate fit params [6]:** A checkbox.
- START GENERATIONS:** A large green button.

Figure 2. Detail of PENUMBRA PROFILE CREATION interface.

The panel [1] allows the user to choose some CSV files that are needed by the algorithm to compute and create the PENUMBRA PROFILES. These CSV are:

1. SSI_Solar_min or SSI_Solar_max: solar spectral irradiance at the minimum or maximum solar activity (it depends to the Sun condition the user want to simulate);
2. Responsivity_temp_dependence_m20degC_to_60degC: the SPS responsivity curves as a function of wavelength and temperature;
3. Ld_coefficients: limb darkening coefficients;
4. Glass_transmittance: SPS glass transmittance in function of the wavelength;
5. Coating_transmittance: coating transmittance in function of the wavelength.

The panel [2] allows the user to set the longitudinal range (ISD) at which the penumbra profile is being calculated. At the end of this process, the result will be a vector containing the longitudinal distances (in meters) obtained by selecting one of the two flags:

- i "Specular at 0": the corresponding panel will be enabled. The user must write the highest longitudinal distance value that will be used to create the penumbra profiles, and the interval from this to its specular value opposite to 0. (i.e. if the user writes -0.5 at the "Start" text box and 0.250 at the "Step" text box, the vector will be: [-0.5, -0.25, 0, 0.25, 0.5]);
- ii "Start-Step-End": the corresponding panel will be enabled. The user must write the minimum

and the maximum longitudinal distance and the step at which sampling the relative distance (i.e. if the user writes -0.5 at the “Start” text box, 0.250 at the “Step” text box and 0.75 at the “End” text box, the vector will be: [-0.5, -0.25, 0, 0.25, 0.5 0.75]);

The panel [3] contains the button to choose the folder where all the CSV will be saved.

[4] is the gauge and the label where the actual penumbra profile being created is shown;

[5] is the gauge and the label where the actual sub-iteration in the current calculation is shown;

At [6] the algorithm is started. If the user wants to save the fit parameters in a CSV file, the related checkbox can be flagged.

7.2 Module Description

The input to this module are the text strings with the full path to point at the five CSV files (number [1] in §6.1) and the array containing the desired longitudinal positions (number [2] in §6.1). This module is based on treatment in [RD01] and [RD02].

The output is the penumbra profile itself and a new CSV file containing the fitting parameters of the paraboloid in the requirement box and in the goal box, together with the H, K and L parameters of the longitudinal fit. The penumbra profile is generated for the fixed longitudinal positions (X) taking into account all the terms that affect the SPS measurement as discussed in [RD01].

8 [XYZ]-TO-[SPS] tab

8.1 Interface description

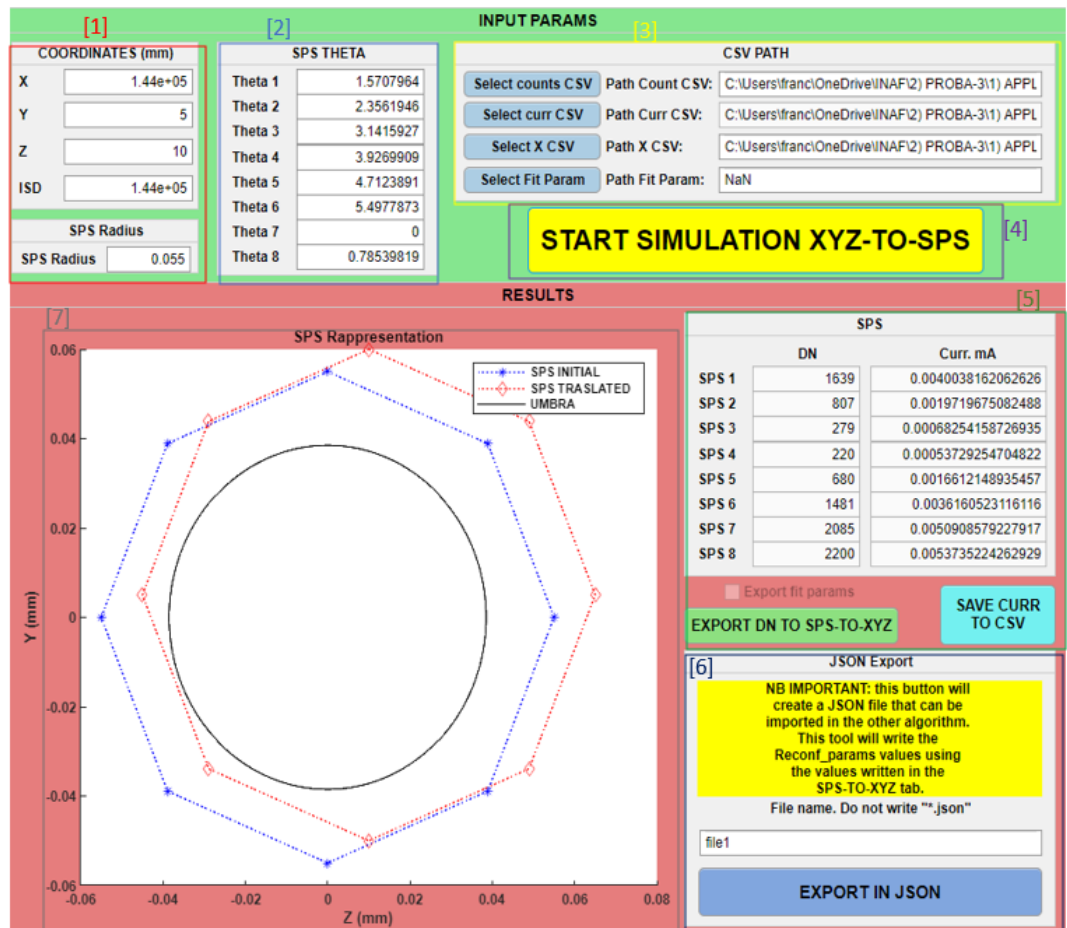


Figure 3. Detail of [XYZ]-TO-[SPS] interface.

Panel [1] lists the coordinates X, Y, Z, the ISD, and SPS radial position for the geometrical configuration being considered in the simulation;

Panel [2] lists the theta angles of each SPS, measured in a counterclockwise way starting from the Z positive axis;

Panel [3] contains four link buttons to open four different CSV files corresponding to the DN counts, the DN currents, the X value (ISD), and the fit parameters that will be used in the algorithm.

At panel [4] the execution of the algorithm can be started.

In panel [5] the DNs and the currents obtained at the end of the execution, are shown. These values (and also X, Y, Z and the fit parameters, if the user wants) can be uploaded in the next calculation module i.e., [SPS]-TO-[XYZ] interface (green button), or exported in a json csv (cyan button).

Panel [6] allows the user to export the full configuration in a json file that can be read by the other modules.

In the graphic area [7], the disposition of the SPS in the reference aligned position (blue color) and in the translated position, corresponding to the reported DNs (red color), is shown.

8.2 Module Description

This module is used to obtain the DNs expected from the eight SPS for a given position X, Y, Z and is based on treatment in [RD01] and [RD02]. It also needs the distance from the SPS to the center of the pupil and the theta angle from the reference horizontal axis: Z positive. The actual SPS disposition is counterclockwise starting from the SPS 1 that is at 90° (1,57 rad) from the reference axis.

Once the user has chosen the initial position, he must select three CSV files:

- The SPS DN of one penumbra profile at a given longitudinal position;
- The SPS currents of one penumbra profile at a given longitudinal position;
- The X array containing data related to longitudinal distances.

These CSV files are created from the previous module (§ 6).

Once the CSV data have been imported, the user presses the “START” to run the XYZ-TO-SPS module. At the end of this run, the program shows the current disposition of the SPS (respect to the reference aligned one) and the new corresponding set of digital readouts are shown.

Once the module run ends, the user can choose to save the results in a json file or to export them to the next calculation module SPS-TO-XYZ (§ 8).

9 [SPS]-TO-[XYZ] tab

9.1 Interface description

The interface is divided into several panels:

- Panel 1 (Raw Data):** Contains a table for SPS irradiance (High Gain, Low Gain) for SPS 1 to SPS 8, and a table for temperature (Nom T1, Nom T2, Red T1, Red T2) and door status (Door, Door_tr).
- Panel 2 (Recon Params):** Contains a table for K coefficients (K1 to K8) and a table for pp_coarse and pp_fine parameters (a_coarse, b_coarse, c_coarse, A_coarse, a_fine, b_fine, c_fine, A_fine, delta_transv, R_sps).
- Panel 3 (XYZ Expected):** Contains input fields for X, Y, and Z expected values.
- Panel 4 (Import params):** Contains an "Import params" button.
- Panel 5 (CYCLE PANEL):** Contains a "Test" field (10), "Only Real Time" checkbox, "Select path CSV" button, "Path CSV" field (<ACTUAL MATLAB PATH>), "Montecarlo" checkbox, and "Percentage" dropdown.
- Panel 6 (START):** Contains a large grid of checkboxes for SPS parameters (SPS1 High/Low, SPS2 High/Low, SPS3 High/Low, SPS4 High/Low, SPS5 High/Low, SPS6 High/Low, SPS7 High/Low, SPS8 High/Low) and a "START" button.

Figure 4. Detail of [SPS]-TO-[XYZ] interface.

In panel [1], the user can write the irradiance of the SPS, the temperature for the nominal and redundant sets, and the FDA/Lid status.

In panel [2], the user can set all the reconfigurable parameters. In the “Algo” panel, the algorithm to be run can be also set to: LINEAR, PSEUDO, BOTH. There is a checkbox which allow the algorithm to self-change the value of the algo param: if the initial value is not LINEAR and the checkbox is checked, there are some conditions that will change the value to LINEAR:

1. $A_coarse > 0$;
2. $A_fine > 0$;
3. The argument of the arccosines is out of range ($arg > 1$ or $arg < -1$).

In panel [3], the user writes the X, Y and Z expected from the calculation.

Pushing button [4] the user can import a full configuration from a json file.

In section [5], the user can modify the rules and options of the tests:

- A** the user must write the number of tests to be executed;
- B** the user must choose the how the algorithm will show the results: in real time (using the right section of the simulator), in CSV or both;

- C the user sets the path for saving the CSV files. The button and the text field will be enabled only if the B part is “Only CSV” or “CSV and real time”;
- D Checkbox named “Montecarlo”: if it is checked, the algorithm will randomly modify the value of the checked parameters (see point [6]) within their range;
- E The user can choose if the variations in range are returned as percentage variations or as absolute variations.

The panel [6] can be used if the user wants to change the value of one (or more) parameter or the DN during the tests: he must click in the checkbox corresponding to the parameters he wants to change and, after that, specifies the range where the parameters can vary.

Pushing the button [7] the algorithm is run. Note that the user can choose to save the relevant data (Nom_X, Nom_Y, Nom_Z, Red_X, Red_Y, Red_Z and the corresponding delta) in a CSV and to plot them in a largest plot (Fig. 6).

9.2 Module Description

The SPS-TO-XYZ algorithm is the MATLAB version of the metrology algorithm still developed by M. Casti at INAF Torino ([RD02], [RD03], [RD04], [RD05], [RD06], code based on [RC01]) and used to calculate the correction to the position X, Y, Z required to maintain aligned the coronagraph spacecraft with respect to the occulter spacecraft, in order to preserve the formation flying. To this end, it needs: the high gain and low gain DN readouts from all the SPS; the temperature and the FDA/Lid status; the full configuration of the reconfigurable parameters.

The major utility of this module comes from the opportunity to vary (by GUI settings) the input parameters to see how these variations will affect the precision and the robustness of the algorithm. Once the user has set the number of tests and the range of variability for each parameter, the simulator calculates the initial value for each variable as:

$$newInitialValue = nominalValue - (nominalValue \frac{range}{numberOfTests}) \quad (1)$$

and the increment being applied to the considered variable after each step (if the user chooses to apply the Montecarlo, the variable will be randomly changed).

For example, in the case we let a_coarse varying:

1. Cycle Number: 100 (so there will be 101 tests, starting from 0);
2. Check a_coarse;
3. Variability range of $\pm 50\%$;
4. Press the start button.

The simulator calculates the “initial value” of a_coarse starting from its nominal value using Eq. (1), as:

$$newItemValue = nominalValue - (nominalValue \frac{range}{100}) = 0.000090688205 - (0.000090688205 \frac{50}{100}) \quad (2)$$

Where $9.0688205 \cdot 10^{-5}$ is the nominal value. The “intermediate value” used to increment the initial value after each completed step is given by:

$$intermediateValue = \frac{nominalValue * range * 0.02}{cycleNumber} = \frac{0.000090688205 * 50 * 0.02}{101} \quad (3)$$

Where 0.02 comes from the need to span the variable range $\pm 50\%$.

After the determination of the “intermediate value” and “initial value”, the simulator will loop the SPS-TO-XYZ for 101 times and will recalculate the `a_coarse` adding the “intermediate value” to the “initial value” at each cycle. At the end of the loop, the results will be displayed and plotted as shown in Fig.5.

10 GENERAL PLOT tab

10.1 Interface description

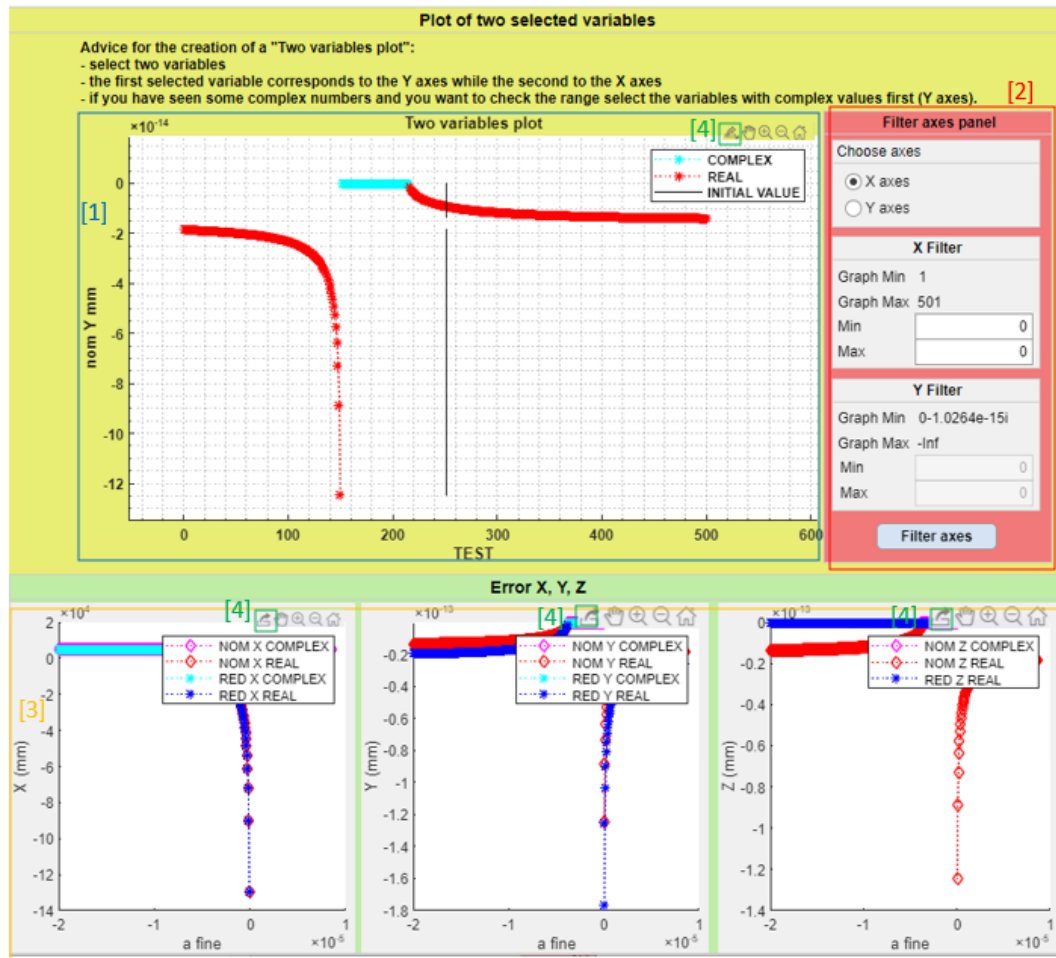


Figure 5. Detail of general 2D plot interface..

The graph [1] is named “Two variable plot” because the user can select two variables from the filter-list in the real time section and plot them in. The instructions on how to create this plot are reported on top of the plot area.

The panel [2] contains the interface to zoom the plot by typing the range for X axis or Y axis.

The three plots [3] show the absolute error in the estimation of the X, Y, Z with respect to the original position (X, Y, Z), returned by each test iteration.

Each plot in [4] can be saved by clicking the correct icon on the top-right.

11 RELEVANT PLOT tab

11.1 Interface description

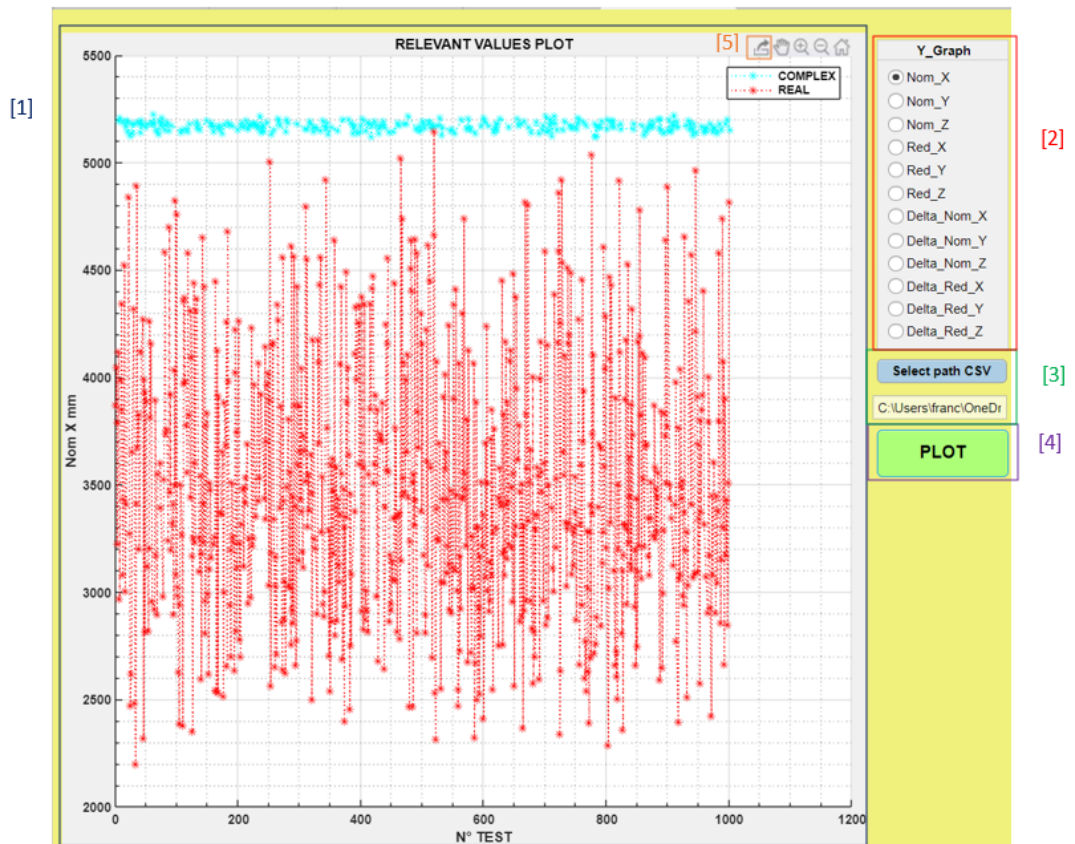


Figure 6. Detail of relevant plot interface.

[1] is the graphic area for plotting selected variables;

The combo box [2] allows the user to choose the data to be plotted in [1] (in the abscissa axis the number of test iteration is reported).

The button [3] allows the user to choose the CSV ([7] in Figure 4) where loading the relevant data from.

The button [4] plots the data.

The plot in [5] shows the variation of the error associated to the coordinate selected in [2] at each simulation step and it can be saved by clicking the correct icon on the top-right.

12 Real TIME section

12.1 Interface description

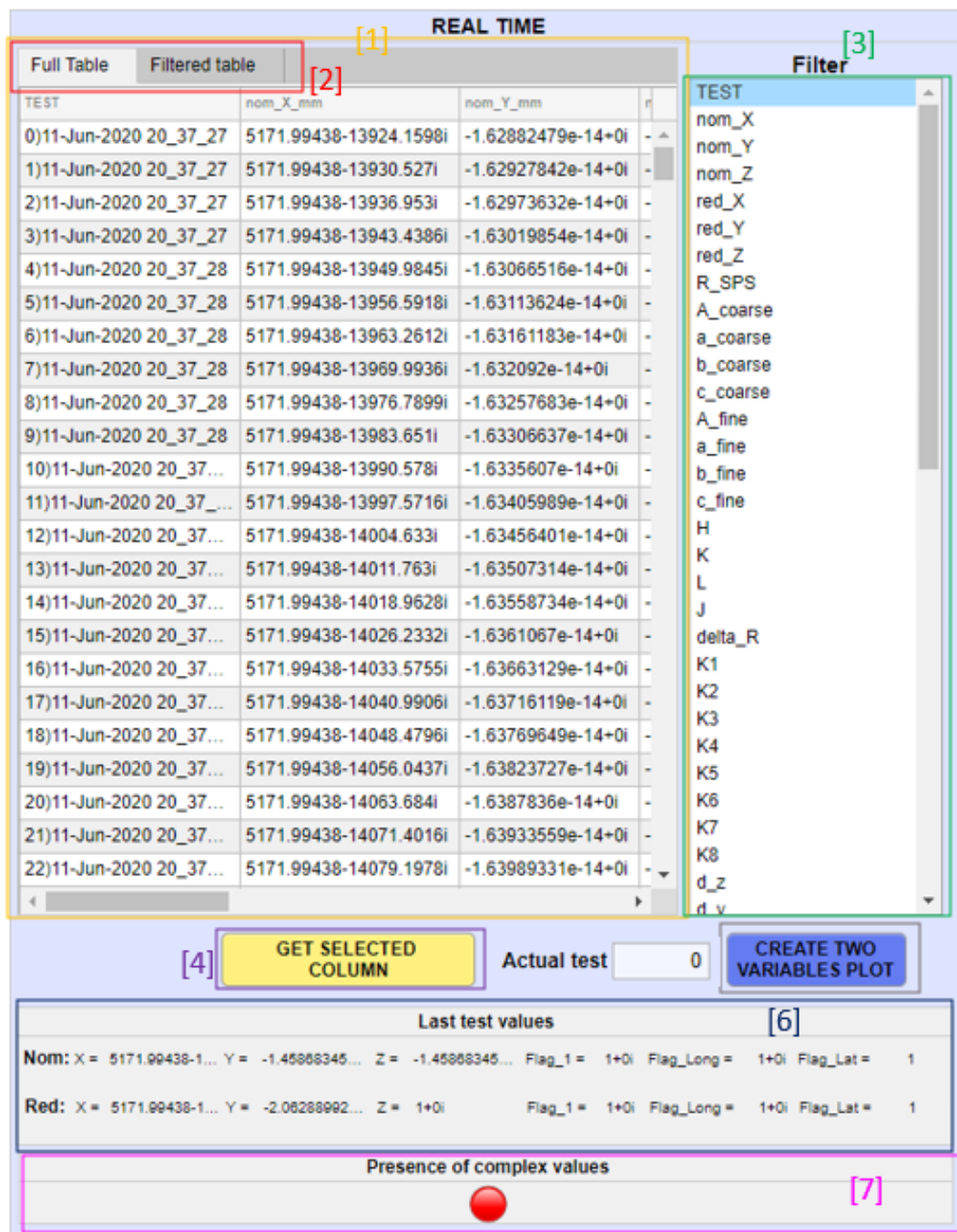


Figure 7. Detail of real time section.

The section [1] contains 2 tabs:

- Full table listing all the relevant variables from each test (one for row);
- Filtered table listing only the variables chosen in the filtered list ([3]), once pressed the button GET SELECTED COLUMN ([4]).

The two tabs [2] permits the user to go through the two tables.

The list [3] contains all the variable names and it is used to create a Filtered Table and a Two Variables Plot.

The button [4] is used to create a filtered table.

The button [5] is used to create a Two Variables Plot.

The panel [6] contains the most relevant values of the last test.

The panel [7] contains a warning light that turns on only if one test returns complex number between its results.

13 Example of JSON file for the SPS-to-XYZ parameter import

```
{
  "K" : [ 1, 1, 1, 1, 1, 1, 1, 1, 1],
  "algo" : 0,
  "pp_coarse" :
  {
    "a_coarse" : 9.0688205E-5,
    "b_coarse" : 0.0010232291,
    "c_coarse" : -0.008504024,
    "A_coarse" : -0.062319517219178
  },
  "pp_fine":
  {
    "a_fine" : -5.7192131E-6,
    "b_fine" : 0.00053501845,
    "c_fine" : -0.0042708996,
    "A_fine" : -0.007240941134938,
    "delta_transv" : 0
  },
  "linear":
  {
    "d_z" : 1.858E+05,
    "d_y" : 1.858E+05
  },
  "long" :
  {
    "H" : 5.2304673E-005,
    "K" : -0.54103895,
    "L" : 1034,
    "J" : 0,
    "delta_R" : 1.1845e+03
  },
  "thresholds":
  {
    "V_max_lat" : 5,
    "V_max_long" : 50,
    "I_max" : 75,
    "delta_flag" : 13.8
  },
  "raw_data":
  {
    "Read": [1015, 203, 1017, 203, 1020, 204, 1017, 203, 1025, 205, 1017, 203, 1010, 202, 1017, 203],
    "Temp": [308, 308, 308, 308],
    "door" : 0,
    "door_tr" : 0
  },
  "Expected_X_Y_Z":
  {
    "X":0,
    "Y":0,
    "Z":0
  }
}
```

Figure 8. Example of a JSON file.

14 Example of a CSV file and description

14.1 MATLAB representation

Function	Nan1	SPS 1 Res.	SPS 1 Res.	SPS 2 Res.	SPS 2 Res.	SPS 3 Res.	SPS 3 Res.	SPS 4 Res.	SPS 4 Res.	SPS 5 Res.	SPS 5 Res.	SPS 6 Res.	SPS 6 Res.	SPS 7 Res.	SPS 7 Res.
Text	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
Function	Nan1	SPS 1 Read.	SPS 1 Read.	SPS 2 Read.	SPS 2 Read.	SPS 3 Read.	SPS 3 Read.	SPS 4 Read.	SPS 4 Read.	SPS 5 Read.	SPS 5 Read.	SPS 6 Read.	SPS 6 Read.	SPS 7 Read.	SPS 7 Read.
Data_acqui...		1017	203	1017	203	1017	203	1017	203	1017	203	1017	203	1017	203
Pre_proces...		1017	203	1017	203	1017	203	1017	203	1017	203	1017	203	1017	203
First_Validi...		1017	203	1017	203	1017	203	1017	203	1017	203	1017	203	1017	203
Relative_po...		1017	203	1017	203	1017	203	1017	203	1017	203	1017	203	1017	203
Second_Val...		1017	203	1017	203	1017	203	1017	203	1017	203	1017	203	1017	203
Output		1017	203	1017	203	1017	203	1017	203	1017	203	1017	203	1017	203
r_sps = 0.055															
NOM X = 3...		NOM Y = -...													

Figure 9. Matlab representation of CSV.

14.2 Excel representation

First View

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Function	Nan1	SPS 1 Reading High Gain	SPS 1 Reading Low Gain	SPS 2 Reading High Gain	SPS 2 Reading Low Gain	SPS 3 Reading High Gain	SPS 3 Reading Low Gain	SPS 4 Reading High Gain	SPS 4 Reading Low Gain	SPS 5 Reading High Gain	SPS 5 Reading Low Gain	SPS 6 Reading High Gain	SPS 6 Reading Low Gain	SPS 7 Reading High Gain	SPS 7 Reading Low Gain
2	Data_acquisition		1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203
3	Pre_processing		1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203
4	First Validity Check		1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203
5	Relative_position Computation		1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203
6	Second Validity Check		1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203
7	Output		1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203	1017,203
8																
9	r_sps = 0.055		A_coarse = -0.06232	a_coarse = 9.0688e-05	b_coarse = 0.0010232	c_coarse = -0.008504	A_fine = 0.14197	a_fine = -5.7192e-06	b_fine = 0.00053502	c_fine = 0.00053502						
10	NOM X = 5171.9944	4406.80911	NOM Y = 0-3.9960773e-14	NOM Z = 0-3.9960773e-14	RED X = 5171.9944	4406.80911	RED Y = 0-5.6513068e-14	RED Z = 0-5.6513068e-14	Final NOM							
11																

Figure 10. Excel representation after the first open.

After the right configuration

	A	B	C	D	E	F	G	H
1	Function	Nan1	SPS 1 Reading High Gain	SPS 1 Reading Low Gain	SPS 2 Reading High Gain	SPS 2 Reading Low Gain	SPS 3 Reading High Gain	SPS 3 Reading Low Gain
2	Data_acquisition			1017	203	1017	203	1017
3	Pre_processing			1017	203	1017	203	1017
4	First Validity Check			1017	203	1017	203	1017
5	Relative_position Computation			1017	203	1017	203	1017
6	Second Validity Check			1017	203	1017	203	1017
7	Output			1017	203	1017	203	1017
8								
9	r_sps = 0.055		A_coarse = -0.06232		a_coarse = 9.0688e-05		b_coarse = 0.0010232	
10	NOM X = 5171.9944	4427.6097	NOM Y = 0-3.9293862e-14		NOM Z = 0-3.9293862e-14		RED X = 5171.9944	4427.6097

Figure 11. Excel representation after the right configuration.

The steps to view the CSV files in the right configuration are: 1. Select the column A by clicking on its index.

1. Select the column A by clicking on its index.

	A	B	C	D	E	F
1	Function,	Nan1,SP5_1_Reading_High_Gain,SP5_1_Reading_Low_Gain,SP				
2	Data_acquisition,	,1017,203,1017,203,1017,203,1017,203,1017				
3	Pre_processing,	,1017,203,1017,203,1017,203,1017,203,1017,				
4	First Validity Check,	,1017,203,1017,203,1017,203,1017,203,				
5	Relative_position_Computation,	,1017,203,1017,203,1017,203,				
6	Second Validity Check,	,1017,203,1017,203,1017,203,1017,2				
7	Output,	,1017,203,1017,203,1017,203,1017,203,1017,				
8						
9		r_sps = 0.055,,A_coarse = -0.06232,,a_coarse = 9.0688e-05,,b_co				
10		NOM_X = 0.171.9944-4406.80911,,NOM_Y = 0-3.9960773e-14,,N				
11						
12						

Figure 12. First step to set the right configuration to a CSV file in Excel.

2. Click on the Data tab on the top of the application.



Figure 13. Second step to set the right configuration to a CSV file in Excel.

3. Select “Text in columns”.

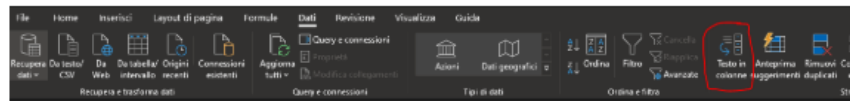


Figure 14. Third step to set the right configuration to a CSV file in Excel.

After this, a Popup will appear.

4. Click on the first circle.

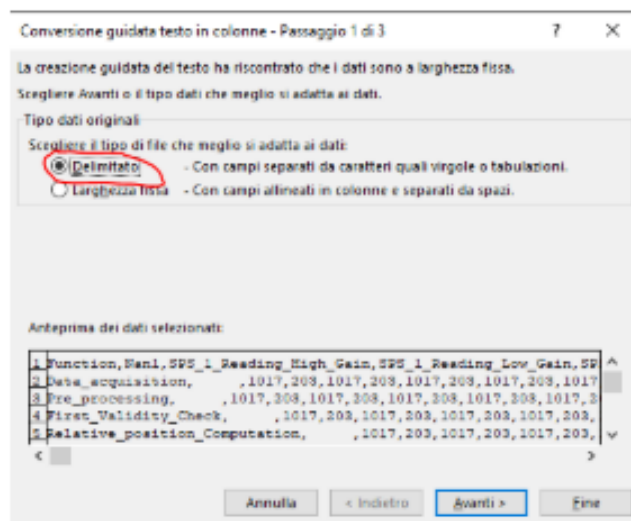


Figure 15. Fourth step to set the right configuration to a CSV file in Excel.

Click “Next” to choose the delimiter.

5. Select only the third checkbox named “Virgola” (Comma in English).

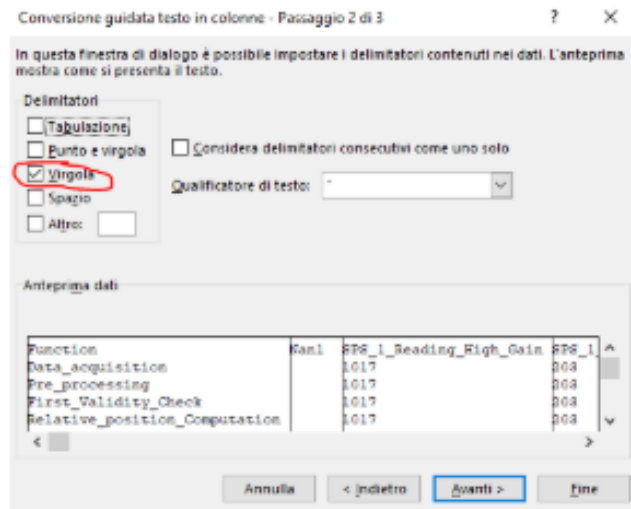


Figure 16. Fifth step to set the right configuration to a CSV file in Excel.

14.3 Gedit representation

```
Function,Nan1,SPS_1_Reading_High_Gain,SPS_1_Reading_Low_Gain,SPS_2_Reading_High_Gain,SPS_2_Reading_Low_Gain
Data_acquisition, ,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203, ,308,308
Pre_processing, ,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203, ,308,308
First_Velocity_Check, ,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203, ,308,308
Relative_position_Computation, ,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203, ,308,308
+0i,0+0i,5171.9944-4520.5762i,0-3.6674804e-14i,0-3.6674804e-14i, ,0+0i,0+0i,5171.9944-4520.5762i,0-5.18
Second_Velocity_Check, ,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203, ,308,308
+0i,0+0i,5171.9944-4520.5762i,0-3.6674804e-14i,0-3.6674804e-14i, ,0+0i,0+0i,5171.9944-4520.5762i,0-5.18
Output, ,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203,1017,203, ,308,308,308,308
+0i,0+0i,5171.9944-4520.5762i,0-3.6674804e-14i,0-3.6674804e-14i, ,0+0i,0+0i,5171.9944-4520.5762i,0-3.18
+0i, ,0,0,0,0,0,5171.9944-4520.5762i,0,0,5171.9944-4520.5762i, ,0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r_sps = 0.055, a_coarse = -0.06232, a_coarse = 0.06088-05, b_coarse = 0.0010232, c_coarse = -0.009504,
l_1, R6 = 1, R7 = 1, R8 = 1, d_x = 185800, d_y = 185900, SPS1 = 1017, SPS2 = 1017, SPS3 = 1017, SPS4 =
0
NOM X = 5171.9944-4520.5762i, NOM Y = 0-3.6674804e-14i, NOM Z = 0-3.6674804e-14i, RED X = 5171.9944-4520
+0i, Final_RED_delta_26 = 0+0i, Final_RED_delta_48 = 0+0i, Final_RED_Flag_Long = 1+0i, Expected_X = 0,
5171.9944-4520.5762i,
```

Figure 17. Gedit representation of a CSV.

14.4 Description of a CSV file

Each CSV contains ten rows:

First row	all the columns name
Second row	all the data once uploaded by the simulator
Third row	all the data updated after the Pre-Processing function of the SPS-TO-XYZ algorithm
Fourth row	all the data updated after the First-Validity-Check function of the SPS-TO-XYZ algorithm
Fifth row	all the data updated after the Pre-Processing function of the SPS-TO-XYZ algorithm
Sixth row	contains all the data updated after the Relative-position-Computation (SPSAlgorithm) function of the SPS-TO-XYZ algorithm
Seventh row	all the data updated after the Second-Validity-Check function of the SPS-TO-XYZ algorithm
Eighth row	all the data updated at the end of the test
Ninth row	the most relevant parameters excluding the information about X, Y, Z
Tenth row	all the information about X, Y, Z

15 Possible exception during the execution of the simulator

This issue can occur when Java runs out of heap memory when running the “Penumbra profile generation” algorithm. Please increase the heap space to try avoiding such crashes. To increase the heap space:

Adjust the Java heap size:

1. On the Home tab, in the Environment section, click Preferences. Select MATLAB > General > Java Heap Memory;
2. Select a Java heap size value using the slider or spin box; NB: Increasing the Java heap size decreases the amount of memory available for storing data in arrays;
3. Click OK;
4. Restart MATLAB.

If the maximum value, allowed by this slider, is too small to satisfy running requirements, it could be directly adjusted by editing:

"C:\Users\%USERNAME%\AppData\Roaming\MathWorks\MATLAB\%matlabversion%\matlab.prf"

Is possible to get this path using: » **prefdir**.

1. Open this file with a text editor.
2. Modify "JavaMemHeapMax" value. It should look like JavaMemHeapMax=1128000.
3. Restart MATLAB for changes to reflect.

The Preferences UI will display a warning in Java Heap Memory, that it is larger than what recommended.

16 Penumbra profile generation algorithm: Notes

16.1 15.1 Constant variables

Variable	Value	Description
dpin_mm	2.5d0	Pinhole Diameter (mm)
dn_stad1	0.014532180059524	microAmpere/DN
dn_stad2	0.002906436011905	microAmpere/DN
sps_temp	2	NB: 0 for -20°C, 1 for 0°C, 2 for +20°C, 3 for 40°C and 4 for 60°C
au	149.597871d6	1 AU (km)
r_sun	6.96342d5	1 Rsun (km)
r_occ	0.71d0	Occulter radius (m)
r_pup	0.025d0	Pupil radius (m)
r_sps	0.055d0	SPS radius disposition (m)
t_occ	0.d0	Occulter inclination over the optical axis of the corona-graph (grad, tilt max 0.5)
dx_min_occ	50.d-6	MIN delta x occulter (m, max 0.01)
max_distance	55.d0 + 100.d0	Extension of the 2D map (this is the distance from optical center along the x or y axis) (mm)
pixres	500.d-6 * 1.d3	Pixel resolution (N micron in mm)
fit_range_goal = find(x_arr .* 1.e3 > 40. & x_arr .* 1.e3 < 55. + 50. & proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));	40.	Lower limit of the goal box
fit_range_goal = find(x_arr .* 1.e3 > 40. & x_arr .* 1.e3 < 55. + 50. & proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));	55- + 50.	Upper limit of the goal box
fit_range_req = find(x_arr .* 1.e3 > 45. & x_arr .* 1.e3 < 65. & proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));	45.	Lower limit of the requirement box
fit_range_req = find(x_arr .* 1.e3 > 45. & x_arr .* 1.e3 < 65. & proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));	65.	Upper limit of the requirement box

16.2 J parameter of longitudinal equation

The J parameter was created to assign the right value to the calculus of the X position, to balance the “deltaR” parameter that was calculated using the fitting parameters in the goal box. The “deltaR” parameter is obtained using the coarse parameter by the formula:

$$\text{deltaR} = \left(\frac{\rho}{a_{coarse}}\right) + \left(\frac{\rho}{b_{coarse}}\right)^2 + \left(\frac{\rho}{c_{coarse}}\right)^3 | \rho = 0.055m \quad (4)$$

The formula for R_{55} is:

$$R_{55} = R_0 - \text{deltaR} \quad (5)$$

And the formula for the X (longitudinal) position is:

$$X = \frac{K - \sqrt{K^2 - 4H(L - R_{55})}}{2H} + (J * d_0^2) \quad (6)$$

J is set in order to give the second addend equal to zero because (in these tests) the “deltaR” parameter is calculated using the fine parameter, so d_0 is no useful.

The formula of “deltaR” obtained using the fine fit parameter is:

$$\text{deltaR} = \left(\frac{\rho}{a_{fine}}\right) + \left(\frac{\rho}{b_{fine}}\right)^2 + \left(\frac{\rho}{c_{fine}}\right)^3 | \rho = 0.055m \quad (7)$$

17 ANNEX 1: Simulator flowchart

17.1 Complete flowchart



Figure 18. Complete Flowchart generated by MATLAB.

17.2 SPS-TO-XYZ Flowchart

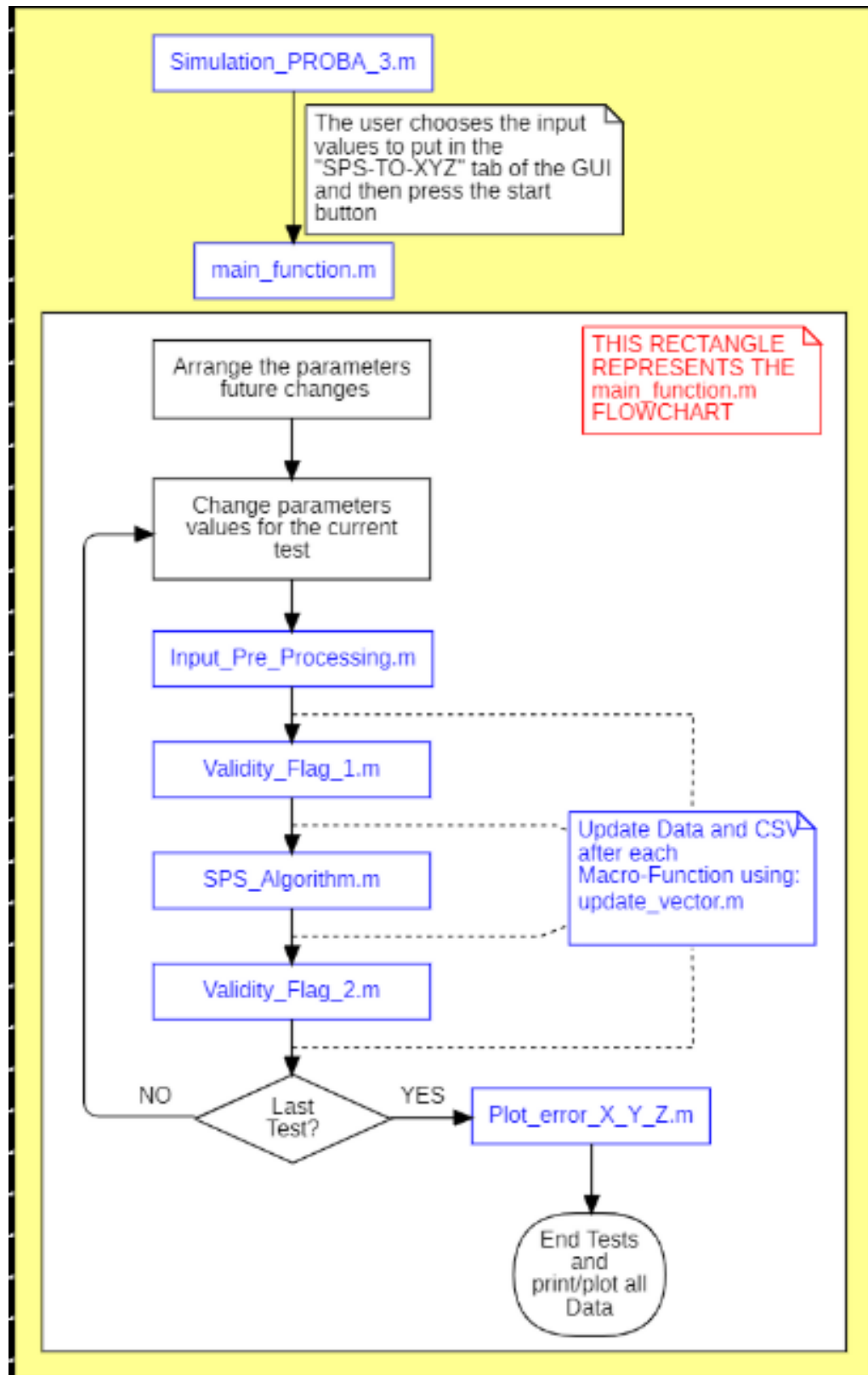


Figure 19. SPS-TO-XYZ Flowchart.

17.3 XYZ-TO-SPS flowchart

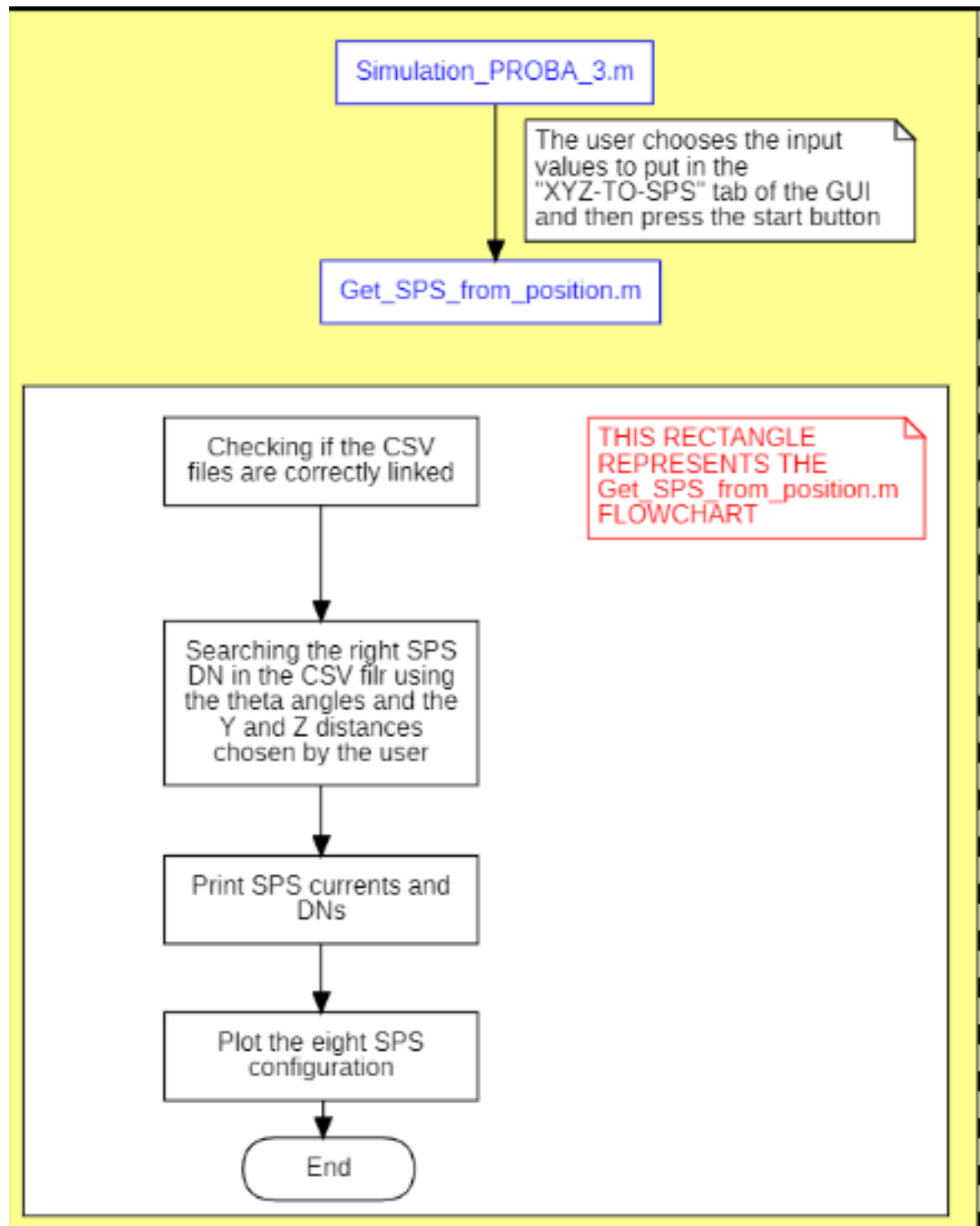


Figure 20. XYZ-TO-SPS flowchart.

17.4 Penumbra profile generation flowchart

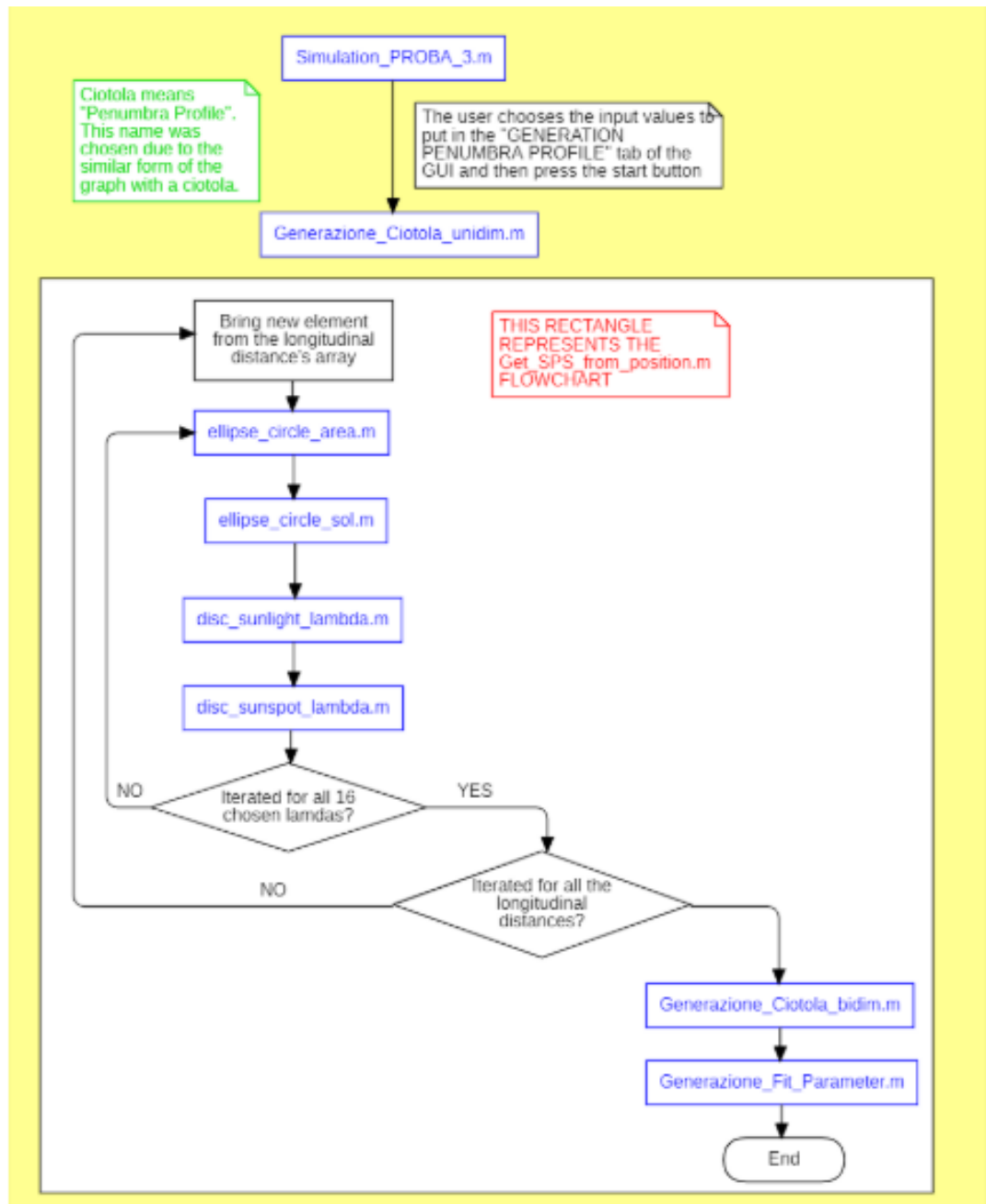


Figure 21. Penumbra profile generation flowchart.

18 ANNEX 2: Simulator code
PAGE INTENTIONALLY LEFT BLANK

```

classdef SIMULATION_PROBA_3_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        SIMULATIONPROBA3UIFigure          matlab.ui.Figure
        REALTIMEPanel                      matlab.ui.container.Panel
        FilterListBoxLabel                 matlab.ui.control.Label
        FilterListBox                      matlab.ui.control.ListBox
        GETSELECTEDCOLUMNButton           matlab.ui.control.StateButton
        TabGroup                           matlab.ui.container.TabGroup
        Tab_FullTable                     matlab.ui.container.Tab
        Full_Table                         matlab.ui.control.Table
        FilteredtableTab                  matlab.ui.container.Tab
        Filtered_Table                    matlab.ui.control.Table
        CREATETWOVARIABLESPLOTButton      matlab.ui.control.StateButton
        LasttestvaluesPanel               matlab.ui.container.Panel
        NomLabel                           matlab.ui.control.Label
        Label                              matlab.ui.control.Label
        XLabel_2                          matlab.ui.control.Label
        Label_nom_x                       matlab.ui.control.Label
        YLabel_2                          matlab.ui.control.Label
        Label_nom_y                       matlab.ui.control.Label
        ZLabel_2                          matlab.ui.control.Label
        Label_nom_z                       matlab.ui.control.Label
        Flag_1Label                       matlab.ui.control.Label
        Label_Flag_1_nom                  matlab.ui.control.Label
        Flag_LongLabel                    matlab.ui.control.Label
        Label_Flag_Long_nom               matlab.ui.control.Label
        Flag_LatLabel                     matlab.ui.control.Label
        Label_Flag_lat_nom                matlab.ui.control.Label
        RedLabel                           matlab.ui.control.Label
        XLabel_3                          matlab.ui.control.Label
        Label_red_x                       matlab.ui.control.Label
        YLabel_3                          matlab.ui.control.Label
        Label_red_y                       matlab.ui.control.Label
        ZLabel_3                          matlab.ui.control.Label
        Label_red_z                       matlab.ui.control.Label
        Flag_1Label_2                     matlab.ui.control.Label
        Label_Flag_1_red                  matlab.ui.control.Label
        Flag_LongLabel_2                  matlab.ui.control.Label
        Label_Flag_Long_red               matlab.ui.control.Label
        Flag_LatLabel_2                   matlab.ui.control.Label
        Label_Flag_lat_red                 matlab.ui.control.Label
        PresenceofcomplexvaluesPanel      matlab.ui.container.Panel
        Lamp                              matlab.ui.control.Lamp
        ActualtestEditFieldLabel           matlab.ui.control.Label
        ActualtestEditField                matlab.ui.control.NumericEditField
        TabGroup2                          matlab.ui.container.TabGroup
        PENUMBRAPROFILECREATIONTab        matlab.ui.container.Tab
        STARTGENERATIONSButton             matlab.ui.control.Button
        IMPORTCSVPANELPanel                matlab.ui.container.Panel
        SSI_solar_min_maxPanel             matlab.ui.container.Panel
        SSI_solarmin_CSV                   matlab.ui.control.EditField
        SelectSSI_solar_min_maxCSVButton   matlab.ui.control.Button
        PathSSI_solar_min_maxCSVLabel      matlab.ui.control.Label
    end

```

```

Responsivity_temp_dependence_m20degC_to_60degCPanel matlab.ui.container.Panel
PathResponsivity_temp_dependence_m20degC_to_60degCLabel matlab.ui.control.↙
Label
resp_t_dep_m20degC_to_60degCCSV matlab.ui.control.EditField
Select_resp_t_dep_m20degC_to_60degCCSVButton matlab.ui.control.Button
Ld_coefficients_CSVPanel matlab.ui.container.Panel
Ld_coefficients_CSV matlab.ui.control.EditField
SelectLd_coefficientsCSVButton matlab.ui.control.Button
PathLd_coefficientsCSVLabel matlab.ui.control.Label
Glass_transmittancePanel matlab.ui.container.Panel
Glass_transmittance_CSV matlab.ui.control.EditField
SelectGlass_transmittanceCSVButton matlab.ui.control.Button
PathGlass_transmittanceCSVLabel matlab.ui.control.Label
Coating_transmittancePanel matlab.ui.container.Panel
Coating_transmittance_CSV matlab.ui.control.EditField
SelectCoating_transmittanceCSVButton matlab.ui.control.Button
PathCoating_transmittanceCSVLabel matlab.ui.control.Label
ALGORITHMPARAMETERSPanel matlab.ui.container.Panel
PATHEXPORTCSVPanel matlab.ui.container.Panel
EXport_CSV_Ciotola matlab.ui.control.EditField
SelectthefolderwheretoeexporttheCSVButton matlab.ui.control.Button
PATHLabel matlab.ui.control.Label
Iteration matlab.ui.control.Label
Gauge matlab.ui.control.LinearGauge
MacroIterationLabel matlab.ui.control.Label
Iteration_2 matlab.ui.control.Label
Gauge_2 matlab.ui.control.LinearGauge
SubIterationLabel matlab.ui.control.Label
LongitudinalRangePanel matlab.ui.container.Panel
Specularat0meterPanel matlab.ui.container.Panel
StartEditField matlab.ui.control.NumericEditField
StepmeterEditFieldLabel matlab.ui.control.Label
StepmeterEditField matlab.ui.control.NumericEditField
StartEditFieldLabel matlab.ui.control.Label
Label_3 matlab.ui.control.Label
ChangeXarrayrangeButtonGroup matlab.ui.container.ButtonGroup
Specularat0Button matlab.ui.control.RadioButton
StartStepEndButton matlab.ui.control.RadioButton
StartStepEndmeterPanel matlab.ui.container.Panel
StartEditField_2 matlab.ui.control.NumericEditField
StepmeterEditFieldLabel_2 matlab.ui.control.Label
StepmeterEditField_2 matlab.ui.control.NumericEditField
StartEditFieldLabel_2 matlab.ui.control.Label
End_Edit_Field matlab.ui.control.NumericEditField
EndEditField matlab.ui.control.Label
Label_4 matlab.ui.control.Label
CalculatefitparamsCheckBox matlab.ui.control.CheckBox
XYZtoSPSTab matlab.ui.container.Tab
RESULTSPanel matlab.ui.container.Panel
PLOT_SPS matlab.ui.control.UIAxes
SPSPanel_3 matlab.ui.container.Panel
SPS1EditFieldLabel matlab.ui.control.Label
SPS1EditField matlab.ui.control.NumericEditField
SPS2EditFieldLabel matlab.ui.control.Label
SPS2EditField matlab.ui.control.NumericEditField

```

SPS3EditFieldLabel	matlab.ui.control.Label
SPS3EditField	matlab.ui.control.NumericEditField
SPS4EditFieldLabel	matlab.ui.control.Label
SPS4EditField	matlab.ui.control.NumericEditField
SPS5EditFieldLabel	matlab.ui.control.Label
SPS5EditField	matlab.ui.control.NumericEditField
SPS6EditFieldLabel	matlab.ui.control.Label
SPS6EditField	matlab.ui.control.NumericEditField
SPS7EditFieldLabel	matlab.ui.control.Label
SPS7EditField	matlab.ui.control.NumericEditField
SPS8EditFieldLabel	matlab.ui.control.Label
SPS8EditField	matlab.ui.control.NumericEditField
SPS1_Curr_EditField	matlab.ui.control.NumericEditField
SPS2_Curr_EditField	matlab.ui.control.NumericEditField
SPS3_Curr_EditField	matlab.ui.control.NumericEditField
SPS4_Curr_EditField	matlab.ui.control.NumericEditField
SPS5_Curr_EditField	matlab.ui.control.NumericEditField
SPS6_Curr_EditField	matlab.ui.control.NumericEditField
SPS7_Curr_EditField	matlab.ui.control.NumericEditField
SPS8_Curr_EditField	matlab.ui.control.NumericEditField
DNLabel	matlab.ui.control.Label
CurrmALabel	matlab.ui.control.Label
EXPORTDNTOSPSTOXYZButton	matlab.ui.control.Button
JSONExportPanel	matlab.ui.container.Panel
EXPORTINJSONButton	matlab.ui.control.Button
Json_filename	matlab.ui.control.EditField
FilenameDonotwritejsonLabel	matlab.ui.control.Label
Label_2	matlab.ui.control.Label
SAVECURRTOCSVButton	matlab.ui.control.Button
ExportfitparamsCheckBox	matlab.ui.control.CheckBox
INPUTPARAMSPanel	matlab.ui.container.Panel
COORDINATESmmPanel	matlab.ui.container.Panel
XEditFieldLabel	matlab.ui.control.Label
XEditField	matlab.ui.control.NumericEditField
ZEditFieldLabel	matlab.ui.control.Label
ZEditField	matlab.ui.control.NumericEditField
YEditFieldLabel	matlab.ui.control.Label
YEditField	matlab.ui.control.NumericEditField
ISDEditFieldLabel	matlab.ui.control.Label
ISDEditField	matlab.ui.control.NumericEditField
SPSTHETAPanel	matlab.ui.container.Panel
Theta1EditFieldLabel	matlab.ui.control.Label
Theta1EditField	matlab.ui.control.NumericEditField
Theta2Label	matlab.ui.control.Label
Theta2EditField	matlab.ui.control.NumericEditField
Theta3EditFieldLabel	matlab.ui.control.Label
Theta3EditField	matlab.ui.control.NumericEditField
Theta4EditFieldLabel	matlab.ui.control.Label
Theta4EditField	matlab.ui.control.NumericEditField
Theta5EditFieldLabel	matlab.ui.control.Label
Theta5EditField	matlab.ui.control.NumericEditField
Theta6EditFieldLabel	matlab.ui.control.Label
Theta6EditField	matlab.ui.control.NumericEditField
Theta7EditFieldLabel	matlab.ui.control.Label
Theta7EditField	matlab.ui.control.NumericEditField

Theta8EditFieldLabel	matlab.ui.control.Label
Theta8EditField	matlab.ui.control.NumericEditField
STARTSIMULATIONXYZTOSPSButton	matlab.ui.control.Button
CSVPATHPanel	matlab.ui.container.Panel
CSV_Count	matlab.ui.control.EditField
SelectcountsCSVButton	matlab.ui.control.Button
PathCountCSVLabel	matlab.ui.control.Label
CSV_Curr	matlab.ui.control.EditField
SelectcurrCSVButton	matlab.ui.control.Button
PathCurrCSVLabel	matlab.ui.control.Label
CSV_X	matlab.ui.control.EditField
SelectZCSVButton	matlab.ui.control.Button
PathZCSVLabel	matlab.ui.control.Label
CSV_fit_param	matlab.ui.control.EditField
SelectFitParamButton	matlab.ui.control.Button
PathFitParamLabel	matlab.ui.control.Label
SPSRadiusPanel	matlab.ui.container.Panel
SPSRadiusEditField	matlab.ui.control.NumericEditField
SPSRadiusEditFieldLabel	matlab.ui.control.Label
SPStoXYZTab	matlab.ui.container.Tab
SETPARAMVALUESPanel	matlab.ui.container.Panel
RawDataPanel	matlab.ui.container.Panel
SPSPanel	matlab.ui.container.Panel
HighGainLabel	matlab.ui.control.Label
LowGainLabel	matlab.ui.control.Label
SPS1Label	matlab.ui.control.Label
EditField_SPS1_high	matlab.ui.control.NumericEditField
EditField_SPS1_low	matlab.ui.control.NumericEditField
SPS2Label	matlab.ui.control.Label
EditField_SPS2_high	matlab.ui.control.NumericEditField
EditField_SPS2_low	matlab.ui.control.NumericEditField
SPS3Label	matlab.ui.control.Label
EditField_SPS3_high	matlab.ui.control.NumericEditField
EditField_SPS3_low	matlab.ui.control.NumericEditField
SPS4Label	matlab.ui.control.Label
EditField_SPS4_high	matlab.ui.control.NumericEditField
EditField_SPS4_low	matlab.ui.control.NumericEditField
SPS5Label	matlab.ui.control.Label
EditField_SPS5_high	matlab.ui.control.NumericEditField
EditField_SPS5_low	matlab.ui.control.NumericEditField
SPS6Label	matlab.ui.control.Label
EditField_SPS6_high	matlab.ui.control.NumericEditField
EditField_SPS6_low	matlab.ui.control.NumericEditField
SPS7Label	matlab.ui.control.Label
EditField_SPS7_high	matlab.ui.control.NumericEditField
EditField_SPS7_low	matlab.ui.control.NumericEditField
SPS8Label	matlab.ui.control.Label
EditField_SPS8_high	matlab.ui.control.NumericEditField
EditField_SPS8_low	matlab.ui.control.NumericEditField
TempPanel	matlab.ui.container.Panel
NomT1Label	matlab.ui.control.Label
EditField_nom_T1	matlab.ui.control.NumericEditField
NomT2Label	matlab.ui.control.Label
EditField_Nom_T2	matlab.ui.control.NumericEditField
RedT1Label	matlab.ui.control.Label

EditField_Red_T1	matlab.ui.control.NumericEditField
RedT2Label	matlab.ui.control.Label
EditField_Red_T2	matlab.ui.control.NumericEditField
DoorPanel	matlab.ui.container.Panel
DoorLabel	matlab.ui.control.Label
Door_trLabel	matlab.ui.control.Label
CheckBox_Door_tr	matlab.ui.control.CheckBox
CheckBox_Door	matlab.ui.control.CheckBox
ReconParamsPanel	matlab.ui.container.Panel
KcoeffPanel_2	matlab.ui.container.Panel
K1Label	matlab.ui.control.Label
K2Label	matlab.ui.control.Label
K3Label	matlab.ui.control.Label
K4Label	matlab.ui.control.Label
K5Label	matlab.ui.control.Label
K6Label	matlab.ui.control.Label
K7Label	matlab.ui.control.Label
K8Label	matlab.ui.control.Label
EditField_K1	matlab.ui.control.NumericEditField
EditField_K2	matlab.ui.control.NumericEditField
EditField_K3	matlab.ui.control.NumericEditField
EditField_K4	matlab.ui.control.NumericEditField
EditField_K5	matlab.ui.control.NumericEditField
EditField_K6	matlab.ui.control.NumericEditField
EditField_K7	matlab.ui.control.NumericEditField
EditField_K8	matlab.ui.control.NumericEditField
pp_coarsePanel	matlab.ui.container.Panel
EditField_a_coarse_value	matlab.ui.control.NumericEditField
a_coarseLabel_2	matlab.ui.control.Label
EditField_b_coarse_value	matlab.ui.control.NumericEditField
b_coarseLabel_2	matlab.ui.control.Label
EditField_c_coarse_value	matlab.ui.control.NumericEditField
c_coarseLabel_2	matlab.ui.control.Label
EditField_A_coarse_value	matlab.ui.control.NumericEditField
A_coarseLabel	matlab.ui.control.Label
pp_finePanel	matlab.ui.container.Panel
EditField_a_fine_value	matlab.ui.control.NumericEditField
a_fineLabel	matlab.ui.control.Label
EditField_b_fine_value	matlab.ui.control.NumericEditField
b_fineLabel	matlab.ui.control.Label
EditField_c_fine_value	matlab.ui.control.NumericEditField
c_fineLabel	matlab.ui.control.Label
EditField_A_fine_value	matlab.ui.control.NumericEditField
A_fineLabel	matlab.ui.control.Label
EditField_delta_transv	matlab.ui.control.NumericEditField
delta_transvLabel	matlab.ui.control.Label
ThresholdsPanel_2	matlab.ui.container.Panel
EditField_V_max_lat	matlab.ui.control.NumericEditField
V_max_latLabel	matlab.ui.control.Label
EditField_V_max_long	matlab.ui.control.NumericEditField
V_max_longLabel	matlab.ui.control.Label
EditField_T_max	matlab.ui.control.NumericEditField
T_maxLabel	matlab.ui.control.Label
EditField_delta_flag	matlab.ui.control.NumericEditField
delta_flagLabel	matlab.ui.control.Label

LinearPanel	matlab.ui.container.Panel
EditField_d_z	matlab.ui.control.NumericEditField
d_zLabel	matlab.ui.control.Label
EditField_d_y	matlab.ui.control.NumericEditField
d_yLabel	matlab.ui.control.Label
LongPanel	matlab.ui.container.Panel
EditField_H	matlab.ui.control.NumericEditField
HLabel	matlab.ui.control.Label
EditField_K	matlab.ui.control.NumericEditField
KLabel	matlab.ui.control.Label
EditField_L	matlab.ui.control.NumericEditField
LLabel	matlab.ui.control.Label
EditField_J	matlab.ui.control.NumericEditField
JLabel	matlab.ui.control.Label
EditField_delta_R	matlab.ui.control.NumericEditField
delta_RLabel	matlab.ui.control.Label
AlgorithmPanel	matlab.ui.container.Panel
AlgoLabel	matlab.ui.control.Label
Box_Algorithm	matlab.ui.control.DropDown
InrunselfchangingalgorithmvalueCheckBox	matlab.ui.control.CheckBox
RadiusSPSPanel	matlab.ui.container.Panel
EditField_R_sps	matlab.ui.control.NumericEditField
R_spsLabel	matlab.ui.control.Label
ImportparamsButton	matlab.ui.control.Button
XYZExpectedPanel	matlab.ui.container.Panel
XLabel	matlab.ui.control.Label
YLabel	matlab.ui.control.Label
EditField_X_Expected	matlab.ui.control.NumericEditField
EditField_Y_Expected	matlab.ui.control.NumericEditField
ZLabel	matlab.ui.control.Label
EditField_Z_Expected	matlab.ui.control.NumericEditField
CYCLEPANELPanel	matlab.ui.container.Panel
TestEditFieldLabel	matlab.ui.control.Label
Test_Field	matlab.ui.control.EditField
pp_coarsePanel_2	matlab.ui.container.Panel
CheckBox_a_coarse	matlab.ui.control.CheckBox
RangeEditField_a_coarse_range	matlab.ui.control.EditField
a_coarseLabel	matlab.ui.control.Label
RangeLabel	matlab.ui.control.Label
CheckBox_b_coarse	matlab.ui.control.CheckBox
RangeEditField_b_coarse_range	matlab.ui.control.EditField
b_coarseLabel	matlab.ui.control.Label
RangeLabel_2	matlab.ui.control.Label
CheckBox_c_coarse	matlab.ui.control.CheckBox
RangeEditField_c_coarse_range	matlab.ui.control.EditField
c_coarseLabel	matlab.ui.control.Label
RangeLabel_3	matlab.ui.control.Label
CheckBox_A_coarse	matlab.ui.control.CheckBox
RangeEditField_A_coarse_range	matlab.ui.control.EditField
A_coarseLabel_2	matlab.ui.control.Label
RangeLabel_4	matlab.ui.control.Label
pp_finePanel_2	matlab.ui.container.Panel
CheckBox_a_fine	matlab.ui.control.CheckBox
RangeEditField_a_fine_range	matlab.ui.control.EditField
a_fineLabel_2	matlab.ui.control.Label

RangeLabel_5	matlab.ui.control.Label
CheckBox_b_fine	matlab.ui.control.CheckBox
RangeEditField_b_fine_range	matlab.ui.control.EditField
b_fineLabel_2	matlab.ui.control.Label
RangeLabel_6	matlab.ui.control.Label
CheckBox_c_fine	matlab.ui.control.CheckBox
RangeEditField_c_fine_range	matlab.ui.control.EditField
c_fineLabel_2	matlab.ui.control.Label
RangeLabel_7	matlab.ui.control.Label
CheckBox_A_fine	matlab.ui.control.CheckBox
RangeEditField_A_fine_range	matlab.ui.control.EditField
A_fineLabel_2	matlab.ui.control.Label
RangeLabel_8	matlab.ui.control.Label
LongPanel_2	matlab.ui.container.Panel
CheckBox_H	matlab.ui.control.CheckBox
RangeEditField_H_range	matlab.ui.control.EditField
HLabel_2	matlab.ui.control.Label
RangeLabel_9	matlab.ui.control.Label
CheckBox_K	matlab.ui.control.CheckBox
RangeEditField_K_range	matlab.ui.control.EditField
KLabel_2	matlab.ui.control.Label
RangeLabel_10	matlab.ui.control.Label
CheckBox_L	matlab.ui.control.CheckBox
RangeEditField_L_range	matlab.ui.control.EditField
LLabel_2	matlab.ui.control.Label
RangeLabel_11	matlab.ui.control.Label
CheckBox_J	matlab.ui.control.CheckBox
RangeEditField_J_range	matlab.ui.control.EditField
JLabel_2	matlab.ui.control.Label
RangeLabel_12	matlab.ui.control.Label
CheckBox_delta_R	matlab.ui.control.CheckBox
RangeEditField_delta_R_range	matlab.ui.control.EditField
delta_RLabel_2	matlab.ui.control.Label
RangeLabel_13	matlab.ui.control.Label
KcoeffPanel	matlab.ui.container.Panel
CheckBox_K_coeff	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range	matlab.ui.control.EditField
K1Label_2	matlab.ui.control.Label
RangeLabel_14	matlab.ui.control.Label
CheckBox_K_coeff_2	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_2	matlab.ui.control.EditField
K2Label_2	matlab.ui.control.Label
RangeLabel_15	matlab.ui.control.Label
CheckBox_K_coeff_3	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_3	matlab.ui.control.EditField
K3Label_2	matlab.ui.control.Label
RangeLabel_16	matlab.ui.control.Label
CheckBox_K_coeff_4	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_4	matlab.ui.control.EditField
K4Label_2	matlab.ui.control.Label
RangeLabel_17	matlab.ui.control.Label
CheckBox_K_coeff_5	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_5	matlab.ui.control.EditField
K5Label_2	matlab.ui.control.Label
RangeLabel_18	matlab.ui.control.Label

CheckBox_K_coeff_6	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_6	matlab.ui.control.EditField
K6Label_2	matlab.ui.control.Label
RangeLabel_19	matlab.ui.control.Label
CheckBox_K_coeff_7	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_7	matlab.ui.control.EditField
K7Label_2	matlab.ui.control.Label
RangeLabel_20	matlab.ui.control.Label
CheckBox_K_coeff_8	matlab.ui.control.CheckBox
RangeEditField_K_coeff_range_8	matlab.ui.control.EditField
K8Label_2	matlab.ui.control.Label
RangeLabel_21	matlab.ui.control.Label
LinearPanel_2	matlab.ui.container.Panel
CheckBox_d_z	matlab.ui.control.CheckBox
RangeEditField_d_z	matlab.ui.control.EditField
d_zLabel_2	matlab.ui.control.Label
RangeLabel_22	matlab.ui.control.Label
CheckBox_d_y	matlab.ui.control.CheckBox
RangeEditField_d_y	matlab.ui.control.EditField
d_yLabel_2	matlab.ui.control.Label
RangeLabel_23	matlab.ui.control.Label
SPSPanel_2	matlab.ui.container.Panel
CheckBox_SPS_high_1	matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_1	matlab.ui.control.EditField
SPS1HighLabel	matlab.ui.control.Label
RangeLabel_24	matlab.ui.control.Label
CheckBox_SPS_Low_1	matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_1	matlab.ui.control.EditField
SPS1LowLabel	matlab.ui.control.Label
RangeLabel_25	matlab.ui.control.Label
CheckBox_SPS_high_2	matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_2	matlab.ui.control.EditField
SPS2HighLabel	matlab.ui.control.Label
RangeLabel_26	matlab.ui.control.Label
CheckBox_SPS_Low_2	matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_2	matlab.ui.control.EditField
SPS2LowLabel	matlab.ui.control.Label
RangeLabel_27	matlab.ui.control.Label
CheckBox_SPS_high_3	matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_3	matlab.ui.control.EditField
SPS3HighLabel	matlab.ui.control.Label
RangeLabel_28	matlab.ui.control.Label
CheckBox_SPS_Low_3	matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_3	matlab.ui.control.EditField
SPS3LowLabel	matlab.ui.control.Label
RangeLabel_29	matlab.ui.control.Label
CheckBox_SPS_high_4	matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_4	matlab.ui.control.EditField
SPS4HighLabel	matlab.ui.control.Label
RangeLabel_30	matlab.ui.control.Label
CheckBox_SPS_Low_4	matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_4	matlab.ui.control.EditField
SPS4LowLabel	matlab.ui.control.Label
RangeLabel_31	matlab.ui.control.Label
CheckBox_SPS_high_5	matlab.ui.control.CheckBox

```
RangeEditField_SPS_high_range_5 matlab.ui.control.EditField
SPS5HighLabel matlab.ui.control.Label
RangeLabel_32 matlab.ui.control.Label
CheckBox_SPS_Low_5 matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_5 matlab.ui.control.EditField
SPS5LowLabel matlab.ui.control.Label
RangeLabel_33 matlab.ui.control.Label
CheckBox_SPS_high_6 matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_6 matlab.ui.control.EditField
SPS6HighLabel matlab.ui.control.Label
RangeLabel_34 matlab.ui.control.Label
CheckBox_SPS_Low_6 matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_6 matlab.ui.control.EditField
SPS6LowLabel matlab.ui.control.Label
RangeLabel_35 matlab.ui.control.Label
CheckBox_SPS_high_7 matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_7 matlab.ui.control.EditField
SPS7HighLabel matlab.ui.control.Label
RangeLabel_36 matlab.ui.control.Label
CheckBox_SPS_Low_7 matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_7 matlab.ui.control.EditField
SPS7LowLabel matlab.ui.control.Label
RangeLabel_37 matlab.ui.control.Label
CheckBox_SPS_high_8 matlab.ui.control.CheckBox
RangeEditField_SPS_high_range_8 matlab.ui.control.EditField
SPS8HighLabel matlab.ui.control.Label
RangeLabel_38 matlab.ui.control.Label
CheckBox_SPS_Low_8 matlab.ui.control.CheckBox
RangeEditField_SPS_Low_range_8 matlab.ui.control.EditField
SPS8LowLabel matlab.ui.control.Label
RangeLabel_39 matlab.ui.control.Label
PathCSVEditField matlab.ui.control.EditField
SelectpathCSVButton matlab.ui.control.Button
MontecarloCheckBox matlab.ui.control.CheckBox
RangeoptionforSPSDropDown matlab.ui.control.DropDown
CSV_REAL_Drop_Down matlab.ui.control.DropDown
PathCSVLabel matlab.ui.control.Label
STARTPanel matlab.ui.container.Panel
STARTSIMULATIONSPSTOXYZButton matlab.ui.control.Button
SaverelavantsvaluesinaCSVCheckBox matlab.ui.control.CheckBox
PathCSV_relevant_values matlab.ui.control.EditField
PathCSVLabel_2 matlab.ui.control.Label
GENERALPLOTSTab matlab.ui.container.Tab
PlotoftwoselectedvariablesPanel matlab.ui.container.Panel
PLOTS_TWO_VARIABLES matlab.ui.control.UIAxes
FilteraxespanelPanel matlab.ui.container.Panel
ChooseaxesButtonGroup matlab.ui.container.ButtonGroup
XaxesButton matlab.ui.control.RadioButton
YaxesButton matlab.ui.control.RadioButton
XFilterPanel matlab.ui.container.Panel
Edit_Min_X matlab.ui.control.NumericEditField
MinLabel matlab.ui.control.Label
Edit_Max_X matlab.ui.control.NumericEditField
MaxLabel matlab.ui.control.Label
GraphMinLabel matlab.ui.control.Label
```

```

GraphMaxLabel          matlab.ui.control.Label
actual_min_x           matlab.ui.control.Label
actual_max_x           matlab.ui.control.Label
YFilterPanel           matlab.ui.container.Panel
Edit_Min_Y             matlab.ui.control.NumericEditField
MinLabel_2             matlab.ui.control.Label
Edit_Max_Y             matlab.ui.control.NumericEditField
MaxLabel_2             matlab.ui.control.Label
GraphMinLabel_2        matlab.ui.control.Label
GraphMaxLabel_2        matlab.ui.control.Label
actual_min_y           matlab.ui.control.Label
actual_max_y           matlab.ui.control.Label
FilteraxesButton       matlab.ui.control.StateButton
AdviseforthecreationofaTwovariablesplotLabel matlab.ui.control.Label
UIAxes2                matlab.ui.control.UIAxes
ErrorXYZPanel          matlab.ui.container.Panel
X_error_plot           matlab.ui.control.UIAxes
Y_error_plot           matlab.ui.control.UIAxes
Z_error_plot           matlab.ui.control.UIAxes
RELEVANTPLOTTab        matlab.ui.container.Tab
RELEVANT_VALUES_PLOT   matlab.ui.control.UIAxes
Y_GraphButtonGroup     matlab.ui.container.ButtonGroup
Nom_XButton            matlab.ui.control.RadioButton
Nom_YButton            matlab.ui.control.RadioButton
Nom_ZButton            matlab.ui.control.RadioButton
Red_XButton            matlab.ui.control.RadioButton
Red_YButton            matlab.ui.control.RadioButton
Red_ZButton            matlab.ui.control.RadioButton
Delta_Nom_XButton      matlab.ui.control.RadioButton
Delta_Nom_YButton      matlab.ui.control.RadioButton
Delta_Nom_ZButton      matlab.ui.control.RadioButton
Delta_Red_XButton      matlab.ui.control.RadioButton
Delta_Red_YButton      matlab.ui.control.RadioButton
Delta_Red_ZButton      matlab.ui.control.RadioButton
PLOTButton            matlab.ui.control.Button
PathCSVEditField_Relevant matlab.ui.control.EditField
SelectpathCSVButton_2  matlab.ui.control.Button

end

```

```

properties (Access = public)
    flag_real_table = 0;
    last_real_table = 0; % Description
    temp_table = 0; % Description
    column_name = ["r_sps", "A_coarse", "a_coarse", "b_coarse", "c_coarse",
    "A_fine", "a_fine", "b_fine", "c_fine", "H", "K", "L", "J", "delta_R", "K1", "K2",
    "K3", "K4", "K5", "K6", "K7", "K8", "d_z", "d_y", "SPS1", "SPS3", "SPS5", "SPS7",
    "SPS2", "SPS4", "SPS6", "SPS8", "Flag_1_Nom", "Flag_1_Red", "Flag_Lat_Nom",
    "Flag_Lat_Red", "Flag_Long_Nom", "Flag_Long_Red", "flag_acos", "NOM_X_mm", "NOM_Y_mm",
    "NOM_Z_mm", "RED_X_mm", "RED_Y_mm", "RED_Z_mm", "Final_NOM_Flag_1",
    "Final_NOM_delta_15", "Final_NOM_delta_37", "Final_RED_Flag_Long", "Final_RED_Flag_1",
    "Final_RED_delta_26", "Final_RED_delta_48", "Final_RED_Flag_Long", "Expected_X_mm",
    "Expected_y_mm", "Expected_Z_mm", "Delta_X_nom_mm", "Delta_Y_nom_mm",
    "Delta_Z_nom_mm", "Delta_X_red_mm", "Delta_Y_red_mm", "Delta_Z_red_mm", "algo_nom",
    "algo_red", "flag_x_0"]; % Description

```



```

methods (Access = private)

% =====
% MAIN FUNCTION
function main_function(app)
    %app.last_real_table = zeros(str2double(app.Test_Field.Value) + 1, app.
column_number_real_time);
    %app.last_real_table = string(app.last_real_table);
    %app.last_real_table(:, :) = "";
    app.STARTSIMULATIONSPSTOXYZButton.Enable = 'off';
    app.CREATE TWO VARIABLES PLOT Button.Enable = 'off';
    app.GETSELECTEDCOLUMNButton.Enable = 'off';
    msgbox("Simulation started, do not change any values until the next
popup", "SPS-TO-XYZ", "warn");
    app.Full_Table.Data = {};
    app.Filtered_Table.Data = {};
    %% Previous Loop Data
    % Nominal Set: Position vector related to the measurements @ t = t_n-1
    %nom_01 = [0, 0, -50, 0, 0];
    % Redundant Set: Position vector related to the measurements @ t = t_n-1
    %red_01 = [0, 0, -50, 0, 0];

    %% Load data and reconf params
    Raw_data = acquisition_raw_data(app);
    Reconf_params = acquisition_reconf_params(app);

    %% Initial Values
    app.initial_raw_data = Raw_data;
    app.initial_reconf_params = Reconf_params;

    r_temporary = ["NaN", "NaN", "NaN", "NaN", "NaN", "NaN", "NaN", "NaN"];
    set_temporary = "NaN";
    temps_temporary = ["NaN", "NaN"];
    flag_1_temporary = ["NaN", "NaN"];
    nom_temporary = ["NaN", "NaN", "NaN", "NaN", "NaN"];
    red_temporary = ["NaN", "NaN", "NaN", "NaN", "NaN"];
    flag_lat_temporary = ["NaN", "NaN"];
    flag_long_temporary = ["NaN", "NaN"];
    nom_final_pos_temporary = ["NaN", "NaN", "NaN", "NaN", "NaN", "NaN",
"NaN", "NaN"];
    red_final_pos_temporary = ["NaN", "NaN", "NaN", "NaN", "NaN", "NaN",
"NaN", "NaN"];
    flag_acos_temporary = "NaN";
    algo_nom_temporary = "NaN";
    algo_red_temporary = "NaN";
    flag_x_0_temporary = "NaN";
    x_y_z_delta_temporary = ["NaN", "NaN", "NaN", "NaN", "NaN", "NaN", "NaN",
"NaN", "NaN"];

    % vector corresponding to table indices
    app.vector_check = [];
    %%CYCLE INFO

```

```

% cycle number
number_cycle = str2double(app.Test_Field.Value);

% R_sps
r_sps = app.EditField_R_sps.Value;

%=====
INTERMEDIATE VALUES SECTION
%=====
% a_coarse
if app.CheckBox_a_coarse.Value
    [Intermediate_value_a_coarse, pp_coarse_a_coarse] = range_and_value(
app, app.RangeEditField_a_coarse_range.Value, Reconf_params.pp_coarse.a_coarse,
number_cycle);
    app.vector_check(end+1) = 10;
end
% b_coarse
if app.CheckBox_b_coarse.Value
    [Intermediate_value_b_coarse, pp_coarse_b_coarse] = range_and_value(
app, app.RangeEditField_b_coarse_range.Value, Reconf_params.pp_coarse.b_coarse,
number_cycle);
    app.vector_check(end+1) = 11;
end
% c_coarse
if app.CheckBox_c_coarse.Value
    [Intermediate_value_c_coarse, pp_coarse_c_coarse] = range_and_value(
app, app.RangeEditField_c_coarse_range.Value, Reconf_params.pp_coarse.c_coarse,
number_cycle);
    app.vector_check(end+1) = 12;
end
% A_coarse
if app.CheckBox_A_coarse.Value
    [Intermediate_value_A_coarse, pp_coarse_A_coarse] = range_and_value(
app, app.RangeEditField_A_coarse_range.Value, Reconf_params.pp_coarse.A_coarse,
number_cycle);
    app.vector_check(end+1) = 9;
end
% a_fine
if app.CheckBox_a_fine.Value
    [Intermediate_value_a_fine, pp_fine_a_fine] = range_and_value(app,
app.RangeEditField_a_fine_range.Value, Reconf_params.pp_fine.a_fine, number_cycle);
    app.vector_check(end+1) = 14;
end
% b_fine
if app.CheckBox_b_fine.Value
    [Intermediate_value_b_fine, pp_fine_b_fine] = range_and_value(app,
app.RangeEditField_b_fine_range.Value, Reconf_params.pp_fine.b_fine, number_cycle);
    app.vector_check(end+1) = 15;
end
% c_fine
if app.CheckBox_c_fine.Value
    [Intermediate_value_c_fine, pp_fine_c_fine] = range_and_value(app,
app.RangeEditField_c_fine_range.Value, Reconf_params.pp_fine.c_fine, number_cycle);

```

```

        app.vector_check(end+1) = 16;
    end
    % A_fine
    if app.CheckBox_A_fine.Value
        [Intermediate_value_A_fine, pp_fine_A_fine] = range_and_value(app, ✓
app.RangeEditField_A_fine_range.Value, Reconf_params.pp_fine.A_fine, number_cycle);
        app.vector_check(end+1) = 13;
    end
    % H
    if app.CheckBox_H.Value
        [Intermediate_value_H, long_H] = range_and_value(app, app.✓
RangeEditField_H_range.Value, Reconf_params.long.H, number_cycle);
        app.vector_check(end+1) = 17;
    end
    % K
    if app.CheckBox_K.Value
        [Intermediate_value_K, long_K] = range_and_value(app, app.✓
RangeEditField_K_range.Value, Reconf_params.long.K, number_cycle);
        app.vector_check(end+1) = 18;
    end
    % L
    if app.CheckBox_L.Value
        [Intermediate_value_L, long_L] = range_and_value(app, app.✓
RangeEditField_L_range.Value, Reconf_params.long.L, number_cycle);
        app.vector_check(end+1) = 19;
    end
    % J
    if app.CheckBox_J.Value
        [Intermediate_value_J, long_J] = range_and_value(app, app.✓
RangeEditField_J_range.Value, Reconf_params.long.J, number_cycle);
        app.vector_check(end+1) = 20;
    end
    % delta_R
    if app.CheckBox_delta_R.Value
        [Intermediate_value_delta_R, long_delta_R] = range_and_value(app, app.✓
RangeEditField_delta_R_range.Value, Reconf_params.long.delta_R, number_cycle);
        app.vector_check(end+1) = 21;
    end
    % K1
    if app.CheckBox_K_coeff.Value
        [Intermediate_value_K_coeff, K_coeff] = range_and_value(app, app.✓
RangeEditField_K_coeff_range.Value, Reconf_params.K(1), number_cycle);
        app.vector_check(end+1) = 22;
    end
    % K2
    if app.CheckBox_K_coeff_2.Value
        [Intermediate_value_K_coeff_2, K_coeff_2] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_2.Value, Reconf_params.K(2), number_cycle);
        app.vector_check(end+1) = 23;
    end
    % K3
    if app.CheckBox_K_coeff_3.Value
        [Intermediate_value_K_coeff_3, K_coeff_3] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_3.Value, Reconf_params.K(3), number_cycle);

```

```

        app.vector_check(end+1) = 24;
    end
    % K4
    if app.CheckBox_K_coeff_4.Value
        [Intermediate_value_K_coeff_4, K_coeff_4] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_4.Value, Reconf_params.K(4), number_cycle);
        app.vector_check(end+1) = 25;
    end
    % K5
    if app.CheckBox_K_coeff_5.Value
        [Intermediate_value_K_coeff_5, K_coeff_5] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_5.Value, Reconf_params.K(5), number_cycle);
        app.vector_check(end+1) = 26;
    end
    % K6
    if app.CheckBox_K_coeff_6.Value
        [Intermediate_value_K_coeff_6, K_coeff_6] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_6.Value, Reconf_params.K(6), number_cycle);
        app.vector_check(end+1) = 27;
    end
    % K7
    if app.CheckBox_K_coeff_7.Value
        [Intermediate_value_K_coeff_7, K_coeff_7] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_7.Value, Reconf_params.K(7), number_cycle);
        app.vector_check(end+1) = 28;
    end
    % K8
    if app.CheckBox_K_coeff_8.Value
        [Intermediate_value_K_coeff_8, K_coeff_8] = range_and_value(app, app.✓
RangeEditField_K_coeff_range_8.Value, Reconf_params.K(8), number_cycle);
        app.vector_check(end+1) = 29;
    end

    % d_z
    if app.CheckBox_d_z.Value
        [Intermediate_value_d_z, linear_d_z] = range_and_value(app, app.✓
RangeEditField_d_z.Value, Reconf_params.linear.d_z, number_cycle);
        app.vector_check(end+1) = 30;
    end
    % d_y
    if app.CheckBox_d_y.Value
        [Intermediate_value_d_y, linear_d_y] = range_and_value(app, app.✓
RangeEditField_d_y.Value, Reconf_params.linear.d_y, number_cycle);
        app.vector_check(end+1) = 31;
    end

    % SPS1_High
    if app.CheckBox_SPS_high_1.Value
        [Intermediate_value_SPS_high_1, SPS_high_1] = range_and_value(app,✓
app.RangeEditField_SPS_high_range_1.Value, Raw_data.Read(1), number_cycle);
        app.vector_check(end+1) = -1;
    end
    % SPS2_High
    if app.CheckBox_SPS_high_2.Value
        [Intermediate_value_SPS_high_2, SPS_high_2] = range_and_value(app,✓

```

```

app.RangeEditField_SPS_high_range_2.Value, Raw_data.Read(3), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS3_High
if app.CheckBox_SPS_high_3.Value
    [Intermediate_value_SPS_high_3, SPS_high_3] = range_and_value(app, ✓
app.RangeEditField_SPS_high_range_3.Value, Raw_data.Read(5), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS4_High
if app.CheckBox_SPS_high_4.Value
    [Intermediate_value_SPS_high_4, SPS_high_4] = range_and_value(app, ✓
app.RangeEditField_SPS_high_range_4.Value, Raw_data.Read(7), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS5_High
if app.CheckBox_SPS_high_5.Value
    [Intermediate_value_SPS_high_5, SPS_high_5] = range_and_value(app, ✓
app.RangeEditField_SPS_high_range_5.Value, Raw_data.Read(9), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS6_High
if app.CheckBox_SPS_high_6.Value
    [Intermediate_value_SPS_high_6, SPS_high_6] = range_and_value(app, ✓
app.RangeEditField_SPS_high_range_6.Value, Raw_data.Read(11), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS7_High
if app.CheckBox_SPS_high_7.Value
    [Intermediate_value_SPS_high_7, SPS_high_7] = range_and_value(app, ✓
app.RangeEditField_SPS_high_range_7.Value, Raw_data.Read(13), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS8_High
if app.CheckBox_SPS_high_8.Value
    [Intermediate_value_SPS_high_8, SPS_high_8] = range_and_value(app, ✓
app.RangeEditField_SPS_high_range_8.Value, Raw_data.Read(15), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS1_Low
if app.CheckBox_SPS_Low_1.Value
    [Intermediate_value_SPS_Low_1, SPS_Low_1] = range_and_value(app, app. ✓
RangeEditField_SPS_Low_range_1.Value, Raw_data.Read(2), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS2_Low
if app.CheckBox_SPS_Low_2.Value
    [Intermediate_value_SPS_Low_2, SPS_Low_2] = range_and_value(app, app. ✓
RangeEditField_SPS_Low_range_2.Value, Raw_data.Read(4), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS3_Low
if app.CheckBox_SPS_Low_3.Value
    [Intermediate_value_SPS_Low_3, SPS_Low_3] = range_and_value(app, app. ✓
RangeEditField_SPS_Low_range_3.Value, Raw_data.Read(6), number_cycle);

```

```
app.vector_check(end+1) = -1;
end
% SPS4_Low
if app.CheckBox_SPS_Low_4.Value
    [Intermediate_value_SPS_Low_4, SPS_Low_4] = range_and_value(app, app.RangeEditField_SPS_Low_range_4.Value, Raw_data.Read(8), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS5_Low
if app.CheckBox_SPS_Low_5.Value
    [Intermediate_value_SPS_Low_5, SPS_Low_5] = range_and_value(app, app.RangeEditField_SPS_Low_range_5.Value, Raw_data.Read(10), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS6_Low
if app.CheckBox_SPS_Low_6.Value
    [Intermediate_value_SPS_Low_6, SPS_Low_6] = range_and_value(app, app.RangeEditField_SPS_Low_range_6.Value, Raw_data.Read(12), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS7_Low
if app.CheckBox_SPS_Low_7.Value
    [Intermediate_value_SPS_Low_7, SPS_Low_7] = range_and_value(app, app.RangeEditField_SPS_Low_range_7.Value, Raw_data.Read(14), number_cycle);
    app.vector_check(end+1) = -1;
end
% SPS8_Low
if app.CheckBox_SPS_Low_8.Value
    [Intermediate_value_SPS_Low_8, SPS_Low_8] = range_and_value(app, app.RangeEditField_SPS_Low_range_8.Value, Raw_data.Read(16), number_cycle);
    app.vector_check(end+1) = -1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CREATE FOLDER
dir = string(pwd) + "\";
if app.PathCSV>EditField.Value ~= "<ACTUAL MATLAB PATH>"
    dir = app.PathCSV>EditField.Value + "\";
end

%%%%%%%%%%%%%%%% START CYCLE
for j=0:number_cycle
    app.n_cycle = j;
    app.ActualTest>EditField.Value = j;
    pause on
    pause(0.00001);
    pause off

% %%%% 
% if(app.real_value == 0)
%     return;
```

```

%           end
%           %%%%%%%%%

% Nominal Set: Position vector related to the measurements @ t = t_n-1
nom_01 = [0, 0, -50, 0, 0];
% Redundant Set: Position vector related to the measurements @ t = ✓
t_n-1
red_01 = [0, 0, -50, 0, 0];

✓
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CHANGE ✓
VALUES SECTION ✓
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ✓
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CHANGE VALUE OF a_coarse in range
if app.CheckBox_a_coarse.Value
    [Reconf_params.pp_coarse.a_coarse] = range_or_montecarlo_value ✓
(app, pp_coarse_a_coarse, Intermediate_value_a_coarse, app. ✓
RangeEditField_a_coarse_range.Value);
end
%CHANGE VALUE OF b_coarse in range
if app.CheckBox_b_coarse.Value
    [Reconf_params.pp_coarse.b_coarse] = range_or_montecarlo_value ✓
(app, pp_coarse_b_coarse, Intermediate_value_b_coarse, app. ✓
RangeEditField_b_coarse_range.Value);
end
%CHANGE VALUE OF c_coarse in range
if app.CheckBox_c_coarse.Value
    [Reconf_params.pp_coarse.c_coarse] = range_or_montecarlo_value ✓
(app, pp_coarse_c_coarse, Intermediate_value_c_coarse, app. ✓
RangeEditField_c_coarse_range.Value);
end
%ASSIGN A_coarse VALUE
if app.CheckBox_A_coarse.Value
    [Reconf_params.pp_coarse.A_coarse] = range_or_montecarlo_value ✓
(app, pp_coarse_A_coarse, Intermediate_value_A_coarse, app. ✓
RangeEditField_A_coarse_range.Value);
else
    Reconf_params.pp_coarse.A_coarse = (Reconf_params.pp_coarse. ✓
c_coarse^3) * ((1 / Reconf_params.pp_coarse.a_coarse)+((2 * r_sps) / (Reconf_params. ✓
pp_coarse.b_coarse^2))+((3 * r_sps^2) / Reconf_params.pp_coarse.c_coarse^3));
end

%CHANGE VALUE OF a_coarse in range
if app.CheckBox_a_fine.Value
    [Reconf_params.pp_fine.a_fine] = range_or_montecarlo_value(app, ✓
pp_fine_a_fine, Intermediate_value_a_fine, app.RangeEditField_a_fine_range.Value);
end
%CHANGE VALUE OF b_fine in range
if app.CheckBox_b_fine.Value
    [Reconf_params.pp_fine.b_fine] = range_or_montecarlo_value(app, ✓
pp_fine_b_fine, Intermediate_value_b_fine, app.RangeEditField_b_fine_range.Value);
end
%CHANGE VALUE OF c_fine in range
if app.CheckBox_c_fine.Value

```

```

[Reconf_params.pp_fine.c_fine] = range_or_montecarlo_value(app,
pp_fine_c_fine, Intermediate_value_c_fine, app.RangeEditField_c_fine_range.Value);
end
%ASSIGN A_fine VALUE
if app.CheckBox_A_fine.Value
[Reconf_params.pp_fine.A_fine] = range_or_montecarlo_value(app,
pp_fine_A_fine, Intermediate_value_A_fine, app.RangeEditField_A_fine_range.Value);
else
Reconf_params.pp_fine.A_fine = (Reconf_params.pp_fine.c_fine^3) *
((1 / Reconf_params.pp_fine.a_fine)+(2 * r_sps) / (Reconf_params.pp_fine.b_fine^2))+
((3 * r_sps^2) / Reconf_params.pp_fine.c_fine^3));
end

%CHANGE VALUE OF long_H in range
if app.CheckBox_H.Value
[Reconf_params.long.H] = range_or_montecarlo_value(app, long_H,
Intermediate_value_H, app.RangeEditField_H_range.Value);
end
%CHANGE VALUE OF long_K in range
if app.CheckBox_K.Value
[Reconf_params.long.K] = range_or_montecarlo_value(app, long_K,
Intermediate_value_K, app.RangeEditField_K_range.Value);
end
%CHANGE VALUE OF long_L in range
if app.CheckBox_L.Value
[Reconf_params.long.L] = range_or_montecarlo_value(app, long_L,
Intermediate_value_L, app.RangeEditField_L_range.Value);
end
%CHANGE VALUE OF long_J in range
if app.CheckBox_J.Value
[Reconf_params.long.J] = range_or_montecarlo_value(app, long_J,
Intermediate_value_J, app.RangeEditField_J_range.Value);
end
%CHANGE VALUE OF long_delta_R in range
if app.CheckBox_delta_R.Value
[Reconf_params.long.delta_R] = range_or_montecarlo_value(app,
long_delta_R, Intermediate_value_delta_R, app.RangeEditField_delta_R_range.Value);
end

%CHANGE VALUE OF K1 in range
if app.CheckBox_K_coeff.Value
[a] = range_or_montecarlo_value(app, K_coeff,
Intermediate_value_K_coeff, app.RangeEditField_K_coeff_range.Value);
Reconf_params.K(1) = a;
end
%CHANGE VALUE OF K2 in range
if app.CheckBox_K_coeff_2.Value
[Reconf_params.K(2)] = range_or_montecarlo_value(app, K_coeff_2,
Intermediate_value_K_coeff_2, app.RangeEditField_K_coeff_range_2.Value);
end
%CHANGE VALUE OF K3 in range
if app.CheckBox_K_coeff_3.Value
[Reconf_params.K(3)] = range_or_montecarlo_value(app, K_coeff_3,
Intermediate_value_K_coeff_3, app.RangeEditField_K_coeff_range_3.Value);
end

```

```

%CHANGE VALUE OF K4 in range
if app.CheckBox_K_coeff_4.Value
    [Reconf_params.K(4)] = range_or_montecarlo_value(app, K_coeff_4, ✓
Intermediate_value_K_coeff_4, app.RangeEditField_K_coeff_range_4.Value);
end
%CHANGE VALUE OF K5 in range
if app.CheckBox_K_coeff_5.Value
    [Reconf_params.K(5)] = range_or_montecarlo_value(app, K_coeff_5, ✓
Intermediate_value_K_coeff_5, app.RangeEditField_K_coeff_range_5.Value);
end
%CHANGE VALUE OF K6 in range
if app.CheckBox_K_coeff_6.Value
    [Reconf_params.K(6)] = range_or_montecarlo_value(app, K_coeff_6, ✓
Intermediate_value_K_coeff_6, app.RangeEditField_K_coeff_range_6.Value);
end
%CHANGE VALUE OF K7 in range
if app.CheckBox_K_coeff_7.Value
    [Reconf_params.K(7)] = range_or_montecarlo_value(app, K_coeff_7, ✓
Intermediate_value_K_coeff_7, app.RangeEditField_K_coeff_range_7.Value);
end
%CHANGE VALUE OF K8 in range
if app.CheckBox_K_coeff_8.Value
    [Reconf_params.K(8)] = range_or_montecarlo_value(app, K_coeff_8, ✓
Intermediate_value_K_coeff_8, app.RangeEditField_K_coeff_range_8.Value);
end

%CHANGE VALUE OF d_z in range
if app.CheckBox_d_z.Value
    [Reconf_params.linear.d_z] = range_or_montecarlo_value(app, ✓
linear_d_z, Intermediate_value_d_z, app.RangeEditField_d_z.Value);
end
%CHANGE VALUE OF d_y in range
if app.CheckBox_d_y.Value
    [Reconf_params.linear.d_y] = range_or_montecarlo_value(app, ✓
linear_d_y, Intermediate_value_d_y, app.RangeEditField_d_y.Value);
end

%CHANGE VALUE OF SPS_high_1 in range
if app.CheckBox_SPS_high_1.Value
    [Raw_data.Read(1)] = range_or_montecarlo_value(app, SPS_high_1, ✓
Intermediate_value_SPS_high_1, app.RangeEditField_SPS_high_range_1.Value);
    Raw_data.Read(2) = Raw_data.Read(1) / 5;
end
%CHANGE VALUE OF SPS_high_2 in range
if app.CheckBox_SPS_high_2.Value
    [Raw_data.Read(3)] = range_or_montecarlo_value(app, SPS_high_2, ✓
Intermediate_value_SPS_high_2, app.RangeEditField_SPS_high_range_2.Value);
    Raw_data.Read(4) = Raw_data.Read(3) / 5;
end
%CHANGE VALUE OF SPS_high_3 in range
if app.CheckBox_SPS_high_3.Value
    [Raw_data.Read(5)] = range_or_montecarlo_value(app, SPS_high_3, ✓
Intermediate_value_SPS_high_3, app.RangeEditField_SPS_high_range_3.Value);
    Raw_data.Read(6) = Raw_data.Read(5) / 5;
end

```

```

%CHANGE VALUE OF SPS_high_4 in range
if app.CheckBox_SPS_high_4.Value
    [Raw_data.Read(7)] = range_or_montecarlo_value(app, SPS_high_4, ✓
Intermediate_value_SPS_high_4, app.RangeEditField_SPS_high_range_4.Value);
    Raw_data.Read(8) = Raw_data.Read(7) / 5;
end
%CHANGE VALUE OF SPS_high_5 in range
if app.CheckBox_SPS_high_5.Value
    [Raw_data.Read(9)] = range_or_montecarlo_value(app, SPS_high_5, ✓
Intermediate_value_SPS_high_5, app.RangeEditField_SPS_high_range_5.Value);
    Raw_data.Read(10) = Raw_data.Read(9) / 5;
end
%CHANGE VALUE OF SPS_high_6 in range
if app.CheckBox_SPS_high_6.Value
    [Raw_data.Read(11)] = range_or_montecarlo_value(app, SPS_high_6, ✓
Intermediate_value_SPS_high_6, app.RangeEditField_SPS_high_range_6.Value);
    Raw_data.Read(12) = Raw_data.Read(11) / 5;
end
%CHANGE VALUE OF SPS_high_7 in range
if app.CheckBox_SPS_high_7.Value
    [Raw_data.Read(13)] = range_or_montecarlo_value(app, SPS_high_7, ✓
Intermediate_value_SPS_high_7, app.RangeEditField_SPS_high_range_7.Value);
    Raw_data.Read(14) = Raw_data.Read(13) / 5;
end
%CHANGE VALUE OF SPS_high_8 in range
if app.CheckBox_SPS_high_8.Value
    [Raw_data.Read(15)] = range_or_montecarlo_value(app, SPS_high_8, ✓
Intermediate_value_SPS_high_8, app.RangeEditField_SPS_high_range_8.Value);
    Raw_data.Read(16) = Raw_data.Read(15) / 5;
end
%CHANGE VALUE OF SPS_Low_1 in range
if app.CheckBox_SPS_Low_1.Value
    [Raw_data.Read(2)] = range_or_montecarlo_value(app, SPS_Low_1, ✓
Intermediate_value_SPS_Low_1, app.RangeEditField_SPS_Low_range_1.Value);
    Raw_data.Read(1) = Raw_data.Read(2) * 5;
end
%CHANGE VALUE OF SPS_Low_2 in range
if app.CheckBox_SPS_Low_2.Value
    [Raw_data.Read(4)] = range_or_montecarlo_value(app, SPS_Low_2, ✓
Intermediate_value_SPS_Low_2, app.RangeEditField_SPS_Low_range_2.Value);
    Raw_data.Read(3) = Raw_data.Read(4) * 5;
end
%CHANGE VALUE OF SPS_Low_3 in range
if app.CheckBox_SPS_Low_3.Value
    [Raw_data.Read(6)] = range_or_montecarlo_value(app, SPS_Low_3, ✓
Intermediate_value_SPS_Low_3, app.RangeEditField_SPS_Low_range_3.Value);
    Raw_data.Read(5) = Raw_data.Read(6) * 5;
end
%CHANGE VALUE OF SPS_Low_4 in range
if app.CheckBox_SPS_Low_4.Value
    [Raw_data.Read(8)] = range_or_montecarlo_value(app, SPS_Low_4, ✓
Intermediate_value_SPS_Low_4, app.RangeEditField_SPS_Low_range_4.Value);
    Raw_data.Read(7) = Raw_data.Read(8) * 5;
end
%CHANGE VALUE OF SPS_Low_5 in range

```

```
%% Input Pre-Processing

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

*** ***

```
x_y_z_delta_temporary, flag_acos_temporary, algo_nom_temporary, algo_red_temporary,
flag_x_0_temporary);
    update_vector(app, "Pre_processing", V, 1);

    %% First Validity Check

    [flag_1] = Validity_Flag_1(app, R, set, temps, Raw_data.door,
Reconf_params.thresholds.T_max);

    %CONVERSION and EXPORT
    [V] = conversion(app, Raw_data, Reconf_params, R, set, temps, flag_1,
nom_temporary, red_temporary, flag_lat_temporary, flag_long_temporary,
nom_final_pos_temporary, red_final_pos_temporary, x_y_z_delta_temporary,
flag_acos_temporary, algo_nom_temporary, algo_red_temporary, flag_x_0_temporary);
    update_vector(app, "First_VValidity_Check", V, 1);

    %% Relative position Computation

    [nom_pos, red_pos, flag_acos, algo_nom, algo_red, flag_x_0] =
SPS_Algorithms (app, R, set, Reconf_params.algo, Reconf_params.pp_coarse,
Reconf_params.pp_fine, Reconf_params.linear, Reconf_params.long);

    %CONVERSION and EXPORT
    [V] = conversion(app, Raw_data, Reconf_params, R, set, temps, flag_1,
nom_pos, red_pos, flag_lat_temporary, flag_long_temporary, nom_final_pos_temporary,
red_final_pos_temporary, x_y_z_delta_temporary, flag_acos, algo_nom, algo_red,
flag_x_0);
    update_vector(app, "Relative_position_Computation", V, 1);

    %% Second Validity Check

    [ flag_lat, flag_long ] = Validity_Flag_2(app, set, nom_01, nom_pos,
red_01, red_pos, Reconf_params.thresholds );

    %CONVERSION and EXPORT
    [V] = conversion(app, Raw_data, Reconf_params, R, set, temps, flag_1,
nom_pos, red_pos, flag_lat, flag_long, nom_final_pos_temporary,
red_final_pos_temporary, x_y_z_delta_temporary, flag_acos, algo_nom, algo_red,
flag_x_0);
    update_vector(app, "Second_VValidity_Check", V, 1);

    %% Output

    nom = [flag_1(1), nom_pos, flag_lat(1), flag_long(1)];
    red = [flag_1(2), red_pos, flag_lat(2), flag_long(2)];

    nom_01 = nom_pos;
    red_01 = red_pos;

    %%%%%%%%%%
    if ~isreal(nom_01(4)) || ~isreal(nom_01(5))
        app.real_value = 0;
    end
    %%%%%%%%%%
```

*These scripts are provided under authorized request

```
% =====
%
% CONVERSION
function [V] = conversion(~, Raw_data, Reconf_params, r_temp, set_temp,
temps_temporary, flag_l_temporary, nom_temporary, red_temporary, flag_lat_temporary,
flag_long_temporary, nom_final_pos_temporary, red_final_pos_temporary, d_ex,
flag_acos, algo_nom, algo_red, flag_x_0_temporary)
    [raw] = [Raw_data.Read.', " ", Raw_data.Temp.', " ", Raw_data.door.', " ",
Raw_data.door_tr.', " "];
    [recon] = [Reconf_params.K.', " ", Reconf_params.algo.', " ", struct2cell(
(Reconf_params.pp_coarse).', " ", struct2cell(Reconf_params.pp_fine).', " ",
struct2cell(Reconf_params.linear).', " ", struct2cell(Reconf_params.long).', " ",
struct2cell(Reconf_params.thresholds).', " ")];
    V = [raw, recon, r_temp, set_temp, " ", temps_temporary, " ",
flag_l_temporary, " ", nom_temporary, " ", red_temporary, " ", flag_lat_temporary, " ",
flag_long_temporary, " ", nom_final_pos_temporary, " ", red_final_pos_temporary, " ",
d_ex, " ", flag_acos, " ", algo_nom, " ", algo_red, " ", flag_x_0_temporary];
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% UPDATE VECTOR
function update_vector(app, function_f, V, status)
    real_time_path = pwd() + "\" + "temp_real_time.csv";
    Vector = [function_f, " ", V];
    %new_tab = array2table(Vector, 'VariableNames', app.column_indices);
    % 0 = finish table
    % 1 = concatenate table
    % 2 = first table
    if status == 1
        app.csv_table = [app.csv_table; Vector];
    elseif status == 2
        app.csv_table = Vector;
    else
        %CREATION OF LAST 3 ROW OF CSV
        x = last_3_rows_CSV(app, Vector);
        % get row real time
        real_time_table = get_row_for_real_time_table(app, x);
        % assign name to extra table values
        x = assign_name_2_values(app, x);
        % CREATE LAST TABLE TO PRINT EVENTUALLY IN CSV
        app.csv_table = [app.csv_table; Vector; x];
        % Check presence of complex values
        if ~isreal(str2double(real_time_table(2:end)))
            app.Lamp.Enable = 'on';
            app.path = app.path + "(complex values)";
        end
        if str2double(real_time_table(63)) == 1
            app.path = app.path + "(arcos flag)";
        end
        % CHECK IF THE DEBUG CONTAINS REAL TIME PART
        if app.CSV_REAL_Drop_Down.Value == "Only Real Time" || app.
CSV_REAL_Drop_Down.Value == "CSV and Real Time"

            % CHECK IF FIRST LINE OF THE TABLE
            if app.flag_real_table == 0
                stringa_temp = "TEST,nom_X_mm,nom_Y_mm,nom_Z_mm,red_X_mm,
red_Y_mm,red_Z_mm,R_SPS,A_coarse,a_coarse,b_coarse,c_coarse,A_fine,a_fine,b_fine,
c_fine,H,K,L,J,delta_R,K1,K2,K3,K4,K5,K6,K7,K8,d_z,d_y,SPS1,SPS3,SPS5,SPS7,SPS2,SPS4,
SPS6,SPS8,Flag_1_Nom,Flag_1_Red,Flag_Lat_Nom,Flag_Lat_Red,Flag_Long_Nom,Flag_Long_Red,
Final_NOM_Flag_1,Final_NOM_delta_15,Final_NOM_delta_37,Final_NOM_Flag_Long,
Final_RED_Flag_1,Final_RED_delta_26,Final_RED_delta_48,Final_RED_Flag_Long,
Expected_X_mm,Expected_Y_mm,Expected_Z_mm,Delta_X_nom_mm,Delta_Y_nom_mm,
Delta_Z_nom_mm,Delta_X_red_mm,Delta_Y_red_mm,Delta_Z_red_mm,flag_acos,algo_nom,
algo_red,flag_x_0\n";

                fid = fopen(real_time_path, "w");
                fprintf(fid, stringa_temp);
                fclose(fid);
                app.flag_real_table = 1;
            end
            stringa_temp = "";
            for a = 1:length(real_time_table)
                if ismissing(real_time_table(a))
                    real_time_table(a) = "NaN";
                end
            end
        end
    end
end

```

```

        if (a == 1)
            stringa_temp = stringa_temp + app.n_cycle + ",";
        elseif (a == length(real_time_table))
            stringa_temp = stringa_temp + real_time_table(a);
        else
            stringa_temp = stringa_temp + real_time_table(a) + ",";
        end
    end
    stringa_temp = stringa_temp + "\n";
    fid = fopen(real_time_path, "a+");
    fprintf(fid, stringa_temp);
    fclose(fid);
    % IF LAST TEST
    if app.n_cycle == str2double(app.Test_Field.Value)
        %app.Full_Table.Data = [app.Full_Table.Data;array2table(app.↵
last_real_table)];

        %app.temp_table = app.last_real_table;
        %app.last_real_table = 0;
        %app.flag_real_table = 0;
        app.Full_Table.Data = readtable(real_time_path);
        fclose(fopen(real_time_path, "r"));
        delete(real_time_path);
        app.temp_table = table2array(app.Full_Table.Data);
        app.last_real_table = 0;
        app.flag_real_table = 0;
        % UPDATE LABEL AND NOT REAL LAMP
        app.Label_Flag_1_nom.Text = Vector(100);
        app.Label_Flag_1_red.Text = Vector(109);
        app.Label_Flag_lat_nom.Text = Vector(94);
        app.Label_Flag_lat_red.Text = Vector(95);
        app.Label_Flag_Long_nom.Text = Vector(107);
        app.Label_Flag_Long_red.Text = Vector(116);
        app.Label_nom_x.Text = Vector(103);
        app.Label_nom_y.Text = Vector(104);
        app.Label_nom_z.Text = Vector(105);
        app.Label_red_x.Text = Vector(112);
        app.Label_red_y.Text = Vector(113);
        app.Label_red_z.Text = Vector(114);
    end
end
% CHECK IF THE DEBUG CONTAINS CSV PART
if app.CSV_REAL_Drop_Down.Value == "Only CSV" || app.↵
CSV_REAL_Drop_Down.Value == "CSV and Real Time"
    app.path = app.path + ".csv";
    table_csv = array2table(app.csv_table, 'VariableNames', app.↵
column_indices);
    writetable(table_csv, app.path);
end
end
end
↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ACQUISITION RAW DATA

```

```

function Raw_data = acquisition_raw_data(app)
    Read = [app.EditField_SPS1_high.Value; app.EditField_SPS1_low.Value; app.
EditField_SPS2_high.Value; app.EditField_SPS2_low.Value; app.EditField_SPS3_high.
Value; app.EditField_SPS3_low.Value; app.EditField_SPS4_high.Value; app.
EditField_SPS4_low.Value; app.EditField_SPS5_high.Value; app.EditField_SPS5_low.Value;
app.EditField_SPS6_high.Value; app.EditField_SPS6_low.Value; app.EditField_SPS7_high.
Value; app.EditField_SPS7_low.Value; app.EditField_SPS8_high.Value; app.
EditField_SPS8_low.Value];
    Raw_data.Read = Read;
    Raw_data.Temp = [app.EditField_nom_T1.Value; app.EditField_Nom_T2.Value;
app.EditField_Red_T1.Value; app.EditField_Red_T2.Value];
    Raw_data.door = double(app.CheckBox_Door.Value);
    Raw_data.door_tr = double(app.CheckBox_Door_tr.Value);
end

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ACQUISITION RECONF PARAMS
function Reconf_params = acquisition_reconf_params(app)
    Reconf_params.K = [app.EditField_K1.Value; app.EditField_K2.Value; app.
EditField_K3.Value; app.EditField_K4.Value; app.EditField_K5.Value; app.EditField_K6.
Value; app.EditField_K7.Value; app.EditField_K8.Value];
    Reconf_params.algo = str2double(app.Box_Algorithm.Value);
    Reconf_params.pp_coarse.a_coarse = app.EditField_a_coarse_value.Value;
    Reconf_params.pp_coarse.b_coarse = app.EditField_b_coarse_value.Value;
    Reconf_params.pp_coarse.c_coarse = app.EditField_c_coarse_value.Value;
    Reconf_params.pp_coarse.A_coarse = app.EditField_A_coarse_value.Value;
    Reconf_params.pp_fine.a_fine = app.EditField_a_fine_value.Value;
    Reconf_params.pp_fine.b_fine = app.EditField_b_fine_value.Value;
    Reconf_params.pp_fine.c_fine = app.EditField_c_fine_value.Value;
    Reconf_params.pp_fine.A_fine = app.EditField_A_fine_value.Value;
    Reconf_params.pp_fine.delta_transv = app.EditField_delta_transv.Value;
    Reconf_params.linear.d_z = app.EditField_d_z.Value;
    Reconf_params.linear.d_y = app.EditField_d_y.Value;
    Reconf_params.long.H = app.EditField_H.Value;
    Reconf_params.long.K = app.EditField_K.Value;
    Reconf_params.long.L = app.EditField_L.Value;
    Reconf_params.long.J = app.EditField_J.Value;
    Reconf_params.long.delta_R = app.EditField_delta_R.Value;
    Reconf_params.thresholds.V_max_lat = app.EditField_V_max_lat.Value;
    Reconf_params.thresholds.V_max_long = app.EditField_V_max_long.Value;
    Reconf_params.thresholds.T_max = app.EditField_T_max.Value;
    Reconf_params.thresholds.delta_flag = app.EditField_delta_flag.Value;
end

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RANGE AND VALUE
function [Intermediate_value, Reconf_params] = range_and_value(app, Range,
params, number_cycle)
    plus_minus_range = str2double(Range);
    if app.RangeoptionforSPSDropDown.Value == "Absolute"
        Intermediate_value = (2 * plus_minus_range) / number_cycle;

```

```

        Reconf_params = params - plus_minus_range;
    else
        Intermediate_value = (params * plus_minus_range * 0.02) / ✓
number_cycle;
        Reconf_params = params - (params * plus_minus_range / 100);
    end
    if (app.MontecarloCheckBox.Value)
        Reconf_params = params;
    end
end

↙
%% =====
%% RANGE OR MONTECARLO VALUE
function param = range_or_montecarlo_value(app, params_input, intermediate, ~)
    if app.MontecarloCheckBox.Value
        r = str2double(app.Test_Field.Value) / 2 + 0.5;
        r = -r + (r + r) * rand();
        % r = str2double(app.Test_Field.Value);
        % r = (randi(r) - 1);
        param = params_input + intermediate * r;
    else
        param = params_input + intermediate * app.n_cycle;
    end
end

↙
%% =====
%% ASSIGN NAME 2 VALUES
function tab = assign_name_2_values(app, x)
    tab = zeros(3, app.column_number_csv);
    tab = string(tab);
    tab(:, :) = "";
    c = 0;
    for i=1:2:77
        c = c + 1;
        tab(2, i) = app.column_name(c) + " = " + x(2,i);
    end
    for i=1:2:51
        c = c + 1;
        tab(3, i) = app.column_name(c) + " = " + x(3,i);
    end
end

↙
%% =====
%% PLOT ERROR
function plot_error_X_Y_Z(app)
    array_nom_X = app.temp_table(:,2);
    array_nom_Y = app.temp_table(:,3);
    array_nom_Z = app.temp_table(:,4);
    array_red_X = app.temp_table(:,5);

```

```

array_red_Y = app.temp_table(:,6);
array_red_Z = app.temp_table(:,7);
array_nom_X = (array_nom_X - app.EditField_X_Expected.Value)';
array_nom_Y = (array_nom_Y - app.EditField_Y_Expected.Value)';
array_nom_Z = (array_nom_Z - app.EditField_Z_Expected.Value)';
array_red_X = (array_red_X - app.EditField_X_Expected.Value)';
array_red_Y = (array_red_Y - app.EditField_Y_Expected.Value)';
array_red_Z = (array_red_Z - app.EditField_Z_Expected.Value)';
if length(app.vector_check) == 1 && ismember(app.vector_check, -1) == 0
    tab = table2array(app.Full_Table.Data);
    x_arr = tab(:,app.vector_check(1));
    X_label = strrep(app.Full_Table.ColumnName(app.vector_check(1)), "_", ↵
" ");
else
    %% NUMBER OF TESTS
    x_arr = 1:str2double(app.Test_Field.Value);
    x_arr(end+1) = x_arr(end) + 1;
    X_label = "N° TEST";
end
%% X
[y_complex_nom, y_real_nom, x_complex_nom, x_real_nom] = ↵
real_complex_vector(app, array_nom_X, x_arr);
[y_complex_red, y_real_red, x_complex_red, x_real_red] = ↵
real_complex_vector(app, array_red_X, x_arr);
if app.MontecarloCheckBox.Value
    x_complex_nom = [];
    y_complex_nom = [];
    y_complex_red = [];
    x_complex_red = [];
end
plot(app.X_error_plot, x_complex_nom, y_complex_nom, 'm:d', x_real_nom, ↵
y_real_nom, 'r:d', x_complex_red, y_complex_red, 'c:*', x_real_red, y_real_red, 'b: ↵
*');

if isempty(x_complex_nom) % NO COMPLEX NOM
    if isempty(x_real_nom) % NO REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                msgbox("If the X graph legend says 'NOTHING' and there are ↵
present some lines, these lines refer to the previous graph", "Warning", "warn");
                legend(app.X_error_plot, 'NOTHING');
            else %YES REAL RED
                legend(app.X_error_plot, 'RED X REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.X_error_plot, 'RED X COMPLEX');
            else %YES REAL RED
                legend(app.X_error_plot, 'RED X COMPLEX', 'RED X REAL');
            end
        end
    else %YES REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.X_error_plot, 'NOM X REAL');
            end
        end
    end
end

```

```

        else %YES REAL RED
            legend(app.X_error_plot, 'NOM X REAL', 'RED X REAL');
        end
    else % YES COMPLEX RED
        if isempty(x_real_red) % NO REAL RED
            legend(app.X_error_plot, 'NOM X REAL', 'RED X COMPLEX');
        else % YES REAL RED
            legend(app.X_error_plot, 'NOM X REAL', 'RED X COMPLEX',
'RED X REAL');
        end
    end
end
else %YES COMPLEX REAL
    if isempty(x_real_nom) % NO REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX');
            else %YES REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'RED X REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'RED X
COMPLEX');
            else %YES REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'RED X COMPLEX',
'RED X REAL');
            end
        end
    else %YES REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'NOM X REAL');
            else %YES REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'NOM X REAL',
'RED X REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'NOM X REAL',
'RED X COMPLEX');
            else % YES REAL RED
                legend(app.X_error_plot, 'NOM X COMPLEX', 'NOM X REAL',
'RED X COMPLEX', 'RED X REAL');
            end
        end
    end
end
end

%% Y
[y_complex_nom, y_real_nom, x_complex_nom, x_real_nom] =
real_complex_vector(app, array_nom_Y, x_arr);
[y_complex_red, y_real_red, x_complex_red, x_real_red] =
real_complex_vector(app, array_red_Y, x_arr);
if app.MontecarloCheckBox.Value

```

```

        x_complex_nom = [];
        y_complex_nom = [];
        y_complex_red = [];
        x_complex_red = [];
    end
    plot(app.Y_error_plot, x_complex_nom, y_complex_nom, 'm:d', x_real_nom,
y_real_nom, 'r:d', x_complex_red, y_complex_red, 'c:*', x_real_red, y_real_red, 'b:
*');

    if isempty(x_complex_nom) % NO COMPLEX NOM
        if isempty(x_real_nom) % NO REAL NOM
            if isempty(x_complex_red) %NO COMPLEX RED
                if isempty(x_real_red) % NO REAL RED
                    msgbox("If the Y graph legend says 'NOTHING' and there are
present some lines, these lines refer to the previous graph", "Warning", "warn");
                    legend(app.Y_error_plot, 'NOTHING');
                else %YES REAL RED
                    legend(app.Y_error_plot, 'RED Y REAL');
                end
            else % YES COMPLEX RED
                if isempty(x_real_red) % NO REAL RED
                    legend(app.Y_error_plot, 'RED Y COMPLEX');
                else %YES REAL RED
                    legend(app.Y_error_plot, 'RED Y COMPLEX', 'RED Y REAL');
                end
            end
        else %YES REAL NOM
            if isempty(x_complex_red) %NO COMPLEX RED
                if isempty(x_real_red) % NO REAL RED
                    legend(app.Y_error_plot, 'NOM Y REAL');
                else %YES REAL RED
                    legend(app.Y_error_plot, 'NOM Y REAL', 'RED Y REAL');
                end
            else % YES COMPLEX RED
                if isempty(x_real_red) % NO REAL RED
                    legend(app.Y_error_plot, 'NOM Y REAL', 'RED Y COMPLEX');
                else % YES REAL RED
                    legend(app.Y_error_plot, 'NOM Y REAL', 'RED Y COMPLEX',
'RED Y REAL');
                end
            end
        end
    else %YES COMPLEX REAL
        if isempty(x_real_nom) % NO REAL NOM
            if isempty(x_complex_red) %NO COMPLEX RED
                if isempty(x_real_red) % NO REAL RED
                    legend(app.Y_error_plot, 'NOM Y COMPLEX');
                else %YES REAL RED
                    legend(app.Y_error_plot, 'NOM Y COMPLEX', 'RED Y REAL');
                end
            else % YES COMPLEX RED
                if isempty(x_real_red) % NO REAL RED
                    legend(app.Y_error_plot, 'NOM Y COMPLEX', 'RED Y
COMPLEX');
                else %YES REAL RED

```

```

        legend(app.Y_error_plot, 'NOM Y COMPLEX', 'RED Y COMPLEX', '
'RED Y REAL');
    end
end
else %YES REAL NOM
    if isempty(x_complex_red) %NO COMPLEX RED
        if isempty(x_real_red) % NO REAL RED
            legend(app.Y_error_plot, 'NOM Y COMPLEX', 'NOM Y REAL');
        else %YES REAL RED
            legend(app.Y_error_plot, 'NOM Y COMPLEX', 'NOM Y REAL',
'RED Y REAL');
        end
    else % YES COMPLEX RED
        if isempty(x_real_red) % NO REAL RED
            legend(app.Y_error_plot, 'NOM Y COMPLEX', 'NOM Y REAL',
'RED Y COMPLEX');
        else % YES REAL RED
            legend(app.Y_error_plot, 'NOM Y COMPLEX', 'NOM Y REAL',
'RED Y COMPLEX', 'RED Y REAL');
        end
    end
end
end

%% Z
[y_complex_nom, y_real_nom, x_complex_nom, x_real_nom] =
real_complex_vector(app, array_nom_Z, x_arr);
[y_complex_red, y_real_red, x_complex_red, x_real_red] =
real_complex_vector(app, array_red_Z, x_arr);
if app.MontecarloCheckBox.Value
    x_complex_nom = [];
    y_complex_nom = [];
    y_complex_red = [];
    x_complex_red = [];
end
plot(app.Z_error_plot, x_complex_nom, y_complex_nom, 'm:d', x_real_nom,
y_real_nom, 'r:d', x_complex_red, y_complex_red, 'c:*', x_real_red, y_real_red, 'b:
*');

if isempty(x_complex_nom) % NO COMPLEX NOM
    if isempty(x_real_nom) % NO REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                msgbox("If the Z graph legend says 'NOTHING' and there are
present some lines, these lines refer to the previous graph", "Warning", "warn");
                legend(app.Z_error_plot, 'NOTHING');
            else %YES REAL RED
                legend(app.Z_error_plot, 'RED Z REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'RED Z COMPLEX');
            else %YES REAL RED
                legend(app.Z_error_plot, 'RED Z COMPLEX', 'RED Z REAL');
            end
        end
    end
end

```

```

        end
    else %YES REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'NOM Z REAL');
            else %YES REAL RED
                legend(app.Z_error_plot, 'NOM Z REAL', 'RED Z REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'NOM Z REAL', 'RED Z COMPLEX');
            else % YES REAL RED
                legend(app.Z_error_plot, 'NOM Z REAL', 'RED Z COMPLEX',
'RED Z REAL');
            end
        end
    end
else %YES COMPLEX REAL
    if isempty(x_real_nom) % NO REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX');
            else %YES REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'RED Z REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'RED Z
COMPLEX');
            else %YES REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'RED Z COMPLEX',
'RED Z REAL');
            end
        end
    else %YES REAL NOM
        if isempty(x_complex_red) %NO COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'NOM Z REAL');
            else %YES REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'NOM Z REAL',
'RED Z REAL');
            end
        else % YES COMPLEX RED
            if isempty(x_real_red) % NO REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'NOM Z REAL',
'RED Z COMPLEX');
            else % YES REAL RED
                legend(app.Z_error_plot, 'NOM Z COMPLEX', 'NOM Z REAL',
'RED Z COMPLEX', 'RED Z REAL');
            end
        end
    end
end
%% LABEL
app.X_error_plot.XLabel.String = X_label;

```

```

app.X_error_plot.YLabel.String = "X (mm)";
app.Y_error_plot.XLabel.String = X_label;
app.Y_error_plot.YLabel.String = "Y (mm)";
app.Z_error_plot.XLabel.String = X_label;
app.Z_error_plot.YLabel.String = "Z (mm)";
end

```

```

% =====
% LAST 3 ROWS CSV
function x = last_3_rows_CSV(app, Vector)
x = zeros(3,app.column_number_csv);
x = string(x);
x(:, :) = "";
x(2,1) = app.EditField_R_sps.Value;
x(2,3) = Vector(43);
x(2,5) = Vector(40);
x(2,7) = Vector(41);
x(2,9) = Vector(42);
x(2,11) = Vector(48);
x(2,13) = Vector(45);
x(2,15) = Vector(46);
x(2,17) = Vector(47);
x(2,19) = Vector(54);
x(2,21) = Vector(55);
x(2,23) = Vector(56);
x(2,25) = Vector(57);
x(2,27) = Vector(58);
x(2,29) = Vector(29);
x(2,31) = Vector(30);
x(2,33) = Vector(31);
x(2,35) = Vector(32);
x(2,37) = Vector(33);
x(2,39) = Vector(34);
x(2,41) = Vector(35);
x(2,43) = Vector(36);
x(2,45) = Vector(51);
x(2,47) = Vector(52);
x(2,49) = Vector(65);
x(2,51) = Vector(66);
x(2,53) = Vector(67);
x(2,55) = Vector(68);
x(2,57) = Vector(69);
x(2,59) = Vector(70);
x(2,61) = Vector(71);
x(2,63) = Vector(72);
x(2,65) = Vector(79);
x(2,67) = Vector(80);
x(2,69) = Vector(94);
x(2,71) = Vector(95);
x(2,73) = Vector(97);
x(2,75) = Vector(98);
x(2,77) = Vector(128);
x(3,1) = Vector(103);

```

```

x(3,3) = Vector(104);
x(3,5) = Vector(105);
x(3,7) = Vector(112);
x(3,9) = Vector(113);
x(3,11) = Vector(114);
x(3,13) = Vector(100);
x(3,15) = Vector(101);
x(3,17) = Vector(102);
x(3,19) = Vector(107);
x(3,21) = Vector(109);
x(3,23) = Vector(110);
x(3,25) = Vector(111);
x(3,27) = Vector(116);
x(3,29) = Vector(118);
x(3,31) = Vector(119);
x(3,33) = Vector(120);
x(3,35) = Vector(121);
x(3,37) = Vector(122);
x(3,39) = Vector(123);
x(3,41) = Vector(124);
x(3,43) = Vector(125);
x(3,45) = Vector(126);
x(3,47) = Vector(130);
x(3,49) = Vector(132);
x(3,51) = Vector(134);
%x = array2table(x, 'VariableNames', app.column_indices);
end

```

```

% =====
% get_row_for_real_time_table
function real_time_table = get_row_for_real_time_table(app, z)
% CREATE TABLE FOR THE REAL TIME
real_time_table = strings(1,app.column_number_real_time);
real_time_table(1,1) = app.filename;
real_time_table(1,2) = z(3,1);
real_time_table(1,3) = z(3,3);
real_time_table(1,4) = z(3,5);
real_time_table(1,5) = z(3,7);
real_time_table(1,6) = z(3,9);
real_time_table(1,7) = z(3,11);
c = 0;
for i=1:2:75
    c = c + 1;
    real_time_table(1, 7 + c) = z(2,i);
end
real_time_table(1,46) = z(3,13);
real_time_table(1,47) = z(3,15);
real_time_table(1,48) = z(3,17);
real_time_table(1,49) = z(3,19);
real_time_table(1,50) = z(3,21);
real_time_table(1,51) = z(3,23);
real_time_table(1,52) = z(3,25);
real_time_table(1,53) = z(3,27);

```

```

real_time_table(1,54) = z(3,29);
real_time_table(1,55) = z(3,31);
real_time_table(1,56) = z(3,33);
real_time_table(1,57) = z(3,35);
real_time_table(1,58) = z(3,37);
real_time_table(1,59) = z(3,39);
real_time_table(1,60) = z(3,41);
real_time_table(1,61) = z(3,43);
real_time_table(1,62) = z(3,45);
real_time_table(1,63) = z(2,77);
real_time_table(1,64) = z(3,47);
real_time_table(1,65) = z(3,49);
real_time_table(1,66) = z(3,51);
end

```

```

↙
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ↘
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ↘
% start_param_chosen
function initial_value = start_param_chosen(app, name, vect)
    switch name
    case "a coarse"
        initial_value = app.initial_reconf_params.pp_coarse.a_coarse;
    case "b coarse"
        initial_value = app.initial_reconf_params.pp_coarse.b_coarse;
    case "c coarse"
        initial_value = app.initial_reconf_params.pp_coarse.c_coarse;
    case "A coarse"
        initial_value = app.initial_reconf_params.pp_coarse.A_coarse;
    case "a fine"
        initial_value = app.initial_reconf_params.pp_fine.a_fine;
    case "b fine"
        initial_value = app.initial_reconf_params.pp_fine.b_fine;
    case "c fine"
        initial_value = app.initial_reconf_params.pp_fine.c_fine;
    case "d z"
        initial_value = app.initial_reconf_params.linear.d_z;
    case "d y"
        initial_value = app.initial_reconf_params.linear.d_y;
    case "H"
        initial_value = app.initial_reconf_params.long.H;
    case "K"
        initial_value = app.initial_reconf_params.long.K;
    case "L"
        initial_value = app.initial_reconf_params.long.L;
    case "J"
        initial_value = app.initial_reconf_params.long.J;
    case "delta r"
        initial_value = app.initial_reconf_params.long.delta_r;
    otherwise
        initial_value = vect(int64(end/2));
    end
end

```

↙

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% REAL COMPLEX VECTOR
function [y_complex, y_real, x_complex, x_real] = real_complex_vector(~,
y_axis_value, x_axis_value)
    y_complex = zeros(size(y_axis_value));
    y_real = zeros(size(y_axis_value));
    x_complex = zeros(size(y_axis_value));
    x_real = zeros(size(y_axis_value));
    c1 = 0;
    c2 = 0;
    for i = 1:length(y_axis_value)
        if ~isreal(y_axis_value(i))
            c1 = c1 + 1;
            y_complex(c1) = y_axis_value(i);
            x_complex(c1) = x_axis_value(i);
        else
            c2 = c2 + 1;
            y_real(c2) = y_axis_value(i);
            x_real(c2) = x_axis_value(i);
        end
    end
    y_complex = y_complex(1:c1);
    y_real = y_real(1:c2);
    x_complex = x_complex(1:c1);
    x_real = x_real(1:c2);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GET SPS FROM POSITION
function Get_SPS_From_Position(app)
    if (app.CSV_Count.Value == "NaN" || app.CSV_X.Value == "NaN" || app.CSV_Curr.Value == "NaN")
        msgbox("Select one Count CSV and one X CSV", "Error", "error");
    else
        msgbox("Simulation started, do not change any values until the next
popup", "XYZ-TO-SPS", "warn");
        app.SPS1EditField.Value = 0;
        app.SPS2EditField.Value = 0;
        app.SPS3EditField.Value = 0;
        app.SPS4EditField.Value = 0;
        app.SPS5EditField.Value = 0;
        app.SPS6EditField.Value = 0;
        app.SPS7EditField.Value = 0;
        app.SPS8EditField.Value = 0;
        app.STARTSIMULATIONXYZTOSPSButton.Enable = "off";
        app.EXPORTDNTOSPSTOXYZButton.Enable = "on";
        sps_rad = app.SPSRadiusEditField.Value;
        sps_theta = [app.Theta1EditField.Value, app.Theta2EditField.Value,
app.Theta3EditField.Value, app.Theta4EditField.Value, app.Theta5EditField.Value, app.
Theta6EditField.Value, app.Theta7EditField.Value, app.Theta8EditField.Value];
        [~, name, ~] = fileparts(app.CSV_Count.Value);
        name(1:19) = [];
    end
end

```

```

ISD = app.ISDEditField.Value;
app.XEditField.Value = ISD - str2double(name) * 10^3;
coordinate.X = str2double(name); %app.XEditField.Value * 10^-3;
coordinate.Y = (app.YEditField.Value * 10^-3);
coordinate.Z = (app.ZEditField.Value * 10^-3);
counts_sim_x0 = table2array(readtable(app.CSV_Count.Value));
curr_sim_x0 = table2array(readtable(app.CSV_Curr.Value));
x = table2array(readtable(app.CSV_X.Value));
% Determine the coordinates of SPS displacement
sps_coord = zeros(2, 8);
sps_dists = zeros(8);
for i=[1,2,3,4,5,6,7,8]
    sps_zcoords_now = sps_rad*cos(sps_theta(i)) + coordinate.Z;
    sps_ycoords_now = sps_rad*sin(sps_theta(i)) + coordinate.Y;
    sps_coord(1,i) = sps_zcoords_now;
    sps_coord(2,i) = sps_ycoords_now;
    sps_dists(i) = sqrt(sps_ycoords_now^2 + sps_zcoords_now^2);
end
% Reflect those displacements in the first quadrant
sps_coords_refl = abs(sps_coord);
% Find positions on the matrix
sps_pos = zeros(2, 8);
for i=[1,2,3,4,5,6,7,8]
    for j=[1,2]
        sps_pos(j,i) = fix(find(abs(sps_coords_refl(j,i)-x) == min(abs(
(sps_coords_refl(j,i)-x))))); %+ 0.5
    end
end
% Extract the SPS signal in output
sps_signal = zeros(2, 8);
sps_curr = zeros(2, 8);

for i=[1,2,3,4,5,6,7,8]
    sps_signal(i) = counts_sim_x0(sps_pos(1,i), sps_pos(2,i)); %+1
    sps_curr(i) = curr_sim_x0(sps_pos(1,i), sps_pos(2,i)); %+1
end
%% Alessandro's Algorithm representation of SPS
app.SPS1EditField.Value = sps_signal(1);
app.SPS2EditField.Value = sps_signal(2);
app.SPS3EditField.Value = sps_signal(3);
app.SPS4EditField.Value = sps_signal(4);
app.SPS5EditField.Value = sps_signal(5);
app.SPS6EditField.Value = sps_signal(6);
app.SPS7EditField.Value = sps_signal(7);
app.SPS8EditField.Value = sps_signal(8);

app.SPS1_Curr_EditField.Value = sps_curr(1);
app.SPS2_Curr_EditField.Value = sps_curr(2);
app.SPS3_Curr_EditField.Value = sps_curr(3);
app.SPS4_Curr_EditField.Value = sps_curr(4);
app.SPS5_Curr_EditField.Value = sps_curr(5);
app.SPS6_Curr_EditField.Value = sps_curr(6);
app.SPS7_Curr_EditField.Value = sps_curr(7);
app.SPS8_Curr_EditField.Value = sps_curr(8);

```

```

        % PLOT
        sps_coord(1,9) = sps_coord(1,1);
        sps_coord(2,9) = sps_coord(2,1);
        sps_theta(9) = sps_theta(1);
        angle = 0:0.1:360;
        radius = 38.5 * 1e-3 + 0.0041 * (app.ISDEditField.Value - app.
XEditField.Value);
        circx = radius*cosd(angle);
        circy = radius*sind(angle);
        plot(app.PLOT_SPS, sps_rad*cos(sps_theta),sps_rad*sin(sps_theta),"b:*",
sps_coord(1,:), sps_coord(2,:), "r:d", circx, circy, "k-");
        legend(app.PLOT_SPS, 'SPS INITIAL', "SPS TRASLATED", "UMBRA");
        app.PLOT_SPS.XLabel.String = "Z (mm)";
        app.PLOT_SPS.YLabel.String = "Y (mm)";

        app.STARTSIMULATIONXYZTOSPSButton.Enable = "on";
        app.Json_filename.Enable = "on";
        app.EXPORTINJSONButton.Enable = "on";
        app.SAVECURRTOCSVButton.Enable = "on";
        if app.CSV_fit_param.Value ~= "NaN"
            app.fit_param = table2array(readtable(app.CSV_fit_param.Value));
            app.ExportfitparamsCheckBox.Enable = "on";
        end
        msgbox("Operation completed", "XYZ-TO-SPS");
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GENERAZIONE CIOTOLA
function Generazione_Ciotola_unidim(app) %RICORDA DI SOSTITUIRE LA Z NELLA X E
VICEVERSA
    if ((mod(2*app.StartEditField.Value, app.StepmeterEditField.Value) ~= 0 &&
app.ChooseXarrayrangeButtonGroup.SelectedObject.Text == "Specular at 0") || (mod
(2*app.StartEditField_2.Value, app.StepmeterEditField_2.Value) ~= 0 && app.
ChooseXarrayrangeButtonGroup.SelectedObject.Text == "Specular at 0"))
        msgbox("(Start value * 2) must be a multiple of step value for a
specular rappresentation", "Error", "error");
        return;
    end
    dpin_mm = 2.5d0; % diametro del pinhole (mm)
    apin_cm2 = pi*(dpin_mm/1.d1/2.d0)^2.; % area del pinhole (cm^2.)
    %sps_filter_reduction = 1.6498171;
    dn_stad1 = 0.014532180059524; % (microAmpere/DN)
    dn_stad2 = 0.002906436011905; % (microAmpere/DN)

    % Temperatura degli SPS
    sps_temp = 2; % NOTA: 0 per -20°C, 1 per 0°C, 2 per +20°C, 3 per +40°C e 4
per 60°C

    % Errore dato dagli SPS
    %sps_error = 1.22d0; % (mV) accuracy of ADC
    %sps_gain = 1.d5;

```

```

% sps_size1 = (0.001)^2.d0 * 1.d4; % (cm^2) quadrato di lato 1 mm
% sps_size2 = (0.003)^2.d0 * 1.d4; % (cm^2) quadrato di lato 3 mm
% sps_size3 = (0.006)^2.d0 * 1.d4; % (cm^2) quadrato di lato 6 mm

% Costanti varie
au = 149.597871d6; % 1 AU (km)
% au_peri = 147.09806d6;
% au_aphe = 152.09768d6;
r_sun = 6.96342d5; % 1 R_sun (km)

% Array ausiliari per la routine disc_sunlight.pro
r = 1.d0;
arr1 = 0:1000;
xsun = 2.d0 * r * arr1 / 1000.d0 -r;
ysun = 2.d0 * r * arr1 / 1000.d0 -r;
xx = repmat(xsun, 1001, 1); % in IDL is REBIN
yy = repmat(ysun, 1001, 1);
ht0 = sqrt(xx.^2 + yy.^2);
theta_now = zeros(1001,1001);
sundisk = zeros(1001,1001);

% 'Starting iterations on longitudinal distance'
if app.ChangeXarrayrangeButtonGroup.SelectedObject.Text == "Specular at 0"
    dz_occ_array = -(abs(app.StartEditField.Value) : abs(app.
StepmeterEditField.Value) : (abs(app.StartEditField.Value)); % (m)
else
    dz_occ_array = app.StartEditField_2.Value : abs(app.
StepmeterEditField_2.Value) : app.End_Edit_Field.Value; % (m)
end
dz_occ_array(dz_occ_array == 0) = [];
dz_occ_array(end+1) = 0;
length_dz_occ_array = length(dz_occ_array);

if length_dz_occ_array ~= 1
    app.Gauge.Limits = [1, length_dz_occ_array];
    app.Gauge.MajorTicks = 1:length_dz_occ_array;
end
for kk=1:length_dz_occ_array % Iterazione su tre spostamenti longitudinali
(-100 + 100)
    %t0 = datestr(now, 'dd-mm-yyyy HH:MM:SS');

    app.Gauge.Value = kk;
    app.Iteration.Text = "" + string(kk) + " / " + string
(length_dz_occ_array);

    pause on
    pause(1);
    pause off
    % Spostamento longitudinale dell'occultatore rispetto alla ISD
    dz_occ = dz_occ_array(kk);

    % Proprietà geometriche dell'occultatore
    r_occ = 0.71d0; % +0.005d-3 ; dimensioni fisiche raggio
dell'occultatore (m)

```

```

r_pup = 0.025d0; % dimensioni fisiche raggio della pupilla (m)
r_sps = 0.055d0; % dimensioni fisiche raggio su cui si trovano gli SPS✓
(m)
d_occ = (r_occ - r_pup) / tan(1.02 * atan(r_sun / au)); % distanza✓
dell'occultatore dal coronografo (m)
alpha_occ = atan(r_occ / d_occ); % angolo di apertura che sottende✓
l'occultatore
alpha_sun = atan(r_sun / au); % angolo di apertura che sottende il✓
sole
r_sun_proj = d_occ * tan(alpha_sun);
r_occ_proj = (d_occ + dz_occ) * tan(alpha_occ);
%dx_max_fullsun = 2 * r_sun_proj + (r_occ_proj - r_sun_proj); %✓
massimo spostamento trasverso dell'occultatore
% d_occ = 144.27567d0; -50.d-3
t_occ = 0.d0; %; 1.d0; 0.5d0 ; tilt dell'occultatore rispetto all'asse✓
ottico del coronografo (gradi, tilt max 0.5)
dx_min_occ = 50.d-6; % MIN delta x occultatore (m, max 0.01)
goalbox_max = double(double(50.d-3 + r_sps) * sqrt(2.d0)); % massimo✓
spostamento trasverso entro la goal box (m)
dx_max_occ = double((50.5d-3 + r_sps) * sqrt(2.d0)); %dx_max_fullsun;✓
70.d-3; 1.3d0; 0.155d0 * sqrt(2.d0) ; 0.1d0 * sqrt(2.d0) ; MAX delta x occultatore (m, max✓
0.01)

% Diminuisce la risoluzione della mappa
dx_res = 1.d0 / 10.d0;
dx_min_occ = dx_min_occ / dx_res;

% Numero di punti con cui viene creata la matrice di illuminazione
% della penombra. Per accelerare il calcolo si usa la risoluzione
% massima impostata entro 150 mm e poi un decimo della risoluzione
% massima impostata.
if dx_max_occ < goalbox_max
    n_dx = int64(dx_max_occ/dx_min_occ+0.5);
    transversal_shift = zeros(1, n_dx);
    for i = 0:n_dx - 1
        transversal_shift(i+1) = double(i) * dx_min_occ + 38.d-3;
    end
else
    n_dx_hires = fix(goalbox_max * sqrt(2.) / dx_min_occ); % + 0.5
    n_dx_lores = fix((dx_max_occ - goalbox_max) * sqrt(2.) / dx_min_occ✓
/ 10. + 0.5); % + 0.5
    n_dx = n_dx_hires + n_dx_lores;
    transversal_shift = zeros(1, n_dx);
    for i = 0:n_dx_hires-1
        transversal_shift(i+1) = double(i) * dx_min_occ + 38.d-3;
    end
    for i = n_dx_hires:n_dx-1
        transversal_shift(i+1) = goalbox_max * sqrt(2.) + 10.d0 * double✓
(i - n_dx_hires) * dx_min_occ + 38.d-3;
    end
end

%transversal_shift(n_dx-1)=dx_max_fullsun

% Trova la posizione degli SPS e del bordo dell'ombra

```

```

%min_val = min(abs(transversal_shift - r_sps));
%arr = abs(transversal_shift - r_sps);
%sps_pos = find(arr==min_val(1,1));

%min_val = min(abs(transversal_shift - 39.d-3));
%arr = abs(transversal_shift - 39.d-3);
%umbra_pos = find(arr==min_val(1,1));

%goto,fine

proba_inlight_nolimb_lambda = zeros(n_dx,17); % illuminazione della
penombra, prima dell'attraversamento vetro e filtro, caso disco piatto con luminosità
uniforme pari a 1
proba_inlight_limb_lambda = zeros(n_dx,17); % illuminazione della
penombra, prima dell'attraversamento vetro e filtro, caso con limb darkening
proba_inlight_spot_lambda = zeros(n_dx,17); % illuminazione della
penombra, prima dell'attraversamento vetro e filtro, caso con limb darkening e sunspot
al lembo
proba_light_nolimb_lambda = zeros(n_dx,17); % illuminazione della
penombra, dopo attraversamento vetro e filtro, caso disco piatto con luminosità
uniforme pari a 1
proba_light_limb_lambda = zeros(n_dx,17); % illuminazione della
penombra, dopo attraversamento vetro e filtro, caso con limb darkening
proba_light_spot_lambda = zeros(n_dx,17); % illuminazione della
penombra, dopo attraversamento vetro e filtro, caso con limb darkening e sunspot al
lembo
proba_denscurr_nolimb_lambda = zeros(n_dx,17); % corrente della
penombra, caso disco piatto con luminosità uniforme pari a 1
proba_denscurr_limb_lambda = zeros(n_dx,17); % corrente della
penombra, caso con limb darkening
proba_denscurr_spot_lambda = zeros(n_dx,17); % corrente della
penombra, caso con limb darkening e sunspot al lembo

area_arr = zeros(n_dx,17);
limbdark_avg_now = zeros(n_dx,17);
sunlight_now = zeros(n_dx,17);

% Parte di codice che legge in input e interpola alla lambda_now:
% 1) solar spectral irradiance
% 2) SPS responsivity
% 3) limb darkening coefficients
% 4) ND filter transmittance
% 5) glass transmittance
% 6) coating transmittance
% per il calcolo della curva di penombra alla singola lambda
%

% Recover the Spectral solar Irradiance data as measured with
composite SORCE data
% retrieved from http://lasp.colorado.
edu/lisird/sorce/sorce_ssi/index.html
s_file = app.SSI_solarmin_CSV.Value;
ss_min = table2array(readtable(s_file)); % (Watt /m^2 /nm)
%ssi_day = ss_min(1,:);
ssi_wavl = ss_min(2,:);

```

```

ssi_irra = ss_min(3,:);
ssi_irra = ssi_irra / 10.d0; % (mW / cm^2 / nm)

% Recover the SPS responsivity curve
s_file = app.resp_t_dep_m20degC_to_60degCCSV.Value;
Responsivity1_temp_dependence_m20degC_to_60degC = table2array(
(readtable(s_file)));
sps_wavl1 = Responsivity1_temp_dependence_m20degC_to_60degC(1,:);
sps_resp1 = Responsivity1_temp_dependence_m20degC_to_60degC(2,:);
sps_wavl2 = Responsivity1_temp_dependence_m20degC_to_60degC(3,:);
sps_resp2 = Responsivity1_temp_dependence_m20degC_to_60degC(4,:);
sps_wavl3 = Responsivity1_temp_dependence_m20degC_to_60degC(5,:);
sps_resp3 = Responsivity1_temp_dependence_m20degC_to_60degC(6,:);
switch sps_temp
case 0
    sps_wavl = sps_wavl1*1.d9; % (nm)
    sps_resp = sps_resp1; % (Ampere/Watt)
case 1
    sps_wavl = sps_wavl2*1.d9; % (nm)
    sps_resp=sps_resp2; % (Ampere/Watt)
case 2
    sps_wavl = sps_wavl3*1.d9; % (nm)
    sps_resp=sps_resp3; % (Ampere/Watt)
case 3
    sps_wavl = sps_wavl4*1.d9; % (nm)
    sps_resp=sps_resp4; % (Ampere/Watt)
case 4
    sps_wavl = sps_wavl5*1.d9; % (nm)
    sps_resp=sps_resp5; % (Ampere/Watt)
end

% Recover the limb darkening coefficients curve given
% by Allen (2000), pp. 356-357, Table 14.17.
s_file = app.Ld_coefficients_CSV.Value;
ld_coefficients = table2array(readtable(s_file));
ld_wav = ld_coefficients(1,:);
ld_u = ld_coefficients(2,:);
ld_v = ld_coefficients(3,:);
ld_wav = ld_wav*1.e3; % (nm)

% Recover the ND filter transmittance curve
ndfilter_trans = 1.d-2 .* sps_wavl ./ sps_wavl; % (%)

% Recover the glass transmittance curve
s_file = app.Glass_transmittance_CSV.Value;
glass_transmittance = table2array(readtable(s_file));
glass_wav = glass_transmittance(1,:);
glass_trans = glass_transmittance(2, :);

% Recover the coating transmittance curve
s_file = app.Coating_transmittance_CSV.Value;
Coating_transmittance = table2array(readtable(s_file));
coating_wav = Coating_transmittance(1,:);
coating_trans = Coating_transmittance(2,:);
coating_trans = coating_trans / 100;

```

```

% Ricostruisce la curva di coating con delta lambda costante
nwav_coating = fix((max(coating_wav) - min(coating_wav)) / 0.1);
arr1 = 0:nwav_coating-1;
coating_wav2 = arr1 * 0.1 + min(coating_wav);
coating_trans2 = interp1(coating_wav,coating_trans,coating_wav2);
coating_trans3 = coating_trans2;

% Prende la parte di curva da 610 a 640 nm e la ricopia tra 640 e 670
nm
    coating_wav_cut1 = find(abs(coating_wav2-600.) == min(abs(
(coating_wav2-600.)));
    coating_wav_cut2 = find(abs(coating_wav2-650.) == min(abs(
(coating_wav2-650.)));
    coating_wav_cut3 = find(abs(coating_wav2-623.) == min(abs(
(coating_wav2-623.)));
    coating_wav_cut4 = find(abs(coating_wav2-673.) == min(abs(
(coating_wav2-673.)));
    coating_trans3(coating_wav_cut3:coating_wav_cut4) = coating_trans2(
(coating_wav_cut1:coating_wav_cut2));

%lambdas=dindgen(41)*10.d0+300.d0
arr = 1:17;
lambdas = (arr - 1) .* 10.d0 + 500.d0; %deve partire da 0

ssi_irra_now = zeros(1, 17);
sps_resp_now = zeros(1, 17);
ndfilter_trans_now = zeros(1, 17);
glass_trans_now = zeros(1, 17);
coating_trans_now = zeros(1, 17);

%lambda0 = 200.;
%lambda1 = 1100.;

%min_val = min(abs(sps_wav1-lambda0));
%arr = abs(sps_wav1-lambda0);
%lambda0_pos = find(arr==min_val(1,1));
%min_val = min(abs(sps_wav1-lambda1));
%arr = abs(sps_wav1-lambda1);
%lambda1_pos = find(arr==min_val(1,1));

%print,total(interp1(ssi_irra,ssi_wav1,sps_wav1(lambda0_pos:
lambda1_pos))*sps_resp(lambda0_pos:lambda1_pos),1)*10. ; (il fattore 10 nm è il
Delta_lambda)
    %print,total(interp1(ssi_irra,ssi_wav1,sps_wav1(lambda0_pos:
lambda1_pos))*sps_resp(lambda0_pos:lambda1_pos)*$
    %   interp1(glass_trans,glass_wav,sps_wav1(lambda0_pos:lambda1_pos))*$
    %   interp1(coating_trans3, coating_wav2,sps_wav1(lambda0_pos:
lambda1_pos)),1)*10.; (il fattore 10 nm è il Delta_lambda)

%t1=systime(/seconds)

%print,'Starting iterations on lambdas after',t1-t0,' seconds'

```

```

for h= 1:16 % iterazione sulle lambda
    % h=0

    app.Gauge_2.Value = h;
    app.Iteration_2.Text = "" + string(h) + " / 16";
    pause on
    pause(1);
    pause off

    lambda_now = lambdas(h); % Lunghezza d'onda (nm)

    ssi_irra_now(h) = interp1(ssi_wavl,ssi_irra,lambda_now); % (mW✓
/cm^2 /nm)

    sps_resp_now(h) = interp1(sps_wavl,sps_resp,lambda_now); % (A/W)
    ld_u_now = interp1(ld_wav, ld_u, lambda_now); %0.827051d0;
    ld_v_now = interp1(ld_wav, ld_v, lambda_now); %-0.184326d0;
    ndfilter_trans_now(h) = interp1(sps_wavl, ndfilter_trans,✓
lambda_now);

    glass_trans_now(h) = interp1(glass_wav, glass_trans, lambda_now);
    coating_trans_now(h) = interp1(coating_wav2, coating_trans3,✓
lambda_now);

    if lambda_now < min(glass_wav)
        glass_trans_now(h) = glass_trans(1);
    end
    if lambda_now < min(coating_wav2)
        coating_trans_now(h) = coating_trans3(1);
    end
    if coating_trans_now(h) == 0.
        coating_trans_now(h) = 0.0001;
    end

    for i = 1:n_dx % iterazione sugli spostamenti spaziali trasversi

        dx_occ_now = transversal_shift(i); %dx_max_fullsun;

        % Proprietà dell'ellisse corrispondente alla proiezione
        % dell'occultatore sul piano
        dx_occ_proj = dx_occ_now * (d_occ / (d_occ + dz_occ));
        dy_occ_proj = 0.d0;

        x0 = dx_occ_proj / r_sun_proj;
        y0 = dy_occ_proj / r_sun_proj;
        a = cos(t_occ / 57.2958) * r_occ_proj / r_sun_proj; %radeg =✓
57.2958

        b = r_occ_proj / r_sun_proj;

        % Trova i punti di intersezione tra il cerchio
        % del disco e l'ellisse dell'occultatore.
        % NOTA: considerati i valori di cui sopra per
        % il dx_max, dy_max e per il tilt max, ci si
        % trova nel caso in cui cerchio ed ellisse hanno
        % al più 2 punti di intersezione e mai 4 punti.
        % Tuttavia, trovare la posizione di questi punti

```

```

% richiede sempre e comunque di trovare le radici
% di un polinomio di quarto grado, cosa che viene
% fatta dalla routine ellipse_circle_sol per via
% numerica. Non è possibile trovare soluzione
% analitica.

[sol_x, sol_y, ~] = ellipse_circle_sol(app, x0, y0, r, a, b);

%t2=systime(/seconds)

if (length(sol_x) < 2 && length(sol_y) < 2) ~= 1
    % Imposta il valore della risposta in corrente
    % attesa dagli SPS tenendo conto della loro
    % curva di responsivity (valore stimato con
    % la routine sps_irradiance.pro).
    %sps_avg_resp=33.9619d0 % risposta dagli SPS (mA/cm^2)
    %sps_avg_resp_temp=[28.0577d0,28.6514d0,29.5885d0,30.0020d0,30.6538d0] % calcolati con sps_irradiance.pro al variare della temperature degli SPS
    sps_avg_resp_now = ssi_irra_now(h) * sps_resp_now(h) *
    glass_trans_now(h) * coating_trans_now(h); % risposta dagli SPS alla lambda considerata
    (mA/cm^2/nm)
    sps_avg_irra_now = ssi_irra_now(h) * glass_trans_now(h) *
    coating_trans_now(h); % illuminazione media nel passa-banda degli SPS (mW/cm^2/nm)

    %
    *****
    % F. LANDINI'S TEST (14/12/2015: caso solo a lambda = 550
    nm,
    % no SPS responsivity function)
    %sps_avg_irra=1.89230 % questa è la TSI a 550 nm (mW/cm^2)
    %sps_avg_resp=1.89230 % assumendo che la SPS response sia
    uguale a 1
    %
    *****
    %
    % Quindi calcola l'area della parte di disco
    % che emerge dietro l'occultatore, data in
    % frazione dell'area di disco. Quest'area
    % corrisponde all'illuminazione ricevuta dal
    % pixel (i,j) assunto che l'illuminazione
    % data dalla parte di disco visibile sia uniforme
    % e pari a 1. Per passare ad unità fisiche (fotoni)
    % basterà quindi moltiplicare la quantità ottenuta
    % per la radianza emessa da tutto il disco integrato.

    area = ellipse_circle_area(app, x0, y0, r, a, b, sol_x,
sol_y);

    proba_denscurr_nolimb_lambda(i,h) = area*sps_avg_resp_now;
    proba_inlight_nolimb_lambda(i,h) = area*sps_avg_irra_now;
    proba_light_nolimb_lambda(i,h) = area*ssi_irra_now(h);
    area_arr(i,h) = area;

    %t3=systime(/seconds)

```

```

% Trova poi l'intensità media per quella frazione
% di disco tenendo conto però del limb darkening al
% bordo. L'area di prima viene quindi moltiplicata
% per un numero < 1 che dà l'illuminazione data data
% dalla parte di disco visibile rispetto all'illuminazione
% del centro del disco (sempre pari a 1). Per passare
% ad unità fisiche (fotoni) basta anche qui moltiplicare
% la quantità ottenuta per la radianza emessa da tutto il
% disco integrato dividendo poi per il valor medio della
% funzione di limb darkening su tutto il disco (vedi
note).

[theta_now1, sundisk1, sunlight, limbdark_avg] = \
disc_sunlight_lambda(app, x0, y0, r, a, b, ht0, xx, yy, ld_u_now, ld_v_now, theta_now, \
sundisk); % forse mettere circle_now, ellipse_now, visible, theta,
theta_now = theta_now1;
sundisk = sundisk1;
%limbdark_avg = 0.82159140d0 % valore calcolato con la \
routine limbdark_average.pro
%limbdark_avg_temp=[0.81417125d0,0.81551092d0,0.81652035\
d0,0.81765500d0,0.81898317d0] % valori calcolati con la routine limbdark_average.pro \
per varie temperature degli SPS
%limbdark_avg_temp_blue=[0.84423026d0,0.84423026d0, \
0.84423026d0,0.84423026d0,0.84423026] % valori calcolati con la routine \
limbdark_average.pro per varie temperature degli SPS tagliando la parte rossa (lambda \
> 680 nm)

%limbdark_avg=limbdark_avg_temp(sps_temp)
%limbdark_avg_blue=limbdark_avg_temp(sps_temp)
%
*****
%limbdark_avg = 0.80504507 % F. LANDINI'S TEST (caso solo \
lambda 550 nm)
%
*****
limbdark_avg_now(i,h) = limbdark_avg;
sunlight_now(i,h) = sunlight;
proba_denscurr_limb_lambda(i,h) = area * sps_avg_resp_now \
* sunlight / limbdark_avg;
proba_inlight_limb_lambda(i,h) = area * ssi_irra_now (h) * \
sunlight / limbdark_avg;
proba_light_limb_lambda(i,h) = area * sps_avg_irra_now * \
sunlight / limbdark_avg;

%t4=systemtime(/seconds)

% Infine trova l'intensità media per quella frazione
% di disco tenendo conto del limb darkening al bordo
% ed aggiungendo anche una grossa sunspot al lembo Est,
% dotata di ombra (intensità 0.7 della fotosfera
% circostante) e di penombra( intensità 0.1 della
% fotosfera circostante). Per passare
% ad unità fisiche (fotoni) basta anche qui moltiplicare
% la quantità ottenuta per la radianza emessa da tutto il
% disco integrato dividendo poi per il valor medio della
% funzione di limb darkening su tutto il disco (vedi

```

```

note).

                                [theta_now2, sundisk2, sunlight_spot] = ✓
disc_sunspot_lambda(app, x0, y0, r, a, b, ht0, xx, yy, ld_u_now, ld_v_now, theta_now, ✓
sundisk);

                                theta_now = theta_now2;
                                sundisk = sundisk2;
                                if isnan(sunlight_spot)
                                    sunlight_spot = -1;
                                end
                                proba_denscurr_spot_lambda(i,h) = area * sps_avg_resp_now ✓
* sunlight_spot / limbdark_avg;
                                proba_inlight_spot_lambda(i,h) = area * ssi_irra_now(h) * ✓
sunlight / limbdark_avg;
                                proba_light_spot_lambda(i,h) = area * sps_avg_irra_now * ✓
sunlight_spot / limbdark_avg;
                                end

                                %print,100.*i/n_dx,'% '
                                end
                                end
                                % Conversione della densità di corrente in corrente
                                % e quindi in DN per il primo e secondo stadio

                                % POTENZE SPETTRALI IN INPUT PRIMA DELL'ATTRAVERSAMENTO DELLE ✓
OTTICHE
                                %
                                proba_inpower_nolimb_lambda = zeros(n_dx, 17);
                                %
                                proba_inpower_limb_lambda = zeros(n_dx, 17);
                                %
                                proba_inpower_spot_lambda = zeros(n_dx, 17);
                                %
                                % POTENZE SPETTRALI IN INPUT DOPO ATTRAVERSAMENTO DELLE OTTICHE
                                %
                                proba_power_nolimb_lambda = zeros(n_dx, 17);
                                %
                                proba_power_limb_lambda = zeros(n_dx, 17);
                                %
                                proba_power_spot_lambda = zeros(n_dx, 17);
                                %
                                % CORRENTI SPETTRALI IN OUTPUT
                                %
                                proba_curr_nolimb_lambda = zeros(n_dx, 17);
                                %
                                proba_curr_limb_lambda = zeros(n_dx, 17);
                                %
                                proba_curr_spot_lambda = zeros(n_dx, 17);

                                dpin_pix_hires = dpin_mm * 1.d-3 / dx_min_occ; % numero di pixels ✓
corrispondenti al diametro del pinhole, zona alta risoluzione

                                for k = 1:16
                                    if dx_max_occ < goalbox_max
                                        for i = 1:n_dx
                                            if fix(i - dpin_pix_hires / 2. + 1) >= 1 && fix(i + ✓
dpin_pix_hires / 2.) < n_dx
                                                a = fix(i - dpin_pix_hires / 2.);
                                                if (a <= 1)
                                                    a = 1;
                                                end
                                                b = fix(i + dpin_pix_hires / 2.);
                                                proba_inpower_nolimb_lambda(i, k) = average(app, ✓
proba_inlight_nolimb_lambda(a:b, k)) * apin_cm2;
                                                %
                                                proba_inpower_limb_lambda(i, k) = average(app, ✓
proba_inlight_limb_lambda(a:b, k)) * apin_cm2;

```

```

%               proba_inpower_spot_lambda(i, k) = average(app,✓
proba_inlight_spot_lambda(a:b, k)) * apin_cm2;
%               proba_power_nolimb_lambda(i, k) = average(app,✓
proba_light_nolimb_lambda(a:b)) * apin_cm2;
%               proba_power_limb_lambda(i, k) = average(app,✓
proba_light_limb_lambda(a:b, k)) * apin_cm2;
%               proba_power_spot_lambda(i, k) = average(app,✓
proba_light_spot_lambda(a:b, k)) * apin_cm2;
%               proba_curr_nolimb_lambda(i, k) = average(app,✓
proba_denscurr_nolimb_lambda(a:b, k)) * apin_cm2;
%               proba_curr_limb_lambda(i, k) = average(app,✓
proba_denscurr_limb_lambda(a:b, k)) * apin_cm2;
%               proba_curr_spot_lambda(i, k) = average(app,✓
proba_denscurr_spot_lambda(a:b, k)) * apin_cm2;
end
else
    for i = 1:n_dx_hires
        if fix(i - dpin_pix_hires / 2. + 1) >= 1 && fix(i +✓
dpin_pix_hires / 2.) < n_dx_hires
            a = fix(i - dpin_pix_hires / 2.);
            if (a <= 1)
                a = 1;
            end
            b = fix(i + dpin_pix_hires / 2.);
            %               proba_inpower_nolimb_lambda(i, k) = average(app,✓
proba_inlight_nolimb_lambda(a:b, k)) * apin_cm2;
            %               proba_inpower_limb_lambda(i, k) = average(app,✓
proba_inlight_limb_lambda(a:b, k)) * apin_cm2;
            %               proba_inpower_spot_lambda(i, k) = average(app,✓
proba_inlight_spot_lambda(a:b, k)) * apin_cm2;
            %               proba_power_nolimb_lambda(i, k) = average(app,✓
proba_light_nolimb_lambda(a:b, k)) * apin_cm2;
            %               proba_power_limb_lambda(i, k) = average(app,✓
proba_light_limb_lambda(a:b, k)) * apin_cm2;
            %               proba_power_spot_lambda(i, k) = average(app,✓
proba_light_spot_lambda(a:b, k)) * apin_cm2;
            %               proba_curr_nolimb_lambda(i, k) = average(app,✓
proba_denscurr_nolimb_lambda(a:b, k)) * apin_cm2;
            %               proba_curr_limb_lambda(i, k) = average(app,✓
proba_denscurr_limb_lambda(a:b, k)) * apin_cm2;
            %               proba_curr_spot_lambda(i, k) = average(app,✓
proba_denscurr_spot_lambda(a:b, k)) * apin_cm2;
            end
        end
        for i = n_dx_hires + 1:n_dx %VALUTARE SE TOGLIERE +1
            proba_inpower_nolimb_lambda(i, k) =✓
proba_inlight_nolimb_lambda(i, k) * apin_cm2;
            %               proba_inpower_limb_lambda(i, k) =✓
proba_inlight_limb_lambda(i, k) * apin_cm2;
            %               proba_inpower_spot_lambda(i, k) =✓
proba_inlight_spot_lambda(i, k) * apin_cm2;
            %               proba_power_nolimb_lambda(i, k) =✓
proba_light_nolimb_lambda(i, k) * apin_cm2;
            %               proba_power_limb_lambda(i, k) = proba_light_limb_lambda✓

```

```

(i, k) * apin_cm2;
%                               proba_power_spot_lambda(i, k) = proba_light_spot_lambda
(i, k) * apin_cm2;
%                               proba_curr_nolimb_lambda(i, k) =
proba_denscurr_nolimb_lambda(i, k) * apin_cm2;
                               proba_curr_limb_lambda(i, k) = proba_denscurr_limb_lambda
(i, k) * apin_cm2;
%                               proba_curr_spot_lambda(i, k) = proba_denscurr_spot_lambda
(i, k) * apin_cm2;
                               end
                               for i = fix(n_dx_hires - dpin_pix_hires / 2.):n_dx_hires %
VALUTARE SE aggiungere -1 a inizio
                               a = fix(i);
                               b = fix(n_dx_hires - dpin_pix_hires / 2. - 1);
                               transversal_shift_now = transversal_shift(i);
%                               proba_inpower_nolimb_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_inpower_nolimb_lambda
(b, k), proba_inpower_nolimb_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_inpower_limb_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_inpower_limb_lambda(b,
k), proba_inpower_limb_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_inpower_spot_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_inpower_spot_lambda(b,
k), proba_inpower_spot_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_power_nolimb_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_power_nolimb_lambda(b,
k), proba_power_nolimb_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_power_limb_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_power_limb_lambda(b,
k), proba_power_limb_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_power_spot_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_power_spot_lambda(b,
k), proba_power_spot_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_curr_nolimb_lambda(a, k) = interp1
([transversal_shift(b), transversal_shift(n_dx_hires)], [proba_curr_nolimb_lambda(b,
k), proba_curr_nolimb_lambda(n_dx_hires, k)], transversal_shift_now);
                               proba_curr_limb_lambda(a, k) = interp1([transversal_shift
(b), transversal_shift(n_dx_hires)], [proba_curr_limb_lambda(b, k),
proba_curr_limb_lambda(n_dx_hires, k)], transversal_shift_now);
%                               proba_curr_spot_lambda(a, k) = interp1([transversal_shift
(b), transversal_shift(n_dx_hires)], [proba_curr_spot_lambda(b, k),
proba_curr_spot_lambda(n_dx_hires, k)], transversal_shift_now);
                               end
                               end
                               end

% POTENZE TOTALI E CORRENTI TOTALI (integrate in lambda) sugli SPS
prima e dopo l'attraversamento delle ottiche
lambdacut = find(lambdas == 500. | lambdas == 650.);

%proba_inpower_nolimb_full = sum(proba_inpower_nolimb_lambda, 2) *
10.; % integrazione in lambda (Delta lambda 10 nm)%FORSE "
%proba_inpower_nolimb_500 = sum(proba_inpower_nolimb_lambda(:,
lambdacut(1):lambdacut(2)), 2) * 10.; % integrazione in lambda (Delta lambda 10 nm)

```

```

        %proba_power_nolimb_full = sum(proba_power_nolimb_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_power_nolimb_500 = sum(proba_power_nolimb_lambda(:, lambdacut(
    (1):lambdacut(2)), 2) * 10.; % integrazione in lambda (Delta lambda 10 nm)
        %proba_curr_nolimb_full = sum(proba_curr_nolimb_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_curr_nolimb_500 = sum(proba_curr_nolimb_lambda(:, lambdacut(1):
    lambdacut(2)), 2) * 10.; % integrazione in lambda (Delta lambda 10 nm)
    %
    %
        %proba_inpower_limb_full = sum(proba_inpower_limb_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_inpower_limb_500 = sum(proba_inpower_limb_lambda(:, lambdacut(
    (1):lambdacut(2)), 2) * 10.; % integrazione in lambda (Delta lambda 10 nm)
    %
        %proba_power_limb_full = sum(proba_power_limb_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_power_limb_500 = sum(proba_power_limb_lambda(:, lambdacut(1):
    lambdacut(2)), 2) * 10.; % integrazione in lambda (Delta lambda 10 nm)
    %
        %proba_curr_limb_full = sum(proba_curr_limb_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_curr_limb_500 = sum(proba_curr_limb_lambda(:, lambdacut(1):
    lambdacut(2)), 2) * 10.; % integrazione in lambda (Delta lambda 10 nm)

        %proba_inpower_spot_full = sum(proba_inpower_spot_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_inpower_spot_500 = sum(proba_inpower_spot_lambda(:, lambdacut(
    (1):lambdacut(2)), 3) * 10.; % integrazione in lambda (Delta lambda 10 nm)
        %proba_power_spot_full = sum(proba_power_spot_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_power_spot_500 = sum(proba_power_spot_lambda(:, lambdacut(1):
    lambdacut(2)), 3) * 10.; % integrazione in lambda (Delta lambda 10 nm)
        %proba_curr_spot_full = sum(proba_curr_spot_lambda, 2) * 10.; %✓
    integrazione in lambda (Delta lambda 10 nm)
        %proba_curr_spot_500 = sum(proba_curr_spot_lambda(:, lambdacut(1):
    lambdacut(2)), 3) * 10.; % integrazione in lambda (Delta lambda 10 nm)

    % CONTEGGI TOTALI IN OUTPUT DAI DUE STADI
    reaction_resistance_factor = 100. / 84.;

        %proba_counts_nolimb_full_stad1 = proba_curr_nolimb_full * 1.e3 /✓
    dn_stad1 * reaction_resistance_factor;
        %proba_counts_nolimb_500_stad1 = proba_curr_nolimb_500 * 1.e3 /✓
    dn_stad1 * reaction_resistance_factor;
        %proba_counts_nolimb_full_stad2 = proba_curr_nolimb_full * 1.e3 /✓
    dn_stad2 * reaction_resistance_factor;
        %proba_counts_nolimb_500_stad2 = proba_curr_nolimb_500 * 1.e3 /✓
    dn_stad2 * reaction_resistance_factor;

    % CONTEGGI TOTALI IN OUTPUT DA ENTRAMBE GLI STADI, NOLIMB
    %proba_counts_nolimb_full = zeros(n_dx, 1);
    %proba_counts_nolimb_500 = zeros(n_dx, 1);
    %proba_counts_nolimb_full(proba_counts_nolimb_full_stad2 < 4096) =✓
    uint32(fix(proba_counts_nolimb_full_stad2(proba_counts_nolimb_full_stad2 < 4096)));✓
    %%uint32
    %proba_counts_nolimb_full(proba_counts_nolimb_full_stad2 >= 4096) =✓

```

```

uint32(fix(proba_counts_nolimb_full_stad1(proba_counts_nolimb_full_stad2 >= 4096)) * 5);
    %proba_counts_nolimb_500(proba_counts_nolimb_500_stad2 < 4096) = 0;
uint32(fix(proba_counts_nolimb_500_stad2(proba_counts_nolimb_500_stad2 < 4096)));
    %proba_counts_nolimb_500(proba_counts_nolimb_500_stad2 >= 4096) = 0;
uint32(fix(proba_counts_nolimb_500_stad1(proba_counts_nolimb_500_stad2 >= 4096)) * 5);

%
    proba_counts_limb_full_stad1 = proba_curr_limb_full * 1.e3 /
dn_stad1 * reaction_resistance_factor;
    proba_counts_limb_500_stad1 = proba_curr_limb_500 * 1.e3 / dn_stad1 *
reaction_resistance_factor;
%
    proba_counts_limb_full_stad2 = proba_curr_limb_full * 1.e3 /
dn_stad2 * reaction_resistance_factor;
    proba_counts_limb_500_stad2 = proba_curr_limb_500 * 1.e3 / dn_stad2 *
reaction_resistance_factor;

% CONTEGGI TOTALI IN OUTPUT DA ENTRAMBE GLI STADI, LIMB
%
    proba_counts_limb_full = zeros(n_dx, 1);
    proba_counts_limb_500 = zeros(n_dx, 1);
%
    proba_counts_limb_full(proba_counts_limb_full_stad2 < 4096) =
uint32(fix(proba_counts_limb_full_stad2(proba_counts_limb_full_stad2 < 4096)));
%
    proba_counts_limb_full(proba_counts_limb_full_stad2 >= 4096) =
uint32(fix(proba_counts_limb_full_stad1(proba_counts_limb_full_stad2 >= 4096)) * 5);
    proba_counts_limb_500(proba_counts_limb_500_stad2 < 4096) = uint32
(fix(proba_counts_limb_500_stad2(proba_counts_limb_500_stad2 < 4096)));
    proba_counts_limb_500(proba_counts_limb_500_stad2 >= 4096) = uint32
(fix(proba_counts_limb_500_stad1(proba_counts_limb_500_stad2 >= 4096)) * 5);
    %proba_counts_limb_500(isnan(proba_counts_limb_500_stad2)) = NaN;

%
    proba_counts_spot_full_stad1 = proba_curr_spot_full * 1.e3 / dn_stad1 *
* reaction_resistance_factor;
    %proba_counts_spot_500_stad1 = proba_curr_spot_500 * 1.e3 / dn_stad1 *
reaction_resistance_factor;
%
    proba_counts_spot_full_stad2 = proba_curr_spot_full * 1.e3 / dn_stad2 *
* reaction_resistance_factor;
    %proba_counts_spot_500_stad2 = proba_curr_spot_500 * 1.e3 / dn_stad2 *
reaction_resistance_factor;

% CONTEGGI TOTALI IN OUTPUT DA ENTRAMBE GLI STADI, SPOT
%
    proba_counts_spot_full = zeros(n_dx, 1);
    proba_counts_spot_500 = zeros(n_dx, 1);
%
    proba_counts_spot_full(proba_counts_spot_full_stad2 < 4096) = uint32
(fix(proba_counts_spot_full_stad2(proba_counts_spot_full_stad2 < 4096)));
%
    proba_counts_spot_full(proba_counts_spot_full_stad2 >= 4096) = uint32
(fix(proba_counts_spot_full_stad1(proba_counts_spot_full_stad2 >= 4096)) * 5);
%
    proba_counts_spot_500(proba_counts_spot_500_stad2 < 4096) = uint32
(fix(proba_counts_spot_500_stad2(proba_counts_spot_500_stad2 < 4096)));
    proba_counts_spot_500(proba_counts_spot_500_stad2 >= 4096) = uint32
(fix(proba_counts_spot_500_stad1(proba_counts_spot_500_stad2 >= 4096)) * 5);

%
% IRRADIANZE TOTALI IN INPUT PRIMA DELL'ATTRAVERSAMENTO DELLE
OTTICHE
%
    %proba_inlight_nolimb_full = sum(proba_inlight_nolimb_lambda, 2) *
10.;

```

```

%          %proba_inlight_nolimb_500 = sum(proba_inlight_nolimb_lambda(:,
lambdacut(1):lambdacut(2)), 2) * 10.;
%          proba_inlight_limb_full = sum(proba_inlight_limb_lambda, 2) * 10.;
%          proba_inlight_limb_500 = sum(proba_inlight_limb_lambda(:, lambdacut
(1):lambdacut(2)), 2) * 10.;
%          %proba_inlight_spot_full = sum(proba_inlight_spot_lambda, 2) * 10.;
%          %proba_inlight_spot_500 = sum(proba_inlight_spot_lambda(:, lambdacut
(1):lambdacut(2)), 2) * 10.;
%
%          % IRRADIANZE TOTALI IN INPUT DOPO L'ATTRAVERSAMENTO DELLE OTTICHE
%          %proba_light_nolimb_full = sum(proba_light_nolimb_lambda, 2) * 10.;
%          %proba_light_nolimb_500 = sum(proba_light_nolimb_lambda(:, lambdacut
(1):lambdacut(2)), 2) * 10.;
%          proba_light_limb_full = sum(proba_light_limb_lambda, 2) * 10.;
%          proba_light_limb_500 = sum(proba_light_limb_lambda(:, lambdacut(1):
lambdacut(2)), 2) * 10.;
%          %proba_light_spot_full = sum(proba_light_spot_lambda, 2) * 10.;
%          %proba_light_spot_500 = sum(proba_light_spot_lambda(:, lambdacut(1):
lambdacut(2)), 2) * 10.;
%
%          % Funzione di trasferimento da Watt ad Ampere
%          %cal_curve_nolimb_full = proba_inpower_nolimb_full ./
proba_curr_nolimb_full;
%          %cal_curve_nolimb_500 = proba_inpower_nolimb_500 ./
proba_curr_nolimb_500;
%          cal_curve_limb_full = proba_inpower_limb_full ./
proba_curr_limb_full;
%          cal_curve_limb_500 = proba_inpower_limb_500 ./ proba_curr_limb_500;
%          %cal_curve_spot_full = proba_inpower_spot_full ./
proba_curr_spot_full;
%          %cal_curve_spot_500 = proba_inpower_spot_500 ./ proba_curr_spot_500;
%
%          %cal_curve_nolimb_full(isfinite(cal_curve_nolimb_full) == 0) = 0.;
%          %cal_curve_nolimb_500(isfinite(cal_curve_nolimb_500) == 0) = 0.;
%          cal_curve_limb_full(~isfinite(cal_curve_limb_full)) = 0.;
%          cal_curve_limb_500(~isfinite(cal_curve_limb_500)) = 0.;
%          %cal_curve_spot_full(isfinite(cal_curve_spot_full) == 0) = 0.;
%          %cal_curve_spot_500(isfinite(cal_curve_spot_500) == 0) = 0.;

%t5 = systime( / seconds)

%print, 'kk = ', kk
%print, 'Elapsed time = ', t5 - t0, ' seconds'
%print, ''

longit_displ = dz_occ_array(kk);

Generazione_Ciotola_bidim(app, transversal_shift, longit_displ,
proba_curr_limb_500, proba_counts_limb_500);
if kk == 1
    mkdir(app.path_CSV_export_ciotola + "folder_for_fit_params");
    stringa_temp = "";
    for columns = 1:n_dx
        stringa_temp = stringa_temp + "col" + num2str(columns);

```

```

        if (columns ~= n_dx)
            stringa_temp = stringa_temp + ",";
        else
            stringa_temp = stringa_temp + "\n";
        end
    end
%
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_curr_limb_full_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_curr_limb_500_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_inlight_limb_full_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_inlight_limb_500_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_light_limb_full_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_light_limb_500_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_cal_curve_limb_full_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_cal_curve_limb_500_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_counts_limb_full_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
%         fid = fopen(app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_counts_limb_500_longit.csv", "w");
%         fprintf(fid, stringa_temp);
%         fclose(fid);
    end
%
%         % proba_curr_limb_full
%         array_to_csv(app, app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_curr_limb_full_longit.csv", proba_curr_limb_full);
%         % proba_curr_limb_500
%         array_to_csv(app, app.path_CSV_export_ciotola +
"folder_for_fit_params\app_proba_curr_limb_500_longit.csv", proba_curr_limb_500);
%         % proba_inlight_limb_full
%         array_to_csv(app, app.path_CSV_export_ciotola +

```

```

"folder_for_fit_params\app_proba_inlight_limb_full_longit.csv",↵
proba_inlight_limb_full);
%           % proba_inlight_limb_500
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_proba_inlight_limb_500_longit.csv",↵
proba_inlight_limb_500);
%           % proba_light_limb_full
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_proba_light_limb_full_longit.csv", proba_light_limb_full);
%           % proba_light_limb_500
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_proba_light_limb_500_longit.csv", proba_light_limb_500);
%           % cal_curve_limb_full
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_cal_curve_limb_full_longit.csv", cal_curve_limb_full);
%           % cal_curve_limb_500
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_cal_curve_limb_500_longit.csv", cal_curve_limb_500);
%           % proba_counts_limb_full
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_proba_counts_limb_full_longit.csv",↵
proba_counts_limb_full);
%           % proba_counts_limb_500
%           array_to_csv(app, app.path_CSV_export_ciotola +↵
"folder_for_fit_params\app_proba_counts_limb_500_longit.csv", proba_counts_limb_500);
end
if (length(dz_occ_array) >= 3 && app.CalculatefitparamsCheckBox.Value)
    longit_displ = dz_occ_array;
    %writetable(array2table(transversal_shift), "transversal_shift.csv");
    %writetable(array2table(longit_displ), "longit_displ.csv");
    %writetable(array2table(app.app_proba_counts_limb_500_longit),↵
"proba_counts_limb_500_longit.csv");
    Generazione_Fit_Parameters(app, transversal_shift, longit_displ, app.↵
path_CSV_export_ciotola + "folder_for_fit_params\app_proba_counts_limb_500_longit.↵
csv");
end
end

↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generazione ciotola bidimensionale
function Generazione_Ciotola_bidim(app, transversal_shift, longit_displ,↵
proba_curr_limb_500_longit, proba_counts_limb_500_longit)
% Ultima versione (25 / 07 / 2017), lavora direttamente sul
% profilo unidimensionale in DN costruito con l'ultime
% versione della routine proba_light_unidim.pro

%goto, plotta
%goto, fatto
%goto, fitta
%goto, newshade
%goto, newplot
%goto, qui

```

```

% Costruisce la ciotola di illuminazione "vera" dalla
% curva calcolata unidimensionale.
x0 = 0.d0;
y0 = 0.d0;

max_distance = 55.d0 + 100.d0; % extension of the 2D map (this is the
distance from optical center along the x or y axis) (mm)
pixres = 500.d-6 * 1.d3; % pixel resolution (N micron in mm)
% dshad = 38.5; % raggio dell'ombra (mm)
n_elem = max_distance / pixres + 1;

arr = 0:n_elem - 1;

x = (arr ./ (n_elem - 1.d0)) .* max_distance ./ 1.d3;

xx = repmat(x, n_elem, 1); % in IDL is REBIN
yy = repmat(x, n_elem, 1)';

% distance = zeros(n_elem, n_elem);
distance = sqrt((xx - x0).^2. + (yy - y0).^2.);

% Recupera il profilo unidimensionale di correnti (mA)
% per poi interpolarlo alla massima risoluzione impostata
% e quindi convertirlo in counts (DN) considerando
% sia lo high che il low gains.

n_elem_diag = max_distance * sqrt(2.) / pixres;
arr = 0:n_elem_diag - 1;
z_arr = pixres .* arr ./ 1.d3 + min(transversal_shift); % (m)
proba_counts = spline(transversal_shift, proba_counts_limb_500_longit,
z_arr);
proba_currs = spline(transversal_shift, proba_curr_limb_500_longit,
z_arr);

counts_sim = zeros(n_elem, n_elem);
curr_sim = zeros(n_elem, n_elem);

% t0 = systime( / seconds)

for i = 1:n_elem
    for j = 1:n_elem
        posnow = find(abs(z_arr - distance(i, j)) == min(abs(z_arr - distance
(i, j))));
        counts_sim(i, j) = proba_counts(posnow);
        curr_sim(i, j) = proba_currs(posnow);
    end
end

counts_sim(~isfinite(counts_sim)) = 0.;
curr_sim(~isfinite(curr_sim)) = 0.;

path_file = app.path_CSV_export_ciotola + longit_displ + "\";
mkdir(app.path_CSV_export_ciotola + longit_displ);
tab_counts = array2table(counts_sim);
count_filename = path_file + "proba_counts_bidim_" + longit_displ + ".

```

```

csv";
    x_longit = array2table(x);
    x_filename = path_file + "Z_" + longit_displ + ".csv";
    tab_curr = array2table(curr_sim);
    curr_filename = path_file + "proba_curr_bidim_" + longit_displ + ".csv";
    writetable(tab_counts, count_filename);
    writetable(x_longit, x_filename);
    writetable(tab_curr, curr_filename);
end

% =====
% Generazione_Fit_Parameters
function Generazione_Fit_Parameters(app, transversal_shift, longit_displ,
path_proba_counts_limb_500_longit)
% Costruisce la ciotola di illuminazione "vera" dalla
% curva calcolata unidimensionale.
%x0 = 0.d0;
%y0 = 0.d0;

    max_distance = 55.d0 + 100.d0; % extension of the 2D map (this is the
distance from optical center along the x or y axis) (mm)
    pixres = 500.d-6 * 1.d3; % pixel resolution (N micron in mm)
    %dshad = 38.5; % raggio dell'ombra (mm)
    %n_elem = max_distance / pixres + 1;

    %arr = 0:n_elem - 1;

    %x = (arr ./ (n_elem - 1.d0)) .* max_distance ./ 1.d3;

    %xx = repmat(x, n_elem1); %in IDL is REBIN
    %yy = repmat(x, n_elem, 1)';

    %distance = sqrt((xx - x0).^2. + (yy - y0).^2.);

    % Recupera il profilo unidimensionale di correnti (mA)
    % per poi interpolarlo alla massima risoluzione impostata
    % e quindi convertirlo in counts (DN) considerando
    % sia lo high che il low gains.

    n_elem_diag = max_distance * sqrt(2.) / pixres;
    arr = 0:n_elem_diag - 1;
    z_arr = pixres .* arr ./ 1.d3 + min(transversal_shift); % (m)
    if length(longit_displ) > 3
        a = fix(1);
        b = fix(2);
        c = length(longit_displ);
    else
        a = fix(1);
        b = fix(2);
        c = fix(3);
    end
end

```

```

        proba_counts_limb_500_longit = table2array(readtable(
(path_proba_counts_limb_500_longit));
        proba_counts_limb_500_longit = proba_counts_limb_500_longit';
        proba_counts_1 = spline(transversal_shift, proba_counts_limb_500_longit(:,
a), z_arr); %fix(end / 2)), z_arr);
        proba_counts_2 = spline(transversal_shift, proba_counts_limb_500_longit(:,
b), z_arr); %fix(end / 2) + 2), z_arr);
        proba_counts_x0 = spline(transversal_shift, proba_counts_limb_500_longit(
(:, c), z_arr); %fix(end / 2) + 1), z_arr);
        proba_counts_vect = zeros(length(proba_counts_1), length(longit_displ));
        for i = 1:length(longit_displ)
            arr = spline(transversal_shift, proba_counts_limb_500_longit(:, i),
z_arr); %fix(end / 2)), z_arr);;
            proba_counts_vect(:, i) = arr;
        end

% FIT TRASVERSALE NELLA REQUIREMENT BOX
%Z_ARR Starts eith 38.5 mm and, with steps of 0.5 mm, it goes
%to 105 mm (55 + 50). The rewuirment box is at 55mm (R_SPS) +
%10mm, so it goes from 45 yo 65.
fit_range_req = find(z_arr .* 1.e3 > 45. & z_arr .* 1.e3 < 65. &
proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));
fit_counts_req = fit(z_arr(fit_range_req)', proba_counts_x0(
(fit_range_req)', "poly3");
fit_counts_req_transv = coeffvalues(fit_counts_req);
%fit_counts_req_transv_curve = fit_counts_req_transv(1) +
fit_counts_req_transv(2) * z_arr(fit_range_req) + fit_counts_req_transv(3) * z_arr(
(fit_range_req)^2. + fit_counts_req_transv(4) * z_arr(fit_range_req)^3.;

% Parametri della ciotola fittante nella REQUIREMENT BOX
R0_req = fit_counts_req_transv(4);
a_req = 1.d0 / fit_counts_req_transv(3);
b_req = 1.d0 / sqrt(fit_counts_req_transv(2));
c_req = - 1.d0 / (abs(fit_counts_req_transv(1))^(1. / 3.));

% FIT TRASVERSALE NELLA GOAL BOX
fit_range_goal = find(z_arr .* 1.e3 > 40. & z_arr .* 1.e3 < 55. + 50. &
proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));
fit_counts_goal = fit(z_arr(fit_range_goal)', proba_counts_x0(
(fit_range_goal)', "poly3");
fit_counts_goal_transv = coeffvalues(fit_counts_goal);
%fit_counts_goal_transv_curve = fit_counts_goal_transv(1) +
fit_counts_goal_transv(2) * z_arr(fit_range_goal) + fit_counts_goal_transv(3) * z_arr(
(fit_range_goal)^2. + fit_counts_goal_transv(4) * z_arr(fit_range_goal)^3.;
% Parametri della ciotola fittante nella GOAL BOX
R0_goal = fit_counts_goal_transv(4);
a_goal = 1.d0 / fit_counts_goal_transv(3);
b_goal = 1.d0 / sqrt(fit_counts_goal_transv(2));
c_goal = - 1.d0 / (abs(fit_counts_goal_transv(1))^(1. / 3.));

% FIT TRASVERSALE NELLA GOAL BOX
fit_range_goal = find(z_arr .* 1.e3 > 65. & z_arr .* 1.e3 < 55. + 35. &
proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));
fit_counts_goal = fit(z_arr(fit_range_goal)', proba_counts_x0(
(fit_range_goal)', "poly3");

```

```

fit_counts_goal_transv = coeffvalues(fit_counts_goal);
% Parametri della ciotola fittante nella GOAL BOX
R0_goal1 = fit_counts_goal_transv(4);
a_goal1 = 1.d0 / fit_counts_goal_transv(3);
b_goal1 = 1.d0 / sqrt(fit_counts_goal_transv(2));
c_goal1 = - 1.d0 / (abs(fit_counts_goal_transv(1))^(1. / 3.));

% FIT TRASVERSALE NELLA GOAL BOX
fit_range_goal = find(z_arr .* 1.e3 > 55. + 35. & z_arr .* 1.e3 < 55. +
50. & proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));
fit_counts_goal = fit(z_arr(fit_range_goal)', proba_counts_x0
(fit_range_goal)', "poly3");
fit_counts_goal_transv = coeffvalues(fit_counts_goal);
% Parametri della ciotola fittante nella GOAL BOX
R0_goal2 = fit_counts_goal_transv(4);
a_goal2 = 1.d0 / fit_counts_goal_transv(3);
b_goal2 = 1.d0 / sqrt(fit_counts_goal_transv(2));
c_goal2 = - 1.d0 / (abs(fit_counts_goal_transv(1))^(1. / 3.));

% FIT TRASVERSALE NELLA GOAL BOX
fit_range_goal = find(z_arr .* 1.e3 > 55. + 10. & z_arr .* 1.e3 < 55. +
50. & proba_counts_x0 > 0. & proba_counts_x0 < max(proba_counts_x0));
fit_counts_goal = fit(z_arr(fit_range_goal)', proba_counts_x0
(fit_range_goal)', "poly3");
fit_counts_goal_transv = coeffvalues(fit_counts_goal);
% Parametri della ciotola fittante nella GOAL BOX
R0_goal3 = fit_counts_goal_transv(4);
a_goal3 = 1.d0 / fit_counts_goal_transv(3);
b_goal3 = 1.d0 / sqrt(fit_counts_goal_transv(2));
c_goal3 = - 1.d0 / (abs(fit_counts_goal_transv(1))^(1. / 3.));

% FIT LONGITUDINALE NELLA REQUIREMENT BOX
r_sps = 55.;
sps_pos = find(abs(z_arr .* 1.e3 - r_sps) == min(abs(z_arr .* 1.e3 -
r_sps)));
proba_counts_longit = zeros(1, length(longit_displ));
longit_displ = longit_displ * 1.e3;
if length(longit_displ) > 3
    for i = 1:length(longit_displ)
        arr = proba_counts_vect(:,i);
        proba_counts_longit(i) = arr(sps_pos);
    end
    fit_counts_req_longit = fit(longit_displ', proba_counts_longit',
"poly2");
    fit_counts_req_longit_coeff = coeffvalues(fit_counts_req_longit);
    H_req = fit_counts_req_longit_coeff(1);
    K_req = fit_counts_req_longit_coeff(2);
    L_req = fit_counts_req_longit_coeff(3);
else
    proba_counts_longit = [proba_counts_1(sps_pos), proba_counts_2
(sps_pos), proba_counts_x0(sps_pos)];
    longit_displ = [longit_displ(a), longit_displ(b), longit_displ(c)];
    H_req = ((proba_counts_longit(2) - proba_counts_longit(3)) +
(proba_counts_longit(1) - proba_counts_longit(2)) * (longit_displ(2) - longit_displ
(3)) / (longit_displ(2) - longit_displ(1))) / ((longit_displ(2)^2. - longit_displ(3))

```

```

^2.) - (longit_displ(2) - longit_displ(3)) * (longit_displ(2) + longit_displ(1));
    K_req = ((proba_counts_longit(2) - proba_counts_longit(1)) - H_req *
(longit_displ(2)^2. - longit_displ(1)^2.)) / (longit_displ(2) - longit_displ(1));
    L_req = proba_counts_longit(1) - H_req * longit_displ(1)^2. -
K_req * longit_displ(1);
    end

    arr = [R0_req, a_req, b_req, c_req, H_req, K_req, L_req, R0_goal,
a_goal, b_goal, c_goal, R0_goal1, a_goal1, b_goal1, c_goal1, R0_goal2, a_goal2,
b_goal2, c_goal2, R0_goal3, a_goal3, b_goal3, c_goal3];
    v_name = {'R0_fine' 'a_fine' 'b_fine' 'c_fine' 'H' 'K' 'L'
'R0_coarse(40_105)' 'a_coarse(40_105)' 'b_coarse(40_105)' 'c_coarse(40_105)'
'R0_coarse(65_90)' 'a_coarse(65_90)' 'b_coarse(65_90)' 'c_coarse(65_90)' 'R0_coarse'
(90_105)' 'a_coarse(90_105)' 'b_coarse(90_105)' 'c_coarse(90_105)' 'R0_coarse'
(65_105)' 'a_coarse(65_105)' 'b_coarse(65_105)' 'c_coarse(65_105)'};
    tab = array2table(arr, "VariableNames", v_name);
    filename_csv = app.path_CSV_export_ciotola + "fit_parameters.csv";
    writetable(tab, filename_csv);
end

% =====
% array_to_csv
function array_to_csv(~, path, array)
    % proba_curr_limb_full
    stringa_temp = "";
    for counter = 1:length(array)
        elem = num2str(array(counter));
        if ismissing(elem)
            elem = "NaN";
        end
        if (counter == length(array))
            stringa_temp = stringa_temp + elem + "\n";
        else
            stringa_temp = stringa_temp + elem + ",";
        end
    end
    fid = fopen(path, "a+");
    fprintf(fid, stringa_temp);
    fclose(fid);
end

% =====
% disc_sunlight_lambda
function [theta, sundisk, sunlight, limbdark_avg] = disc_sunlight_lambda(app,
x0, y0, r, a, b, ht0, xx, yy, u_limb, v_limb, thetal, sundisk1)
    % Programma per la determinazione della frazione di luce
    % del disco (misurata rispetto alla luce al centro) emessa
    % dalla "mezza luna" emergente dall'occultatore (data dall'
    % intersezione tra il cerchio del disco e l'ellisse dell'
    % occultatore).

```

```

%x0=-0.2
%y0=0.05
%r=1.00000
%a=1.0600756
%b=1.0600353
%
%% Array ausiliari per la routine disc_sunlight.pro
%r=1.d0
%xsun=2.d0*r*dindgen(1001)/1000.d0-r
%ysun=2.d0*r*dindgen(1001)/1000.d0-r
%xx=rebin(xsun,1001,1001)
%yy=rebin(reform(ysun,1,1001),1001,1001)
%ht0=sqrt(xx^2+yy^2)
%theta=dblarr(1001,1001)
%sundisk=dblarr(1001,1001)
%u_limb=0.97000003
%v_limb=-0.22000000

% Imposta le costanti per il limb darkening calcolate
% da Allen (2000) mediando sul profilo di responsivity
% degli SPS (calcolo effettuato dalla routine sps_irradiance.pro)
%u_limb=0.833458d0
%v_limb=-0.19874d0
%
%
%*****
% F. LANDINI'S TEST (14/12/2015: caso solo a lambda = 550 nm
%u_limb=0.93
%v_limb=-0.23
%*****

% Determina tutti i punti che si trovano entro al
% cerchio del disco e fuori dall'ellisse dell'
% occultatore (lavorando col disco al centro e
% l'ellisse shiftata).
circle_now = ht0.^2. - r^2.;
ellipse_now = (xx+x0).^2. / a^2. + (yy+y0).^2. / b^2.-1.d0;
% Lavora solo se ci si trova nella "mezza luna" di disco
% che sporge da dietro l'occultatore.

%visible_1 = zeros(1, length(circle_now(1,:)));
%visible_2 = zeros(1, length(ellipse_now(1,:)));
%for k=1:length(circle_now(1,:))
%    visible_1(k, circle_now(k,:) < 0.) = 1;
%end
%for l=1:length(ellipse_now(1,:))
%    visible_2(l, ellipse_now(l,:) > 0.) = 1;
%end
%visible = zeros(size(circle_now));
%x = find(circle_now < 0.);
%y = find(ellipse_now > 0.);
%z = ismember(x,y);
%for k=1:length(ellipse_now(1,:))
%    visible_1(l, circle_now(k,:) < 0.) = 1;

```

```

% visible_2(1, ellipse_now(k,:) > 0.) = 1;
% a = find(visible_1);
% b = find(visible_2);
% if(isempty(a) ~=1 && isempty(b) ~= 1)
%     c = ismember(a,b);
%     if (isempty(find(c, 1)) ~= 1)
%         visible(k,c(c)) = 1;
%     end
% end
% end
theta = thetal;
sundisk = sundisk1;
[row, col] = find(circle_now < 0. & ellipse_now > 0.);

for k = 1:length(row)
    theta(row(k), col(k)) = asin(ht0(row(k), col(k)) / r);
    % Funzione del limb darkening presa da Allen (2000)
    sundisk(row(k), col(k)) = 1.d0 - u_limb - v_limb + u_limb * cos(theta)✓
    (row(k), col(k)) + v_limb * (cos(theta(row(k), col(k))))^2.;
end

% Integra quindi tutta la "mezza luna" di disco col suo
% profilo di limb darkening.
sunlight = average(app, sundisk(sundisk > 0.)); %forse mettere find prima✓
di ultimo sundisk

% Fa poi la media della funzione di limb darkening su tutto il disco
[row, col] = find(circle_now < 0.);
for k = 1:length(row)
    a = asin(ht0(row(k), col(k)) / r);
    theta(row(k), col(k)) = a;
    % Funzione del limb darkening presa da Allen (2000)
    a = 1.d0 - u_limb - v_limb + u_limb * cos(theta(row(k), col(k))) +✓
    v_limb * (cos(theta(row(k), col(k))))^2.;
    sundisk(row(k), col(k)) = a;
end
limbdark_avg = average(app, sundisk(sundisk > 0.)); %forse mettere find✓
prima di ultimo sundisk

for k = 1:length(row)
    sundisk(row(k), col(k)) = 0.;
    theta(row(k), col(k)) = 0.;
end

end

✓
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%✓
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% disc_sunspot_lambda
function [theta, sundisk, sunlight] = disc_sunspot_lambda(app, x0, y0, r, a,✓
b, ht0, xx, yy, u_limb, v_limb, theta_now1, sundisk1)
% Programma per la determinazione della frazione di luce
% del disco (misurata rispetto alla luce al centro) emessa
% dalla "mezza luna" emergente dall'occultatore (data dall'

```

```

% intersezione tra il cerchio del disco e l'ellisse dell'
% occultatore). Il programma inoltre aggiunge al lembo una
% sunspot proiettata con raggio tipico di 2 raggi terrestri
% ed intensità pari a 0.1 dell'intensità della fotosfera
% circostante nella zona di ombra e 0.7 dell'intensità
% fotosferica nella zona di penombra.

% x0=-0.2
% y0=0.05
% r=1.00000
% a=1.0600756
% b=1.0600353

% Imposta le costanti per il limb darkening calcolate
% da Allen (2000) mediando sul profilo di responsivity
% degli SPS (calcolo effettuato dalla routine sps_irradiance.pro)
% u_limb=0.833458d0
% v_limb=-0.19874d0

% Imposta le costanti che definiscono la finta
% macchia solare al lembo Est
r_penum = 0.15; % raggio penombra (Rsun)
r_umbra = 0.10; % raggio ombra (Rsun)
i_penum = 0.7; %intensità della penombra
i_umbra = 0.15; %intensità dell'ombra
xc_umbra = 0.96;
yc_umbra = 0.;
a_penum = r_penum * sqrt(1. - xc_umbra^2.);
b_penum = r_penum;
a_umbra = r_umbra * sqrt(1. - xc_umbra^2.);
b_umbra = r_umbra;

% Determina tutti i punti che si trovano entro al
% cerchio del disco e fuori dall'ellisse dell'
% occultatore (lavorando col disco al centro e
% l'ellisse shiftata).
circle_now = ht0.^2. - r^2.;
ellipse_now = (xx+x0).^2. / a^2. + (yy+y0).^2. / b^2.-1.d0;
% Lavora solo se ci si trova nella "mezza luna" di disco
% che sporge da dietro l'occultatore.
[row, col] = find(circle_now < 0. & ellipse_now > 0.);
sundisk = sundisk1;
theta = theta_now1;
for k = 1:length(row)
    theta(row(k), col(k)) = asin(ht0(row(k), col(k)) / r);
    % Funzione del limb darkening presa da Allen (2000)
    sundisk(row(k), col(k)) = 1.d0 - u_limb - v_limb + u_limb * cos(theta)✓
    (row(k), col(k)) + v_limb * (cos(theta(row(k), col(k))))^2.;
end

%umbra_ellipse_now = zeros(zeros(size(circle_now)));
%penum_ellipse_now = zeros(zeros(size(circle_now)));
% Lavora solo se ci si trova dentro allo sunspot

```

```

        umbra_ellipse_now = (xx - xc_umbra).^2. / a_umbra^2. + (yy - yc_umbra).^2. / b_umbra^2. - 1.;
        penum_ellipse_now = (xx - xc_umbra).^2. / a_penum^2. + (yy - yc_umbra).^2. / b_penum^2. - 1.;

        [row, col] = find(umbra_ellipse_now <= 0.);
        [row2, col2] = find(umbra_ellipse_now > 0. & penum_ellipse_now <= 0.);
        for k = 1:length(row)
            sundisk(row(k), col(k)) = sundisk(row(k), col(k)) * i_umbra;
        end
        for k = 1:length(row2)
            sundisk(row2(k), col2(k)) = sundisk(row2(k), col2(k)) * i_penum;
        end

        % Integra quindi tutta la "mezza luna" di disco col suo
        % profilo di limb darkening.
        sunlight = average(app, sundisk(sundisk > 0.)); % Programma per la
determinazione della frazione di luce, FORSE CON FIND PRIMA DI ULTIMO SUNDISK
    end

% =====
% eclipse circle area
function area = ellipse_circle_area(app, x0, y0, r, a, b, sol_x, sol_y)
    % Programma per il calcolo dell'area di disco (cerchio) che emerge
    % da dietro all'occultatore (ellisse), dando l'illuminazione. L'area
    % è calcolata come differenza tra il segmento circolare ed il segmento
    % ellittico sotteso dai due punti di intersezione.

    %x0=0.085104662
    %y0=0.
    %r=1.0000000
    %a=1.0600756
    %b=1.0600
    %sol_x=[0.76921368,    0.76921368]
    %sol_y=[0.72937966,   -0.72937966]

    if length(sol_x) == 2 && length(sol_y) == 2
        xa = sol_x(1);
        xb = sol_x(2);
        ya = sol_y(1);
        yb = sol_y(2);

        % Determina gli angoli corrispondenti ai due punti A e B
        % misurati da 0 a 2*!PI rispetto al centro del cerchio, per
        % il calcolo dell'area del settore circolare.
        if (ya - y0) >= 0. && (xa - x0) > 0.
            theta_a_circ = atan((ya - y0) / (xa - x0));
        end
        if (xa - x0) < 0.
            theta_a_circ = atan((ya - y0) / (xa - x0)) + pi;
        end
        if (xa - x0) == 0. && (ya-y0) > 0.
            theta_a_circ = pi / 2.d0;

```

```

end
if (xa - x0) == 0. && (ya-y0) < 0.
    theta_a_circ = 3.d0 * pi / 2.d0;
end
if (ya - y0) <= 0. && (xa-x0) > 0.
    theta_a_circ = atan((ya - y0) / (xa - x0)) + 2.d0 * pi;
end

if (yb - y0) >= 0. && (xb - x0) > 0.
    theta_b_circ = atan((yb - y0) / (xb - x0));
end
if (xb - x0) < 0.
    theta_b_circ = atan((yb - y0) / (xb - x0)) + pi;
end
if (xb - x0) == 0. && (yb - y0) > 0.
    theta_b_circ = pi / 2.d0;
end
if (xb - x0) == 0. && (yb - y0) < 0.
    theta_b_circ = 3.d0 * pi / 2.d0;
end
if (yb - y0) <= 0. && (xb - x0) > 0.
    theta_b_circ = atan((yb - y0) / (xb - x0)) + 2.d0 * pi;
end

% Angolo compreso tra A e B rispetto al centro del cerchio
theta_ab = abs(theta_a_circ - theta_b_circ);
if theta_ab > pi
    theta_ab = 2.d0 * pi - theta_ab;
end

% Lunghezza del segmento AB
length_ab = sqrt((xa - xb)^2. + (ya - yb)^2.);

% Distanza del segmento AB dal centro del cerchio
if xa ~= xb
    line_ab_m = (ya - yb) / (xa - xb);
    line_ab_q = ya - line_ab_m * xa;
    dist_ab = abs(y0-line_ab_m * x0 - line_ab_q) / sqrt(1. + line_ab_m^2.);
else
    dist_ab = abs(xa - x0);
end

% Area del settore circolare
circle_sector_area = theta_ab*r^2. / 2.d0;

% Area del segmento circolare
circle_segment_area = circle_sector_area - length_ab * dist_ab / 2.d0;

% Determina gli angoli corrispondenti ai due punti A e B
% misurati da 0 a 2*!PI rispetto al centro dell'ellisse, per
% il calcolo dell'area del settore ellittico.
if ya > 0. && xa > 0.
    theta_a_ellips = atan(ya / xa);
end

```

```

    if xa < 0.
        theta_a_ellips = atan(ya / xa) + pi;
    end
    if ya < 0. && xa > 0.
        theta_a_ellips = atan(ya / xa) + 2.d0 * pi;
    end

    if yb > 0. && xb > 0.
        theta_b_ellips = atan(yb / xb);
    end
    if xb < 0.
        theta_b_ellips = atan(yb / xb) + pi;
    end
    if yb < 0. && xb > 0.
        theta_b_ellips = atan(yb / xb) + 2.d0 * pi;
    end

    % Calcola l'area del settore ellittico tra A e B

    ellipt_area_max = ellipt_sector(app, a, b, max([theta_a_ellips,
theta_b_ellips]));
    ellipt_area_min = ellipt_sector(app, a, b, min([theta_a_ellips,
theta_b_ellips]));

    % Verifica di prendere sempre l'area compresa nella zona
    % dove l'angolo del settore ellittico è minore di !PI
    ellipse_sector_area1 = (ellipt_area_max - ellipt_area_min);
    ellipse_sector_area2 = pi * a * b - (ellipt_area_max -
ellipt_area_min);
    ellipse_sector_area = min([ellipse_sector_area1,
ellipse_sector_area2]);

    % Area del segmento ellittico
    ellipse_segment_area = ellipse_sector_area - abs(xa * yb - xb * ya) /
2.d0;

    % Ci sono due casi possibili a seconda che il centro della
circonferenza
    % sia ad una distanza dal centro dell'ellisse minore (CASO 1) o
maggiore
    % (CASO 2) della distanza tra il segmento AB ed il centro
dell'ellisse.

    % Distanza del segmento AB dal centro dell'ellisse (origine degli
assi)
    if xa ~= xb
        dist_ab_orig = abs(line_ab_q) / sqrt(1. + line_ab_m^2.);
    else
        dist_ab_orig = abs(xa);
    end

    % CASO 1
    if dist_ab_orig >= sqrt(x0^2.+y0^2.)
        % AREA FINALE = (area del segmento circolare) - (area del
segmento ellittico)

```

```

        area = circle_segment_area - ellipse_segment_area;
    else
        % CASO 2
        % AREA FINALE = (area del cerchio) - (area del segmento
circolare + area del segmento ellittico)
        area = pi * r^2. - (circle_segment_area + ellipse_segment_area);
    end
else
    area = 0.d0;
end

% L'area è normalizzata all'area del disco.
area = area/(pi * r^2.);
end

%=====
%=====
% eclipt sector
function area = ellipt_sector(~, a, b, theta)
% Programma per il calcolo della funzione F(theta) che
% dà l'area spazzata dal raggio vettore nell'ellisse
% centrata a partire dall'asse x fino all'angolo theta
% dato in input (usata per calcolare l'area del settore
% ellittico).

    area = (a * b / 2.d0) * (theta - atan(((b - a) * sin(2.d0 * theta)) / ((b
+ a) + (b - a) * cos(2.d0 * theta))));
end

%=====
%=====
% ellipse circle sol
function [sol_x, sol_y, sol_pos] = ellipse_circle_sol(~, x0, y0, r, a, b) %
FORSE AGGIUNGERE SOL POS
% Programma per la determinazione dei punti di intersezione
% tra un'ellisse centrata con semiassi a e b ed una circonferenza
% traslata di raggio r e centro (x0,y0). Il programma usa la
% subroutine poly4roots per calcolare le radici esatte del
% polinomio di quarto grado che dà i punti di intersezione,
% quindi non introduce alcuna approssimazione e trova le
% coordinate esatte dei punti. Una parte (ora commentata)
% del programma calcolava invece le posizioni dei punti
% approssimate per via numerica.
%
%
% x0=2.05723d0
% y0=0.d0
% r=1.d0
% a=1.0572301d0
% b=1.0572301d0
%
% x0=0.011171604d0
% y0=0.15640246d0

```

```

%r=1.0000000d0
%a=1.0581650d0
%b=1.0581649d0

%x0=0.d0
%y0=-0.10429721d0
%r=1.d0
%a=1.0578717d0
%b=1.0577106d0
% CASO 1: |x0|>0 e |y0|>0
if abs(x0) > 0. && abs(y0) > 0.

    % Calcola i coefficienti del polinomio di quarto grado
    % che dà le soluzioni per le posizioni y dei punti di
    % intersezione
    A_par = (1.d0 - a^2. / b^2.) / (2. * x0);
    B_par = - y0 / x0;
    C_par = (x0^2. + y0^2. - r^2. + a^2.) / (2. * x0);

    coeff0 = C_par^2. - a^2.;
    coeff1 = 2. * B_par * C_par;
    coeff2 = B_par^2. + 2. * A_par * C_par + a^2. / b^2.;
    coeff3 = 2. * A_par * B_par;
    coeff4 = A_par^2.;

    % % SOLUZIONE APPROSSIMATA del polinomio quarto grado per via numerica
    %
    % % Polinomio soluzione
    % y_arr_sol=(dindgen(10001)/5000.-1.d0)*b*1.2d0
    % sol_poly=coeff0+coeff1*y_arr_sol+coeff2*y_arr_sol^2.
+coeff3*y_arr_sol^3.+coeff4*y_arr_sol^4.
    %
    % % Determina per quali elementi il polinomio cambia segno passando dallo
zero
    % sign_poly=dblarr(length(sol_poly)-1)
    % for i=0,length(sol_poly)-2 do sign_poly(i)=sol_poly(i+1)*sol_poly(i)
/abs(sol_poly(i+1)*sol_poly(i))
    % sol_pos=where(sign_poly lt 0.)
    %
    % if sol_pos(0) eq -1. then begin
    %     sol_x=0.d0
    %     sol_y=0.d0
    %     sol_pos=-1.
    %     goto,skip
    % endif
    %
    % sol_y=dblarr(length(sol_pos))
    %
    % for i=0,length(sol_pos)-1 do begin
    %     % Coordinate y dei punti, calcolate come il
    %     % valor medio della y a cavallo dei 2 punti
    %     % dove il polinomio cambia segno.
    %     sol_y(i)=(y_arr_sol(sol_pos(i))+y_arr_sol(sol_pos(i)+1))/2.
    % endfor

```

```

% SOLUZIONE ESATTA del polinomio quarto grado per via algebrica
sol_y = roots([coeff4, coeff3, coeff2, coeff1, coeff0]);

if sum(finite(sol_y), 1) > 0
    % Trova quindi le ascisse dei punti soluzione corrispondenti
    % sol_x=dblarr(length(sol_pos))
    sol_x = zeros(1, length(sol_y));

    % for i=0,length(sol_pos)-1 do begin
    for i = 1:length(sol_y)
        % Coordinate x dei punti: poichè per ogni y fissata ci sono
        % 2 possibili x corrispondenti sia sul cerchio che sull'ellisse,
        % trova le x dei punti di intersezione calcolando le x di tutti
        % e 4 i punti corrispondenti alla data y (2 sull'ellisse e 2
        % sulla circonferenza) e trova il punto corretto come quello
        % che dà lo scarto minore tra la x sulla circonferenza e la x
        % sull'ellisse.
        sol_x1_ellipse = a * sqrt(1.d0 - sol_y(i)^2. / b^2.);
        sol_x2_ellipse = - a * sqrt(1.d0 - sol_y(i)^2. / b^2.);
        sol_x1_circle = x0 + sqrt(abs(r^2. - (sol_y(i) - y0)^2.));
        sol_x2_circle = x0 - sqrt(abs(r^2. - (sol_y(i) - y0)^2.));
        delta_sol_x = [sol_x1_ellipse - sol_x1_circle, sol_x1_ellipse -
sol_x2_circle, sol_x2_ellipse - sol_x1_circle, sol_x2_ellipse - sol_x2_circle];
        delta_sol_x_pos = find(abs(delta_sol_x) == min(abs(
(delta_sol_x)));

        % if delta_sol_x_pos eq -1 then begin
        % sol_x=0.d0
        % sol_y=0.d0
        % sol_pos=-1.
        % goto, skip
        % endif

        switch delta_sol_x_pos
        case 1
            sol_x(i) = (sol_x1_ellipse + sol_x1_circle) / 2.;
        case 2
            sol_x(i) = (sol_x1_ellipse + sol_x2_circle) / 2.;
        case 3
            sol_x(i) = (sol_x2_ellipse + sol_x1_circle) / 2.;
        case 4
            sol_x(i) = (sol_x2_ellipse + sol_x2_circle) / 2.;
        end
    end
else
    sol_x = 0.d0;
    sol_y = 0.d0;
    sol_pos = -1.;%FORSE 2
    return;
end
end

% CASO 2: x0=0 e |y0|>0
if abs(x0) == 0. && abs(y0) > 0.
    A_par = 1.d0 - a^2. / b^2.;
    B_par = -2. * y0;

```

```

C_par = a^2. + y0^2. - r^2.;
delta = B_par^2. - 4. * A_par * C_par;
if delta < 0.
    sol_x = 0.d0;
    sol_y = 0.d0;
    sol_pos = -1.;
    return;
else
    sol_pos = 1.;
end
if delta == 0.
    sol_y = -B_par / (2. * A_par);
    sol_x = 0.d0;
    sol_pos = 1.;
    return;
end
if delta > 0. && A_par ~= 0.
    sol_y_pair = [( - B_par + sqrt(delta)) / (2. * A_par), ( - B_par -
sqrt(delta)) / (2. * A_par)];
    sol_y_pos = find(sol_y_pair < b * 1.01 & sol_y_pair > -b * 1.01);
    %sol_y_pos_1 = find(sol_y_pair < b * 1.01);
    %sol_y_pos_2 = find(sol_y_pair > -b * 1.01);
    %sol_y_pos = sol_y_pos_1(ismember(sol_y_pos_1, sol_y_pos_2));
    if isempty(sol_y_pos)
        sol_y = [sol_y_pair(sol_y_pos), sol_y_pair(sol_y_pos)];
        sol_x = [sqrt(a^2.-sol_y(1)^2.), -sqrt(a^2.-sol_y(1)^2.)];
        sol_pos = 1.;
    return;
    else
        sol_x = 0.d0;
        sol_y = 0.d0;
        sol_pos = -1.;
    return;
end
end
if delta > 0. && A_par == 0. && abs(y0) > abs(a-r)
    sol_y = [(y0^2. + a^2. - r^2.) / (2.d0 * y0), (y0^2. + a^2. - r^2.) /
(2.d0 * y0)];
    sol_x = [sqrt(a^2. - sol_y(1)^2.), - sqrt(a^2. - sol_y(1)^2.)];
    sol_pos = 1.;
    return;
end
if delta > 0. && A_par == 0. && abs(y0) <= abs(a-r)
    sol_y = (y0^2. + a^2. - r^2.) / (2.d0 * y0);
    sol_x = 0.;
    sol_pos = 1.;
    return;
end
if length(sol_y) == 2
    if sol_y(1) ~= sol_y(2)
        sol_x = zeros(1, 4);
        sol_x(1) = a * sqrt(1.d0 - sol_y(1)^2. / b^2.);
        sol_x(2) = - a * sqrt(1.d0 - sol_y(1)^2. / b^2.);
        sol_x(3) = a * sqrt(1.d0 - sol_y(2)^2. / b^2.);
        sol_x(4) = - a * sqrt(1.d0-sol_y(2)^2./b^2.);
    end
end

```

```

        sol_y = [sol_y(1), sol_y(1), sol_y(2), sol_y(2)];
        sol_pos = 1.;
        return;
    else
        sol_y = sol_y(1);
        sol_x = 0.;
        sol_pos = 1.;
        return;
    end
end

if length(sol_y) == 1 && abs(y0) <= abs(b-r)
    sol_x = 0.d0;
    sol_pos = 1.;
    return;
end
if length(sol_y) == 1 && abs(y0) > abs(b-r)
    sol_x = zeros(1, 2);
    sol_x(1) = a * sqrt(1.d0 - sol_y^2. / b^2.);
    sol_x(2) = -a * sqrt(1.d0 - sol_y^2. / b^2.);
    sol_y = [sol_y, sol_y];
    sol_pos = 1.;
    return;
end
if length(sol_x) == 2 && length(sol_y) == 1
    sol_y=[sol_y(1),sol_y(1)];
end
end
% CASO 3: |x0|>0 e y0=0
if abs(x0) > 0. && abs(y0) == 0.
    A_par = 1.d0 - b^2. / a^2.;
    B_par = -2. * x0;
    C_par = b^2. + x0^2. - r^2.;
    delta = B_par^2. - 4. * A_par * C_par;
    if delta < 0.
        sol_x = 0.d0;
        sol_y = 0.d0;
        sol_pos = - 1.;
        return;
    else
        sol_pos = 1.; %%FORSE DUE
    end
    if delta == 0.
        sol_x = -B_par / (2. * A_par);
    end
    if delta > 0. && A_par ~= 0.
        sol_x_pair = [( - B_par + sqrt(delta)) / (2. * A_par), (-B_par - sqrt(
(delta)) / (2. * A_par)];
        sol_x_pos = find(sol_x_pair < a * 1.01 & sol_x_pair > -a * 1.01);
        %sol_x_pos_1 = find(sol_x_pair < a * 1.01);
        %sol_x_pos_2 = find(sol_x_pair > -a * 1.01);
        %sol_x_pos = sol_x_pos_1(ismember(sol_x_pos_1, sol_x_pos_2));
        if isempty(sol_x_pos)
            sol_x = sol_x_pair(sol_x_pos);
        else

```

```

        sol_x = 0.d0;
        sol_y = 0.d0;
        sol_pos = - 1.;
        return;
    end
end
if delta > 0. && A_par == 0. && abs(x0) > abs(a-r)
    sol_x = [(x0^2. + a^2. - r^2.) / (2.d0 * x0), (x0^2. + a^2. - r^2.) / ✓
(2.d0 * x0)];
    sol_y = [sqrt(a^2. - sol_x(1)^2.), - sqrt(a^2. - sol_x(1)^2.)];
    sol_pos = 1.;
    return;
end
if delta > 0. && A_par == 0. && abs(x0) <= abs(a-r)
    sol_x = (x0^2. + a^2. - r^2.) / (2.d0 * x0);
    sol_y = 0.;
    sol_pos = 1.;
    return;
end
if length(sol_x) == 2
    if sol_x(1) ~= sol_x(2)
        sol_y = zeros(1, 4);
        sol_y(1) = b * sqrt(1.d0 - sol_x(1)^2. / a^2.);
        sol_y(2) = -b * sqrt(1.d0 - sol_x(1)^2. / a^2.);
        sol_y(3) = b*sqrt(1.d0 - sol_x(2)^2. / a^2.);
        sol_y(4) = - b * sqrt(1.d0 - sol_x(2)^2. / a^2.);
        sol_x = [sol_x(1), sol_x(1), sol_x(2), sol_x(2)];
        sol_pos = 1.;
    else
        sol_x = sol_y(1);
        sol_y = 0.;
        sol_pos = 1.;
        return;
    end
end
if length(sol_x) == 1 && abs(x0) <= abs(a-r)
    sol_y = 0.d0;
    sol_pos = 1.;
    return;
end
if length(sol_x) == 1 && abs(x0) > abs(a-r)
    sol_y = zeros(1,2);
    sol_y(1) = b * sqrt(1.d0 - sol_x^2. / a^2.);
    sol_y(2) = - b * sqrt(1.d0 - sol_x^2. / a^2.);
    sol_x=[sol_x,sol_x];
    sol_pos=1.;
    return;
end
end
% CASO 4: x0=0 e y0=0
if x0 == 0. && y0 == 0.
    if r == a
        sol_x = [-a, a];
        sol_y = [0.d0, 0.d0];
        sol_pos = 1.;
    end
end

```

```

end
if r == b
    sol_x = [0.d0, 0.d0];
    sol_y = [-b, b];
    sol_pos = 1.;
end
if r ~= a && r ~= b && a ~= b
    sol_y = [sqrt((a^2. - r^2.) / (a^2. / b^2. - 1.d0)), -sqrt((a^2. - r^2.) / (a^2. / b^2. - 1.d0))];
    sol_x = [a * sqrt(1.d0 - sol_y(1)^2. / b^2.), -a * sqrt(1.d0 - sol_y(1)^2. / b^2.)];
    sol_y = [sol_y(1), sol_y(1), sol_y(2), sol_y(2)];
    sol_x = [sol_x(1), sol_x(2), sol_x(1), sol_x(2)];
    sol_pos = 1.;
end
if a == b
    sol_x = 0.d0;
    sol_y = 0.d0;
    sol_pos = -1.;
end
end

%% Grafico per la visualizzazione di ellisse, circonferenza
%% e delle soluzioni per le intersezioni.

z_arr=(dindgen(10001)/5000.-1.)*2.*r
%!P.MULTI=0
%window,0
%plot,z_arr,y0+sqrt(r^2.-(z_arr-x0)^2.),xrange=[-2.*r,2.*r],yrange=[-2.*r,2.*r],/isotropic
%oplot,z_arr,y0-sqrt(r^2.-(z_arr-x0)^2.)
%oplot,z_arr,b*sqrt(1.-z_arr^2./a^2.),line=2
%oplot,z_arr,-b*sqrt(1.-z_arr^2./a^2.),line=2
%%if sol_pos(0) ne -1 then begin
%   plots,sol_x,sol_y,psym=2
%   if length(sol_x) eq 2 and length(sol_y) eq 2 then plots,[sol_x(0),sol_x(1)],[sol_y(0),sol_y(1)]
%%endif
%plots,0,0,psym=3
%plots,x0,y0,psym=1

%% Grafico per la visualizzazione del polinomio di quarto grado risolvente
%window,1
%plot,y_arr_sol,sol_poly,yrange=[-5.,5.]
%plots,[-b*1.2,b*1.2],[0,0]
end

% =====
% average
function w = average(~, ARRAY)
    ARRAY(isnan(ARRAY)) = [];
    w = mean(ARRAY);
end

```

end

```

↙
%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%%

```

```

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

```

```

% Callbacks that handle component events
methods (Access = private)

```

```

% Code that executes after component creation

```

```

function startupFcn(app)

```

```

    pause on;
    app.f = waitbar(0, "Starting app, please wait...", 'windowstyle', 'modal');
    frames = java.awt.Frame.getFrames();
    frames(end).setAlwaysOnTop(1);
    waitbar(0.03, app.f, "Loading app component...");
    format long
    drawnow;
    app.SIMULATIONPROBA3UIFigure.WindowState = 'maximized';
    pause(2);
    waitbar(0.75, app.f, "Resizing app component...");
    pause(10);
    waitbar(1, app.f, "Finishing, do not press anything");
    pause off
    close(app.f);

```

```

end

```

```

% Value changed function: CheckBox_a_coarse

```

```

function CheckBox_a_coarseValueChanged(app, ~)

```

```

    value = app.CheckBox_a_coarse.Value;
    if value
        app.RangeEditField_a_coarse_range.Enable = 'on';
        app.RangeEditField_A_coarse_range.Enable = 'off';
        app.CheckBox_A_coarse.Value = 0;
    else
        app.RangeEditField_a_coarse_range.Enable = 'off';
    end

```

```

end

```

```

% Button pushed function: STARTSIMULATIONSPSTOXYZButton

```

```

function STARTSIMULATIONSPSTOXYZButtonPushed(app, ~)

```

```

    app.Label_Flag_l_nom.Text = "";
    app.Label_Flag_l_red.Text = "";
    app.Label_Flag_lat_nom.Text = "";
    app.Label_Flag_lat_red.Text = "";
    app.Label_Flag_Long_nom.Text = "";
    app.Label_Flag_Long_red.Text = "";
    app.Label_nom_x.Text = "";
    app.Label_nom_y.Text = "";
    app.Label_nom_z.Text = "";
    app.Label_red_x.Text = "";
    app.Label_red_y.Text = "";
    app.Label_red_z.Text = "";

```

```

app.Lamp.Enable = 'off';
app.X_error_plot.NextPlot = 'replaceall';
app.Y_error_plot.NextPlot = 'replaceall';
app.Z_error_plot.NextPlot = 'replaceall';
main_function(app);
app.flag_real_table = 0;
#####
if(app.real_value == 0)
    app.real_value = 1;
    msgbox("Complex solutions");
    % ENABLE BUTTON
    app.STARTSIMULATIONSPSTOXYZButton.Enable = 'on';
    if app.CSV_REAL_Drop_Down.Value ~= "Only CSV"
        app.GETSELECTEDCOLUMNButton.Enable = 'on';
        app.CREATE TWO VARIABLES PLOT Button.Enable = 'on';
    else
        app.GETSELECTEDCOLUMNButton.Enable = 'off';
        app.CREATE TWO VARIABLES PLOT Button.Enable = 'off';
    end
    return;
end
#####
if app.Save relevant values in a CSV CheckBox.Value
    dir = string(pwd) + "\";
    if app.PathCSV_relevant_values.Value ~= "<ACTUAL MATLAB PATH>"
        dir = app.PathCSV_relevant_values.Value + "\";
    end
    date = datestr(datetime);
    DATE_STR = strrep(date, ',', '_');
    filename_csv = dir + "X = " + string(app.EditField_X_Expected.Value) + <
+ string(app.EditField_Y_Expected.Value) + " Z = " + string(app.<
ld_Z_Expected.Value) + "_" + DATE_STR + "_relevant_values.csv";
    tab = table2array(app.Full_Table.Data);
    test = 1:(str2double(app.Test_Field.Value) + 1);
    arr = [test', tab(:,2), tab(:,3), tab(:,4), tab(:,5), tab(:,6), tab(:,<
(:,57), tab(:,58), tab(:,59), tab(:,60), tab(:,61), tab(:,62), tab(:,10), tab<
tab(:,12), tab(:,9), tab(:,14), tab(:,15), tab(:,16), tab(:,13), tab(:,32),<
l), tab(:,17), tab(:,18), tab(:,20), tab(:,19), tab(:,21)];
    v_name = {'TEST' 'Nom_X_mm' 'Nom_Y_mm' 'Nom_Z_mm' 'Red_X_mm'<
mm' 'Red_Z_mm' 'Delta_Nom_X_mm' 'Delta_Nom_Y_mm' 'Delta_Nom_Z_mm'<
Red_X_mm' 'Delta_Red_Y_mm' 'Delta_Red_Z_mm' 'a_coarse' 'b_coarse' 'c_coarse'<
se' 'a_fine' 'b_fine' 'c_fine' 'A_fine' 'd_y' 'd_z' 'H' 'K' 'J' 'L' 'delta_R'};
    table_csv = array2table(arr, "VariableNames", v_name);
    writetable(table_csv, filename_csv);
end
msgbox("Operation Completed", "SPS-TO-XYZ");
end

% Value changed function: CheckBox_b_coarse
function CheckBox_b_coarseValueChange(app, ~)
    value = app.CheckBox_b_coarse.Value;
    if value
        app.RangeEditField_b_coarse_range.Enable = 'on';
        app.RangeEditField_A_coarse_range.Enable = 'off';
        app.CheckBox_A_coarse.Value = 0;
    end
end

```

```

else
    app.RangeEditField_b_coarse_range.Enable = 'off';
end
end

% Value changed function: CheckBox_c_coarse
function CheckBox_c_coarseValueChanged(app, ~)
    value = app.CheckBox_c_coarse.Value;
    if value
        app.RangeEditField_c_coarse_range.Enable = 'on';
        app.RangeEditField_A_coarse_range.Enable = 'off';
        app.CheckBox_A_coarse.Value = 0;
    else
        app.RangeEditField_c_coarse_range.Enable = 'off';
    end
end

% Button pushed function: ImportparamsButton
function ImportparamsButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.json');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        All_Data = jsondecode(fileread(filename_temp));
        app.EditField_SPS1_high.Value = All_Data.raw_data.Read(1);
        app.EditField_SPS1_low.Value = All_Data.raw_data.Read(2);
        app.EditField_SPS2_high.Value = All_Data.raw_data.Read(3);
        app.EditField_SPS2_low.Value = All_Data.raw_data.Read(4);
        app.EditField_SPS3_high.Value = All_Data.raw_data.Read(5);
        app.EditField_SPS3_low.Value = All_Data.raw_data.Read(6);
        app.EditField_SPS4_high.Value = All_Data.raw_data.Read(7);
        app.EditField_SPS4_low.Value = All_Data.raw_data.Read(8);
        app.EditField_SPS5_high.Value = All_Data.raw_data.Read(9);
        app.EditField_SPS5_low.Value = All_Data.raw_data.Read(10);
        app.EditField_SPS6_high.Value = All_Data.raw_data.Read(11);
        app.EditField_SPS6_low.Value = All_Data.raw_data.Read(12);
        app.EditField_SPS7_high.Value = All_Data.raw_data.Read(13);
        app.EditField_SPS7_low.Value = All_Data.raw_data.Read(14);
        app.EditField_SPS8_high.Value = All_Data.raw_data.Read(15);
        app.EditField_SPS8_low.Value = All_Data.raw_data.Read(16);
        app.EditField_nom_T1.Value = All_Data.raw_data.Temp(1);
        app.EditField_Nom_T2.Value = All_Data.raw_data.Temp(2);
        app.EditField_Red_T1.Value = All_Data.raw_data.Temp(3);
        app.EditField_Red_T2.Value = All_Data.raw_data.Temp(4);
        app.CheckBox_Door.Value = logical(All_Data.raw_data.door);
        app.CheckBox_Door_tr.Value = logical(All_Data.raw_data.door_tr);
        app.EditField_K1.Value = All_Data.K(1);
        app.EditField_K2.Value = All_Data.K(2);
        app.EditField_K3.Value = All_Data.K(3);
        app.EditField_K4.Value = All_Data.K(4);
        app.EditField_K5.Value = All_Data.K(5);
        app.EditField_K6.Value = All_Data.K(6);
        app.EditField_K7.Value = All_Data.K(7);
        app.EditField_K8.Value = All_Data.K(8);
        if All_Data.algo == 2

```

```

        app.Box_Algorithm.Value = '2';
    elseif All_Data.algo == 1
        app.Box_Algorithm.Value = '1';
    else
        app.Box_Algorithm.Value = '0';
    end
    app.EditField_a_coarse_value.Value = All_Data.pp_coarse.a_coarse;
    app.EditField_b_coarse_value.Value = All_Data.pp_coarse.b_coarse;
    app.EditField_c_coarse_value.Value = All_Data.pp_coarse.c_coarse;
    app.EditField_A_coarse_value.Value = All_Data.pp_coarse.A_coarse;
    app.EditField_a_fine_value.Value = All_Data.pp_fine.a_fine;
    app.EditField_b_fine_value.Value = All_Data.pp_fine.b_fine;
    app.EditField_c_fine_value.Value = All_Data.pp_fine.c_fine;
    app.EditField_A_fine_value.Value = All_Data.pp_fine.A_fine;
    app.EditField_delta_transv.Value = All_Data.pp_fine.delta_transv;
    app.EditField_d_z.Value = All_Data.linear.d_z;
    app.EditField_d_y.Value = All_Data.linear.d_y;
    app.EditField_H.Value = All_Data.long.H;
    app.EditField_J.Value = All_Data.long.J;
    app.EditField_K.Value = All_Data.long.K;
    app.EditField_L.Value = All_Data.long.L;
    app.EditField_delta_R.Value = All_Data.long.delta_R;
    app.EditField_V_max_lat.Value = All_Data.thresholds.V_max_lat;
    app.EditField_V_max_long.Value = All_Data.thresholds.V_max_long;
    app.EditField_T_max.Value = All_Data.thresholds.T_max;
    app.EditField_delta_flag.Value = All_Data.thresholds.delta_flag;
    app.EditField_X_Expected.Value = All_Data.Expected_X_Y_Z.X;
    app.EditField_Y_Expected.Value = All_Data.Expected_X_Y_Z.Y;
    app.EditField_Z_Expected.Value = All_Data.Expected_X_Y_Z.Z;
end
end

% Button pushed function: SelectpathCSVButton
function SelectpathCSVButtonPushed(app, ~)
    dir = uigetdir;
    figure(app.SIMULATIONPROBA3UIFigure);
    if dir ~= 0
        app.PathCSVEditField.Value = string(dir);
    end
end

% Value changed function: CheckBox_A_coarse
function CheckBox_A_coarseValueChanged(app, ~)
    value = app.CheckBox_A_coarse.Value;
    if value
        app.CheckBox_a_coarse.Value = 0;
        app.CheckBox_b_coarse.Value = 0;
        app.CheckBox_c_coarse.Value = 0;
        app.RangeEditField_a_coarse_range.Enable = 'off';
        app.RangeEditField_b_coarse_range.Enable = 'off';
        app.RangeEditField_c_coarse_range.Enable = 'off';
        app.RangeEditField_A_coarse_range.Enable = 'on';
    else
        app.RangeEditField_A_coarse_range.Enable = 'off';
    end
end

```

```

end

% Value changed function: CSV_REAL_Drop_Down
function CSV_REAL_Drop_DownValueChanged(app, ~)
    value = app.CSV_REAL_Drop_Down.Value;
    if value == "Only Real Time"
        app.PathCSVEditField.Enable = 'off';
        app.SelectpathCSVButton.Enable = 'off';
        app.Full_Table.Enable = 'on';
        app.Filtered_Table.Enable = 'on';
        app.FilterListBox.Enable = 'on';
        app.GETSELECTEDCOLUMNButton.Enable = 'on';
        app.CREATETWOVARIABLESPLOTButton.Enable = 'on';
    elseif value == "Only CSV"
        app.PathCSVEditField.Enable = 'on';
        app.SelectpathCSVButton.Enable = 'on';
        app.Full_Table.Enable = 'off';
        app.Filtered_Table.Enable = 'off';
        app.FilterListBox.Enable = 'off';
        app.GETSELECTEDCOLUMNButton.Enable = 'off';
        app.CREATETWOVARIABLESPLOTButton.Enable = 'off';
    else
        app.PathCSVEditField.Enable = 'on';
        app.SelectpathCSVButton.Enable = 'on';
        app.Full_Table.Enable = 'on';
        app.Filtered_Table.Enable = 'on';
        app.FilterListBox.Enable = 'on';
        app.GETSELECTEDCOLUMNButton.Enable = 'on';
        app.CREATETWOVARIABLESPLOTButton.Enable = 'on';
    end
end

% Value changed function: CheckBox_a_fine
function CheckBox_a_fineValueChanged(app, ~)
    value = app.CheckBox_a_fine.Value;
    if value
        app.RangeEditField_a_fine_range.Enable = 'on';
        app.RangeEditField_A_fine_range.Enable = 'off';
        app.CheckBox_A_fine.Value = 0;
    else
        app.RangeEditField_a_fine_range.Enable = 'off';
    end
end

% Value changed function: CheckBox_b_fine
function CheckBox_b_fineValueChanged(app, ~)
    value = app.CheckBox_b_fine.Value;
    if value
        app.RangeEditField_b_fine_range.Enable = 'on';
        app.RangeEditField_A_fine_range.Enable = 'off';
        app.CheckBox_A_fine.Value = 0;
    else
        app.RangeEditField_b_fine_range.Enable = 'off';
    end
end

```

```
% Value changed function: CheckBox_c_fine
function CheckBox_c_fineValueChanged(app, ~)
    value = app.CheckBox_c_fine.Value;
    if value
        app.RangeEditField_c_fine_range.Enable = 'on';
        app.RangeEditField_A_fine_range.Enable = 'off';
        app.CheckBox_A_fine.Value = 0;
    else
        app.RangeEditField_c_fine_range.Enable = 'off';
    end
end

% Value changed function: CheckBox_A_fine
function CheckBox_A_fineValueChanged(app, ~)
    value = app.CheckBox_A_fine.Value;
    if value
        app.CheckBox_a_fine.Value = 0;
        app.CheckBox_b_fine.Value = 0;
        app.CheckBox_c_fine.Value = 0;
        app.RangeEditField_a_fine_range.Enable = 'off';
        app.RangeEditField_b_fine_range.Enable = 'off';
        app.RangeEditField_c_fine_range.Enable = 'off';
        app.RangeEditField_A_fine_range.Enable = 'on';
    else
        app.RangeEditField_A_fine_range.Enable = 'off';
    end
end

% Value changed function: CheckBox_H
function CheckBox_HValueChanged(app, ~)
    app.RangeEditField_H_range.Enable = app.CheckBox_H.Value;
end

% Value changed function: CheckBox_K
function CheckBox_KValueChanged(app, ~)
    app.RangeEditField_K_range.Enable = app.CheckBox_K.Value;
end

% Value changed function: CheckBox_L
function CheckBox_LValueChanged(app, ~)
    app.RangeEditField_L_range.Enable = app.CheckBox_L.Value;
end

% Value changed function: CheckBox_J
function CheckBox_JValueChanged(app, ~)
    app.RangeEditField_J_range.Enable = app.CheckBox_J.Value;
end

% Value changed function: CheckBox_delta_R
function CheckBox_delta_RValueChanged(app, ~)
    app.RangeEditField_delta_R_range.Enable = app.CheckBox_delta_R.Value;
end

% Value changed function: CheckBox_K_coeff
```

```

function CheckBox_K_coeffValueChanged(app, ~)
    app.RangeEditField_K_coeff_range.Enable = app.CheckBox_K_coeff.Value;
end

% Value changed function: CheckBox_K_coeff_2
function CheckBox_K_coeff_2ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_2.Enable = app.CheckBox_K_coeff_2.Value;
end

% Value changed function: CheckBox_K_coeff_3
function CheckBox_K_coeff_3ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_3.Enable = app.CheckBox_K_coeff_3.Value;
end

% Value changed function: CheckBox_K_coeff_4
function CheckBox_K_coeff_4ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_4.Enable = app.CheckBox_K_coeff_4.Value;
end

% Value changed function: CheckBox_K_coeff_5
function CheckBox_K_coeff_5ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_5.Enable = app.CheckBox_K_coeff_5.Value;
end

% Value changed function: CheckBox_K_coeff_6
function CheckBox_K_coeff_6ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_6.Enable = app.CheckBox_K_coeff_6.Value;
end

% Value changed function: CheckBox_K_coeff_7
function CheckBox_K_coeff_7ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_7.Enable = app.CheckBox_K_coeff_7.Value;
end

% Value changed function: CheckBox_K_coeff_8
function CheckBox_K_coeff_8ValueChanged(app, ~)
    app.RangeEditField_K_coeff_range_8.Enable = app.CheckBox_K_coeff_8.Value;
end

% Value changed function: CheckBox_d_z
function CheckBox_d_zValueChanged(app, ~)
    app.RangeEditField_d_z.Enable = app.CheckBox_d_z.Value;
end

% Value changed function: CheckBox_d_y
function CheckBox_d_yValueChanged(app, ~)
    app.RangeEditField_d_y.Enable = app.CheckBox_d_y.Value;
end

% Value changed function: CheckBox_SPS_high_1
function CheckBox_SPS_high_1ValueChanged(app, ~)
    app.CheckBox_SPS_Low_1.Value = 0;
    app.RangeEditField_SPS_high_range_1.Enable = app.CheckBox_SPS_high_1.Value;
    app.RangeEditField_SPS_Low_range_1.Enable = "off";
Value;

```

```

end

% Value changed function: CheckBox_SPS_high_2
function CheckBox_SPS_high_2ValueChanged(app, ~)
    app.CheckBox_SPS_Low_2.Value = 0;
    app.RangeEditField_SPS_high_range_2.Enable = app.CheckBox_SPS_high_2.✔
Value;
    app.RangeEditField_SPS_Low_range_2.Enable = "off";
end

% Value changed function: CheckBox_SPS_high_3
function CheckBox_SPS_high_3ValueChanged(app, ~)
    app.CheckBox_SPS_Low_3.Value = 0;
    app.RangeEditField_SPS_high_range_3.Enable = app.CheckBox_SPS_high_3.✔
Value;
    app.RangeEditField_SPS_Low_range_3.Enable = "off";
end

% Value changed function: CheckBox_SPS_high_4
function CheckBox_SPS_high_4ValueChanged(app, ~)
    app.CheckBox_SPS_Low_4.Value = 0;
    app.RangeEditField_SPS_high_range_4.Enable = app.CheckBox_SPS_high_4.✔
Value;
    app.RangeEditField_SPS_Low_range_4.Enable = "off";
end

% Value changed function: CheckBox_SPS_high_5
function CheckBox_SPS_high_5ValueChanged(app, ~)
    app.CheckBox_SPS_Low_5.Value = 0;
    app.RangeEditField_SPS_high_range_5.Enable = app.CheckBox_SPS_high_5.✔
Value;
    app.RangeEditField_SPS_Low_range_5.Enable = "off";
end

% Value changed function: CheckBox_SPS_high_7
function CheckBox_SPS_high_7ValueChanged(app, ~)
    app.CheckBox_SPS_Low_7.Value = 0;
    app.RangeEditField_SPS_high_range_7.Enable = app.CheckBox_SPS_high_7.✔
Value;
    app.RangeEditField_SPS_Low_range_7.Enable = "off";
end

% Value changed function: CheckBox_SPS_high_8
function CheckBox_SPS_high_8ValueChanged(app, ~)
    app.CheckBox_SPS_Low_8.Value = 0;
    app.RangeEditField_SPS_high_range_8.Enable = app.CheckBox_SPS_high_8.✔
Value;
    app.RangeEditField_SPS_Low_range_8.Enable = "off";
end

% Value changed function: CheckBox_SPS_high_6
function CheckBox_SPS_high_6ValueChanged(app, ~)
    app.CheckBox_SPS_Low_6.Value = 0;
    app.RangeEditField_SPS_high_range_6.Enable = app.CheckBox_SPS_high_6.✔
Value;

```

```
    app.RangeEditField_SPS_Low_range_6.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_1
function CheckBox_SPS_Low_1ValueChanged(app, ~)
    app.CheckBox_SPS_high_1.Value = 0;
    app.RangeEditField_SPS_Low_range_1.Enable = app.CheckBox_SPS_Low_1.Value;
    app.RangeEditField_SPS_high_range_1.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_2
function CheckBox_SPS_Low_2ValueChanged(app, ~)
    app.CheckBox_SPS_high_2.Value = 0;
    app.RangeEditField_SPS_Low_range_2.Enable = app.CheckBox_SPS_Low_2.Value;
    app.RangeEditField_SPS_high_range_2.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_3
function CheckBox_SPS_Low_3ValueChanged(app, ~)
    app.CheckBox_SPS_high_3.Value = 0;
    app.RangeEditField_SPS_Low_range_3.Enable = app.CheckBox_SPS_Low_3.Value;
    app.RangeEditField_SPS_high_range_3.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_4
function CheckBox_SPS_Low_4ValueChanged(app, ~)
    app.CheckBox_SPS_high_4.Value = 0;
    app.RangeEditField_SPS_Low_range_4.Enable = app.CheckBox_SPS_Low_4.Value;
    app.RangeEditField_SPS_high_range_4.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_5
function CheckBox_SPS_Low_5ValueChanged(app, ~)
    app.CheckBox_SPS_high_5.Value = 0;
    app.RangeEditField_SPS_Low_range_5.Enable = app.CheckBox_SPS_Low_5.Value;
    app.RangeEditField_SPS_high_range_5.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_6
function CheckBox_SPS_Low_6ValueChanged(app, ~)
    app.CheckBox_SPS_high_6.Value = 0;
    app.RangeEditField_SPS_Low_range_6.Enable = app.CheckBox_SPS_Low_6.Value;
    app.RangeEditField_SPS_high_range_6.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_7
function CheckBox_SPS_Low_7ValueChanged(app, ~)
    app.CheckBox_SPS_high_7.Value = 0;
    app.RangeEditField_SPS_Low_range_7.Enable = app.CheckBox_SPS_Low_7.Value;
    app.RangeEditField_SPS_high_range_7.Enable = "off";
end

% Value changed function: CheckBox_SPS_Low_8
function CheckBox_SPS_Low_8ValueChanged(app, ~)
    app.CheckBox_SPS_high_8.Value = 0;
```

```
app.RangeEditField_SPS_Low_range_8.Enable = app.CheckBox_SPS_Low_8.Value;  
app.RangeEditField_SPS_high_range_8.Enable = "off";  
end
```

```
% Value changed function: MontecarloCheckBox  
function MontecarloCheckBoxValueChanged(app, ~)  
    value = app.MontecarloCheckBox.Value;  
    % checkbox  
    app.CheckBox_a_coarse.Value = value;  
    app.CheckBox_b_coarse.Value = value;  
    app.CheckBox_c_coarse.Value = value;  
    app.CheckBox_a_fine.Value = value;  
    app.CheckBox_b_fine.Value = value;  
    app.CheckBox_c_fine.Value = value;  
    app.CheckBox_d_y.Value = value;  
    app.CheckBox_d_z.Value = value;  
    app.CheckBox_delta_R.Value = value;  
    app.CheckBox_H.Value = value;  
    app.CheckBox_J.Value = value;  
    app.CheckBox_K.Value = value;  
    app.CheckBox_L.Value = value;  
    app.CheckBox_K_coeff.Value = value;  
    app.CheckBox_K_coeff_2.Value = value;  
    app.CheckBox_K_coeff_3.Value = value;  
    app.CheckBox_K_coeff_4.Value = value;  
    app.CheckBox_K_coeff_5.Value = value;  
    app.CheckBox_K_coeff_6.Value = value;  
    app.CheckBox_K_coeff_7.Value = value;  
    app.CheckBox_K_coeff_8.Value = value;  
    app.CheckBox_SPS_high_1.Value = value;  
    app.CheckBox_SPS_high_2.Value = value;  
    app.CheckBox_SPS_high_3.Value = value;  
    app.CheckBox_SPS_high_4.Value = value;  
    app.CheckBox_SPS_high_5.Value = value;  
    app.CheckBox_SPS_high_6.Value = value;  
    app.CheckBox_SPS_high_7.Value = value;  
    app.CheckBox_SPS_high_8.Value = value;  
    % range  
    app.RangeEditField_a_coarse_range.Enable = value;  
    app.RangeEditField_b_coarse_range.Enable = value;  
    app.RangeEditField_c_coarse_range.Enable = value;  
    app.RangeEditField_a_fine_range.Enable = value;  
    app.RangeEditField_b_fine_range.Enable = value;  
    app.RangeEditField_c_fine_range.Enable = value;  
    app.RangeEditField_H_range.Enable = value;  
    app.RangeEditField_J_range.Enable = value;  
    app.RangeEditField_K_range.Enable = value;  
    app.RangeEditField_L_range.Enable = value;  
    app.RangeEditField_delta_R_range.Enable = value;  
    app.RangeEditField_d_z.Enable = value;  
    app.RangeEditField_d_y.Enable = value;  
    app.RangeEditField_K_coeff_range.Enable = value;  
    app.RangeEditField_K_coeff_range_2.Enable = value;  
    app.RangeEditField_K_coeff_range_3.Enable = value;  
    app.RangeEditField_K_coeff_range_4.Enable = value;
```

```

app.RangeEditField_K_coeff_range_5.Enable = value;
app.RangeEditField_K_coeff_range_6.Enable = value;
app.RangeEditField_K_coeff_range_7.Enable = value;
app.RangeEditField_K_coeff_range_8.Enable = value;
app.RangeEditField_SPS_high_range_1.Enable = value;
app.RangeEditField_SPS_high_range_2.Enable = value;
app.RangeEditField_SPS_high_range_3.Enable = value;
app.RangeEditField_SPS_high_range_4.Enable = value;
app.RangeEditField_SPS_high_range_5.Enable = value;
app.RangeEditField_SPS_high_range_6.Enable = value;
app.RangeEditField_SPS_high_range_7.Enable = value;
app.RangeEditField_SPS_high_range_8.Enable = value;
end

% Value changed function: GETSELECTEDCOLUMNButton
function GETSELECTEDCOLUMNButtonValueChanged(app, ~)
    filtered_value = app.FilterListBox.Value;
    c = str2double(filtered_value);
    names = app.Full_Table.ColumnNames;
    app.Filtered_Table.ColumnNames = names(c);
    app.Filtered_Table.Data = app.Full_Table.Data(:,c);
    msgbox("Filtration Completed", "Filter");
end

% Value changed function: CREATETWOVARIABLESPLOTButton
function CREATETWOVARIABLESPLOTButtonValueChanged(app, ~)
    value = app.FilterListBox.Value;
    if length(value) == 2
        column_x = str2double(value(2));
        column_y = str2double(value(1));
        tab = table2array(app.Full_Table.Data);
        if (column_x == 1)
            x_axis_value = 1:str2double(app.Test_Field.Value);
            x_axis_value(end+1) = x_axis_value(end) + 1;
        else
            x_axis_value = tab(:,column_x);
        end
        if (column_y == 1)
            y_axis_value = 1:str2double(app.Test_Field.Value);
            y_axis_value(end+1) = y_axis_value(end) + 1;
        else
            y_axis_value = tab(:,column_y);
        end
        app.x_axes_values_filter = x_axis_value';
        app.y_axes_values_filter = y_axis_value';
        app.actual_max_x.Text = "" + max(x_axis_value);
        app.actual_max_y.Text = "" + max(y_axis_value);
        app.actual_min_x.Text = "" + min(x_axis_value);
        app.actual_min_y.Text = "" + min(y_axis_value);

        %%label
        app.PLOTS_TWO_VARIABLES.XLabel.String = strrep(app.Full_Table.
ColumnNames(column_x), "_", " ");
        app.PLOTS_TWO_VARIABLES.YLabel.String = strrep(app.Full_Table.
ColumnNames(column_y), "_", " ");

```

```

%%complex value for y axes
[y_complex, y_real, x_complex, x_real] = real_complex_vector(app,
y_axis_value, x_axis_value);

%% starter value
app.actual_x_choose_params = start_param_chosen(app, app.
PLOTS_TWO_VARIABLES.XLabel.String, x_axis_value);
line = ones(size(y_axis_value)) * app.actual_x_choose_params;

%%plot
line_y_axis_value = y_axis_value;
line_y_axis_value(1) = line_y_axis_value(1) - 0.01 * line_y_axis_value
(1);
line_y_axis_value(end) = line_y_axis_value(end) + 0.01 *
line_y_axis_value(end);
plot(app.PLOTS_TWO_VARIABLES, x_real, y_real, 'r:', x_complex,
y_complex, 'c:', line, line_y_axis_value, 'k-');
if isempty(x_complex)
    legend(app.PLOTS_TWO_VARIABLES, 'REAL', 'INITIAL VALUE');
elseif isempty(x_real)
    legend(app.PLOTS_TWO_VARIABLES, 'COMPLEX', 'INITIAL VALUE');
else
    legend(app.PLOTS_TWO_VARIABLES, 'REAL', 'COMPLEX', 'INITIAL
VALUE');
end
else
    msgbox("SELECT TWO VARIABLES", "Error", "error");
end
end

% Selection changed function: ChooseaxesButtonGroup
function ChooseaxesButtonGroupSelectionChanged(app, ~)
    selectedButton = app.ChooseaxesButtonGroup.SelectedObject.Text;
    if selectedButton == "X axes"
        app.Edit_Max_X.Enable = 'on';
        app.Edit_Min_X.Enable = 'on';
        app.Edit_Max_Y.Enable = 'off';
        app.Edit_Min_Y.Enable = 'off';
    else
        app.Edit_Max_X.Enable = 'off';
        app.Edit_Min_X.Enable = 'off';
        app.Edit_Max_Y.Enable = 'on';
        app.Edit_Min_Y.Enable = 'on';
    end
end

% Value changed function: FilteraxesButton
function FilteraxesButtonValueChanged(app, ~)
    min_x = app.Edit_Min_X.Value;
    max_x = app.Edit_Max_X.Value;
    min_y = app.Edit_Min_Y.Value;
    max_y = app.Edit_Max_Y.Value;

    if app.ChooseaxesButtonGroup.SelectedObject.Text == "X axes"

```

```

        if str2double(app.actual_min_x.Text) > app.Edit_Min_X.Value ||
str2double(app.actual_max_x.Text) < app.Edit_Max_X.Value
        msgbox("Choose values in range", "Error", "error");
    else
        pos_min_x = find(app.x_axes_values_filter <= min_x);
        pos_max_x = find(app.x_axes_values_filter >= max_x);
        if pos_min_x(end) > pos_max_x(1)
            msgbox("Invert these two values for this variable", "Error",
"error");
        else
            x_axis_value = app.x_axes_values_filter(pos_min_x(end);
pos_max_x(1));
            y_axis_value = app.y_axes_values_filter(pos_min_x(end);
pos_max_x(1));

            %%complex value for y axes
            [y_complex, y_real, x_complex, x_real] = real_complex_vector
(app, y_axis_value, x_axis_value);

            %% starter value
            app.actual_x_choose_params = start_param_chosen(app, app.
PLOTS_TWO_VARIABLES.XLabel.String, x_axis_value);
            line = ones(size(y_axis_value)) * app.actual_x_choose_params;

            %%plot
            plot(app.PLOTS_TWO_VARIABLES, x_complex, y_complex, 'c:*',
x_real, y_real, 'r:*', line, y_axis_value, 'k-');
            if isempty(x_complex)
                legend(app.PLOTS_TWO_VARIABLES, 'REAL', 'INITIAL VALUE');
            elseif isempty(x_real)
                legend(app.PLOTS_TWO_VARIABLES, 'COMPLEX', 'INITIAL
VALUE');
            else
                legend(app.PLOTS_TWO_VARIABLES, 'COMPLEX', 'REAL',
'INITIAL VALUE');
            end

            %plot(app.PLOTS_TWO_VARIABLES, app.x_axes_values_filter
(pos_min_x(end):pos_max_x(1)), app.y_axes_values_filter(pos_min_x(end):pos_max_x(1)),
'r*');
        end
    end
else
    if str2double(app.actual_min_y.Text) > app.Edit_Min_Y.Value ||
str2double(app.actual_max_y.Text) < app.Edit_Max_Y.Value
    msgbox("Choose values in range", "Error", "error");
    else
        pos_min_y = find(app.y_axes_values_filter <= min_y);
        pos_max_y = find(app.y_axes_values_filter >= max_y);
        if pos_min_y(end) > pos_max_y(1)
            msgbox("Invert these two values for this variable", "Error",
"error");

```

```

else
    x_axis_value = app.x_axes_values_filter(pos_min_y(end):✓
pos_max_y(1));
    y_axis_value = app.y_axes_values_filter(pos_min_y(end):✓
pos_max_y(1));

    %%label
    app.PLOTS_TWO_VARIABLES.XLabel.String = strrep(app.Full_Table.✓
ColumnName(column_x), " ", " ");
    app.PLOTS_TWO_VARIABLES.YLabel.String = strrep(app.Full_Table.✓
ColumnName(column_y), " ", " ");

    %%complex value for y axes
    [y_complex, y_real, x_complex, x_real] = real_complex_vector✓
(app, y_axis_value, x_axis_value);

    %% starter value
    app.actual_x_choose_params = start_param_chosen(app, app.✓
PLOTS_TWO_VARIABLES.XLabel.String, x_axis_value);
    line = ones(size(y_axis_value)) * app.actual_x_choose_params;

    %%plot
    plot(app.PLOTS_TWO_VARIABLES, x_complex, y_complex, 'c:~', ✓
x_real, y_real, 'r:~', line, y_axis_value, 'k-');
    if isempty(x_complex)
        legend(app.PLOTS_TWO_VARIABLES, 'REAL', 'INITIAL VALUE');
    elseif isempty(x_real)
        legend(app.PLOTS_TWO_VARIABLES, 'COMPLEX', 'INITIAL ✓
VALUE');
    else
        legend(app.PLOTS_TWO_VARIABLES, 'COMPLEX', 'REAL', ✓
'INITIAL VALUE');
    end

    %plot(app.PLOTS_TWO_VARIABLES, app.x_axes_values_filter✓
(pos_min_y(end):pos_max_y(1)), app.y_axes_values_filter(pos_min_y(end):pos_max_y(1)), ✓
'r-~');
end
end
end

% Button pushed function: STARTSIMULATIONXYZTOSPSButton
function STARTSIMULATIONXYZTOSPSButtonPushed(app, ~)
    Get_SPS_From_Position(app);
end

% Button pushed function: SelectcountsCSVButton
function SelectcountsCSVButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        app.CSV_Count.Value = filename_temp;
    end
end

```

```

        end
    end

    % Button pushed function: SelectcurrCSVButton
    function SelectcurrCSVButtonPushed(app, ~)
        [file, path_temp] = uigetfile('*.csv');
        figure(app.SIMULATIONPROBA3UIFigure);
        if file ~= 0
            filename_temp = "" + string(path_temp) + string(file);
            app.CSV_Curr.Value = filename_temp;
        end
    end

    % Button pushed function: SelectZCSVButton
    function SelectZCSVButtonPushed(app, ~)
        [file, path_temp] = uigetfile('*.csv');
        figure(app.SIMULATIONPROBA3UIFigure);
        if file ~= 0
            filename_temp = "" + string(path_temp) + string(file);
            app.CSV_X.Value = filename_temp;
        end
    end

    % Button pushed function: EXPORTDNTOSPSTOXYZButton
    function EXPORTDNTOSPSTOXYZButtonPushed(app, ~)
        app.EditField_X_Expected.Value = app.ISEditField.Value - app.XEditField.✓
Value;
        app.EditField_Y_Expected.Value = 0 - app.YEditField.Value;
        app.EditField_Z_Expected.Value = 0 - app.ZEditField.Value;
        app.EditField_SPS1_high.Value = app.SPS1EditField.Value;
        app.EditField_SPS1_low.Value = fix(app.SPS1EditField.Value / 5);
        app.EditField_SPS2_high.Value = app.SPS2EditField.Value;
        app.EditField_SPS2_low.Value = fix(double(app.SPS2EditField.Value) / 5);
        app.EditField_SPS3_high.Value = app.SPS3EditField.Value;
        app.EditField_SPS3_low.Value = fix(double(app.SPS3EditField.Value) / 5);
        app.EditField_SPS4_high.Value = app.SPS4EditField.Value;
        app.EditField_SPS4_low.Value = fix(double(app.SPS4EditField.Value) / 5);
        app.EditField_SPS5_high.Value = app.SPS5EditField.Value;
        app.EditField_SPS5_low.Value = fix(double(app.SPS5EditField.Value) / 5);
        app.EditField_SPS6_high.Value = app.SPS6EditField.Value;
        app.EditField_SPS6_low.Value = fix(double(app.SPS6EditField.Value) / 5);
        app.EditField_SPS7_high.Value = app.SPS7EditField.Value;
        app.EditField_SPS7_low.Value = fix(double(app.SPS7EditField.Value) / 5);
        app.EditField_SPS8_high.Value = app.SPS8EditField.Value;
        app.EditField_SPS8_low.Value = fix(double(app.SPS8EditField.Value) / 5);
        app.EditField_R_sps.Value = app.SPSRadiusEditField.Value;
        if app.ExportfitparamsCheckBox.Value
            r_sps = 0.055;
            app.EditField_a_coarse_value.Value = app.fit_param(21);
            app.EditField_b_coarse_value.Value = app.fit_param(22);
            app.EditField_c_coarse_value.Value = app.fit_param(23);
            app.EditField_A_coarse_value.Value = (app.EditField_c_coarse_value.✓
Value^3) * ((1 / app.EditField_a_coarse_value.Value) + ((2 * r_sps) / (app.✓
EditField_b_coarse_value.Value^2)) + ((3 * r_sps^2) / app.EditField_c_coarse_value.✓
Value^3));

```

```

        app.EditField_a_fine_value.Value = app.fit_param(2);
        app.EditField_b_fine_value.Value = app.fit_param(3);
        app.EditField_c_fine_value.Value = app.fit_param(4);
        app.EditField_A_fine_value.Value = (app.EditField_c_fine_value.Value^3) * ((1 / app.EditField_a_fine_value.Value)+(2 * r_sps) / (app.EditField_b_fine_value.Value^2))+((3 * r_sps^2) / app.EditField_c_fine_value.Value^3));
        app.EditField_H.Value = app.fit_param(5);
        app.EditField_K.Value = app.fit_param(6);
        app.EditField_L.Value = app.fit_param(7);
        app.EditField_J.Value = 0;
        app.EditField_delta_R.Value = - (0.055 / app.EditField_a_fine_value.Value) - (0.055 / app.EditField_b_fine_value.Value)^2 - (0.055 / app.EditField_c_fine_value.Value)^3;
    end
    msgbox("Exportation completed", "Export");
end

% Button pushed function: EXPORTINJSONButton
function EXPORTINJSONButtonPushed(app, ~)
    dir = uigetdir();
    figure(app.SIMULATIONPROBA3UIFigure);
    if dir ~= 0
        %raw
        Read = [app.SPS1EditField.Value; fix(app.SPS1EditField.Value / 5); app.SPS2EditField.Value; fix(app.SPS2EditField.Value / 5); app.SPS3EditField.Value; fix(app.SPS3EditField.Value / 5); app.SPS4EditField.Value; fix(app.SPS4EditField.Value / 5); app.SPS5EditField.Value; fix(app.SPS5EditField.Value / 5); app.SPS6EditField.Value; fix(app.SPS6EditField.Value / 5); app.SPS7EditField.Value; fix(app.SPS7EditField.Value / 5); app.SPS8EditField.Value; fix(app.SPS8EditField.Value / 5)];
        raw_data.Read = Read;
        raw_data.Temp = [app.EditField_nom_T1.Value; app.EditField_Nom_T2.Value; app.EditField_Red_T1.Value; app.EditField_Red_T2.Value];
        raw_data.door = double(app.CheckBox_Door.Value);
        raw_data.door_tr = double(app.CheckBox_Door_tr.Value);
        %reconf
        complete_struct.K = [app.EditField_K1.Value; app.EditField_K2.Value; app.EditField_K3.Value; app.EditField_K4.Value; app.EditField_K5.Value; app.EditField_K6.Value; app.EditField_K7.Value; app.EditField_K8.Value];
        complete_struct.algo = str2double(app.Box_Algorithm.Value);
        complete_struct.pp_coarse.a_coarse = app.EditField_a_coarse_value.Value;
        complete_struct.pp_coarse.b_coarse = app.EditField_b_coarse_value.Value;
        complete_struct.pp_coarse.c_coarse = app.EditField_c_coarse_value.Value;
        complete_struct.pp_coarse.A_coarse = app.EditField_A_coarse_value.Value;
        complete_struct.pp_fine.a_fine = app.EditField_a_fine_value.Value;
        complete_struct.pp_fine.b_fine = app.EditField_b_fine_value.Value;
        complete_struct.pp_fine.c_fine = app.EditField_c_fine_value.Value;
        complete_struct.pp_fine.A_fine = app.EditField_A_fine_value.Value;
        complete_struct.pp_fine.delta_transv = app.EditField_delta_transv.Value;
        complete_struct.linear.d_z = app.EditField_d_z.Value;
    end
end

```

```

        complete_struct.linear.d_y = app.EditField_d_y.Value;
        complete_struct.long.H = app.EditField_H.Value;
        complete_struct.long.K = app.EditField_K.Value;
        complete_struct.long.L = app.EditField_L.Value;
        complete_struct.long.J = app.EditField_J.Value;
        complete_struct.long.delta_R = app.EditField_delta_R.Value;
        complete_struct.thresholds.V_max_lat = app.EditField_V_max_lat.Value;
        complete_struct.thresholds.V_max_long = app.EditField_V_max_long.Value;
Value;

        complete_struct.thresholds.T_max = app.EditField_T_max.Value;
        complete_struct.thresholds.delta_flag = app.EditField_delta_flag.Value;
Value;

        %Expected_X_Y_Z
        Expected_X_Y_Z.X = app.ISDEditField.Value - app.XEditField.Value;
        Expected_X_Y_Z.Y = 0 - app.YEditField.Value;
        Expected_X_Y_Z.Z = 0 - app.ZEditField.Value;
        %to_struct
        complete_struct.raw_data = raw_data;
        complete_struct.Expected_X_Y_Z = Expected_X_Y_Z;
        json = jsonencode(complete_struct);
        filename_json = string(dir) + "\" + app.Json_filename.Value + ".json";
        fileID = fopen(filename_json, 'w');
        fprintf(fileID, json);
        fclose(fileID);
    else
        msgbox("The file will not be created", "File not created", "warn");
    end
end

% Button pushed function: STARTGENERATIONSButton
function STARTGENERATIONSButtonPushed(app, ~)
    if (app.SSI_solarmin_CSV.Value == "NaN" || app.
        resp_t_dep_m20degC_to_60degCCSV.Value == "NaN" || app.Id_coefficients_CSV.Value ==
        "NaN" || app.Glass_trasmittance_CSV.Value == "NaN" || app.Coating_trasmittance_CSV.
        Value == "NaN")
        msgbox("IMPORT ALL CSV", "Error", "error");
    else
        msgbox("Generation started, please note that this process could need
        hours. Do not change any values", "Warning", "warn");
        app.STARTGENERATIONSButton.Enable = "off";
        date = datestr(datetime);
        DATE_STR = strrep(date, ':', '_');
        app.path_CSV_export_ciotola = string(pwd) + "\";
        if app.Export_CSV_Ciotola.Value ~= "<ACTUAL MATLAB PATH>"
            app.path_CSV_export_ciotola = app.Export_CSV_Ciotola.Value + "\";
        end
        mkdir(app.path_CSV_export_ciotola + DATE_STR);
        app.path_CSV_export_ciotola = app.path_CSV_export_ciotola + DATE_STR +
        "\";

        Generazione_Ciotola_unidim(app);
        msgbox("Generation completed");
        app.STARTGENERATIONSButton.Enable = "on";
    end
end

```

```

% Button pushed function: SelectSSI_solar_min_maxCSVButton
function SelectSSI_solar_min_maxCSVButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        app.SSI_solarmin_CSV.Value = filename_temp;
    end
end

% Button pushed function:
% Select_resp_t_dep_m20degC_to_60degCCSVButton
function Select_resp_t_dep_m20degC_to_60degCCSVButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        app.resp_t_dep_m20degC_to_60degCCSV.Value = filename_temp;
    end
end

% Button pushed function:
% SelectCoating_trasmittanceCSVButton
function SelectCoating_trasmittanceCSVButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        app.Coating_trasmittance_CSV.Value = filename_temp;
    end
end

% Button pushed function: SelectLd_coefficientsCSVButton
function SelectLd_coefficientsCSVButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        app.Ld_coefficients_CSV.Value = filename_temp;
    end
end

% Button pushed function: SelectGlass_trasmittanceCSVButton
function SelectGlass_trasmittanceCSVButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        filename_temp = "" + string(path_temp) + string(file);
        app.Glass_trasmittance_CSV.Value = filename_temp;
    end
end

% Button pushed function: SAVECURRTOCSVButton
function SAVECURRTOCSVButtonPushed(app, ~)
    dir = uigetdir;

```

```

figure(app.SIMULATIONPROBA3UIFigure);
if dir ~= 0
    date = datestr(datetime);
    DATE_STR = strrep(date, ':', '_');
    percorso = string(dir) + "\" + DATE_STR + "__X_" + string(app.XEditField.Value) + "__Y_" + string(app.YEditField.Value) + "__Z_" + string(app.ZEditField.Value) + "__ISD_" + string(app.ISDEditField.Value) + ".csv";
    arr = ["NaN", "NaN", "X = " + string(double(app.XEditField.Value)), "Y = " + string(double(app.YEditField.Value)), "Z = " + string(double(app.ZEditField.Value)), "ISD = " + string(double(app.ISDEditField.Value)), "NaN", "NaN"];
    arr2 = [string(double(app.SPS1EditField.Value)), string(double(app.SPS2EditField.Value)), string(double(app.SPS3EditField.Value)), string(double(app.SPS4EditField.Value)), string(double(app.SPS5EditField.Value)), string(double(app.SPS6EditField.Value)), string(double(app.SPS7EditField.Value)), string(double(app.SPS8EditField.Value))];
    arr3 = [string(double(app.SPS1_Curr_EditField.Value)), string(double(app.SPS2_Curr_EditField.Value)), string(double(app.SPS3_Curr_EditField.Value)), string(app.SPS4_Curr_EditField.Value), string(double(app.SPS5_Curr_EditField.Value)), string(double(app.SPS6_Curr_EditField.Value)), string(double(app.SPS7_Curr_EditField.Value)), string(double(app.SPS8_Curr_EditField.Value))];
    v_name = {'Configuration' 'SPS_DN_1_TO_8' 'SPS_curr_mA_1_TO_8'};
    table_csv = array2table([arr, arr2, arr3], 'VariableNames', v_name);
    writetable(table_csv, percorso);
end
end

% Value changed function: SaverelvantvaluesinaCSVCheckBox
function SaverelvantvaluesinaCSVCheckBoxValueChanged(app, ~)
    value = app.SaverelvantvaluesinaCSVCheckBox.Value;
    if (value)
        app.PathCSV_relevant_values.Enable = "on";
        dir = uigetdir;
        figure(app.SIMULATIONPROBA3UIFigure);
        if dir ~= 0
            app.PathCSV_relevant_values.Value = string(dir);
        else
            app.PathCSV_relevant_values.Value = "<ACTUAL MATLAB PATH>";
        end
    else
        app.PathCSV_relevant_values.Enable = "off";
    end
end

% Button pushed function: PLOTButton
function PLOTButtonPushed(app, ~)
    filename_temp = app.PathCSV_EditField_Relevant.Value;
    tab = table2array(readtable(filename_temp));
    switch app.Y_GraphButtonGroup.SelectedObject.Text
        case "Nom_X"
            lab = "Nom X mm";
            y_axis_value = tab(:, 2);
        case "Nom_Y"
            lab = "Nom Y mm";
            y_axis_value = tab(:, 3);
        case "Nom_Z"

```

```

        lab = "Nom Z mm";
        y_axis_value = tab(:, 4);
    case "Red_X"
        lab = "Red X mm";
        y_axis_value = tab(:, 5);
    case "Red_Y"
        lab = "Red Y mm";
        y_axis_value = tab(:, 6);
    case "Red_Z"
        lab = "Red Z mm";
        y_axis_value = tab(:, 7);
    case "Delta_Nom_X"
        lab = "Delta Nom X mm";
        y_axis_value = tab(:, 8);
    case "Delta_Nom_Y"
        lab = "Delta Nom Y mm";
        y_axis_value = tab(:, 9);
    case "Delta_Nom_Z"
        lab = "Delta Nom Z mm";
        y_axis_value = tab(:, 10);
    case "Delta_Red_X"
        lab = "Delta Red X mm";
        y_axis_value = tab(:, 11);
    case "Delta_Red_Y"
        lab = "Delta Red Y mm";
        y_axis_value = tab(:, 12);
    case "Delta_Red_Z"
        lab = "Delta Red Z mm";
        y_axis_value = tab(:, 13);
    end
    x_axis_value = tab(:, 1);
    [y_complex, y_real, x_complex, x_real] = real_complex_vector(app, ✓
y_axis_value, x_axis_value);
    plot(app.RELEVANT_VALUES_PLOT, x_complex, y_complex, 'c:*', x_real, ✓
y_real, 'r:*');
    if isempty(x_complex)
        legend(app.RELEVANT_VALUES_PLOT, 'REAL');
    elseif isempty(x_real)
        legend(app.RELEVANT_VALUES_PLOT, 'COMPLEX');
    else
        legend(app.RELEVANT_VALUES_PLOT, 'COMPLEX', 'REAL');
    end
    app.RELEVANT_VALUES_PLOT.XLabel.String = "N° TEST";
    app.RELEVANT_VALUES_PLOT.YLabel.String = lab;
end

% Button pushed function: SelectpathCSVButton_2
function SelectpathCSVButton_2Pushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        app.PathCSVEditField_Relevant.Value = "" + string(path_temp) + string✓
(file);
        app.PLOTButton.Enable = "on";
    end
end

```

```

end

% Button pushed function:
% Select the folder where to export the CSV Button
function SelectTheFolderWhereToExportTheCSVButtonPushed(app, ~)
    dir = uigetdir;
    figure(app.SIMULATIONPROBA3UIFigure);
    if dir ~= 0
        app.EXport_CSV_Ciotola.Value = string(dir);
    end
end

% Button pushed function: SelectFitParamButton
function SelectFitParamButtonPushed(app, ~)
    [file, path_temp] = uigetfile('*.csv');
    figure(app.SIMULATIONPROBA3UIFigure);
    if file ~= 0
        app.CSV_fit_param.Value = "" + string(path_temp) + string(file);
    end
end

% Selection changed function: ChangeXarrayrangeButtonGroup
function ChangeXarrayrangeButtonGroupSelectionChanged(app, ~)
    selectedButton = app.ChangeXarrayrangeButtonGroup.SelectedObject.Text;
    if selectedButton == "Specular at 0"
        app.StartEditField.Enable = 'on';
        app.StepmeterEditField.Enable = 'on';
        app.End_Edit_Field.Enable = 'off';
        app.StepmeterEditField_2.Enable = 'off';
        app.StartEditField_2.Enable = 'off';
    else
        app.StartEditField.Enable = 'off';
        app.StepmeterEditField.Enable = 'off';
        app.End_Edit_Field.Enable = 'on';
        app.StepmeterEditField_2.Enable = 'on';
        app.StartEditField_2.Enable = 'on';
    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create SIMULATIONPROBA3UIFigure and hide until all components are ✓
created
    app.SIMULATIONPROBA3UIFigure = uifigure('Visible', 'off');
    app.SIMULATIONPROBA3UIFigure.Color = [0.9804 0.9098 0.9098];
    app.SIMULATIONPROBA3UIFigure.Position = [100 100 1513 811];
    app.SIMULATIONPROBA3UIFigure.Name = 'SIMULATION PROBA-3';
    app.SIMULATIONPROBA3UIFigure.Scrollable = 'on';

    % Create REALTIMEPanel

```

```

app.REALTIMEPanel = uipanel(app.SIMULATIONPROBA3UIFigure);
app.REALTIMEPanel.TitlePosition = 'centertop';
app.REALTIMEPanel.Title = 'REAL TIME';
app.REALTIMEPanel.BackgroundColor = [0.8706 0.8941 1];
app.REALTIMEPanel.FontWeight = 'bold';
app.REALTIMEPanel.FontSize = 14;
app.REALTIMEPanel.Position = [883 1 631 811];

% Create FilterListBoxLabel
app.FilterListBoxLabel = uilabel(app.REALTIMEPanel);
app.FilterListBoxLabel.HorizontalAlignment = 'right';
app.FilterListBoxLabel.FontSize = 15;
app.FilterListBoxLabel.FontWeight = 'bold';
app.FilterListBoxLabel.Position = [512 764 42 22];
app.FilterListBoxLabel.Text = 'Filter';

% Create FilterListBox
app.FilterListBox = uilistbox(app.REALTIMEPanel);
app.FilterListBox.Items = {'TEST', 'nom_X', 'nom_Y', 'nom_Z', 'red_X', ↵
'red_Y', 'red_Z', 'R_SPS', 'A_coarse', 'a_coarse', 'b_coarse', 'c_coarse', 'A_fine', ↵
'a_fine', 'b_fine', 'c_fine', 'H', 'K', 'L', 'J', 'delta_R', 'K1', 'K2', 'K3', 'K4', ↵
'K5', 'K6', 'K7', 'K8', 'd_z', 'd_y', 'SPS1', 'SPS3', 'SPS5', 'SPS7', 'SPS2', 'SPS4', ↵
'SPS6', 'SPS8', 'Flag_1_Nom', 'Flag_1_Red', 'Flag_Lat_Nom', 'Flag_Lat_Red', ↵
'Flag_Long_Nom', 'Flag_Long_Red', 'Final_NOM_Flag_1', 'Final_NOM_delta_15', ↵
'Final_NOM_delta_37', 'Final_NOM_Flag_Long', 'Final_RED_Flag_1', 'Final_RED_delta_26', ↵
'Final_RED_delta_48', 'Final_RED_Flag_Long', 'Expected_X', 'Expected_y', 'Expected_Z', ↵
'Delta_X_nom', 'Delta_Y_nom', 'Delta_Z_nom', 'Delta_X_red', 'Delta_Y_red', ↵
'Delta_Z_red', 'flag_acos', 'algo_nom', 'algo_red', 'flag_X_0'};
app.FilterListBox.ItemsData = {'1', '2', '3', '4', '5', '6', '7', '8', ↵
'9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', ↵
'23', '24', '25', '26', '27', '28', '29', '30', '31', '32', '33', '34', '35', '36', ↵
'37', '38', '39', '40', '41', '42', '43', '44', '45', '46', '47', '48', '49', '50', ↵
'51', '52', '53', '54', '55', '56', '57', '58', '59', '60', '61', '62', '63', '64', ↵
'65', '66'};
app.FilterListBox.Multiselect = 'on';
app.FilterListBox.Position = [449 212 168 554];
app.FilterListBox.Value = {'1'};

% Create GETSELECTEDCOLUMNButton
app.GETSELECTEDCOLUMNButton = uibutton(app.REALTIMEPanel, 'state');
app.GETSELECTEDCOLUMNButton.ValueChangedFcn = createCallbackFcn(app, ↵
@GETSELECTEDCOLUMNButtonValueChanged, true);
app.GETSELECTEDCOLUMNButton.Enable = 'off';
app.GETSELECTEDCOLUMNButton.Text = {'GET SELECTED '; 'COLUMN'};
app.GETSELECTEDCOLUMNButton.BackgroundColor = [1 0.9412 0.502];
app.GETSELECTEDCOLUMNButton.FontWeight = 'bold';
app.GETSELECTEDCOLUMNButton.Position = [10 168 168 36];

% Create TabGroup
app.TabGroup = uitabgroup(app.REALTIMEPanel);
app.TabGroup.Position = [5 212 437 574];

% Create Tab_FullTable
app.Tab_FullTable = uitab(app.TabGroup);
app.Tab_FullTable.Title = 'Full Table';

```

```

% Create Full_Table
app.Full_Table = uitable(app.Tab_FullTable);
app.Full_Table.ColumnName = {'TEST'; 'nom_X_mm'; 'nom_Y_mm'; 'nom_Z_mm';↵
'red_X_mm'; 'red_Y_mm'; 'red_Z_mm'; 'R_SPS'; 'A_coarse'; 'a_coarse'; 'b_coarse';↵
'c_coarse'; 'A_fine'; 'a_fine'; 'b_fine'; 'c_fine'; 'H'; 'K'; 'L'; 'J'; 'delta_R';↵
'K1'; 'K2'; 'K3'; 'K4'; 'K5'; 'K6'; 'K7'; 'K8'; 'd_z'; 'd_y'; 'SPS1'; 'SPS3'; 'SPS5';↵
'SPS7'; 'SPS2'; 'SPS4'; 'SPS6'; 'SPS8'; 'Flag_1_Nom'; 'Flag_1_Red'; 'Flag_Lat_Nom';↵
'Flag_Lat_Red'; 'Flag_Long_Nom'; 'Flag_Long_Red'; 'Final_NOM_Flag_1';↵
'Final_NOM_delta_15'; 'Final_NOM_delta_37'; 'Final_NOM_Flag_Long'; 'Final_RED_Flag_1';↵
'Final_RED_delta_26'; 'Final_RED_delta_48'; 'Final_RED_Flag_Long'; 'Expected_X_mm';↵
'Expected_Y_mm'; 'Expected_Z_mm'; 'Delta_X_nom_mm'; 'Delta_Y_nom_mm';↵
'Delta_Z_nom_mm'; 'Delta_X_red_mm'; 'Delta_Y_red_mm'; 'Delta_Z_red_mm'; 'flag_acos';↵
'algo_nom'; 'algo_red'; 'flag_X_0'};
app.Full_Table.RowName = {};
app.Full_Table.Position = [1 0 436 548];

% Create FilteredtableTab
app.FilteredtableTab = uitab(app.TabGroup);
app.FilteredtableTab.Title = 'Filtered table';

% Create Filtered_Table
app.Filtered_Table = uitable(app.FilteredtableTab);
app.Filtered_Table.ColumnName = {' '};
app.Filtered_Table.RowName = {};
app.Filtered_Table.Position = [1 1 436 550];

% Create CREATETWOVARIABLESPLOTButton
app.CREATETWOVARIABLESPLOTButton = uibutton(app.REALTIMEPanel, 'state');
app.CREATETWOVARIABLESPLOTButton.ValueChangedFcn = createCallbackFcn(app,↵
@CREATETWOVARIABLESPLOTButtonValueChanged, true);
app.CREATETWOVARIABLESPLOTButton.Enable = 'off';
app.CREATETWOVARIABLESPLOTButton.Text = {'CREATE TWO'; 'VARIABLES PLOT'};
app.CREATETWOVARIABLESPLOTButton.BackgroundColor = [0.3961 0.4863 0.9412];
app.CREATETWOVARIABLESPLOTButton.FontWeight = 'bold';
app.CREATETWOVARIABLESPLOTButton.Position = [473 170 119 36];

% Create LasttestvaluesPanel
app.LasttestvaluesPanel = uipanel(app.REALTIMEPanel);
app.LasttestvaluesPanel.TitlePosition = 'centertop';
app.LasttestvaluesPanel.Title = 'Last test values';
app.LasttestvaluesPanel.FontWeight = 'bold';
app.LasttestvaluesPanel.Position = [4 60 620 92];

% Create NomLabel
app.NomLabel = uilabel(app.LasttestvaluesPanel);
app.NomLabel.FontWeight = 'bold';
app.NomLabel.Position = [5 47 39 22];
app.NomLabel.Text = 'Nom: ';

% Create Label
app.Label = uilabel(app.LasttestvaluesPanel);
app.Label.Position = [70 47 25 22];
app.Label.Text = '';

```

```
% Create XLabel_2
app.XLabel_2 = uilabel(app.LasttestvaluesPanel);
app.XLabel_2.FontSize = 10;
app.XLabel_2.Position = [39 47 25 22];
app.XLabel_2.Text = 'X =';

% Create Label_nom_x
app.Label_nom_x = uilabel(app.LasttestvaluesPanel);
app.Label_nom_x.FontSize = 10;
app.Label_nom_x.Position = [62 47 71 22];
app.Label_nom_x.Text = '';

% Create YLabel_2
app.YLabel_2 = uilabel(app.LasttestvaluesPanel);
app.YLabel_2.FontSize = 10;
app.YLabel_2.Position = [138 47 25 22];
app.YLabel_2.Text = 'Y =';

% Create Label_nom_y
app.Label_nom_y = uilabel(app.LasttestvaluesPanel);
app.Label_nom_y.FontSize = 10;
app.Label_nom_y.Position = [163 47 69 22];
app.Label_nom_y.Text = '';

% Create ZLabel_2
app.ZLabel_2 = uilabel(app.LasttestvaluesPanel);
app.ZLabel_2.FontSize = 10;
app.ZLabel_2.Position = [238 47 25 22];
app.ZLabel_2.Text = 'Z =';

% Create Label_nom_z
app.Label_nom_z = uilabel(app.LasttestvaluesPanel);
app.Label_nom_z.FontSize = 10;
app.Label_nom_z.Position = [263 47 69 22];
app.Label_nom_z.Text = '';

% Create Flag_1Label
app.Flag_1Label = uilabel(app.LasttestvaluesPanel);
app.Flag_1Label.FontSize = 10;
app.Flag_1Label.Position = [337 47 53 22];
app.Flag_1Label.Text = 'Flag_1 =';

% Create Label_Flag_1_nom
app.Label_Flag_1_nom = uilabel(app.LasttestvaluesPanel);
app.Label_Flag_1_nom.FontSize = 10;
app.Label_Flag_1_nom.Position = [389 47 25 22];
app.Label_Flag_1_nom.Text = '';

% Create Flag_LongLabel
app.Flag_LongLabel = uilabel(app.LasttestvaluesPanel);
app.Flag_LongLabel.FontSize = 10;
app.Flag_LongLabel.Position = [418 47 73 22];
app.Flag_LongLabel.Text = 'Flag_Long =';

% Create Label_Flag_Long_nom
```

```
app.Label_Flag_Long_nom = uilabel(app.LasttestvaluesPanel);
app.Label_Flag_Long_nom.FontSize = 10;
app.Label_Flag_Long_nom.Position = [490 47 25 22];
app.Label_Flag_Long_nom.Text = '';

% Create Flag_LatLabel
app.Flag_LatLabel = uilabel(app.LasttestvaluesPanel);
app.Flag_LatLabel.FontSize = 10;
app.Flag_LatLabel.Position = [517 47 53 22];
app.Flag_LatLabel.Text = 'Flag_Lat =';

% Create Label_Flag_lat_nom
app.Label_Flag_lat_nom = uilabel(app.LasttestvaluesPanel);
app.Label_Flag_lat_nom.FontSize = 10;
app.Label_Flag_lat_nom.Position = [589 47 25 22];
app.Label_Flag_lat_nom.Text = '';

% Create RedLabel
app.RedLabel = uilabel(app.LasttestvaluesPanel);
app.RedLabel.FontWeight = 'bold';
app.RedLabel.Position = [6 15 35 22];
app.RedLabel.Text = 'Red: ';

% Create XLabel_3
app.XLabel_3 = uilabel(app.LasttestvaluesPanel);
app.XLabel_3.FontSize = 10;
app.XLabel_3.Position = [40 15 25 22];
app.XLabel_3.Text = 'X =';

% Create Label_red_x
app.Label_red_x = uilabel(app.LasttestvaluesPanel);
app.Label_red_x.FontSize = 10;
app.Label_red_x.Position = [63 15 71 22];
app.Label_red_x.Text = '';

% Create YLabel_3
app.YLabel_3 = uilabel(app.LasttestvaluesPanel);
app.YLabel_3.FontSize = 10;
app.YLabel_3.Position = [139 15 25 22];
app.YLabel_3.Text = 'Y =';

% Create Label_red_y
app.Label_red_y = uilabel(app.LasttestvaluesPanel);
app.Label_red_y.FontSize = 10;
app.Label_red_y.Position = [164 15 69 22];
app.Label_red_y.Text = '';

% Create ZLabel_3
app.ZLabel_3 = uilabel(app.LasttestvaluesPanel);
app.ZLabel_3.FontSize = 10;
app.ZLabel_3.Position = [239 15 25 22];
app.ZLabel_3.Text = 'Z =';

% Create Label_red_z
app.Label_red_z = uilabel(app.LasttestvaluesPanel);
```

```
app.Label_red_z.FontSize = 10;
app.Label_red_z.Position = [264 15 69 22];
app.Label_red_z.Text = '';

% Create Flag_1Label_2
app.Flag_1Label_2 = uilabel(app.LasttestvaluesPanel);
app.Flag_1Label_2.FontSize = 10;
app.Flag_1Label_2.Position = [338 15 53 22];
app.Flag_1Label_2.Text = 'Flag_1 =';

% Create Label_Flag_1_red
app.Label_Flag_1_red = uilabel(app.LasttestvaluesPanel);
app.Label_Flag_1_red.FontSize = 10;
app.Label_Flag_1_red.Position = [390 15 25 22];
app.Label_Flag_1_red.Text = '';

% Create Flag_LongLabel_2
app.Flag_LongLabel_2 = uilabel(app.LasttestvaluesPanel);
app.Flag_LongLabel_2.FontSize = 10;
app.Flag_LongLabel_2.Position = [419 15 73 22];
app.Flag_LongLabel_2.Text = 'Flag_Long =';

% Create Label_Flag_Long_red
app.Label_Flag_Long_red = uilabel(app.LasttestvaluesPanel);
app.Label_Flag_Long_red.FontSize = 10;
app.Label_Flag_Long_red.Position = [491 15 25 22];
app.Label_Flag_Long_red.Text = '';

% Create Flag_LatLabel_2
app.Flag_LatLabel_2 = uilabel(app.LasttestvaluesPanel);
app.Flag_LatLabel_2.FontSize = 10;
app.Flag_LatLabel_2.Position = [518 15 53 22];
app.Flag_LatLabel_2.Text = 'Flag_Lat =';

% Create Label_Flag_lat_red
app.Label_Flag_lat_red = uilabel(app.LasttestvaluesPanel);
app.Label_Flag_lat_red.FontSize = 10;
app.Label_Flag_lat_red.Position = [590 15 25 22];
app.Label_Flag_lat_red.Text = '';

% Create PresenceofcomplexvaluesPanel
app.PresenceofcomplexvaluesPanel = uipanel(app.REALTIMEPanel);
app.PresenceofcomplexvaluesPanel.TitlePosition = 'centertop';
app.PresenceofcomplexvaluesPanel.Title = 'Presence of complex values';
app.PresenceofcomplexvaluesPanel.FontWeight = 'bold';
app.PresenceofcomplexvaluesPanel.Position = [6 8 618 48];

% Create Lamp
app.Lamp = uilamp(app.PresenceofcomplexvaluesPanel);
app.Lamp.Enable = 'off';
app.Lamp.Position = [298 1 25 25];
app.Lamp.Color = [1 0 0];

% Create ActualtestEditFieldLabel
app.ActualtestEditFieldLabel = uilabel(app.REALTIMEPanel);
```

```

app.ActualtestEditFieldLabel.HorizontalAlignment = 'right';
app.ActualtestEditFieldLabel.Position = [236 175 62 22];
app.ActualtestEditFieldLabel.Text = 'Actual test';

% Create ActualtestEditField
app.ActualtestEditField = uicontrol(app.REALTIMEPanel, 'numeric');
app.ActualtestEditField.Eitable = 'off';
app.ActualtestEditField.Position = [303 176 138 22];

% Create TabGroup2
app.TabGroup2 = uitabgroup(app.SIMULATIONPROBA3UIFigure);
app.TabGroup2.Position = [0 1 880 811];

% Create PENUMBRAPROFILECREATIONTab
app.PENUMBRAPROFILECREATIONTab = uitab(app.TabGroup2);
app.PENUMBRAPROFILECREATIONTab.Title = 'PENUMBRA-PROFILE CREATION';
app.PENUMBRAPROFILECREATIONTab.BackgroundColor = [0.902 0.902 0.5059];

% Create STARTGENERATIONSButton
app.STARTGENERATIONSButton = uicontrol(app.PENUMBRAPROFILECREATIONTab,
'push');
app.STARTGENERATIONSButton.ButtonPushedFcn = createCallbackFcn(app,
@STARTGENERATIONSButtonPushed, true);
app.STARTGENERATIONSButton.BackgroundColor = [0.3373 0.9608 0.3373];
app.STARTGENERATIONSButton.FontSize = 30;
app.STARTGENERATIONSButton.FontWeight = 'bold';
app.STARTGENERATIONSButton.Position = [270 8 344 59];
app.STARTGENERATIONSButton.Text = 'START GENERATIONS';

% Create IMPORTCSVPANELPanel
app.IMPORTCSVPANELPanel = uipanel(app.PENUMBRAPROFILECREATIONTab);
app.IMPORTCSVPANELPanel.TitlePosition = 'centertop';
app.IMPORTCSVPANELPanel.Title = 'IMPORT CSV PANEL';
app.IMPORTCSVPANELPanel.BackgroundColor = [0.549 0.8784 0.9608];
app.IMPORTCSVPANELPanel.FontWeight = 'bold';
app.IMPORTCSVPANELPanel.FontSize = 15;
app.IMPORTCSVPANELPanel.Position = [15 501 852 275];

% Create SSI_solar_min_maxPanel
app.SSI_solar_min_maxPanel = uipanel(app.IMPORTCSVPANELPanel);
app.SSI_solar_min_maxPanel.TitlePosition = 'centertop';
app.SSI_solar_min_maxPanel.Title = 'SSI_solar_min_max';
app.SSI_solar_min_maxPanel.FontWeight = 'bold';
app.SSI_solar_min_maxPanel.Position = [11 139 260 108];

% Create SSI_solarmin_CSV
app.SSI_solarmin_CSV = uicontrol(app.SSI_solar_min_maxPanel, 'text');
app.SSI_solarmin_CSV.Eitable = 'off';
app.SSI_solarmin_CSV.Position = [5 6 251 22];
app.SSI_solarmin_CSV.Value = 'NaN';

% Create SelectSSI_solar_min_maxCSVButton
app.SelectSSI_solar_min_maxCSVButton = uicontrol(app,
SSI_solar_min_maxPanel, 'push');
app.SelectSSI_solar_min_maxCSVButton.ButtonPushedFcn = createCallbackFcn

```

```

(app, @SelectSSI_solar_min_maxCSVButtonPushed, true);
    app.SelectSSI_solar_min_maxCSVButton.BackgroundColor = [0.6745 0.8039
0.8902];
    app.SelectSSI_solar_min_maxCSVButton.FontWeight = 'bold';
    app.SelectSSI_solar_min_maxCSVButton.Position = [31.5 57 197 22];
    app.SelectSSI_solar_min_maxCSVButton.Text = 'Select SSI_solar_min_max
CSV';

    % Create PathSSI_solar_min_maxCSVLabel
    app.PathSSI_solar_min_maxCSVLabel = uilabel(app.SSI_solar_min_maxPanel);
    app.PathSSI_solar_min_maxCSVLabel.HorizontalAlignment = 'center';
    app.PathSSI_solar_min_maxCSVLabel.FontWeight = 'bold';
    app.PathSSI_solar_min_maxCSVLabel.Position = [33 29 196 22];
    app.PathSSI_solar_min_maxCSVLabel.Text = 'Path SSI_solar_min_max CSV: ';

    % Create Responsivity_temp_dependence_m20degC_to_60degCPanel
    app.Responsivity_temp_dependence_m20degC_to_60degCPanel = uipanel(app.
IMPORTCSVPanelPanel);
    app.Responsivity_temp_dependence_m20degC_to_60degCPanel.TitlePosition =
'centertop';
    app.Responsivity_temp_dependence_m20degC_to_60degCPanel.Title =
'Responsivity_temp_dependence_m20degC_to_60degC';
    app.Responsivity_temp_dependence_m20degC_to_60degCPanel.FontWeight =
'bold';
    app.Responsivity_temp_dependence_m20degC_to_60degCPanel.Position = [289
139 542 108];

    % Create PathResponsivity_temp_dependence_m20degC_to_60degCLabel
    app.PathResponsivity_temp_dependence_m20degC_to_60degCLabel = uilabel(app.
Responsivity_temp_dependence_m20degC_to_60degCPanel);
    app.PathResponsivity_temp_dependence_m20degC_to_60degCLabel.
HorizontalAlignment = 'center';
    app.PathResponsivity_temp_dependence_m20degC_to_60degCLabel.FontWeight =
'bold';
    app.PathResponsivity_temp_dependence_m20degC_to_60degCLabel.Position = [30
32 508 22];
    app.PathResponsivity_temp_dependence_m20degC_to_60degCLabel.Text = 'Path
Responsivity_temp_dependence_m20degC_to_60degC: ';

    % Create resp_t_dep_m20degC_to_60degCCSV
    app.resp_t_dep_m20degC_to_60degCCSV = uieditfield(app.
Responsivity_temp_dependence_m20degC_to_60degCPanel, 'text');
    app.resp_t_dep_m20degC_to_60degCCSV.Editable = 'off';
    app.resp_t_dep_m20degC_to_60degCCSV.Position = [8 10 530 22];
    app.resp_t_dep_m20degC_to_60degCCSV.Value = 'NaN';

    % Create Select_resp_t_dep_m20degC_to_60degCCSVButton
    app.Select_resp_t_dep_m20degC_to_60degCCSVButton = uibutton(app.
Responsivity_temp_dependence_m20degC_to_60degCPanel, 'push');
    app.Select_resp_t_dep_m20degC_to_60degCCSVButton.ButtonPushedFcn =
createCallbackFcn(app, @Select_resp_t_dep_m20degC_to_60degCCSVButtonPushed, true);
    app.Select_resp_t_dep_m20degC_to_60degCCSVButton.BackgroundColor = [0.6745
0.8039 0.8902];
    app.Select_resp_t_dep_m20degC_to_60degCCSVButton.FontWeight = 'bold';
    app.Select_resp_t_dep_m20degC_to_60degCCSVButton.Position = [8 57 530 22];

```

```

    app.Select_resp_t_dep_m20degC_to_60degCCSVButton.Text = 'Select
Responsivity_temp_dependence_m20degC_to_60degC CSV';

    % Create Ld_coefficients_CSVPanel
    app.Ld_coefficients_CSVPanel = uipanel(app.IMPORTCSV_PANEL);
    app.Ld_coefficients_CSVPanel.TitlePosition = 'centertop';
    app.Ld_coefficients_CSVPanel.Title = 'Ld_coefficients_CSV';
    app.Ld_coefficients_CSVPanel.FontWeight = 'bold';
    app.Ld_coefficients_CSVPanel.Position = [7 11 260 107];

    % Create Ld_coefficients_CSV
    app.Ld_coefficients_CSV = uieditfield(app.Ld_coefficients_CSVPanel,
'text');
    app.Ld_coefficients_CSV.Editable = 'off';
    app.Ld_coefficients_CSV.Position = [5 8 251 22];
    app.Ld_coefficients_CSV.Value = 'NaN';

    % Create SelectLd_coefficientsCSVButton
    app.SelectLd_coefficientsCSVButton = uibutton(app.
Ld_coefficients_CSVPanel, 'push');
    app.SelectLd_coefficientsCSVButton.ButtonPushedFcn = createCallbackFcn
(app, @SelectLd_coefficientsCSVButtonPushed, true);
    app.SelectLd_coefficientsCSVButton.BackgroundColor = [0.6745 0.8039
0.8902];
    app.SelectLd_coefficientsCSVButton.FontWeight = 'bold';
    app.SelectLd_coefficientsCSVButton.Position = [45 59 170 22];
    app.SelectLd_coefficientsCSVButton.Text = 'Select Ld_coefficients CSV';

    % Create PathLd_coefficientsCSVLabel
    app.PathLd_coefficientsCSVLabel = uilabel(app.Ld_coefficients_CSVPanel);
    app.PathLd_coefficientsCSVLabel.FontWeight = 'bold';
    app.PathLd_coefficientsCSVLabel.Position = [52 34 158 22];
    app.PathLd_coefficientsCSVLabel.Text = 'Path Ld_coefficients CSV: ';

    % Create Glass_trasmittancePanel
    app.Glass_trasmittancePanel = uipanel(app.IMPORTCSV_PANEL);
    app.Glass_trasmittancePanel.TitlePosition = 'centertop';
    app.Glass_trasmittancePanel.Title = 'Glass_trasmittance';
    app.Glass_trasmittancePanel.FontWeight = 'bold';
    app.Glass_trasmittancePanel.Position = [289 11 260 107];

    % Create Glass_trasmittance_CSV
    app.Glass_trasmittance_CSV = uieditfield(app.Glass_trasmittancePanel,
'text');
    app.Glass_trasmittance_CSV.Editable = 'off';
    app.Glass_trasmittance_CSV.Position = [5 4 251 22];
    app.Glass_trasmittance_CSV.Value = 'NaN';

    % Create SelectGlass_trasmittanceCSVButton
    app.SelectGlass_trasmittanceCSVButton = uibutton(app.
Glass_trasmittancePanel, 'push');
    app.SelectGlass_trasmittanceCSVButton.ButtonPushedFcn = createCallbackFcn
(app, @SelectGlass_trasmittanceCSVButtonPushed, true);
    app.SelectGlass_trasmittanceCSVButton.BackgroundColor = [0.6745 0.8039
0.8902];

```

```

app.SelectGlass_trasmittanceCSVButton.FontWeight = 'bold';
app.SelectGlass_trasmittanceCSVButton.Position = [33 56 194 22];
app.SelectGlass_trasmittanceCSVButton.Text = 'Select Glass_trasmittance
CSV';

% Create PathGlass_trasmittanceCSVLabel
app.PathGlass_trasmittanceCSVLabel = uilabel(app.Glass_trasmittancePanel);
app.PathGlass_trasmittanceCSVLabel.FontWeight = 'bold';
app.PathGlass_trasmittanceCSVLabel.Position = [41 30 180 22];
app.PathGlass_trasmittanceCSVLabel.Text = 'Path Glass_trasmittance CSV: ';

% Create Coating_trasmittancePanel
app.Coating_trasmittancePanel = uipanel(app.IMPORTCSVPanel);
app.Coating_trasmittancePanel.TitlePosition = 'centertop';
app.Coating_trasmittancePanel.Title = 'Coating_trasmittance';
app.Coating_trasmittancePanel.FontWeight = 'bold';
app.Coating_trasmittancePanel.Position = [571 11 260 107];

% Create Coating_trasmittance_CSV
app.Coating_trasmittance_CSV = uieditfield(app.Coating_trasmittancePanel,
'text');
app.Coating_trasmittance_CSV.Editable = 'off';
app.Coating_trasmittance_CSV.Position = [5 4 251 22];
app.Coating_trasmittance_CSV.Value = 'NaN';

% Create SelectCoating_trasmittanceCSVButton
app.SelectCoating_trasmittanceCSVButton = uibutton(app.
Coating_trasmittancePanel, 'push');
app.SelectCoating_trasmittanceCSVButton.ButtonPushedFcn =
createCallbackFcn(app, @SelectCoating_trasmittanceCSVButtonPushed, true);
app.SelectCoating_trasmittanceCSVButton.BackgroundColor = [0.6745 0.8039
0.8902];
app.SelectCoating_trasmittanceCSVButton.FontWeight = 'bold';
app.SelectCoating_trasmittanceCSVButton.Position = [27 56 206 22];
app.SelectCoating_trasmittanceCSVButton.Text = 'Select
Coating_trasmittance CSV';

% Create PathCoating_trasmittanceCSVLabel
app.PathCoating_trasmittanceCSVLabel = uilabel(app.
Coating_trasmittancePanel);
app.PathCoating_trasmittanceCSVLabel.FontWeight = 'bold';
app.PathCoating_trasmittanceCSVLabel.Position = [34 30 193 22];
app.PathCoating_trasmittanceCSVLabel.Text = 'Path Coating_trasmittance
CSV: ';

% Create ALGORITHMPARAMETERSPanel
app.ALGORITHMPARAMETERSPanel = uipanel(app.PENUMBRAPROFILECREATIONTab);
app.ALGORITHMPARAMETERSPanel.TitlePosition = 'centertop';
app.ALGORITHMPARAMETERSPanel.Title = 'ALGORITHM PARAMETERS';
app.ALGORITHMPARAMETERSPanel.BackgroundColor = [0.9804 0.6078 0.5882];
app.ALGORITHMPARAMETERSPanel.FontWeight = 'bold';
app.ALGORITHMPARAMETERSPanel.FontSize = 15;
app.ALGORITHMPARAMETERSPanel.Position = [15 98 852 392];

% Create PATHEXPORTCSVPanel

```

```

app.PATHEXPORTCSVPanel = uipanel(app.ALGORITHMPARAMETERSPanel);
app.PATHEXPORTCSVPanel.TitlePosition = 'centertop';
app.PATHEXPORTCSVPanel.Title = 'PATH EXPORT CSV';
app.PATHEXPORTCSVPanel.FontWeight = 'bold';
app.PATHEXPORTCSVPanel.Position = [16 98 813 104];

% Create EXport_CSV_Ciotola
app.EXport_CSV_Ciotola = uieditfield(app.PATHEXPORTCSVPanel, 'text');
app.EXport_CSV_Ciotola.Editable = 'off';
app.EXport_CSV_Ciotola.Position = [10 3 792 22];
app.EXport_CSV_Ciotola.Value = '<ACTUAL MATLAB PATH>';

% Create SelectthefolderwheretoeexporttheCSVButton
app.SelectthefolderwheretoeexporttheCSVButton = uibutton(app.
PATHEXPORTCSVPanel, 'push');
app.SelectthefolderwheretoeexporttheCSVButton.ButtonPushedFcn =
createCallbackFcn(app, @SelectthefolderwheretoeexporttheCSVButtonPushed, true);
app.SelectthefolderwheretoeexporttheCSVButton.BackgroundColor = [0.6745
0.8039 0.8902];
app.SelectthefolderwheretoeexporttheCSVButton.FontSize = 17;
app.SelectthefolderwheretoeexporttheCSVButton.FontWeight = 'bold';
app.SelectthefolderwheretoeexporttheCSVButton.Position = [10 47 792 33];
app.SelectthefolderwheretoeexporttheCSVButton.Text = 'Select the folder
where to export the CSV';

% Create PATHLabel
app.PATHLabel = uilabel(app.PATHEXPORTCSVPanel);
app.PATHLabel.HorizontalAlignment = 'center';
app.PATHLabel.FontWeight = 'bold';
app.PATHLabel.Position = [8 22 792 22];
app.PATHLabel.Text = 'PATH:';

% Create Iteration
app.Iteration = uilabel(app.ALGORITHMPARAMETERSPanel);
app.Iteration.BackgroundColor = [1 1 1];
app.Iteration.HorizontalAlignment = 'center';
app.Iteration.FontWeight = 'bold';
app.Iteration.FontColor = [1 0.3216 0.3216];
app.Iteration.Position = [19 55 101 22];
app.Iteration.Text = '1 / N';

% Create Gauge
app.Gauge = uigauge(app.ALGORITHMPARAMETERSPanel, 'linear');
app.Gauge.Limits = [1 5];
app.Gauge.MajorTicks = [1 2 3 4 5];
app.Gauge.MinorTicks = [];
app.Gauge.FontWeight = 'bold';
app.Gauge.FontColor = [1 0.3216 0.3216];
app.Gauge.Position = [130 48 701 40];

% Create MacroIterationLabel
app.MacroIterationLabel = uilabel(app.ALGORITHMPARAMETERSPanel);
app.MacroIterationLabel.HorizontalAlignment = 'center';
app.MacroIterationLabel.FontWeight = 'bold';
app.MacroIterationLabel.Position = [19 77 101 22];

```

```

app.MacroIterationLabel.Text = 'Macro - Iteration';

% Create Iteration_2
app.Iteration_2 = uilabel(app.ALGORITHMPARAMETERSPanel);
app.Iteration_2.BackgroundColor = [1 1 1];
app.Iteration_2.HorizontalAlignment = 'center';
app.Iteration_2.FontWeight = 'bold';
app.Iteration_2.FontColor = [0.5098 0.702 0.9804];
app.Iteration_2.Position = [16 4 106 22];
app.Iteration_2.Text = '1 / 16';

% Create Gauge_2
app.Gauge_2 = uigauge(app.ALGORITHMPARAMETERSPanel, 'linear');
app.Gauge_2.Limits = [1 16];
app.Gauge_2.MajorTicks = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];
app.Gauge_2.MinorTicks = [];
app.Gauge_2.FontWeight = 'bold';
app.Gauge_2.FontColor = [0.5098 0.702 0.9804];
app.Gauge_2.Position = [130 4 701 40];

% Create SubIterationLabel
app.SubIterationLabel = uilabel(app.ALGORITHMPARAMETERSPanel);
app.SubIterationLabel.HorizontalAlignment = 'center';
app.SubIterationLabel.FontWeight = 'bold';
app.SubIterationLabel.Position = [16 25 106 22];
app.SubIterationLabel.Text = 'Sub - Iteration';

% Create LongitudinalRangePanel
app.LongitudinalRangePanel = uipanel(app.ALGORITHMPARAMETERSPanel);
app.LongitudinalRangePanel.TitlePosition = 'centertop';
app.LongitudinalRangePanel.Title = 'Longitudinal Range';
app.LongitudinalRangePanel.BackgroundColor = [0.949 0.8275 0.5216];
app.LongitudinalRangePanel.FontWeight = 'bold';
app.LongitudinalRangePanel.Position = [16 207 811 157];

% Create Specularat0meterPanel
app.Specularat0meterPanel = uipanel(app.LongitudinalRangePanel);
app.Specularat0meterPanel.TitlePosition = 'centertop';
app.Specularat0meterPanel.Title = 'Specular at 0 (meter)';
app.Specularat0meterPanel.FontWeight = 'bold';
app.Specularat0meterPanel.Position = [220 5 290 128];

% Create StartEditField
app.StartEditField = uieditfield(app.Specularat0meterPanel, 'numeric');
app.StartEditField.Position = [188 30 62 22];
app.StartEditField.Value = -0.075;

% Create StepmeterEditFieldLabel
app.StepmeterEditFieldLabel = uilabel(app.Specularat0meterPanel);
app.StepmeterEditFieldLabel.HorizontalAlignment = 'right';
app.StepmeterEditFieldLabel.FontWeight = 'bold';
app.StepmeterEditFieldLabel.Position = [26 7 75 22];
app.StepmeterEditFieldLabel.Text = 'Step (meter)';

% Create StepmeterEditField

```

```

        app.StepmeterEditField = uieditfield(app.Specularat0meterPanel,
'numeric');
        app.StepmeterEditField.Position = [188 7 62 22];
        app.StepmeterEditField.Value = 0.05;

        % Create StartEditFieldLabel
        app.StartEditFieldLabel = uilabel(app.Specularat0meterPanel);
        app.StartEditFieldLabel.HorizontalAlignment = 'right';
        app.StartEditFieldLabel.FontWeight = 'bold';
        app.StartEditFieldLabel.Position = [26 28 76 22];
        app.StartEditFieldLabel.Text = 'Start (meter)';

        % Create Label_3
        app.Label_3 = uilabel(app.Specularat0meterPanel);
        app.Label_3.HorizontalAlignment = 'center';
        app.Label_3.VerticalAlignment = 'top';
        app.Label_3.FontSize = 11;
        app.Label_3.Position = [19 55 253 49];
        app.Label_3.Text = {'To generate the fit parameters the algorithm needs '
'at least 3 X values (ex with start at -0.025 ' ; 'and step at 0.05 the algo will '
'generates three csv [-0.025 0 0.025])'};

        % Create ChangeXarrayrangeButtonGroup
        app.ChangeXarrayrangeButtonGroup = uibuttongroup(app.
LongitudinalRangePanel);
        app.ChangeXarrayrangeButtonGroup.SelectionChangedFcn = createCallbackFcn
(app, @ChangeXarrayrangeButtonGroupSelectionChanged, true);
        app.ChangeXarrayrangeButtonGroup.TitlePosition = 'centertop';
        app.ChangeXarrayrangeButtonGroup.Title = 'Change X array range';
        app.ChangeXarrayrangeButtonGroup.FontWeight = 'bold';
        app.ChangeXarrayrangeButtonGroup.Position = [21 29 174 70];

        % Create Specularat0Button
        app.Specularat0Button = uiradiobutton(app.ChangeXarrayrangeButtonGroup);
        app.Specularat0Button.Text = 'Specular at 0';
        app.Specularat0Button.Position = [11 24 92 22];
        app.Specularat0Button.Value = true;

        % Create StartStepEndButton
        app.StartStepEndButton = uiradiobutton(app.ChangeXarrayrangeButtonGroup);
        app.StartStepEndButton.Text = 'Start-Step-End';
        app.StartStepEndButton.Position = [11 2 101 22];

        % Create StartStepEndmeterPanel
        app.StartStepEndmeterPanel = uipanel(app.LongitudinalRangePanel);
        app.StartStepEndmeterPanel.TitlePosition = 'centertop';
        app.StartStepEndmeterPanel.Title = 'Start - Step - End (meter)';
        app.StartStepEndmeterPanel.FontWeight = 'bold';
        app.StartStepEndmeterPanel.Position = [532 5 260 128];

        % Create StartEditField_2
        app.StartEditField_2 = uieditfield(app.StartStepEndmeterPanel, 'numeric');
        app.StartEditField_2.Enable = 'off';
        app.StartEditField_2.Position = [181 46 62 22];
        app.StartEditField_2.Value = -0.075;

```

```

% Create StepmeterEditFieldLabel_2
app.StepmeterEditFieldLabel_2 = uilabel(app.StartStepEndmeterPanel);
app.StepmeterEditFieldLabel_2.FontWeight = 'bold';
app.StepmeterEditFieldLabel_2.Position = [30 23 75 22];
app.StepmeterEditFieldLabel_2.Text = 'Step (meter)';

% Create StepmeterEditField_2
app.StepmeterEditField_2 = uieditfield(app.StartStepEndmeterPanel,
'numeric');
app.StepmeterEditField_2.Enable = 'off';
app.StepmeterEditField_2.Position = [181 23 62 22];
app.StepmeterEditField_2.Value = 0.05;

% Create StartEditFieldLabel_2
app.StartEditFieldLabel_2 = uilabel(app.StartStepEndmeterPanel);
app.StartEditFieldLabel_2.FontWeight = 'bold';
app.StartEditFieldLabel_2.Position = [31 46 76 22];
app.StartEditFieldLabel_2.Text = 'Start (meter)';

% Create End_Edit_Field
app.End_Edit_Field = uieditfield(app.StartStepEndmeterPanel, 'numeric');
app.End_Edit_Field.Enable = 'off';
app.End_Edit_Field.Position = [181 0 62 22];
app.End_Edit_Field.Value = 0.075;

% Create EndEditField
app.EndEditField = uilabel(app.StartStepEndmeterPanel);
app.EndEditField.FontWeight = 'bold';
app.EndEditField.Position = [32 0 72 22];
app.EndEditField.Text = 'End (meter)';

% Create Label_4
app.Label_4 = uilabel(app.StartStepEndmeterPanel);
app.Label_4.HorizontalAlignment = 'center';
app.Label_4.VerticalAlignment = 'top';
app.Label_4.FontSize = 11;
app.Label_4.Position = [4 67 253 37];
app.Label_4.Text = {'- Initialize the range at "Start"; '- Step it with a
value equal to "Step"; '- End it at "End" (putting 0 in the middle)'};

% Create CalculatefitparamsCheckBox
app.CalculatefitparamsCheckBox = uicheckbox(app.
PENUMBRAPROFILECREATIONTab);
app.CalculatefitparamsCheckBox.Text = 'Calculate fit params';
app.CalculatefitparamsCheckBox.FontSize = 22;
app.CalculatefitparamsCheckBox.FontWeight = 'bold';
app.CalculatefitparamsCheckBox.Position = [325 66 233 26];

% Create XYZtoSPSTab
app.XYZtoSPSTab = uitab(app.TabGroup2);
app.XYZtoSPSTab.Title = '[X, Y, Z] to [SPS]';

% Create RESULTSPanel
app.RESULTSPanel = uipanel(app.XYZtoSPSTab);

```

```
app.RESULTSPanel.TitlePosition = 'centertop';
app.RESULTSPanel.Title = 'RESULTS';
app.RESULTSPanel.BackgroundColor = [0.902 0.4902 0.4902];
app.RESULTSPanel.FontWeight = 'bold';
app.RESULTSPanel.FontSize = 14;
app.RESULTSPanel.Position = [1 0 878 554];

% Create PLOT_SPS
app.PLOT_SPS = uiaxes(app.RESULTSPanel);
title(app.PLOT_SPS, 'SPS Representation')
xlabel(app.PLOT_SPS, '')
ylabel(app.PLOT_SPS, '')
app.PLOT_SPS.FontWeight = 'bold';
app.PLOT_SPS.BackgroundColor = [0.902 0.4902 0.4902];
app.PLOT_SPS.Position = [12 9 537 508];

% Create SPSPanel_3
app.SPSPanel_3 = uipanel(app.RESULTSPanel);
app.SPSPanel_3.TitlePosition = 'centertop';
app.SPSPanel_3.Title = 'SPS ';
app.SPSPanel_3.FontWeight = 'bold';
app.SPSPanel_3.Position = [560 305 308 223];

% Create SPSPanel_3_EditFieldLabel
app.SPSPanel_3_EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPSPanel_3_EditFieldLabel.HorizontalAlignment = 'right';
app.SPSPanel_3_EditFieldLabel.FontWeight = 'bold';
app.SPSPanel_3_EditFieldLabel.Position = [6 156 40 22];
app.SPSPanel_3_EditFieldLabel.Text = 'SPS 1';

% Create SPSPanel_3_EditField
app.SPSPanel_3_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPSPanel_3_EditField.ValueDisplayFormat = '%11.14g';
app.SPSPanel_3_EditField.Editable = 'off';
app.SPSPanel_3_EditField.Position = [61 156 87 22];

% Create SPSPanel_3_EditFieldLabel
app.SPSPanel_3_EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPSPanel_3_EditFieldLabel.HorizontalAlignment = 'right';
app.SPSPanel_3_EditFieldLabel.FontWeight = 'bold';
app.SPSPanel_3_EditFieldLabel.Position = [6 135 40 22];
app.SPSPanel_3_EditFieldLabel.Text = 'SPS 2';

% Create SPSPanel_3_EditField
app.SPSPanel_3_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPSPanel_3_EditField.ValueDisplayFormat = '%11.14g';
app.SPSPanel_3_EditField.Editable = 'off';
app.SPSPanel_3_EditField.Position = [61 135 87 22];

% Create SPSPanel_3_EditFieldLabel
app.SPSPanel_3_EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPSPanel_3_EditFieldLabel.HorizontalAlignment = 'right';
app.SPSPanel_3_EditFieldLabel.FontWeight = 'bold';
app.SPSPanel_3_EditFieldLabel.Position = [6 114 40 22];
app.SPSPanel_3_EditFieldLabel.Text = 'SPS 3';
```

```
% Create SPS3EditField
app.SPS3EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS3EditField.ValueDisplayFormat = '%11.14g';
app.SPS3EditField.Eitable = 'off';
app.SPS3EditField.Position = [61 114 87 22];

% Create SPS4EditFieldLabel
app.SPS4EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPS4EditFieldLabel.HorizontalAlignment = 'right';
app.SPS4EditFieldLabel.FontWeight = 'bold';
app.SPS4EditFieldLabel.Position = [6 92 40 22];
app.SPS4EditFieldLabel.Text = 'SPS 4';

% Create SPS4EditField
app.SPS4EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS4EditField.ValueDisplayFormat = '%11.14g';
app.SPS4EditField.Eitable = 'off';
app.SPS4EditField.Position = [61 92 87 22];

% Create SPS5EditFieldLabel
app.SPS5EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPS5EditFieldLabel.HorizontalAlignment = 'right';
app.SPS5EditFieldLabel.FontWeight = 'bold';
app.SPS5EditFieldLabel.Position = [6 71 40 22];
app.SPS5EditFieldLabel.Text = 'SPS 5';

% Create SPS5EditField
app.SPS5EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS5EditField.ValueDisplayFormat = '%11.14g';
app.SPS5EditField.Eitable = 'off';
app.SPS5EditField.Position = [61 71 87 22];

% Create SPS6EditFieldLabel
app.SPS6EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPS6EditFieldLabel.HorizontalAlignment = 'right';
app.SPS6EditFieldLabel.FontWeight = 'bold';
app.SPS6EditFieldLabel.Position = [6 50 40 22];
app.SPS6EditFieldLabel.Text = 'SPS 6';

% Create SPS6EditField
app.SPS6EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS6EditField.ValueDisplayFormat = '%11.14g';
app.SPS6EditField.Eitable = 'off';
app.SPS6EditField.Position = [61 50 87 22];

% Create SPS7EditFieldLabel
app.SPS7EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPS7EditFieldLabel.HorizontalAlignment = 'right';
app.SPS7EditFieldLabel.FontWeight = 'bold';
app.SPS7EditFieldLabel.Position = [6 29 40 22];
app.SPS7EditFieldLabel.Text = 'SPS 7';

% Create SPS7EditField
app.SPS7EditField = uieditfield(app.SPSPanel_3, 'numeric');
```

```
app.SPS7EditField.ValueDisplayFormat = '%11.14g';
app.SPS7EditField.Editable = 'off';
app.SPS7EditField.Position = [61 29 87 22];

% Create SPS8EditFieldLabel
app.SPS8EditFieldLabel = uilabel(app.SPSPanel_3);
app.SPS8EditFieldLabel.HorizontalAlignment = 'right';
app.SPS8EditFieldLabel.FontWeight = 'bold';
app.SPS8EditFieldLabel.Position = [6 8 40 22];
app.SPS8EditFieldLabel.Text = 'SPS 8';

% Create SPS8EditField
app.SPS8EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS8EditField.ValueDisplayFormat = '%11.14g';
app.SPS8EditField.Editable = 'off';
app.SPS8EditField.Position = [61 8 87 22];

% Create SPS1_Curr_EditField
app.SPS1_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS1_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS1_Curr_EditField.Editable = 'off';
app.SPS1_Curr_EditField.Position = [157 156 143 22];

% Create SPS2_Curr_EditField
app.SPS2_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS2_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS2_Curr_EditField.Editable = 'off';
app.SPS2_Curr_EditField.Position = [157 135 143 22];

% Create SPS3_Curr_EditField
app.SPS3_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS3_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS3_Curr_EditField.Editable = 'off';
app.SPS3_Curr_EditField.Position = [157 114 143 22];

% Create SPS4_Curr_EditField
app.SPS4_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS4_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS4_Curr_EditField.Editable = 'off';
app.SPS4_Curr_EditField.Position = [157 92 143 22];

% Create SPS5_Curr_EditField
app.SPS5_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS5_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS5_Curr_EditField.Editable = 'off';
app.SPS5_Curr_EditField.Position = [157 71 143 22];

% Create SPS6_Curr_EditField
app.SPS6_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS6_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS6_Curr_EditField.Editable = 'off';
app.SPS6_Curr_EditField.Position = [157 50 143 22];

% Create SPS7_Curr_EditField
app.SPS7_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
```

```

app.SPS7_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS7_Curr_EditField.Editable = 'off';
app.SPS7_Curr_EditField.Position = [157 29 143 22];

% Create SPS8_Curr_EditField
app.SPS8_Curr_EditField = uieditfield(app.SPSPanel_3, 'numeric');
app.SPS8_Curr_EditField.ValueDisplayFormat = '%11.14g';
app.SPS8_Curr_EditField.Editable = 'off';
app.SPS8_Curr_EditField.Position = [157 8 143 22];

% Create DNLabel
app.DNLabel = uilabel(app.SPSPanel_3);
app.DNLabel.HorizontalAlignment = 'center';
app.DNLabel.FontWeight = 'bold';
app.DNLabel.Position = [93 177 25 22];
app.DNLabel.Text = 'DN';

% Create CurrmALabel
app.CurrmALabel = uilabel(app.SPSPanel_3);
app.CurrmALabel.HorizontalAlignment = 'center';
app.CurrmALabel.FontWeight = 'bold';
app.CurrmALabel.Position = [216 177 56 22];
app.CurrmALabel.Text = 'Curr. mA';

% Create EXPORTDNTOSPSTOXYZButton
app.EXPORTDNTOSPSTOXYZButton = uibutton(app.RESULTSPanel, 'push');
app.EXPORTDNTOSPSTOXYZButton.ButtonPushedFcn = createCallbackFcn(app,
@EXPORTDNTOSPSTOXYZButtonPushed, true);
app.EXPORTDNTOSPSTOXYZButton.BackgroundColor = [0.5529 0.8784 0.502];
app.EXPORTDNTOSPSTOXYZButton.FontSize = 13;
app.EXPORTDNTOSPSTOXYZButton.FontWeight = 'bold';
app.EXPORTDNTOSPSTOXYZButton.Enable = 'off';
app.EXPORTDNTOSPSTOXYZButton.Position = [560 248 191 31];
app.EXPORTDNTOSPSTOXYZButton.Text = 'EXPORT DN TO SPS-TO-XYZ';

% Create JSONExportPanel
app.JSONExportPanel = uipanel(app.RESULTSPanel);
app.JSONExportPanel.TitlePosition = 'centertop';
app.JSONExportPanel.Title = 'JSON Export';
app.JSONExportPanel.BackgroundColor = [0.9412 0.9412 0.9412];
app.JSONExportPanel.FontWeight = 'bold';
app.JSONExportPanel.Position = [560 9 308 234];

% Create EXPORTINJSONButton
app.EXPORTINJSONButton = uibutton(app.JSONExportPanel, 'push');
app.EXPORTINJSONButton.ButtonPushedFcn = createCallbackFcn(app,
@EXPORTINJSONButtonPushed, true);
app.EXPORTINJSONButton.BackgroundColor = [0.5059 0.651 0.8706];
app.EXPORTINJSONButton.FontSize = 15;
app.EXPORTINJSONButton.FontWeight = 'bold';
app.EXPORTINJSONButton.Enable = 'off';
app.EXPORTINJSONButton.Position = [13 9 283 41];
app.EXPORTINJSONButton.Text = 'EXPORT IN JSON';

% Create Json_filename

```

```

app.Json_filename = uicontrolfield(app.JSONExportPanel, 'text');
app.Json_filename.Enable = 'off';
app.Json_filename.Position = [14 60 282 22];
app.Json_filename.Value = 'file!';

% Create FilenameDonotwritejsonLabel
app.FilenameDonotwritejsonLabel = uicontrol(app.JSONExportPanel);
app.FilenameDonotwritejsonLabel.HorizontalAlignment = 'center';
app.FilenameDonotwritejsonLabel.FontWeight = 'bold';
app.FilenameDonotwritejsonLabel.Position = [14 89 282 22];
app.FilenameDonotwritejsonLabel.Text = 'File name. Do not write ".json"';

% Create Label_2
app.Label_2 = uicontrol(app.JSONExportPanel);
app.Label_2.BackgroundColor = [1 1 0];
app.Label_2.HorizontalAlignment = 'center';
app.Label_2.VerticalAlignment = 'top';
app.Label_2.FontWeight = 'bold';
app.Label_2.Position = [12 110 284 98];
app.Label_2.Text = {'NB IMPORTANT: this button will ' ; 'create a JSON file'
that can be ' ; 'imported in the other algorithm. ' ; 'This tool will write the ' ;
'Reconf_params values using ' ; 'the values written in the ' ; 'SPS-TO-XYZ tab.'};

% Create SAVECURRTOCSVButton
app.SAVECURRTOCSVButton = uicontrol(app.RESULTSPanel, 'push');
app.SAVECURRTOCSVButton.ButtonPushedFcn = createCallbackFcn(app,
@SAVECURRTOCSVButtonPushed, true);
app.SAVECURRTOCSVButton.BackgroundColor = [0.451 0.9412 0.9412];
app.SAVECURRTOCSVButton.FontSize = 13;
app.SAVECURRTOCSVButton.FontWeight = 'bold';
app.SAVECURRTOCSVButton.Enable = 'off';
app.SAVECURRTOCSVButton.Position = [759 247 109 45];
app.SAVECURRTOCSVButton.Text = {'SAVE CURR' ; 'TO CSV'};

% Create ExportfitparamsCheckBox
app.ExportfitparamsCheckBox = uicontrol(app.RESULTSPanel);
app.ExportfitparamsCheckBox.Enable = 'off';
app.ExportfitparamsCheckBox.Text = 'Export fit params';
app.ExportfitparamsCheckBox.FontWeight = 'bold';
app.ExportfitparamsCheckBox.Position = [596 281 121 22];

% Create INPUTPARAMSPanel
app.INPUTPARAMSPanel = uicontrol(app.XYZtoSPSTab);
app.INPUTPARAMSPanel.TitlePosition = 'centertop';
app.INPUTPARAMSPanel.Title = 'INPUT PARAMS';
app.INPUTPARAMSPanel.BackgroundColor = [0.5216 0.902 0.5608];
app.INPUTPARAMSPanel.FontWeight = 'bold';
app.INPUTPARAMSPanel.FontSize = 14;
app.INPUTPARAMSPanel.Position = [1 553 878 233];

% Create COORDINATESmmPanel
app.COORDINATESmmPanel = uicontrol(app.INPUTPARAMSPanel);
app.COORDINATESmmPanel.TitlePosition = 'centertop';
app.COORDINATESmmPanel.Title = 'COORDINATES (mm)';
app.COORDINATESmmPanel.FontWeight = 'bold';

```

```
app.COORDINATESmmPanel.Position = [4 62 152 142];

% Create XEditFieldLabel
app.XEditFieldLabel = uilabel(app.COORDINATESmmPanel);
app.XEditFieldLabel.FontWeight = 'bold';
app.XEditFieldLabel.Position = [7 97 25 22];
app.XEditFieldLabel.Text = 'X';

% Create XEditField
app.XEditField = uieditfield(app.COORDINATESmmPanel, 'numeric');
app.XEditField.Position = [47 97 100 22];
app.XEditField.Value = 144000;

% Create ZEditFieldLabel
app.ZEditFieldLabel = uilabel(app.COORDINATESmmPanel);
app.ZEditFieldLabel.FontWeight = 'bold';
app.ZEditFieldLabel.Position = [7 38 25 22];
app.ZEditFieldLabel.Text = 'Z';

% Create ZEditField
app.ZEditField = uieditfield(app.COORDINATESmmPanel, 'numeric');
app.ZEditField.Position = [47 38 100 22];

% Create YEditFieldLabel
app.YEditFieldLabel = uilabel(app.COORDINATESmmPanel);
app.YEditFieldLabel.FontWeight = 'bold';
app.YEditFieldLabel.Position = [7 69 25 22];
app.YEditFieldLabel.Text = 'Y';

% Create YEditField
app.YEditField = uieditfield(app.COORDINATESmmPanel, 'numeric');
app.YEditField.Position = [47 69 100 22];

% Create ISDEditFieldLabel
app.ISDEditFieldLabel = uilabel(app.COORDINATESmmPanel);
app.ISDEditFieldLabel.FontWeight = 'bold';
app.ISDEditFieldLabel.Position = [7 7 26 22];
app.ISDEditFieldLabel.Text = 'ISD';

% Create ISDEditField
app.ISDEditField = uieditfield(app.COORDINATESmmPanel, 'numeric');
app.ISDEditField.Position = [47 7 100 22];
app.ISDEditField.Value = 144000;

% Create SPSTHETAPanel
app.SPSTHETAPanel = uipanel(app.INPUTPARAMSPanel);
app.SPSTHETAPanel.TitlePosition = 'centertop';
app.SPSTHETAPanel.Title = 'SPS THETA';
app.SPSTHETAPanel.FontWeight = 'bold';
app.SPSTHETAPanel.Position = [172 6 183 198];

% Create Theta1EditFieldLabel
app.Theta1EditFieldLabel = uilabel(app.SPSTHETAPanel);
app.Theta1EditFieldLabel.HorizontalAlignment = 'right';
app.Theta1EditFieldLabel.FontWeight = 'bold';
```

```
app.Theta1EditFieldLabel.Position = [8 153 48 22];
app.Theta1EditFieldLabel.Text = 'Theta 1';

% Create Theta1EditField
app.Theta1EditField = uieditfield(app.SPSTHETAPanel, 'numeric');
app.Theta1EditField.ValueDisplayFormat = '%11.14g';
app.Theta1EditField.Position = [71 153 100 22];
app.Theta1EditField.Value = 1.5707964;

% Create Theta2Label
app.Theta2Label = uilabel(app.SPSTHETAPanel);
app.Theta2Label.HorizontalAlignment = 'right';
app.Theta2Label.FontWeight = 'bold';
app.Theta2Label.Position = [8 132 48 22];
app.Theta2Label.Text = 'Theta 2';

% Create Theta2EditField
app.Theta2EditField = uieditfield(app.SPSTHETAPanel, 'numeric');
app.Theta2EditField.ValueDisplayFormat = '%11.14g';
app.Theta2EditField.Position = [71 132 100 22];
app.Theta2EditField.Value = 2.3561946;

% Create Theta3EditFieldLabel
app.Theta3EditFieldLabel = uilabel(app.SPSTHETAPanel);
app.Theta3EditFieldLabel.HorizontalAlignment = 'right';
app.Theta3EditFieldLabel.FontWeight = 'bold';
app.Theta3EditFieldLabel.Position = [8 111 48 22];
app.Theta3EditFieldLabel.Text = 'Theta 3';

% Create Theta3EditField
app.Theta3EditField = uieditfield(app.SPSTHETAPanel, 'numeric');
app.Theta3EditField.ValueDisplayFormat = '%11.14g';
app.Theta3EditField.Position = [71 111 100 22];
app.Theta3EditField.Value = 3.1415927;

% Create Theta4EditFieldLabel
app.Theta4EditFieldLabel = uilabel(app.SPSTHETAPanel);
app.Theta4EditFieldLabel.HorizontalAlignment = 'right';
app.Theta4EditFieldLabel.FontWeight = 'bold';
app.Theta4EditFieldLabel.Position = [8 89 48 22];
app.Theta4EditFieldLabel.Text = 'Theta 4';

% Create Theta4EditField
app.Theta4EditField = uieditfield(app.SPSTHETAPanel, 'numeric');
app.Theta4EditField.ValueDisplayFormat = '%11.14g';
app.Theta4EditField.Position = [71 89 100 22];
app.Theta4EditField.Value = 3.9269909;

% Create Theta5EditFieldLabel
app.Theta5EditFieldLabel = uilabel(app.SPSTHETAPanel);
app.Theta5EditFieldLabel.HorizontalAlignment = 'right';
app.Theta5EditFieldLabel.FontWeight = 'bold';
app.Theta5EditFieldLabel.Position = [8 68 48 22];
app.Theta5EditFieldLabel.Text = 'Theta 5';
```

```

% Create Theta5EditField
app.Theta5EditField = uicontrol(app.SPSTHETAPanel, 'numeric');
app.Theta5EditField.ValueDisplayFormat = '%11.14g';
app.Theta5EditField.Position = [71 68 100 22];
app.Theta5EditField.Value = 4.7123891;

% Create Theta6EditFieldLabel
app.Theta6EditFieldLabel = uicontrol(app.SPSTHETAPanel);
app.Theta6EditFieldLabel.HorizontalAlignment = 'right';
app.Theta6EditFieldLabel.FontWeight = 'bold';
app.Theta6EditFieldLabel.Position = [8 47 48 22];
app.Theta6EditFieldLabel.Text = 'Theta 6';

% Create Theta6EditField
app.Theta6EditField = uicontrol(app.SPSTHETAPanel, 'numeric');
app.Theta6EditField.ValueDisplayFormat = '%11.14g';
app.Theta6EditField.Position = [71 47 100 22];
app.Theta6EditField.Value = 5.4977873;

% Create Theta7EditFieldLabel
app.Theta7EditFieldLabel = uicontrol(app.SPSTHETAPanel);
app.Theta7EditFieldLabel.HorizontalAlignment = 'right';
app.Theta7EditFieldLabel.FontWeight = 'bold';
app.Theta7EditFieldLabel.Position = [8 26 48 22];
app.Theta7EditFieldLabel.Text = 'Theta 7';

% Create Theta7EditField
app.Theta7EditField = uicontrol(app.SPSTHETAPanel, 'numeric');
app.Theta7EditField.ValueDisplayFormat = '%11.14g';
app.Theta7EditField.Position = [71 26 100 22];

% Create Theta8EditFieldLabel
app.Theta8EditFieldLabel = uicontrol(app.SPSTHETAPanel);
app.Theta8EditFieldLabel.HorizontalAlignment = 'right';
app.Theta8EditFieldLabel.FontWeight = 'bold';
app.Theta8EditFieldLabel.Position = [8 5 48 22];
app.Theta8EditFieldLabel.Text = 'Theta 8';

% Create Theta8EditField
app.Theta8EditField = uicontrol(app.SPSTHETAPanel, 'numeric');
app.Theta8EditField.ValueDisplayFormat = '%11.14g';
app.Theta8EditField.Position = [71 5 100 22];
app.Theta8EditField.Value = 0.78539819;

% Create STARTSIMULATIONXYZTOSPSButton
app.STARTSIMULATIONXYZTOSPSButton = uicontrol(app.INPUTPARAMSPanel,
'push');
app.STARTSIMULATIONXYZTOSPSButton.ButtonPushedFcn = createCallbackFcn(app,
@STARTSIMULATIONXYZTOSPSButtonPushed, true);
app.STARTSIMULATIONXYZTOSPSButton.BackgroundColor = [1 1 0];
app.STARTSIMULATIONXYZTOSPSButton.FontSize = 24;
app.STARTSIMULATIONXYZTOSPSButton.FontWeight = 'bold';
app.STARTSIMULATIONXYZTOSPSButton.Position = [416 8 406 56];
app.STARTSIMULATIONXYZTOSPSButton.Text = 'START SIMULATION XYZ-TO-SPS';

```

```

% Create CSVPATHPanel
app.CSVPATHPanel = uipanel(app.INPUTPARAMSPanel);
app.CSVPATHPanel.TitlePosition = 'centertop';
app.CSVPATHPanel.Title = 'CSV PATH';
app.CSVPATHPanel.FontWeight = 'bold';
app.CSVPATHPanel.Position = [369 69 499 135];

% Create CSV_Count
app.CSV_Count = uieditfield(app.CSVPATHPanel, 'text');
app.CSV_Count.Position = [236 88 251 22];
app.CSV_Count.Value = 'NaN';

% Create SelectcountsCSVButton
app.SelectcountsCSVButton = uibutton(app.CSVPATHPanel, 'push');
app.SelectcountsCSVButton.ButtonPushedFcn = createCallbackFcn(app, @SelectcountsCSVButtonPushed, true);
app.SelectcountsCSVButton.BackgroundColor = [0.6745 0.8039 0.8902];
app.SelectcountsCSVButton.FontWeight = 'bold';
app.SelectcountsCSVButton.Position = [9 88 122 22];
app.SelectcountsCSVButton.Text = 'Select counts CSV';

% Create PathCountCSVLabel
app.PathCountCSVLabel = uilabel(app.CSVPATHPanel);
app.PathCountCSVLabel.FontWeight = 'bold';
app.PathCountCSVLabel.Position = [135 88 104 22];
app.PathCountCSVLabel.Text = 'Path Count CSV: ';

% Create CSV_Curr
app.CSV_Curr = uieditfield(app.CSVPATHPanel, 'text');
app.CSV_Curr.Editable = 'off';
app.CSV_Curr.Position = [236 62 251 22];
app.CSV_Curr.Value = 'NaN';

% Create SelectcurrCSVButton
app.SelectcurrCSVButton = uibutton(app.CSVPATHPanel, 'push');
app.SelectcurrCSVButton.ButtonPushedFcn = createCallbackFcn(app, @SelectcurrCSVButtonPushed, true);
app.SelectcurrCSVButton.BackgroundColor = [0.6745 0.8039 0.8902];
app.SelectcurrCSVButton.FontWeight = 'bold';
app.SelectcurrCSVButton.Position = [9 62 122 22];
app.SelectcurrCSVButton.Text = 'Select curr CSV';

% Create PathCurrCSVLabel
app.PathCurrCSVLabel = uilabel(app.CSVPATHPanel);
app.PathCurrCSVLabel.FontWeight = 'bold';
app.PathCurrCSVLabel.Position = [135 62 95 22];
app.PathCurrCSVLabel.Text = 'Path Curr CSV: ';

% Create CSV_X
app.CSV_X = uieditfield(app.CSVPATHPanel, 'text');
app.CSV_X.Position = [236 35 251 22];
app.CSV_X.Value = 'NaN';

% Create SelectZCSVButton
app.SelectZCSVButton = uibutton(app.CSVPATHPanel, 'push');

```

```

        app.SelectZCSVButton.ButtonPushedFcn = createCallbackFcn(app, ✓
@SelectZCSVButtonPushed, true);
        app.SelectZCSVButton.BackgroundColor = [0.6745 0.8039 0.8902];
        app.SelectZCSVButton.FontWeight = 'bold';
        app.SelectZCSVButton.Position = [9 35 122 22];
        app.SelectZCSVButton.Text = 'Select Z CSV';

% Create PathZCSVLabel
app.PathZCSVLabel = uilabel(app.CSVPATHPanel);
app.PathZCSVLabel.FontWeight = 'bold';
app.PathZCSVLabel.Position = [135 35 97 22];
app.PathZCSVLabel.Text = 'Path Z CSV: ';

% Create CSV_fit_param
app.CSV_fit_param = uieditfield(app.CSVPATHPanel, 'text');
app.CSV_fit_param.Position = [236 7 251 22];
app.CSV_fit_param.Value = 'NaN';

% Create SelectFitParamButton
app.SelectFitParamButton = uibutton(app.CSVPATHPanel, 'push');
app.SelectFitParamButton.ButtonPushedFcn = createCallbackFcn(app, ✓
@SelectFitParamButtonPushed, true);
app.SelectFitParamButton.BackgroundColor = [0.6745 0.8039 0.8902];
app.SelectFitParamButton.FontWeight = 'bold';
app.SelectFitParamButton.Position = [9 7 122 22];
app.SelectFitParamButton.Text = 'Select Fit Param';

% Create PathFitParamLabel
app.PathFitParamLabel = uilabel(app.CSVPATHPanel);
app.PathFitParamLabel.FontWeight = 'bold';
app.PathFitParamLabel.Position = [135 7 94 22];
app.PathFitParamLabel.Text = 'Path Fit Param: ';

% Create SPSRadiusPanel
app.SPSRadiusPanel = uipanel(app.INPUTPARAMSPanel);
app.SPSRadiusPanel.TitlePosition = 'centertop';
app.SPSRadiusPanel.Title = 'SPS Radius';
app.SPSRadiusPanel.FontWeight = 'bold';
app.SPSRadiusPanel.Position = [4 8 152 46];

% Create SPSRadiusEditField
app.SPSRadiusEditField = uieditfield(app.SPSRadiusPanel, 'numeric');
app.SPSRadiusEditField.Position = [85 1 60 22];
app.SPSRadiusEditField.Value = 0.055;

% Create SPSRadiusEditFieldLabel
app.SPSRadiusEditFieldLabel = uilabel(app.SPSRadiusPanel);
app.SPSRadiusEditFieldLabel.FontWeight = 'bold';
app.SPSRadiusEditFieldLabel.Position = [8 1 73 22];
app.SPSRadiusEditFieldLabel.Text = 'SPS Radius';

% Create SPStoXYZTab
app.SPStoXYZTab = uitab(app.TabGroup2);
app.SPStoXYZTab.Title = '[SPS] to [X, Y, Z]';

```

```
% Create SETPARAMVALUESPanel
app.SETPARAMVALUESPanel = uipanel(app.SPStoXYZTab);
app.SETPARAMVALUESPanel.TitlePosition = 'centertop';
app.SETPARAMVALUESPanel.Title = 'SET PARAM VALUES';
app.SETPARAMVALUESPanel.BackgroundColor = [0.949 0.949 0.6667];
app.SETPARAMVALUESPanel.FontWeight = 'bold';
app.SETPARAMVALUESPanel.FontSize = 14;
app.SETPARAMVALUESPanel.Position = [4 400 875 386];

% Create RawDataPanel
app.RawDataPanel = uipanel(app.SETPARAMVALUESPanel);
app.RawDataPanel.TitlePosition = 'centertop';
app.RawDataPanel.Title = 'Raw Data';
app.RawDataPanel.BackgroundColor = [1 0.6706 0.6706];
app.RawDataPanel.FontWeight = 'bold';
app.RawDataPanel.Position = [4 7 211 353];

% Create SPSPanel
app.SPSPanel = uipanel(app.RawDataPanel);
app.SPSPanel.TitlePosition = 'centertop';
app.SPSPanel.Title = 'SPS';
app.SPSPanel.BackgroundColor = [0.9412 0.9412 0.9412];
app.SPSPanel.FontWeight = 'bold';
app.SPSPanel.Position = [3 116 202 215];

% Create HighGainLabel
app.HighGainLabel = uilabel(app.SPSPanel);
app.HighGainLabel.FontWeight = 'bold';
app.HighGainLabel.Position = [70 171 62 22];
app.HighGainLabel.Text = 'High Gain';

% Create LowGainLabel
app.LowGainLabel = uilabel(app.SPSPanel);
app.LowGainLabel.FontWeight = 'bold';
app.LowGainLabel.Position = [140 171 59 22];
app.LowGainLabel.Text = 'Low Gain';

% Create SPSP1Label
app.SPSP1Label = uilabel(app.SPSPanel);
app.SPSP1Label.FontWeight = 'bold';
app.SPSP1Label.Position = [3 150 40 22];
app.SPSP1Label.Text = 'SPS 1';

% Create EditField_SPSP1_high
app.EditField_SPSP1_high = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPSP1_high.Position = [60 150 69 22];
app.EditField_SPSP1_high.Value = 1034;

% Create EditField_SPSP1_low
app.EditField_SPSP1_low = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPSP1_low.Position = [138 150 58 22];
app.EditField_SPSP1_low.Value = 207;

% Create SPSP2Label
app.SPSP2Label = uilabel(app.SPSPanel);
```

```
app.SPS2Label.FontWeight = 'bold';
app.SPS2Label.Position = [3 129 40 22];
app.SPS2Label.Text = 'SPS 2';

% Create EditField_SPS2_high
app.EditField_SPS2_high = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS2_high.Position = [60 129 69 22];
app.EditField_SPS2_high.Value = 1034;

% Create EditField_SPS2_low
app.EditField_SPS2_low = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS2_low.Position = [138 129 58 22];
app.EditField_SPS2_low.Value = 207;

% Create SPS3Label
app.SPS3Label = uicontrolfield(app.SPSPanel);
app.SPS3Label.FontWeight = 'bold';
app.SPS3Label.Position = [3 108 40 22];
app.SPS3Label.Text = 'SPS 3';

% Create EditField_SPS3_high
app.EditField_SPS3_high = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS3_high.Position = [60 108 69 22];
app.EditField_SPS3_high.Value = 1034;

% Create EditField_SPS3_low
app.EditField_SPS3_low = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS3_low.Position = [138 108 58 22];
app.EditField_SPS3_low.Value = 207;

% Create SPS4Label
app.SPS4Label = uicontrolfield(app.SPSPanel);
app.SPS4Label.FontWeight = 'bold';
app.SPS4Label.Position = [3 87 40 22];
app.SPS4Label.Text = 'SPS 4';

% Create EditField_SPS4_high
app.EditField_SPS4_high = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS4_high.Position = [60 87 69 22];
app.EditField_SPS4_high.Value = 1034;

% Create EditField_SPS4_low
app.EditField_SPS4_low = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS4_low.Position = [138 87 58 22];
app.EditField_SPS4_low.Value = 207;

% Create SPS5Label
app.SPS5Label = uicontrolfield(app.SPSPanel);
app.SPS5Label.FontWeight = 'bold';
app.SPS5Label.Position = [3 66 40 22];
app.SPS5Label.Text = 'SPS 5';

% Create EditField_SPS5_high
app.EditField_SPS5_high = uicontrolfield(app.SPSPanel, 'numeric');
app.EditField_SPS5_high.Position = [60 66 69 22];
```

```
app.EditField_SPS5_high.Value = 1034;

% Create EditField_SPS5_low
app.EditField_SPS5_low = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS5_low.Position = [138 66 58 22];
app.EditField_SPS5_low.Value = 207;

% Create SPS6Label
app.SPS6Label = uilabel(app.SPSPanel);
app.SPS6Label.FontWeight = 'bold';
app.SPS6Label.Position = [3 45 40 22];
app.SPS6Label.Text = 'SPS 6';

% Create EditField_SPS6_high
app.EditField_SPS6_high = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS6_high.Position = [60 45 69 22];
app.EditField_SPS6_high.Value = 1034;

% Create EditField_SPS6_low
app.EditField_SPS6_low = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS6_low.Position = [138 45 58 22];
app.EditField_SPS6_low.Value = 207;

% Create SPS7Label
app.SPS7Label = uilabel(app.SPSPanel);
app.SPS7Label.FontWeight = 'bold';
app.SPS7Label.Position = [3 24 40 22];
app.SPS7Label.Text = 'SPS 7';

% Create EditField_SPS7_high
app.EditField_SPS7_high = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS7_high.Position = [60 24 69 22];
app.EditField_SPS7_high.Value = 1034;

% Create EditField_SPS7_low
app.EditField_SPS7_low = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS7_low.Position = [138 24 58 22];
app.EditField_SPS7_low.Value = 207;

% Create SPS8Label
app.SPS8Label = uilabel(app.SPSPanel);
app.SPS8Label.FontWeight = 'bold';
app.SPS8Label.Position = [3 3 40 22];
app.SPS8Label.Text = 'SPS 8';

% Create EditField_SPS8_high
app.EditField_SPS8_high = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS8_high.Position = [60 3 69 22];
app.EditField_SPS8_high.Value = 1034;

% Create EditField_SPS8_low
app.EditField_SPS8_low = uieditfield(app.SPSPanel, 'numeric');
app.EditField_SPS8_low.Position = [138 3 58 22];
app.EditField_SPS8_low.Value = 207;
```

```
% Create TempPanel
app.TempPanel = uipanel(app.RawDataPanel);
app.TempPanel.TitlePosition = 'centertop';
app.TempPanel.Title = 'Temp';
app.TempPanel.FontWeight = 'bold';
app.TempPanel.Position = [3 3 100 109];

% Create NomT1Label
app.NomT1Label = uilabel(app.TempPanel);
app.NomT1Label.FontWeight = 'bold';
app.NomT1Label.Position = [2 65 50 22];
app.NomT1Label.Text = 'Nom T1';

% Create EditField_nom_T1
app.EditField_nom_T1 = uieditfield(app.TempPanel, 'numeric');
app.EditField_nom_T1.Position = [61 65 35 22];
app.EditField_nom_T1.Value = 308;

% Create NomT2Label
app.NomT2Label = uilabel(app.TempPanel);
app.NomT2Label.FontWeight = 'bold';
app.NomT2Label.Position = [1 45 50 22];
app.NomT2Label.Text = 'Nom T2';

% Create EditField_Nom_T2
app.EditField_Nom_T2 = uieditfield(app.TempPanel, 'numeric');
app.EditField_Nom_T2.Position = [61 44 35 22];
app.EditField_Nom_T2.Value = 308;

% Create RedT1Label
app.RedT1Label = uilabel(app.TempPanel);
app.RedT1Label.FontWeight = 'bold';
app.RedT1Label.Position = [3 24 46 22];
app.RedT1Label.Text = 'Red T1';

% Create EditField_Red_T1
app.EditField_Red_T1 = uieditfield(app.TempPanel, 'numeric');
app.EditField_Red_T1.Position = [61 23 35 22];
app.EditField_Red_T1.Value = 308;

% Create RedT2Label
app.RedT2Label = uilabel(app.TempPanel);
app.RedT2Label.FontWeight = 'bold';
app.RedT2Label.Position = [3 3 46 22];
app.RedT2Label.Text = 'Red T2';

% Create EditField_Red_T2
app.EditField_Red_T2 = uieditfield(app.TempPanel, 'numeric');
app.EditField_Red_T2.Position = [61 2 35 22];
app.EditField_Red_T2.Value = 308;

% Create DoorPanel
app.DoorPanel = uipanel(app.RawDataPanel);
app.DoorPanel.TitlePosition = 'centertop';
app.DoorPanel.Title = 'Door';
```

```
app.DoorPanel.FontWeight = 'bold';
app.DoorPanel.Position = [105 3 100 109];

% Create DoorLabel
app.DoorLabel = uilabel(app.DoorPanel);
app.DoorLabel.FontWeight = 'bold';
app.DoorLabel.Position = [9 53 33 22];
app.DoorLabel.Text = 'Door';

% Create Door_trLabel
app.Door_trLabel = uilabel(app.DoorPanel);
app.Door_trLabel.FontWeight = 'bold';
app.Door_trLabel.Position = [9 14 49 22];
app.Door_trLabel.Text = 'Door_tr';

% Create CheckBox_Door_tr
app.CheckBox_Door_tr = uicheckbox(app.DoorPanel);
app.CheckBox_Door_tr.Text = '';
app.CheckBox_Door_tr.Position = [68 14 25 22];

% Create CheckBox_Door
app.CheckBox_Door = uicheckbox(app.DoorPanel);
app.CheckBox_Door.Text = '';
app.CheckBox_Door.Position = [68 53 25 22];

% Create ReconParamsPanel
app.ReconParamsPanel = uipanel(app.SETPARAMVALUESPanel);
app.ReconParamsPanel.TitlePosition = 'centertop';
app.ReconParamsPanel.Title = 'Recon Params';
app.ReconParamsPanel.BackgroundColor = [0.7098 0.9216 0.8784];
app.ReconParamsPanel.FontWeight = 'bold';
app.ReconParamsPanel.Position = [221 7 513 353];

% Create KcoeffPanel_2
app.KcoeffPanel_2 = uipanel(app.ReconParamsPanel);
app.KcoeffPanel_2.TitlePosition = 'centertop';
app.KcoeffPanel_2.Title = 'K coeff';
app.KcoeffPanel_2.BackgroundColor = [0.902 0.902 0.902];
app.KcoeffPanel_2.FontWeight = 'bold';
app.KcoeffPanel_2.Position = [2 140 100 191];

% Create K1Label
app.K1Label = uilabel(app.KcoeffPanel_2);
app.K1Label.FontWeight = 'bold';
app.K1Label.Position = [6 147 25 22];
app.K1Label.Text = 'K1';

% Create K2Label
app.K2Label = uilabel(app.KcoeffPanel_2);
app.K2Label.FontWeight = 'bold';
app.K2Label.Position = [6 126 25 22];
app.K2Label.Text = 'K2';

% Create K3Label
app.K3Label = uilabel(app.KcoeffPanel_2);
```

```
app.K3Label.FontWeight = 'bold';
app.K3Label.Position = [6 105 25 22];
app.K3Label.Text = 'K3';

% Create K4Label
app.K4Label = uilabel(app.KcoeffPanel_2);
app.K4Label.FontWeight = 'bold';
app.K4Label.Position = [6 84 25 22];
app.K4Label.Text = 'K4';

% Create K5Label
app.K5Label = uilabel(app.KcoeffPanel_2);
app.K5Label.FontWeight = 'bold';
app.K5Label.Position = [5 63 25 22];
app.K5Label.Text = 'K5';

% Create K6Label
app.K6Label = uilabel(app.KcoeffPanel_2);
app.K6Label.FontWeight = 'bold';
app.K6Label.Position = [6 42 25 22];
app.K6Label.Text = 'K6';

% Create K7Label
app.K7Label = uilabel(app.KcoeffPanel_2);
app.K7Label.FontWeight = 'bold';
app.K7Label.Position = [6 21 25 22];
app.K7Label.Text = 'K7';

% Create K8Label
app.K8Label = uilabel(app.KcoeffPanel_2);
app.K8Label.FontWeight = 'bold';
app.K8Label.Position = [6 1 25 22];
app.K8Label.Text = 'K8';

% Create EditField_K1
app.EditField_K1 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K1.Position = [53 147 40 22];
app.EditField_K1.Value = 1;

% Create EditField_K2
app.EditField_K2 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K2.Position = [53 126 40 22];
app.EditField_K2.Value = 1;

% Create EditField_K3
app.EditField_K3 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K3.Position = [53 105 40 22];
app.EditField_K3.Value = 1;

% Create EditField_K4
app.EditField_K4 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K4.Position = [53 84 40 22];
app.EditField_K4.Value = 1;

% Create EditField_K5
```

```
app.EditField_K5 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K5.Position = [53 63 40 22];
app.EditField_K5.Value = 1;

% Create EditField_K6
app.EditField_K6 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K6.Position = [53 42 40 22];
app.EditField_K6.Value = 1;

% Create EditField_K7
app.EditField_K7 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K7.Position = [53 21 40 22];
app.EditField_K7.Value = 1;

% Create EditField_K8
app.EditField_K8 = uieditfield(app.KcoeffPanel_2, 'numeric');
app.EditField_K8.Position = [53 1 40 22];
app.EditField_K8.Value = 1;

% Create pp_coarsePanel
app.pp_coarsePanel = uipanel(app.ReconParamsPanel);
app.pp_coarsePanel.TitlePosition = 'centertop';
app.pp_coarsePanel.Title = 'pp_coarse';
app.pp_coarsePanel.FontWeight = 'bold';
app.pp_coarsePanel.Position = [106 221 194 110];

% Create EditField_a_coarse_value
app.EditField_a_coarse_value = uieditfield(app.pp_coarsePanel, 'numeric');
app.EditField_a_coarse_value.ValueDisplayFormat = '%11.16g';
app.EditField_a_coarse_value.Position = [61 64 123 22];
app.EditField_a_coarse_value.Value = -8.4911629078079e-06;

% Create a_coarseLabel_2
app.a_coarseLabel_2 = uilabel(app.pp_coarsePanel);
app.a_coarseLabel_2.FontWeight = 'bold';
app.a_coarseLabel_2.Position = [4 65 58 22];
app.a_coarseLabel_2.Text = 'a_coarse';

% Create EditField_b_coarse_value
app.EditField_b_coarse_value = uieditfield(app.pp_coarsePanel, 'numeric');
app.EditField_b_coarse_value.ValueDisplayFormat = '%11.16g';
app.EditField_b_coarse_value.Position = [61 43 123 22];
app.EditField_b_coarse_value.Value = 0.000621592577773211;

% Create b_coarseLabel_2
app.b_coarseLabel_2 = uilabel(app.pp_coarsePanel);
app.b_coarseLabel_2.FontWeight = 'bold';
app.b_coarseLabel_2.Position = [4 44 58 22];
app.b_coarseLabel_2.Text = 'b_coarse';

% Create EditField_c_coarse_value
app.EditField_c_coarse_value = uieditfield(app.pp_coarsePanel, 'numeric');
app.EditField_c_coarse_value.ValueDisplayFormat = '%11.14g';
app.EditField_c_coarse_value.Position = [61 22 123 22];
app.EditField_c_coarse_value.Value = -0.00494398761782694;
```

```
% Create c_coarseLabel_2
app.c_coarseLabel_2 = uilabel(app.pp_coarsePanel);
app.c_coarseLabel_2.FontWeight = 'bold';
app.c_coarseLabel_2.Position = [4 23 58 22];
app.c_coarseLabel_2.Text = 'c_coarse';

% Create EditField_A_coarse_value
app.EditField_A_coarse_value = uieditfield(app.pp_coarsePanel, 'numeric');
app.EditField_A_coarse_value.ValueDisplayFormat = '%11.16g';
app.EditField_A_coarse_value.Position = [61 1 123 22];
app.EditField_A_coarse_value.Value = -0.0110973673133897;

% Create A_coarseLabel
app.A_coarseLabel = uilabel(app.pp_coarsePanel);
app.A_coarseLabel.FontWeight = 'bold';
app.A_coarseLabel.Position = [4 3 60 22];
app.A_coarseLabel.Text = 'A_coarse';

% Create pp_finePanel
app.pp_finePanel = uipanel(app.ReconParamsPanel);
app.pp_finePanel.TitlePosition = 'centertop';
app.pp_finePanel.Title = 'pp_fine';
app.pp_finePanel.FontWeight = 'bold';
app.pp_finePanel.Position = [303 196 206 135];

% Create EditField_a_fine_value
app.EditField_a_fine_value = uieditfield(app.pp_finePanel, 'numeric');
app.EditField_a_fine_value.ValueDisplayFormat = '%11.16g';
app.EditField_a_fine_value.Position = [77 91 122 22];
app.EditField_a_fine_value.Value = -4.12828072223287e-06;

% Create a_fineLabel
app.a_fineLabel = uilabel(app.pp_finePanel);
app.a_fineLabel.FontWeight = 'bold';
app.a_fineLabel.Position = [7 92 40 22];
app.a_fineLabel.Text = 'a_fine';

% Create EditField_b_fine_value
app.EditField_b_fine_value = uieditfield(app.pp_finePanel, 'numeric');
app.EditField_b_fine_value.ValueDisplayFormat = '%11.16g';
app.EditField_b_fine_value.Position = [77 70 122 22];
app.EditField_b_fine_value.Value = 0.000456794446297276;

% Create b_fineLabel
app.b_fineLabel = uilabel(app.pp_finePanel);
app.b_fineLabel.FontWeight = 'bold';
app.b_fineLabel.Position = [7 71 41 22];
app.b_fineLabel.Text = 'b_fine';

% Create EditField_c_fine_value
app.EditField_c_fine_value = uieditfield(app.pp_finePanel, 'numeric');
app.EditField_c_fine_value.ValueDisplayFormat = '%11.14g';
app.EditField_c_fine_value.Position = [77 49 122 22];
app.EditField_c_fine_value.Value = -0.00362182084132099;
```

```
% Create c_fineLabel
app.c_fineLabel = uilabel(app.pp_finePanel);
app.c_fineLabel.FontWeight = 'bold';
app.c_fineLabel.Position = [7 50 40 22];
app.c_fineLabel.Text = 'c_fine';

% Create EditField_A_fine_value
app.EditField_A_fine_value = uieditfield(app.pp_finePanel, 'numeric');
app.EditField_A_fine_value.ValueDisplayFormat = '%11.16g';
app.EditField_A_fine_value.Position = [77 28 122 22];
app.EditField_A_fine_value.Value = -0.00446231603379374;

% Create A_fineLabel
app.A_fineLabel = uilabel(app.pp_finePanel);
app.A_fineLabel.FontWeight = 'bold';
app.A_fineLabel.Position = [7 29 42 22];
app.A_fineLabel.Text = 'A_fine';

% Create EditField_delta_transv
app.EditField_delta_transv = uieditfield(app.pp_finePanel, 'numeric');
app.EditField_delta_transv.ValueDisplayFormat = '%11.16g';
app.EditField_delta_transv.Position = [77 7 122 22];
app.EditField_delta_transv.Value = 13.8;

% Create delta_transvLabel
app.delta_transvLabel = uilabel(app.pp_finePanel);
app.delta_transvLabel.FontWeight = 'bold';
app.delta_transvLabel.Position = [7 8 76 22];
app.delta_transvLabel.Text = 'delta_transv';

% Create ThresholdsPanel_2
app.ThresholdsPanel_2 = uipanel(app.ReconParamsPanel);
app.ThresholdsPanel_2.TitlePosition = 'centertop';
app.ThresholdsPanel_2.Title = 'Thresholds';
app.ThresholdsPanel_2.FontWeight = 'bold';
app.ThresholdsPanel_2.Position = [3 6 140 128];

% Create EditField_V_max_lat
app.EditField_V_max_lat = uieditfield(app.ThresholdsPanel_2, 'numeric');
app.EditField_V_max_lat.ValueDisplayFormat = '%11.16g';
app.EditField_V_max_lat.Position = [86 76 45 22];
app.EditField_V_max_lat.Value = 5;

% Create V_max_latLabel
app.V_max_latLabel = uilabel(app.ThresholdsPanel_2);
app.V_max_latLabel.FontWeight = 'bold';
app.V_max_latLabel.Position = [6 77 65 22];
app.V_max_latLabel.Text = 'V_max_lat';

% Create EditField_V_max_long
app.EditField_V_max_long = uieditfield(app.ThresholdsPanel_2, 'numeric');
app.EditField_V_max_long.ValueDisplayFormat = '%11.16g';
app.EditField_V_max_long.Position = [86 55 45 22];
app.EditField_V_max_long.Value = 50;
```

```
% Create V_max_longLabel
app.V_max_longLabel = uilabel(app.ThresholdsPanel_2);
app.V_max_longLabel.FontWeight = 'bold';
app.V_max_longLabel.Position = [6 56 76 22];
app.V_max_longLabel.Text = 'V_max_long';

% Create EditField_T_max
app.EditField_T_max = uieditfield(app.ThresholdsPanel_2, 'numeric');
app.EditField_T_max.ValueDisplayFormat = '%11.14g';
app.EditField_T_max.Position = [86 34 45 22];
app.EditField_T_max.Value = 348;

% Create T_maxLabel
app.T_maxLabel = uilabel(app.ThresholdsPanel_2);
app.T_maxLabel.FontWeight = 'bold';
app.T_maxLabel.Position = [6 35 44 22];
app.T_maxLabel.Text = 'T_max';

% Create EditField_delta_flag
app.EditField_delta_flag = uieditfield(app.ThresholdsPanel_2, 'numeric');
app.EditField_delta_flag.ValueDisplayFormat = '%11.16g';
app.EditField_delta_flag.Position = [86 13 45 22];
app.EditField_delta_flag.Value = 13.8;

% Create delta_flagLabel
app.delta_flagLabel = uilabel(app.ThresholdsPanel_2);
app.delta_flagLabel.FontWeight = 'bold';
app.delta_flagLabel.Position = [6 14 62 22];
app.delta_flagLabel.Text = 'delta_flag';

% Create LinearPanel
app.LinearPanel = uipanel(app.ReconParamsPanel);
app.LinearPanel.TitlePosition = 'centertop';
app.LinearPanel.Title = 'Linear';
app.LinearPanel.FontWeight = 'bold';
app.LinearPanel.Position = [147 8 124 126];

% Create EditField_d_z
app.EditField_d_z = uieditfield(app.LinearPanel, 'numeric');
app.EditField_d_z.ValueDisplayFormat = '%11.16g';
app.EditField_d_z.Position = [60 61 56 22];
app.EditField_d_z.Value = 185800;

% Create d_zLabel
app.d_zLabel = uilabel(app.LinearPanel);
app.d_zLabel.FontWeight = 'bold';
app.d_zLabel.Position = [4 62 26 22];
app.d_zLabel.Text = 'd_z';

% Create EditField_d_y
app.EditField_d_y = uieditfield(app.LinearPanel, 'numeric');
app.EditField_d_y.ValueDisplayFormat = '%11.16g';
app.EditField_d_y.Position = [60 19 56 22];
app.EditField_d_y.Value = 185800;
```

```
% Create d_yLabel
app.d_yLabel = uilabel(app.LinearPanel);
app.d_yLabel.FontWeight = 'bold';
app.d_yLabel.Position = [4 20 26 22];
app.d_yLabel.Text = 'd_y';

% Create LongPanel
app.LongPanel = uipanel(app.ReconParamsPanel);
app.LongPanel.TitlePosition = 'centertop';
app.LongPanel.Title = 'Long';
app.LongPanel.FontWeight = 'bold';
app.LongPanel.Position = [278 5 231 129];

% Create EditField_H
app.EditField_H = uieditfield(app.LongPanel, 'numeric');
app.EditField_H.ValueDisplayFormat = '%11.16g';
app.EditField_H.Position = [86 86 138 22];
app.EditField_H.Value = 5.8413563419434e-05;

% Create HLabel
app.HLabel = uilabel(app.LongPanel);
app.HLabel.FontWeight = 'bold';
app.HLabel.Position = [6 87 25 22];
app.HLabel.Text = 'H';

% Create EditField_K
app.EditField_K = uieditfield(app.LongPanel, 'numeric');
app.EditField_K.ValueDisplayFormat = '%11.16g';
app.EditField_K.Position = [86 65 138 22];
app.EditField_K.Value = -0.542178741406663;

% Create KLabel
app.KLabel = uilabel(app.LongPanel);
app.KLabel.FontWeight = 'bold';
app.KLabel.Position = [6 66 25 22];
app.KLabel.Text = 'K';

% Create EditField_L
app.EditField_L = uieditfield(app.LongPanel, 'numeric');
app.EditField_L.ValueDisplayFormat = '%11.14g';
app.EditField_L.Position = [86 44 138 22];
app.EditField_L.Value = 1034;

% Create LLabel
app.LLabel = uilabel(app.LongPanel);
app.LLabel.FontWeight = 'bold';
app.LLabel.Position = [6 45 25 22];
app.LLabel.Text = 'L';

% Create EditField_J
app.EditField_J = uieditfield(app.LongPanel, 'numeric');
app.EditField_J.ValueDisplayFormat = '%11.16g';
app.EditField_J.Position = [86 23 138 22];
```

```

% Create JLabel
app.JLabel = uilabel(app.LongPanel);
app.JLabel.FontWeight = 'bold';
app.JLabel.Position = [6 24 25 22];
app.JLabel.Text = 'J';

% Create EditField_delta_R
app.EditField_delta_R = uieditfield(app.LongPanel, 'numeric');
app.EditField_delta_R.ValueDisplayFormat = '%11.16g';
app.EditField_delta_R.Position = [86 2 138 22];
app.EditField_delta_R.Value = 2327.47786519536;

% Create delta_RLabel
app.delta_RLabel = uilabel(app.LongPanel);
app.delta_RLabel.FontWeight = 'bold';
app.delta_RLabel.Position = [6 3 49 22];
app.delta_RLabel.Text = 'delta_R';

% Create AlgorithmPanel
app.AlgorithmPanel = uipanel(app.ReconParamsPanel);
app.AlgorithmPanel.TitlePosition = 'centertop';
app.AlgorithmPanel.Title = 'Algorithm';
app.AlgorithmPanel.FontWeight = 'bold';
app.AlgorithmPanel.Position = [103 141 197 78];

% Create AlgoLabel
app.AlgoLabel = uilabel(app.AlgorithmPanel);
app.AlgoLabel.FontWeight = 'bold';
app.AlgoLabel.Position = [6 34 32 22];
app.AlgoLabel.Text = 'Algo';

% Create Box_Algorithm
app.Box_Algorithm = uidropdown(app.AlgorithmPanel);
app.Box_Algorithm.Items = {'BOTH', 'P_PARAB', 'LINEAR', ' '};
app.Box_Algorithm.ItemsData = {'0', '1', '2'};
app.Box_Algorithm.Position = [46 34 138 22];
app.Box_Algorithm.Value = '0';

% Create InrunselfchangingalgorithmvalueCheckBox
app.InrunselfchangingalgorithmvalueCheckBox = uicheckbox(app.
AlgorithmPanel);
app.InrunselfchangingalgorithmvalueCheckBox.Text = 'In run self-changing
algorithm value';
app.InrunselfchangingalgorithmvalueCheckBox.FontSize = 10;
app.InrunselfchangingalgorithmvalueCheckBox.FontWeight = 'bold';
app.InrunselfchangingalgorithmvalueCheckBox.Position = [4 7 195 22];

% Create RadiusSPSPanel
app.RadiusSPSPanel = uipanel(app.ReconParamsPanel);
app.RadiusSPSPanel.TitlePosition = 'centertop';
app.RadiusSPSPanel.Title = 'Radius SPS';
app.RadiusSPSPanel.FontWeight = 'bold';
app.RadiusSPSPanel.Position = [309 141 200 51];

% Create EditField_R_sps

```

```
app.EditField_R_sps = uieditfield(app.RadiusSPSPanel, 'numeric');
app.EditField_R_sps.ValueDisplayFormat = '%11.16g';
app.EditField_R_sps.Position = [75 7 118 22];
app.EditField_R_sps.Value = 0.055;

% Create R_spsLabel
app.R_spsLabel = uilabel(app.RadiusSPSPanel);
app.R_spsLabel.FontWeight = 'bold';
app.R_spsLabel.Position = [8 8 42 22];
app.R_spsLabel.Text = 'R_sps';

% Create ImportparamsButton
app.ImportparamsButton = uibutton(app.SETPARAMVALUESPanel, 'push');
app.ImportparamsButton.ButtonPushedFcn = createCallbackFcn(app, @ImportparamsButtonPushed, true);
app.ImportparamsButton.BackgroundColor = [0.6118 0.6588 0.9412];
app.ImportparamsButton.FontSize = 16;
app.ImportparamsButton.FontWeight = 'bold';
app.ImportparamsButton.Position = [742 10 127 26];
app.ImportparamsButton.Text = 'Import params';

% Create XYZExpectedPanel
app.XYZExpectedPanel = uipanel(app.SETPARAMVALUESPanel);
app.XYZExpectedPanel.TitlePosition = 'centertop';
app.XYZExpectedPanel.Title = 'X Y Z Expected';
app.XYZExpectedPanel.BackgroundColor = [0.8824 0.6745 0.8902];
app.XYZExpectedPanel.FontWeight = 'bold';
app.XYZExpectedPanel.Position = [745 42 124 138];

% Create XLabel
app.XLabel = uilabel(app.XYZExpectedPanel);
app.XLabel.FontWeight = 'bold';
app.XLabel.Position = [8 87 25 22];
app.XLabel.Text = 'X';

% Create YLabel
app.YLabel = uilabel(app.XYZExpectedPanel);
app.YLabel.FontWeight = 'bold';
app.YLabel.Position = [8 48 25 22];
app.YLabel.Text = 'Y';

% Create EditField_X_Expected
app.EditField_X_Expected = uieditfield(app.XYZExpectedPanel, 'numeric');
app.EditField_X_Expected.ValueDisplayFormat = '%11.16g';
app.EditField_X_Expected.Position = [63 87 56 22];

% Create EditField_Y_Expected
app.EditField_Y_Expected = uieditfield(app.XYZExpectedPanel, 'numeric');
app.EditField_Y_Expected.ValueDisplayFormat = '%11.16g';
app.EditField_Y_Expected.Position = [63 48 56 22];

% Create ZLabel
app.ZLabel = uilabel(app.XYZExpectedPanel);
app.ZLabel.FontWeight = 'bold';
app.ZLabel.Position = [8 10 25 22];
```

```

app.ZLabel.Text = 'Z';

% Create EditField_Z_Expected
app.EditField_Z_Expected = uieditfield(app.XYZExpectedPanel, 'numeric');
app.EditField_Z_Expected.ValueDisplayFormat = '%11.16g';
app.EditField_Z_Expected.Position = [63 10 56 22];

% Create CYCLEPANELPanel
app.CYCLEPANELPanel = uipanel(app.SPStoXYZTab);
app.CYCLEPANELPanel.TitlePosition = 'centertop';
app.CYCLEPANELPanel.Title = 'CYCLE PANEL';
app.CYCLEPANELPanel.BackgroundColor = [0.8902 1 0.7804];
app.CYCLEPANELPanel.FontWeight = 'bold';
app.CYCLEPANELPanel.FontSize = 14;
app.CYCLEPANELPanel.Position = [4 0 874 401];

% Create TestEditFieldLabel
app.TestEditFieldLabel = uilabel(app.CYCLEPANELPanel);
app.TestEditFieldLabel.HorizontalAlignment = 'right';
app.TestEditFieldLabel.FontWeight = 'bold';
app.TestEditFieldLabel.Position = [6 353 29 22];
app.TestEditFieldLabel.Text = 'Test';

% Create Test_Field
app.Test_Field = uieditfield(app.CYCLEPANELPanel, 'text');
app.Test_Field.HorizontalAlignment = 'right';
app.Test_Field.FontWeight = 'bold';
app.Test_Field.Position = [39 353 60 22];
app.Test_Field.Value = '10';

% Create pp_coarsePanel_2
app.pp_coarsePanel_2 = uipanel(app.CYCLEPANELPanel);
app.pp_coarsePanel_2.TitlePosition = 'centertop';
app.pp_coarsePanel_2.Title = 'pp_coarse';
app.pp_coarsePanel_2.FontWeight = 'bold';
app.pp_coarsePanel_2.Position = [5 233 260 117];

% Create CheckBox_a_coarse
app.CheckBox_a_coarse = uicheckbox(app.pp_coarsePanel_2);
app.CheckBox_a_coarse.ValueChangedFcn = createCallbackFcn(app, ↵
@CheckBox_a_coarseValueChanged, true);
app.CheckBox_a_coarse.Text = 'Active';
app.CheckBox_a_coarse.Position = [60 68 55 22];

% Create RangeEditField_a_coarse_range
app.RangeEditField_a_coarse_range = uieditfield(app.pp_coarsePanel_2, ↵
'text');
app.RangeEditField_a_coarse_range.HorizontalAlignment = 'right';
app.RangeEditField_a_coarse_range.Enable = 'off';
app.RangeEditField_a_coarse_range.Position = [193 68 57 22];
app.RangeEditField_a_coarse_range.Value = '0';

% Create a_coarseLabel
app.a_coarseLabel = uilabel(app.pp_coarsePanel_2);
app.a_coarseLabel.FontWeight = 'bold';

```

```

app.a_coarseLabel.Position = [3 68 58 22];
app.a_coarseLabel.Text = 'a_coarse';

% Create RangeLabel
app.RangeLabel = uilabel(app.pp_coarsePanel_2);
app.RangeLabel.Position = [139 68 55 22];
app.RangeLabel.Text = 'Range +-';

% Create CheckBox_b_coarse
app.CheckBox_b_coarse = uicontrol(app.pp_coarsePanel_2);
app.CheckBox_b_coarse.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_b_coarseValueChanged, true);
app.CheckBox_b_coarse.Text = 'Active';
app.CheckBox_b_coarse.Position = [60 46 55 22];

% Create RangeEditField_b_coarse_range
app.RangeEditField_b_coarse_range = uicontrol(app.pp_coarsePanel_2, ✓
'text');
app.RangeEditField_b_coarse_range.HorizontalAlignment = 'right';
app.RangeEditField_b_coarse_range.Enable = 'off';
app.RangeEditField_b_coarse_range.Position = [193 46 57 22];
app.RangeEditField_b_coarse_range.Value = '0';

% Create b_coarseLabel
app.b_coarseLabel = uilabel(app.pp_coarsePanel_2);
app.b_coarseLabel.FontWeight = 'bold';
app.b_coarseLabel.Position = [3 46 58 22];
app.b_coarseLabel.Text = 'b_coarse';

% Create RangeLabel_2
app.RangeLabel_2 = uilabel(app.pp_coarsePanel_2);
app.RangeLabel_2.Position = [139 46 55 22];
app.RangeLabel_2.Text = 'Range +-';

% Create CheckBox_c_coarse
app.CheckBox_c_coarse = uicontrol(app.pp_coarsePanel_2);
app.CheckBox_c_coarse.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_c_coarseValueChanged, true);
app.CheckBox_c_coarse.Text = 'Active';
app.CheckBox_c_coarse.Position = [60 24 55 22];

% Create RangeEditField_c_coarse_range
app.RangeEditField_c_coarse_range = uicontrol(app.pp_coarsePanel_2, ✓
'text');
app.RangeEditField_c_coarse_range.HorizontalAlignment = 'right';
app.RangeEditField_c_coarse_range.Enable = 'off';
app.RangeEditField_c_coarse_range.Position = [193 24 57 22];
app.RangeEditField_c_coarse_range.Value = '0';

% Create c_coarseLabel
app.c_coarseLabel = uilabel(app.pp_coarsePanel_2);
app.c_coarseLabel.FontWeight = 'bold';
app.c_coarseLabel.Position = [3 24 58 22];
app.c_coarseLabel.Text = 'c_coarse';

```

```

% Create RangeLabel_3
app.RangeLabel_3 = uilabel(app.pp_coarsePanel_2);
app.RangeLabel_3.Position = [139 24 55 22];
app.RangeLabel_3.Text = 'Range +-';

% Create CheckBox_A_coarse
app.CheckBox_A_coarse = uicheckbox(app.pp_coarsePanel_2);
app.CheckBox_A_coarse.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_A_coarseValueChanged, true);
app.CheckBox_A_coarse.Text = 'Active';
app.CheckBox_A_coarse.Position = [60 3 55 22];

% Create RangeEditField_A_coarse_range
app.RangeEditField_A_coarse_range = uieditfield(app.pp_coarsePanel_2,
'text');
app.RangeEditField_A_coarse_range.HorizontalAlignment = 'right';
app.RangeEditField_A_coarse_range.Enable = 'off';
app.RangeEditField_A_coarse_range.Position = [193 3 57 22];
app.RangeEditField_A_coarse_range.Value = '0';

% Create A_coarseLabel_2
app.A_coarseLabel_2 = uilabel(app.pp_coarsePanel_2);
app.A_coarseLabel_2.FontWeight = 'bold';
app.A_coarseLabel_2.Position = [3 3 60 22];
app.A_coarseLabel_2.Text = 'A_coarse';

% Create RangeLabel_4
app.RangeLabel_4 = uilabel(app.pp_coarsePanel_2);
app.RangeLabel_4.Position = [139 3 55 22];
app.RangeLabel_4.Text = 'Range +-';

% Create pp_finePanel_2
app.pp_finePanel_2 = uipanel(app.CYCLEPANELPanel);
app.pp_finePanel_2.TitlePosition = 'centertop';
app.pp_finePanel_2.Title = 'pp_fine';
app.pp_finePanel_2.FontWeight = 'bold';
app.pp_finePanel_2.Position = [303 233 272 116];

% Create CheckBox_a_fine
app.CheckBox_a_fine = uicheckbox(app.pp_finePanel_2);
app.CheckBox_a_fine.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_a_fineValueChanged, true);
app.CheckBox_a_fine.Text = 'Active';
app.CheckBox_a_fine.Position = [63 68 55 22];

% Create RangeEditField_a_fine_range
app.RangeEditField_a_fine_range = uieditfield(app.pp_finePanel_2, 'text');
app.RangeEditField_a_fine_range.HorizontalAlignment = 'right';
app.RangeEditField_a_fine_range.Enable = 'off';
app.RangeEditField_a_fine_range.Position = [206 68 57 22];
app.RangeEditField_a_fine_range.Value = '0';

% Create a_fineLabel_2
app.a_fineLabel_2 = uilabel(app.pp_finePanel_2);
app.a_fineLabel_2.FontWeight = 'bold';

```

```

app.a_fineLabel_2.Position = [19 67 40 22];
app.a_fineLabel_2.Text = 'a_fine';

% Create RangeLabel_5
app.RangeLabel_5 = uilabel(app.pp_finePanel_2);
app.RangeLabel_5.Position = [152 68 55 22];
app.RangeLabel_5.Text = 'Range +-';

% Create CheckBox_b_fine
app.CheckBox_b_fine = uicontrol(app.pp_finePanel_2);
app.CheckBox_b_fine.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_b_fineValueChanged, true);
app.CheckBox_b_fine.Text = 'Active';
app.CheckBox_b_fine.Position = [63 46 55 22];

% Create RangeEditField_b_fine_range
app.RangeEditField_b_fine_range = uieditfield(app.pp_finePanel_2, 'text');
app.RangeEditField_b_fine_range.HorizontalAlignment = 'right';
app.RangeEditField_b_fine_range.Enable = 'off';
app.RangeEditField_b_fine_range.Position = [206 46 57 22];
app.RangeEditField_b_fine_range.Value = '0';

% Create b_fineLabel_2
app.b_fineLabel_2 = uilabel(app.pp_finePanel_2);
app.b_fineLabel_2.FontWeight = 'bold';
app.b_fineLabel_2.Position = [19 45 41 22];
app.b_fineLabel_2.Text = 'b_fine';

% Create RangeLabel_6
app.RangeLabel_6 = uilabel(app.pp_finePanel_2);
app.RangeLabel_6.Position = [152 46 55 22];
app.RangeLabel_6.Text = 'Range +-';

% Create CheckBox_c_fine
app.CheckBox_c_fine = uicontrol(app.pp_finePanel_2);
app.CheckBox_c_fine.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_c_fineValueChanged, true);
app.CheckBox_c_fine.Text = 'Active';
app.CheckBox_c_fine.Position = [63 24 55 22];

% Create RangeEditField_c_fine_range
app.RangeEditField_c_fine_range = uieditfield(app.pp_finePanel_2, 'text');
app.RangeEditField_c_fine_range.HorizontalAlignment = 'right';
app.RangeEditField_c_fine_range.Enable = 'off';
app.RangeEditField_c_fine_range.Position = [206 24 57 22];
app.RangeEditField_c_fine_range.Value = '0';

% Create c_fineLabel_2
app.c_fineLabel_2 = uilabel(app.pp_finePanel_2);
app.c_fineLabel_2.FontWeight = 'bold';
app.c_fineLabel_2.Position = [19 23 40 22];
app.c_fineLabel_2.Text = 'c_fine';

% Create RangeLabel_7
app.RangeLabel_7 = uilabel(app.pp_finePanel_2);

```

```

app.RangeLabel_7.Position = [152 24 55 22];
app.RangeLabel_7.Text = 'Range +-';

% Create CheckBox_A_fine
app.CheckBox_A_fine = uicontrol(app.pp_finePanel_2);
app.CheckBox_A_fine.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_A_fineValueChanged, true);
app.CheckBox_A_fine.Enable = 'off';
app.CheckBox_A_fine.Text = 'Active';
app.CheckBox_A_fine.Position = [62 3 55 22];

% Create RangeEditField_A_fine_range
app.RangeEditField_A_fine_range = uieditfield(app.pp_finePanel_2, 'text');
app.RangeEditField_A_fine_range.HorizontalAlignment = 'right';
app.RangeEditField_A_fine_range.Enable = 'off';
app.RangeEditField_A_fine_range.Position = [206 3 57 22];
app.RangeEditField_A_fine_range.Value = '0';

% Create A_fineLabel_2
app.A_fineLabel_2 = uilabel(app.pp_finePanel_2);
app.A_fineLabel_2.FontWeight = 'bold';
app.A_fineLabel_2.Enable = 'off';
app.A_fineLabel_2.Position = [18 3 42 22];
app.A_fineLabel_2.Text = 'A_fine';

% Create RangeLabel_8
app.RangeLabel_8 = uilabel(app.pp_finePanel_2);
app.RangeLabel_8.Position = [152 3 55 22];
app.RangeLabel_8.Text = 'Range +-';

% Create LongPanel_2
app.LongPanel_2 = uipanel(app.CYCLEPANELPanel);
app.LongPanel_2.TitlePosition = 'centertop';
app.LongPanel_2.Title = 'Long';
app.LongPanel_2.FontWeight = 'bold';
app.LongPanel_2.Position = [606 218 260 132];

% Create CheckBox_H
app.CheckBox_H = uicontrol(app.LongPanel_2);
app.CheckBox_H.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_HValueChanged, true);
app.CheckBox_H.Text = 'Active';
app.CheckBox_H.Position = [60 88 55 22];

% Create RangeEditField_H_range
app.RangeEditField_H_range = uieditfield(app.LongPanel_2, 'text');
app.RangeEditField_H_range.HorizontalAlignment = 'right';
app.RangeEditField_H_range.Enable = 'off';
app.RangeEditField_H_range.Position = [194 88 57 22];
app.RangeEditField_H_range.Value = '0';

% Create HLabel_2
app.HLabel_2 = uilabel(app.LongPanel_2);
app.HLabel_2.FontWeight = 'bold';
app.HLabel_2.Position = [3 88 25 22];

```

```

app.HLabel_2.Text = 'H';

% Create RangeLabel_9
app.RangeLabel_9 = uilabel(app.LongPanel_2);
app.RangeLabel_9.Position = [140 88 55 22];
app.RangeLabel_9.Text = 'Range +-';

% Create CheckBox_K
app.CheckBox_K = uicheckbox(app.LongPanel_2);
app.CheckBox_K.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_KValueChanged, true);
app.CheckBox_K.Text = 'Active';
app.CheckBox_K.Position = [60 66 55 22];

% Create RangeEditField_K_range
app.RangeEditField_K_range = uieditfield(app.LongPanel_2, 'text');
app.RangeEditField_K_range.HorizontalAlignment = 'right';
app.RangeEditField_K_range.Enable = 'off';
app.RangeEditField_K_range.Position = [194 66 57 22];
app.RangeEditField_K_range.Value = '0';

% Create KLabel_2
app.KLabel_2 = uilabel(app.LongPanel_2);
app.KLabel_2.FontWeight = 'bold';
app.KLabel_2.Position = [3 66 25 22];
app.KLabel_2.Text = 'K';

% Create RangeLabel_10
app.RangeLabel_10 = uilabel(app.LongPanel_2);
app.RangeLabel_10.Position = [140 66 55 22];
app.RangeLabel_10.Text = 'Range +-';

% Create CheckBox_L
app.CheckBox_L = uicheckbox(app.LongPanel_2);
app.CheckBox_L.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_LValueChanged, true);
app.CheckBox_L.Text = 'Active';
app.CheckBox_L.Position = [60 44 55 22];

% Create RangeEditField_L_range
app.RangeEditField_L_range = uieditfield(app.LongPanel_2, 'text');
app.RangeEditField_L_range.HorizontalAlignment = 'right';
app.RangeEditField_L_range.Enable = 'off';
app.RangeEditField_L_range.Position = [194 44 57 22];
app.RangeEditField_L_range.Value = '0';

% Create LLabel_2
app.LLabel_2 = uilabel(app.LongPanel_2);
app.LLabel_2.FontWeight = 'bold';
app.LLabel_2.Position = [3 44 25 22];
app.LLabel_2.Text = 'L';

% Create RangeLabel_11
app.RangeLabel_11 = uilabel(app.LongPanel_2);
app.RangeLabel_11.Position = [140 44 55 22];

```

```

app.RangeLabel_11.Text = 'Range +-';

% Create CheckBox_J
app.CheckBox_J = uicontrol(app.LongPanel_2);
app.CheckBox_J.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_JValueChanged, true);
app.CheckBox_J.Text = 'Active';
app.CheckBox_J.Position = [60 23 55 22];

% Create RangeEditField_J_range
app.RangeEditField_J_range = uieditfield(app.LongPanel_2, 'text');
app.RangeEditField_J_range.HorizontalAlignment = 'right';
app.RangeEditField_J_range.Enable = 'off';
app.RangeEditField_J_range.Position = [194 23 57 22];
app.RangeEditField_J_range.Value = '0';

% Create JLabel_2
app.JLabel_2 = uilabel(app.LongPanel_2);
app.JLabel_2.FontWeight = 'bold';
app.JLabel_2.Position = [3 23 25 22];
app.JLabel_2.Text = 'J';

% Create RangeLabel_12
app.RangeLabel_12 = uilabel(app.LongPanel_2);
app.RangeLabel_12.Position = [140 23 55 22];
app.RangeLabel_12.Text = 'Range +-';

% Create CheckBox_delta_R
app.CheckBox_delta_R = uicontrol(app.LongPanel_2);
app.CheckBox_delta_R.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_delta_RValueChanged, true);
app.CheckBox_delta_R.Text = 'Active';
app.CheckBox_delta_R.Position = [60 3 55 22];

% Create RangeEditField_delta_R_range
app.RangeEditField_delta_R_range = uieditfield(app.LongPanel_2, 'text');
app.RangeEditField_delta_R_range.HorizontalAlignment = 'right';
app.RangeEditField_delta_R_range.Enable = 'off';
app.RangeEditField_delta_R_range.Position = [194 3 57 22];
app.RangeEditField_delta_R_range.Value = '0';

% Create delta_RLabel_2
app.delta_RLabel_2 = uilabel(app.LongPanel_2);
app.delta_RLabel_2.FontWeight = 'bold';
app.delta_RLabel_2.Position = [3 3 49 22];
app.delta_RLabel_2.Text = 'delta_R';

% Create RangeLabel_13
app.RangeLabel_13 = uilabel(app.LongPanel_2);
app.RangeLabel_13.Position = [140 3 55 22];
app.RangeLabel_13.Text = 'Range +-';

% Create KcoeffPanel
app.KcoeffPanel = uipanel(app.CYCLEPANELPanel);
app.KcoeffPanel.TitlePosition = 'centertop';

```

```

app.KcoeffPanel.Title = 'K coeff';
app.KcoeffPanel.FontWeight = 'bold';
app.KcoeffPanel.Position = [7 121 382 109];

% Create CheckBox_K_coeff
app.CheckBox_K_coeff = uicontrol(app.KcoeffPanel);
app.CheckBox_K_coeff.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_K_coeffValueChanged, true);
app.CheckBox_K_coeff.Text = 'Active';
app.CheckBox_K_coeff.Position = [27 63 55 22];

% Create RangeEditField_K_coeff_range
app.RangeEditField_K_coeff_range = uicontrol(app.KcoeffPanel, 'text');
app.RangeEditField_K_coeff_range.HorizontalAlignment = 'right';
app.RangeEditField_K_coeff_range.Enable = 'off';
app.RangeEditField_K_coeff_range.Position = [135 63 57 22];
app.RangeEditField_K_coeff_range.Value = '0';

% Create K1Label_2
app.K1Label_2 = uicontrol(app.KcoeffPanel);
app.K1Label_2.FontWeight = 'bold';
app.K1Label_2.Position = [8 63 25 22];
app.K1Label_2.Text = 'K1';

% Create RangeLabel_14
app.RangeLabel_14 = uicontrol(app.KcoeffPanel);
app.RangeLabel_14.Position = [81 63 55 22];
app.RangeLabel_14.Text = 'Range +-';

% Create CheckBox_K_coeff_2
app.CheckBox_K_coeff_2 = uicontrol(app.KcoeffPanel);
app.CheckBox_K_coeff_2.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_K_coeff_2ValueChanged, true);
app.CheckBox_K_coeff_2.Text = 'Active';
app.CheckBox_K_coeff_2.Position = [27 42 55 22];

% Create RangeEditField_K_coeff_range_2
app.RangeEditField_K_coeff_range_2 = uicontrol(app.KcoeffPanel, 'text');
app.RangeEditField_K_coeff_range_2.HorizontalAlignment = 'right';
app.RangeEditField_K_coeff_range_2.Enable = 'off';
app.RangeEditField_K_coeff_range_2.Position = [135 42 57 22];
app.RangeEditField_K_coeff_range_2.Value = '0';

% Create K2Label_2
app.K2Label_2 = uicontrol(app.KcoeffPanel);
app.K2Label_2.FontWeight = 'bold';
app.K2Label_2.Position = [8 42 25 22];
app.K2Label_2.Text = 'K2';

% Create RangeLabel_15
app.RangeLabel_15 = uicontrol(app.KcoeffPanel);
app.RangeLabel_15.Position = [81 42 55 22];
app.RangeLabel_15.Text = 'Range +-';

% Create CheckBox_K_coeff_3

```

```

        app.CheckBox_K_coeff_3 = uicontrol(app.KcoeffPanel);
        app.CheckBox_K_coeff_3.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_K_coeff_3ValueChanged, true);
        app.CheckBox_K_coeff_3.Text = 'Active';
        app.CheckBox_K_coeff_3.Position = [27 21 55 22];

        % Create RangeEditField_K_coeff_range_3
        app.RangeEditField_K_coeff_range_3 = uicontrol(app.KcoeffPanel, 'text');
        app.RangeEditField_K_coeff_range_3.HorizontalAlignment = 'right';
        app.RangeEditField_K_coeff_range_3.Enable = 'off';
        app.RangeEditField_K_coeff_range_3.Position = [135 21 57 22];
        app.RangeEditField_K_coeff_range_3.Value = '0';

        % Create K3Label_2
        app.K3Label_2 = uicontrol(app.KcoeffPanel);
        app.K3Label_2.FontWeight = 'bold';
        app.K3Label_2.Position = [8 21 25 22];
        app.K3Label_2.Text = 'K3';

        % Create RangeLabel_16
        app.RangeLabel_16 = uicontrol(app.KcoeffPanel);
        app.RangeLabel_16.Position = [81 21 55 22];
        app.RangeLabel_16.Text = 'Range +-';

        % Create CheckBox_K_coeff_4
        app.CheckBox_K_coeff_4 = uicontrol(app.KcoeffPanel);
        app.CheckBox_K_coeff_4.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_K_coeff_4ValueChanged, true);
        app.CheckBox_K_coeff_4.Text = 'Active';
        app.CheckBox_K_coeff_4.Position = [27 1 55 22];

        % Create RangeEditField_K_coeff_range_4
        app.RangeEditField_K_coeff_range_4 = uicontrol(app.KcoeffPanel, 'text');
        app.RangeEditField_K_coeff_range_4.HorizontalAlignment = 'right';
        app.RangeEditField_K_coeff_range_4.Enable = 'off';
        app.RangeEditField_K_coeff_range_4.Position = [135 1 57 22];
        app.RangeEditField_K_coeff_range_4.Value = '0';

        % Create K4Label_2
        app.K4Label_2 = uicontrol(app.KcoeffPanel);
        app.K4Label_2.FontWeight = 'bold';
        app.K4Label_2.Position = [8 1 25 22];
        app.K4Label_2.Text = 'K4';

        % Create RangeLabel_17
        app.RangeLabel_17 = uicontrol(app.KcoeffPanel);
        app.RangeLabel_17.Position = [81 1 55 22];
        app.RangeLabel_17.Text = 'Range +-';

        % Create CheckBox_K_coeff_5
        app.CheckBox_K_coeff_5 = uicontrol(app.KcoeffPanel);
        app.CheckBox_K_coeff_5.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_K_coeff_5ValueChanged, true);
        app.CheckBox_K_coeff_5.Text = 'Active';
        app.CheckBox_K_coeff_5.Position = [215 63 55 22];

```

```

% Create RangeEditField_K_coeff_range_5
app.RangeEditField_K_coeff_range_5 = uieditfield(app.KcoeffPanel, 'text');
app.RangeEditField_K_coeff_range_5.HorizontalAlignment = 'right';
app.RangeEditField_K_coeff_range_5.Enable = 'off';
app.RangeEditField_K_coeff_range_5.Position = [322 63 57 22];
app.RangeEditField_K_coeff_range_5.Value = '0';

% Create K5Label_2
app.K5Label_2 = uilabel(app.KcoeffPanel);
app.K5Label_2.FontWeight = 'bold';
app.K5Label_2.Position = [197 63 25 22];
app.K5Label_2.Text = 'K5';

% Create RangeLabel_18
app.RangeLabel_18 = uilabel(app.KcoeffPanel);
app.RangeLabel_18.Position = [268 63 55 22];
app.RangeLabel_18.Text = 'Range +-';

% Create CheckBox_K_coeff_6
app.CheckBox_K_coeff_6 = uicheckbox(app.KcoeffPanel);
app.CheckBox_K_coeff_6.ValueChangedFcn = createCallbackFcn(app, ↵
@CheckBox_K_coeff_6ValueChanged, true);
app.CheckBox_K_coeff_6.Text = 'Active';
app.CheckBox_K_coeff_6.Position = [215 43 55 22];

% Create RangeEditField_K_coeff_range_6
app.RangeEditField_K_coeff_range_6 = uieditfield(app.KcoeffPanel, 'text');
app.RangeEditField_K_coeff_range_6.HorizontalAlignment = 'right';
app.RangeEditField_K_coeff_range_6.Enable = 'off';
app.RangeEditField_K_coeff_range_6.Position = [322 43 57 22];
app.RangeEditField_K_coeff_range_6.Value = '0';

% Create K6Label_2
app.K6Label_2 = uilabel(app.KcoeffPanel);
app.K6Label_2.FontWeight = 'bold';
app.K6Label_2.Position = [197 43 25 22];
app.K6Label_2.Text = 'K6';

% Create RangeLabel_19
app.RangeLabel_19 = uilabel(app.KcoeffPanel);
app.RangeLabel_19.Position = [268 43 55 22];
app.RangeLabel_19.Text = 'Range +-';

% Create CheckBox_K_coeff_7
app.CheckBox_K_coeff_7 = uicheckbox(app.KcoeffPanel);
app.CheckBox_K_coeff_7.ValueChangedFcn = createCallbackFcn(app, ↵
@CheckBox_K_coeff_7ValueChanged, true);
app.CheckBox_K_coeff_7.Text = 'Active';
app.CheckBox_K_coeff_7.Position = [216 22 55 22];

% Create RangeEditField_K_coeff_range_7
app.RangeEditField_K_coeff_range_7 = uieditfield(app.KcoeffPanel, 'text');
app.RangeEditField_K_coeff_range_7.HorizontalAlignment = 'right';
app.RangeEditField_K_coeff_range_7.Enable = 'off';

```

```

app.RangeEditField_K_coeff_range_7.Position = [322 22 57 22];
app.RangeEditField_K_coeff_range_7.Value = '0';

% Create K7Label_2
app.K7Label_2 = uilabel(app.KcoeffPanel);
app.K7Label_2.FontWeight = 'bold';
app.K7Label_2.Position = [198 22 25 22];
app.K7Label_2.Text = 'K7';

% Create RangeLabel_20
app.RangeLabel_20 = uilabel(app.KcoeffPanel);
app.RangeLabel_20.Position = [268 22 55 22];
app.RangeLabel_20.Text = 'Range +-';

% Create CheckBox_K_coeff_8
app.CheckBox_K_coeff_8 = uicontrol(app.KcoeffPanel);
app.CheckBox_K_coeff_8.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_K_coeff_8ValueChanged, true);
app.CheckBox_K_coeff_8.Text = 'Active';
app.CheckBox_K_coeff_8.Position = [216 1 55 22];

% Create RangeEditField_K_coeff_range_8
app.RangeEditField_K_coeff_range_8 = uieditfield(app.KcoeffPanel, 'text');
app.RangeEditField_K_coeff_range_8.HorizontalAlignment = 'right';
app.RangeEditField_K_coeff_range_8.Enable = 'off';
app.RangeEditField_K_coeff_range_8.Position = [322 1 57 22];
app.RangeEditField_K_coeff_range_8.Value = '0';

% Create K8Label_2
app.K8Label_2 = uilabel(app.KcoeffPanel);
app.K8Label_2.FontWeight = 'bold';
app.K8Label_2.Position = [198 1 25 22];
app.K8Label_2.Text = 'K8';

% Create RangeLabel_21
app.RangeLabel_21 = uilabel(app.KcoeffPanel);
app.RangeLabel_21.Position = [268 1 55 22];
app.RangeLabel_21.Text = 'Range +-';

% Create LinearPanel_2
app.LinearPanel_2 = uipanel(app.CYCLEPANELPanel);
app.LinearPanel_2.TitlePosition = 'centertop';
app.LinearPanel_2.Title = 'Linear';
app.LinearPanel_2.FontWeight = 'bold';
app.LinearPanel_2.Position = [393 135 195 84];

% Create CheckBox_d_z
app.CheckBox_d_z = uicontrol(app.LinearPanel_2);
app.CheckBox_d_z.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_d_zValueChanged, true);
app.CheckBox_d_z.Text = 'Active';
app.CheckBox_d_z.Position = [32 33 55 22];

% Create RangeEditField_d_z
app.RangeEditField_d_z = uieditfield(app.LinearPanel_2, 'text');

```

```

app.RangeEditField_d_z.HorizontalAlignment = 'right';
app.RangeEditField_d_z.Enable = 'off';
app.RangeEditField_d_z.Position = [136 33 57 22];
app.RangeEditField_d_z.Value = '0';

% Create d_zLabel_2
app.d_zLabel_2 = uilabel(app.LinearPanel_2);
app.d_zLabel_2.FontWeight = 'bold';
app.d_zLabel_2.Position = [8 33 26 22];
app.d_zLabel_2.Text = 'd_z';

% Create RangeLabel_22
app.RangeLabel_22 = uilabel(app.LinearPanel_2);
app.RangeLabel_22.Position = [85 33 55 22];
app.RangeLabel_22.Text = 'Range +-';

% Create CheckBox_d_y
app.CheckBox_d_y = uicheckbox(app.LinearPanel_2);
app.CheckBox_d_y.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_d_yValueChanged, true);
app.CheckBox_d_y.Text = 'Active';
app.CheckBox_d_y.Position = [32 12 55 22];

% Create RangeEditField_d_y
app.RangeEditField_d_y = uieditfield(app.LinearPanel_2, 'text');
app.RangeEditField_d_y.HorizontalAlignment = 'right';
app.RangeEditField_d_y.Enable = 'off';
app.RangeEditField_d_y.Position = [136 12 57 22];
app.RangeEditField_d_y.Value = '0';

% Create d_yLabel_2
app.d_yLabel_2 = uilabel(app.LinearPanel_2);
app.d_yLabel_2.FontWeight = 'bold';
app.d_yLabel_2.Position = [8 12 26 22];
app.d_yLabel_2.Text = 'd_y';

% Create RangeLabel_23
app.RangeLabel_23 = uilabel(app.LinearPanel_2);
app.RangeLabel_23.Position = [85 12 55 22];
app.RangeLabel_23.Text = 'Range +-';

% Create SPSPanel_2
app.SPSPanel_2 = uipanel(app.CYCLEPANELPanel);
app.SPSPanel_2.TitlePosition = 'centertop';
app.SPSPanel_2.Title = 'SPS';
app.SPSPanel_2.FontWeight = 'bold';
app.SPSPanel_2.Position = [4 3 864 116];

% Create CheckBox_SPS_high_1
app.CheckBox_SPS_high_1 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_high_1.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_1ValueChanged, true);
app.CheckBox_SPS_high_1.Text = 'Active';
app.CheckBox_SPS_high_1.Position = [70 65 55 22];

```

```

% Create RangeEditField_SPS_high_range_1
app.RangeEditField_SPS_high_range_1 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_1.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_1.Enable = 'off';
app.RangeEditField_SPS_high_range_1.Position = [179 65 39 22];
app.RangeEditField_SPS_high_range_1.Value = '0';

% Create SPS1HighLabel
app.SPS1HighLabel = uilabel(app.SPSPanel_2);
app.SPS1HighLabel.FontWeight = 'bold';
app.SPS1HighLabel.Position = [6 65 66 22];
app.SPS1HighLabel.Text = 'SPS1 High';

% Create RangeLabel_24
app.RangeLabel_24 = uilabel(app.SPSPanel_2);
app.RangeLabel_24.Position = [125 65 55 22];
app.RangeLabel_24.Text = 'Range +-';

% Create CheckBox_SPS_Low_1
app.CheckBox_SPS_Low_1 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_Low_1.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_1ValueChanged, true);
app.CheckBox_SPS_Low_1.Text = 'Active';
app.CheckBox_SPS_Low_1.Position = [70 44 55 22];

% Create RangeEditField_SPS_Low_range_1
app.RangeEditField_SPS_Low_range_1 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_1.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_1.Enable = 'off';
app.RangeEditField_SPS_Low_range_1.Position = [179 44 39 22];
app.RangeEditField_SPS_Low_range_1.Value = '0';

% Create SPS1LowLabel
app.SPS1LowLabel = uilabel(app.SPSPanel_2);
app.SPS1LowLabel.FontWeight = 'bold';
app.SPS1LowLabel.Position = [6 44 64 22];
app.SPS1LowLabel.Text = 'SPS1 Low';

% Create RangeLabel_25
app.RangeLabel_25 = uilabel(app.SPSPanel_2);
app.RangeLabel_25.Position = [125 44 55 22];
app.RangeLabel_25.Text = 'Range +-';

% Create CheckBox_SPS_high_2
app.CheckBox_SPS_high_2 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_high_2.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_2ValueChanged, true);
app.CheckBox_SPS_high_2.Text = 'Active';
app.CheckBox_SPS_high_2.Position = [70 23 55 22];

% Create RangeEditField_SPS_high_range_2
app.RangeEditField_SPS_high_range_2 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_2.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_2.Enable = 'off';
app.RangeEditField_SPS_high_range_2.Position = [179 23 39 22];

```

```

app.RangeEditField_SPS_high_range_2.Value = '0';

% Create SPS2HighLabel
app.SPS2HighLabel = uilabel(app.SPSPanel_2);
app.SPS2HighLabel.FontWeight = 'bold';
app.SPS2HighLabel.Position = [6 23 66 22];
app.SPS2HighLabel.Text = 'SPS2 High';

% Create RangeLabel_26
app.RangeLabel_26 = uilabel(app.SPSPanel_2);
app.RangeLabel_26.Position = [125 23 55 22];
app.RangeLabel_26.Text = 'Range +-';

% Create CheckBox_SPS_Low_2
app.CheckBox_SPS_Low_2 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_Low_2.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_2ValueChanged, true);
app.CheckBox_SPS_Low_2.Text = 'Active';
app.CheckBox_SPS_Low_2.Position = [70 2 55 22];

% Create RangeEditField_SPS_Low_range_2
app.RangeEditField_SPS_Low_range_2 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_2.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_2.Enable = 'off';
app.RangeEditField_SPS_Low_range_2.Position = [179 2 39 22];
app.RangeEditField_SPS_Low_range_2.Value = '0';

% Create SPS2LowLabel
app.SPS2LowLabel = uilabel(app.SPSPanel_2);
app.SPS2LowLabel.FontWeight = 'bold';
app.SPS2LowLabel.Position = [6 2 64 22];
app.SPS2LowLabel.Text = 'SPS2 Low';

% Create RangeLabel_27
app.RangeLabel_27 = uilabel(app.SPSPanel_2);
app.RangeLabel_27.Position = [125 2 55 22];
app.RangeLabel_27.Text = 'Range +-';

% Create CheckBox_SPS_high_3
app.CheckBox_SPS_high_3 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_high_3.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_3ValueChanged, true);
app.CheckBox_SPS_high_3.Text = 'Active';
app.CheckBox_SPS_high_3.Position = [286 65 55 22];

% Create RangeEditField_SPS_high_range_3
app.RangeEditField_SPS_high_range_3 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_3.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_3.Enable = 'off';
app.RangeEditField_SPS_high_range_3.Position = [395 65 40 22];
app.RangeEditField_SPS_high_range_3.Value = '0';

% Create SPS3HighLabel
app.SPS3HighLabel = uilabel(app.SPSPanel_2);
app.SPS3HighLabel.FontWeight = 'bold';

```

```

app.SPS3HighLabel.Position = [222 65 66 22];
app.SPS3HighLabel.Text = 'SPS3 High';

% Create RangeLabel_28
app.RangeLabel_28 = uilabel(app.SPSPanel_2);
app.RangeLabel_28.Position = [341 65 55 22];
app.RangeLabel_28.Text = 'Range +-';

% Create CheckBox_SPS_Low_3
app.CheckBox_SPS_Low_3 = uicontrol(app.SPSPanel_2);
app.CheckBox_SPS_Low_3.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_3ValueChanged, true);
app.CheckBox_SPS_Low_3.Text = 'Active';
app.CheckBox_SPS_Low_3.Position = [286 44 55 22];

% Create RangeEditField_SPS_Low_range_3
app.RangeEditField_SPS_Low_range_3 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_3.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_3.Enable = 'off';
app.RangeEditField_SPS_Low_range_3.Position = [395 44 40 22];
app.RangeEditField_SPS_Low_range_3.Value = '0';

% Create SPS3LowLabel
app.SPS3LowLabel = uilabel(app.SPSPanel_2);
app.SPS3LowLabel.FontWeight = 'bold';
app.SPS3LowLabel.Position = [222 44 64 22];
app.SPS3LowLabel.Text = 'SPS3 Low';

% Create RangeLabel_29
app.RangeLabel_29 = uilabel(app.SPSPanel_2);
app.RangeLabel_29.Position = [341 44 55 22];
app.RangeLabel_29.Text = 'Range +-';

% Create CheckBox_SPS_high_4
app.CheckBox_SPS_high_4 = uicontrol(app.SPSPanel_2);
app.CheckBox_SPS_high_4.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_4ValueChanged, true);
app.CheckBox_SPS_high_4.Text = 'Active';
app.CheckBox_SPS_high_4.Position = [286 23 55 22];

% Create RangeEditField_SPS_high_range_4
app.RangeEditField_SPS_high_range_4 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_4.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_4.Enable = 'off';
app.RangeEditField_SPS_high_range_4.Position = [395 23 40 22];
app.RangeEditField_SPS_high_range_4.Value = '0';

% Create SPS4HighLabel
app.SPS4HighLabel = uilabel(app.SPSPanel_2);
app.SPS4HighLabel.FontWeight = 'bold';
app.SPS4HighLabel.Position = [222 23 66 22];
app.SPS4HighLabel.Text = 'SPS4 High';

% Create RangeLabel_30
app.RangeLabel_30 = uilabel(app.SPSPanel_2);

```

```

app.RangeLabel_30.Position = [341 23 55 22];
app.RangeLabel_30.Text = 'Range +-';

% Create CheckBox_SPS_Low_4
app.CheckBox_SPS_Low_4 = uicontrol(app.SPSPanel_2);
app.CheckBox_SPS_Low_4.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_SPS_Low_4ValueChanged, true);
app.CheckBox_SPS_Low_4.Text = 'Active';
app.CheckBox_SPS_Low_4.Position = [286 2 55 22];

% Create RangeEditField_SPS_Low_range_4
app.RangeEditField_SPS_Low_range_4 = uicontrol(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_4.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_4.Enable = 'off';
app.RangeEditField_SPS_Low_range_4.Position = [395 2 40 22];
app.RangeEditField_SPS_Low_range_4.Value = '0';

% Create SPS4LowLabel
app.SPS4LowLabel = uicontrol(app.SPSPanel_2);
app.SPS4LowLabel.FontWeight = 'bold';
app.SPS4LowLabel.Position = [222 2 64 22];
app.SPS4LowLabel.Text = 'SPS4 Low';

% Create RangeLabel_31
app.RangeLabel_31 = uicontrol(app.SPSPanel_2);
app.RangeLabel_31.Position = [341 2 55 22];
app.RangeLabel_31.Text = 'Range +-';

% Create CheckBox_SPS_high_5
app.CheckBox_SPS_high_5 = uicontrol(app.SPSPanel_2);
app.CheckBox_SPS_high_5.ValueChangedFcn = createCallbackFcn(app,
@CheckBox_SPS_high_5ValueChanged, true);
app.CheckBox_SPS_high_5.Text = 'Active';
app.CheckBox_SPS_high_5.Position = [498 65 55 22];

% Create RangeEditField_SPS_high_range_5
app.RangeEditField_SPS_high_range_5 = uicontrol(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_5.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_5.Enable = 'off';
app.RangeEditField_SPS_high_range_5.Position = [607 65 39 22];
app.RangeEditField_SPS_high_range_5.Value = '0';

% Create SPS5HighLabel
app.SPS5HighLabel = uicontrol(app.SPSPanel_2);
app.SPS5HighLabel.FontWeight = 'bold';
app.SPS5HighLabel.Position = [434 65 66 22];
app.SPS5HighLabel.Text = 'SPS5 High';

% Create RangeLabel_32
app.RangeLabel_32 = uicontrol(app.SPSPanel_2);
app.RangeLabel_32.Position = [553 65 55 22];
app.RangeLabel_32.Text = 'Range +-';

% Create CheckBox_SPS_Low_5
app.CheckBox_SPS_Low_5 = uicontrol(app.SPSPanel_2);

```

```

        app.CheckBox_SPS_Low_5.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_5ValueChanged, true);
        app.CheckBox_SPS_Low_5.Text = 'Active';
        app.CheckBox_SPS_Low_5.Position = [498 44 55 22];

        % Create RangeEditField_SPS_Low_range_5
        app.RangeEditField_SPS_Low_range_5 = uieditfield(app.SPSPanel_2, 'text');
        app.RangeEditField_SPS_Low_range_5.HorizontalAlignment = 'right';
        app.RangeEditField_SPS_Low_range_5.Enable = 'off';
        app.RangeEditField_SPS_Low_range_5.Position = [607 44 39 22];
        app.RangeEditField_SPS_Low_range_5.Value = '0';

        % Create SPS5LowLabel
        app.SPS5LowLabel = uilabel(app.SPSPanel_2);
        app.SPS5LowLabel.FontWeight = 'bold';
        app.SPS5LowLabel.Position = [434 44 64 22];
        app.SPS5LowLabel.Text = 'SPS5 Low';

        % Create RangeLabel_33
        app.RangeLabel_33 = uilabel(app.SPSPanel_2);
        app.RangeLabel_33.Position = [553 44 55 22];
        app.RangeLabel_33.Text = 'Range +-';

        % Create CheckBox_SPS_high_6
        app.CheckBox_SPS_high_6 = uicheckbox(app.SPSPanel_2);
        app.CheckBox_SPS_high_6.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_6ValueChanged, true);
        app.CheckBox_SPS_high_6.Text = 'Active';
        app.CheckBox_SPS_high_6.Position = [498 23 55 22];

        % Create RangeEditField_SPS_high_range_6
        app.RangeEditField_SPS_high_range_6 = uieditfield(app.SPSPanel_2, 'text');
        app.RangeEditField_SPS_high_range_6.HorizontalAlignment = 'right';
        app.RangeEditField_SPS_high_range_6.Enable = 'off';
        app.RangeEditField_SPS_high_range_6.Position = [607 23 39 22];
        app.RangeEditField_SPS_high_range_6.Value = '0';

        % Create SPS6HighLabel
        app.SPS6HighLabel = uilabel(app.SPSPanel_2);
        app.SPS6HighLabel.FontWeight = 'bold';
        app.SPS6HighLabel.Position = [434 23 66 22];
        app.SPS6HighLabel.Text = 'SPS6 High';

        % Create RangeLabel_34
        app.RangeLabel_34 = uilabel(app.SPSPanel_2);
        app.RangeLabel_34.Position = [553 23 55 22];
        app.RangeLabel_34.Text = 'Range +-';

        % Create CheckBox_SPS_Low_6
        app.CheckBox_SPS_Low_6 = uicheckbox(app.SPSPanel_2);
        app.CheckBox_SPS_Low_6.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_6ValueChanged, true);
        app.CheckBox_SPS_Low_6.Text = 'Active';
        app.CheckBox_SPS_Low_6.Position = [498 2 55 22];

```

```

% Create RangeEditField_SPS_Low_range_6
app.RangeEditField_SPS_Low_range_6 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_6.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_6.Enable = 'off';
app.RangeEditField_SPS_Low_range_6.Position = [607 2 39 22];
app.RangeEditField_SPS_Low_range_6.Value = '0';

% Create SPS6LowLabel
app.SPS6LowLabel = uilabel(app.SPSPanel_2);
app.SPS6LowLabel.FontWeight = 'bold';
app.SPS6LowLabel.Position = [434 2 64 22];
app.SPS6LowLabel.Text = 'SPS6 Low';

% Create RangeLabel_35
app.RangeLabel_35 = uilabel(app.SPSPanel_2);
app.RangeLabel_35.Position = [553 2 55 22];
app.RangeLabel_35.Text = 'Range +-';

% Create CheckBox_SPS_high_7
app.CheckBox_SPS_high_7 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_high_7.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_7ValueChanged, true);
app.CheckBox_SPS_high_7.Text = 'Active';
app.CheckBox_SPS_high_7.Position = [713 65 55 22];

% Create RangeEditField_SPS_high_range_7
app.RangeEditField_SPS_high_range_7 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_7.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_7.Enable = 'off';
app.RangeEditField_SPS_high_range_7.Position = [822 65 39 22];
app.RangeEditField_SPS_high_range_7.Value = '0';

% Create SPS7HighLabel
app.SPS7HighLabel = uilabel(app.SPSPanel_2);
app.SPS7HighLabel.FontWeight = 'bold';
app.SPS7HighLabel.Position = [649 65 66 22];
app.SPS7HighLabel.Text = 'SPS7 High';

% Create RangeLabel_36
app.RangeLabel_36 = uilabel(app.SPSPanel_2);
app.RangeLabel_36.Position = [768 65 55 22];
app.RangeLabel_36.Text = 'Range +-';

% Create CheckBox_SPS_Low_7
app.CheckBox_SPS_Low_7 = uicheckbox(app.SPSPanel_2);
app.CheckBox_SPS_Low_7.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_7ValueChanged, true);
app.CheckBox_SPS_Low_7.Text = 'Active';
app.CheckBox_SPS_Low_7.Position = [713 44 55 22];

% Create RangeEditField_SPS_Low_range_7
app.RangeEditField_SPS_Low_range_7 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_7.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_7.Enable = 'off';
app.RangeEditField_SPS_Low_range_7.Position = [822 44 39 22];

```

```

app.RangeEditField_SPS_Low_range_7.Value = '0';

% Create SPS7LowLabel
app.SPS7LowLabel = uilabel(app.SPSPanel_2);
app.SPS7LowLabel.FontWeight = 'bold';
app.SPS7LowLabel.Position = [649 44 64 22];
app.SPS7LowLabel.Text = 'SPS7 Low';

% Create RangeLabel_37
app.RangeLabel_37 = uilabel(app.SPSPanel_2);
app.RangeLabel_37.Position = [768 44 55 22];
app.RangeLabel_37.Text = 'Range +/-';

% Create CheckBox_SPS_high_8
app.CheckBox_SPS_high_8 = uicontrol(app.SPSPanel_2);
app.CheckBox_SPS_high_8.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_high_8ValueChanged, true);
app.CheckBox_SPS_high_8.Text = 'Active';
app.CheckBox_SPS_high_8.Position = [713 23 55 22];

% Create RangeEditField_SPS_high_range_8
app.RangeEditField_SPS_high_range_8 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_high_range_8.HorizontalAlignment = 'right';
app.RangeEditField_SPS_high_range_8.Enable = 'off';
app.RangeEditField_SPS_high_range_8.Position = [822 23 39 22];
app.RangeEditField_SPS_high_range_8.Value = '0';

% Create SPS8HighLabel
app.SPS8HighLabel = uilabel(app.SPSPanel_2);
app.SPS8HighLabel.FontWeight = 'bold';
app.SPS8HighLabel.Position = [649 23 66 22];
app.SPS8HighLabel.Text = 'SPS8 High';

% Create RangeLabel_38
app.RangeLabel_38 = uilabel(app.SPSPanel_2);
app.RangeLabel_38.Position = [768 23 55 22];
app.RangeLabel_38.Text = 'Range +/-';

% Create CheckBox_SPS_Low_8
app.CheckBox_SPS_Low_8 = uicontrol(app.SPSPanel_2);
app.CheckBox_SPS_Low_8.ValueChangedFcn = createCallbackFcn(app, ✓
@CheckBox_SPS_Low_8ValueChanged, true);
app.CheckBox_SPS_Low_8.Text = 'Active';
app.CheckBox_SPS_Low_8.Position = [713 2 55 22];

% Create RangeEditField_SPS_Low_range_8
app.RangeEditField_SPS_Low_range_8 = uieditfield(app.SPSPanel_2, 'text');
app.RangeEditField_SPS_Low_range_8.HorizontalAlignment = 'right';
app.RangeEditField_SPS_Low_range_8.Enable = 'off';
app.RangeEditField_SPS_Low_range_8.Position = [822 2 39 22];
app.RangeEditField_SPS_Low_range_8.Value = '0';

% Create SPS8LowLabel
app.SPS8LowLabel = uilabel(app.SPSPanel_2);
app.SPS8LowLabel.FontWeight = 'bold';

```

```

app.SPS8LowLabel.Position = [649 2 64 22];
app.SPS8LowLabel.Text = 'SPS8 Low';

% Create RangeLabel_39
app.RangeLabel_39 = uilabel(app.SPSPanel_2);
app.RangeLabel_39.Position = [768 2 55 22];
app.RangeLabel_39.Text = 'Range +/-';

% Create PathCSVEditField
app.PathCSVEditField = uieditfield(app.CYCLEPANELPanel, 'text');
app.PathCSVEditField.Enable = 'off';
app.PathCSVEditField.Position = [468 353 159 22];
app.PathCSVEditField.Value = '<ACTUAL MATLAB PATH>';

% Create SelectpathCSVButton
app.SelectpathCSVButton = uibutton(app.CYCLEPANELPanel, 'push');
app.SelectpathCSVButton.ButtonPushedFcn = createCallbackFcn(app, ↵
@SelectpathCSVButtonPushed, true);
app.SelectpathCSVButton.BackgroundColor = [0.6745 0.8039 0.8902];
app.SelectpathCSVButton.FontWeight = 'bold';
app.SelectpathCSVButton.Enable = 'off';
app.SelectpathCSVButton.Position = [277 353 106 22];
app.SelectpathCSVButton.Text = 'Select path CSV';

% Create MontecarloCheckBox
app.MontecarloCheckBox = uicheckbox(app.CYCLEPANELPanel);
app.MontecarloCheckBox.ValueChangedFcn = createCallbackFcn(app, ↵
@MontecarloCheckBoxValueChanged, true);
app.MontecarloCheckBox.Text = 'Montecarlo';
app.MontecarloCheckBox.FontWeight = 'bold';
app.MontecarloCheckBox.Position = [640 353 86 22];

% Create RangeoptionforSPSDropDown
app.RangeoptionforSPSDropDown = uidropdown(app.CYCLEPANELPanel);
app.RangeoptionforSPSDropDown.Items = {'Absolute', 'Percentage'};
app.RangeoptionforSPSDropDown.FontWeight = 'bold';
app.RangeoptionforSPSDropDown.Position = [740 353 100 22];
app.RangeoptionforSPSDropDown.Value = 'Percentage';

% Create CSV_REAL_Drop_Down
app.CSV_REAL_Drop_Down = uidropdown(app.CYCLEPANELPanel);
app.CSV_REAL_Drop_Down.Items = {'Only Real Time', 'Only CSV', 'CSV and ↵
Real Time'};
app.CSV_REAL_Drop_Down.ValueChangedFcn = createCallbackFcn(app, ↵
@CSV_REAL_Drop_DownValueChanged, true);
app.CSV_REAL_Drop_Down.Position = [110 353 153 22];
app.CSV_REAL_Drop_Down.Value = 'Only Real Time';

% Create PathCSVLabel
app.PathCSVLabel = uilabel(app.CYCLEPANELPanel);
app.PathCSVLabel.FontWeight = 'bold';
app.PathCSVLabel.Position = [399 353 66 22];
app.PathCSVLabel.Text = 'Path CSV: ';

% Create STARTPanel

```

```

app.STARTPanel = uipanel(app.CYCLEPANELPanel);
app.STARTPanel.TitlePosition = 'centertop';
app.STARTPanel.Title = 'START';
app.STARTPanel.BackgroundColor = [0.4157 0.9216 0.7922];
app.STARTPanel.FontWeight = 'bold';
app.STARTPanel.Position = [605 122 259 95];

% Create STARTSIMULATIONSPSTOXYZButton
app.STARTSIMULATIONSPSTOXYZButton = uibutton(app.STARTPanel, 'push');
app.STARTSIMULATIONSPSTOXYZButton.ButtonPushedFcn = createCallbackFcn(app,
@STARTSIMULATIONSPSTOXYZButtonPushed, true);
app.STARTSIMULATIONSPSTOXYZButton.BackgroundColor = [0.9686 0.8039
0.4157];
app.STARTSIMULATIONSPSTOXYZButton.FontSize = 14;
app.STARTSIMULATIONSPSTOXYZButton.FontWeight = 'bold';
app.STARTSIMULATIONSPSTOXYZButton.Position = [4 3 245 30];
app.STARTSIMULATIONSPSTOXYZButton.Text = 'START SIMULATION SPS-TO-XYZ';

% Create SaverelantsvaluesinaCSVCheckBox
app.SaverelantsvaluesinaCSVCheckBox = uicheckbox(app.STARTPanel);
app.SaverelantsvaluesinaCSVCheckBox.ValueChangedFcn = createCallbackFcn
(app, @SaverelantsvaluesinaCSVCheckBoxValueChanged, true);
app.SaverelantsvaluesinaCSVCheckBox.Text = 'Save relevant values in a
CSV ';
app.SaverelantsvaluesinaCSVCheckBox.Position = [30 52 194 22];

% Create PathCSV_relevant_values
app.PathCSV_relevant_values = uieditfield(app.STARTPanel, 'text');
app.PathCSV_relevant_values.Enable = 'off';
app.PathCSV_relevant_values.Position = [82 34 159 22];
app.PathCSV_relevant_values.Value = '<ACTUAL MATLAB PATH>';

% Create PathCSVLabel_2
app.PathCSVLabel_2 = uilabel(app.STARTPanel);
app.PathCSVLabel_2.FontWeight = 'bold';
app.PathCSVLabel_2.Position = [13 34 66 22];
app.PathCSVLabel_2.Text = 'Path CSV: ';

% Create GENERALPLOTSTab
app.GENERALPLOTSTab = uitab(app.TabGroup2);
app.GENERALPLOTSTab.Title = 'GENERAL PLOTS';

% Create PlotoftwoselectedvariablesPanel
app.PlotoftwoselectedvariablesPanel = uipanel(app.GENERALPLOTSTab);
app.PlotoftwoselectedvariablesPanel.TitlePosition = 'centertop';
app.PlotoftwoselectedvariablesPanel.Title = 'Plot of two selected
variables';
app.PlotoftwoselectedvariablesPanel.BackgroundColor = [0.9098 0.9294
0.451];
app.PlotoftwoselectedvariablesPanel.FontWeight = 'bold';
app.PlotoftwoselectedvariablesPanel.FontSize = 14;
app.PlotoftwoselectedvariablesPanel.Position = [2 315 876 471];

% Create PLOTS_TWO_VARIABLES
app.PLOTS_TWO_VARIABLES = uiaxes(app.PlotoftwoselectedvariablesPanel);

```

```

title(app.PLOTS_TWO_VARIABLES, 'Two variables plot')
xlabel(app.PLOTS_TWO_VARIABLES, '')
ylabel(app.PLOTS_TWO_VARIABLES, '')
app.PLOTS_TWO_VARIABLES.FontWeight = 'bold';
app.PLOTS_TWO_VARIABLES.XMinorTick = 'on';
app.PLOTS_TWO_VARIABLES.YMinorTick = 'on';
app.PLOTS_TWO_VARIABLES.XMinorGrid = 'on';
app.PLOTS_TWO_VARIABLES.YMinorGrid = 'on';
app.PLOTS_TWO_VARIABLES.BackgroundColor = [0.9098 0.9294 0.451];
app.PLOTS_TWO_VARIABLES.Position = [60 10 617 374];

% Create FilteraxespanelPanel
app.FilteraxespanelPanel = uipanel(app.PlotoftwoselectedvariablesPanel);
app.FilteraxespanelPanel.TitlePosition = 'centertop';
app.FilteraxespanelPanel.Title = 'Filter axes panel';
app.FilteraxespanelPanel.BackgroundColor = [0.949 0.4745 0.4745];
app.FilteraxespanelPanel.FontWeight = 'bold';
app.FilteraxespanelPanel.Position = [676 10 190 375];

% Create ChooseaxesButtonGroup
app.ChooseaxesButtonGroup = uibuttongroup(app.FilteraxespanelPanel);
app.ChooseaxesButtonGroup.SelectionChangedFcn = createCallbackFcn(app, ✓
@ChooseaxesButtonGroupSelectionChanged, true);
app.ChooseaxesButtonGroup.Title = 'Choose axes';
app.ChooseaxesButtonGroup.Position = [7 278 173 73];

% Create XaxesButton
app.XaxesButton = uiradiobutton(app.ChooseaxesButtonGroup);
app.XaxesButton.Text = 'X axes';
app.XaxesButton.Position = [11 27 59 22];
app.XaxesButton.Value = true;

% Create YaxesButton
app.YaxesButton = uiradiobutton(app.ChooseaxesButtonGroup);
app.YaxesButton.Text = 'Y axes';
app.YaxesButton.Position = [11 5 65 22];

% Create XFilterPanel
app.XFilterPanel = uipanel(app.FilteraxespanelPanel);
app.XFilterPanel.TitlePosition = 'centertop';
app.XFilterPanel.Title = 'X Filter';
app.XFilterPanel.FontWeight = 'bold';
app.XFilterPanel.Position = [7 161 173 110];

% Create Edit_Min_X
app.Edit_Min_X = uieditfield(app.XFilterPanel, 'numeric');
app.Edit_Min_X.Position = [69 24 98 22];

% Create MinLabel
app.MinLabel = uilabel(app.XFilterPanel);
app.MinLabel.Position = [5 24 25 22];
app.MinLabel.Text = 'Min';

% Create Edit_Max_X
app.Edit_Max_X = uieditfield(app.XFilterPanel, 'numeric');

```

```
app.Edit_Max_X.Position = [69 3 98 22];

% Create MaxLabel
app.MaxLabel = uilabel(app.XFilterPanel);
app.MaxLabel.Position = [5 3 28 22];
app.MaxLabel.Text = 'Max';

% Create GraphMinLabel
app.GraphMinLabel = uilabel(app.XFilterPanel);
app.GraphMinLabel.Position = [5 65 62 22];
app.GraphMinLabel.Text = 'Graph Min';

% Create GraphMaxLabel
app.GraphMaxLabel = uilabel(app.XFilterPanel);
app.GraphMaxLabel.Position = [5 44 65 22];
app.GraphMaxLabel.Text = 'Graph Max';

% Create actual_min_x
app.actual_min_x = uilabel(app.XFilterPanel);
app.actual_min_x.Position = [72 65 94 22];
app.actual_min_x.Text = '';

% Create actual_max_x
app.actual_max_x = uilabel(app.XFilterPanel);
app.actual_max_x.Position = [71 44 94 22];
app.actual_max_x.Text = '';

% Create YFilterPanel
app.YFilterPanel = uipanel(app.FilteraxespanelPanel);
app.YFilterPanel.TitlePosition = 'centertop';
app.YFilterPanel.Title = 'Y Filter';
app.YFilterPanel.FontWeight = 'bold';
app.YFilterPanel.Position = [7 43 173 110];

% Create Edit_Min_Y
app.Edit_Min_Y = uieditfield(app.YFilterPanel, 'numeric');
app.Edit_Min_Y.Enable = 'off';
app.Edit_Min_Y.Position = [69 25 98 22];

% Create MinLabel_2
app.MinLabel_2 = uilabel(app.YFilterPanel);
app.MinLabel_2.Position = [7 25 28 22];
app.MinLabel_2.Text = 'Min ';

% Create Edit_Max_Y
app.Edit_Max_Y = uieditfield(app.YFilterPanel, 'numeric');
app.Edit_Max_Y.Enable = 'off';
app.Edit_Max_Y.Position = [69 4 98 22];

% Create MaxLabel_2
app.MaxLabel_2 = uilabel(app.YFilterPanel);
app.MaxLabel_2.Position = [7 4 32 22];
app.MaxLabel_2.Text = 'Max ';

% Create GraphMinLabel_2
```

```

app.GraphMinLabel_2 = uilabel(app.YFilterPanel);
app.GraphMinLabel_2.Position = [5 66 62 22];
app.GraphMinLabel_2.Text = 'Graph Min';

% Create GraphMaxLabel_2
app.GraphMaxLabel_2 = uilabel(app.YFilterPanel);
app.GraphMaxLabel_2.Position = [5 45 65 22];
app.GraphMaxLabel_2.Text = 'Graph Max';

% Create actual_min_y
app.actual_min_y = uilabel(app.YFilterPanel);
app.actual_min_y.Position = [72 66 94 22];
app.actual_min_y.Text = '';

% Create actual_max_y
app.actual_max_y = uilabel(app.YFilterPanel);
app.actual_max_y.Position = [71 45 94 22];
app.actual_max_y.Text = '';

% Create FilteraxesButton
app.FilteraxesButton = uibutton(app.FilteraxespanelPanel, 'state');
app.FilteraxesButton.ValueChangedFcn = createCallbackFcn(app, ✓
@FilteraxesButtonValueChanged, true);
app.FilteraxesButton.Text = 'Filter axes';
app.FilteraxesButton.BackgroundColor = [0.8353 0.8902 0.9608];
app.FilteraxesButton.FontWeight = 'bold';
app.FilteraxesButton.Position = [44 11 100 22];

% Create AdviseforthectionofaTwovariablesplotLabel
app.AdviseforthectionofaTwovariablesplotLabel = uilabel(app. ✓
PlotoftwoselectedvariablesPanel);
app.AdviseforthectionofaTwovariablesplotLabel.VerticalAlignment = ✓
'top';
app.AdviseforthectionofaTwovariablesplotLabel.FontWeight = 'bold';
app.AdviseforthectionofaTwovariablesplotLabel.Position = [60 387 766 ✓
57];
app.AdviseforthectionofaTwovariablesplotLabel.Text = {'Advice for the ✓
creation of a "Two variables plot":'; '- select two variables'; '- the first selected ✓
variable corresponds to the Y axes while the second to the X axis'; '- if you have ✓
seen some complex numbers and you want to check the range select the variables with ✓
complex values first (Y axis).'};

% Create UIAxes2
app.UIAxes2 = uiaxes(app.GENERALPLOTSTab);
title(app.UIAxes2, 'Title')
xlabel(app.UIAxes2, 'X')
ylabel(app.UIAxes2, 'Y')
app.UIAxes2.Position = [62 107 300 185];

% Create ErrorXYZPanel
app.ErrorXYZPanel = uipanel(app.GENERALPLOTSTab);
app.ErrorXYZPanel.TitlePosition = 'centertop';
app.ErrorXYZPanel.Title = 'Error X, Y, Z';
app.ErrorXYZPanel.BackgroundColor = [0.749 0.9294 0.651];
app.ErrorXYZPanel.FontWeight = 'bold';

```

```
app.ErrorXYZPanel.FontSize = 14;
app.ErrorXYZPanel.Position = [3 1 875 315];

% Create X_error_plot
app.X_error_plot = uiaxes(app.ErrorXYZPanel);
title(app.X_error_plot, 'X error plot')
xlabel(app.X_error_plot, 'N° TEST')
ylabel(app.X_error_plot, 'X')
app.X_error_plot.DataAspectRatio = [1 1 1];
app.X_error_plot.PlotBoxAspectRatio = [1 1 1];
app.X_error_plot.FontSize = 13;
app.X_error_plot.FontWeight = 'bold';
app.X_error_plot.XMinorTick = 'on';
app.X_error_plot.YMinorTick = 'on';
app.X_error_plot.NextPlot = 'add';
app.X_error_plot.XMinorGrid = 'on';
app.X_error_plot.YMinorGrid = 'on';
app.X_error_plot.BackgroundColor = [0.749 0.9294 0.651];
app.X_error_plot.Position = [1 1 286 290];

% Create Y_error_plot
app.Y_error_plot = uiaxes(app.ErrorXYZPanel);
title(app.Y_error_plot, 'Y error plot')
xlabel(app.Y_error_plot, 'N° TEST')
ylabel(app.Y_error_plot, 'Y')
app.Y_error_plot.DataAspectRatio = [1 1 1];
app.Y_error_plot.PlotBoxAspectRatio = [1 1 1];
app.Y_error_plot.FontSize = 13;
app.Y_error_plot.FontWeight = 'bold';
app.Y_error_plot.XMinorTick = 'on';
app.Y_error_plot.YMinorTick = 'on';
app.Y_error_plot.NextPlot = 'add';
app.Y_error_plot.XMinorGrid = 'on';
app.Y_error_plot.YMinorGrid = 'on';
app.Y_error_plot.BackgroundColor = [0.749 0.9294 0.651];
app.Y_error_plot.Position = [292 1 288 290];

% Create Z_error_plot
app.Z_error_plot = uiaxes(app.ErrorXYZPanel);
title(app.Z_error_plot, 'Z error plot')
xlabel(app.Z_error_plot, 'N° TEST')
ylabel(app.Z_error_plot, 'Z')
app.Z_error_plot.DataAspectRatio = [1 1 1];
app.Z_error_plot.PlotBoxAspectRatio = [1 1 1];
app.Z_error_plot.FontSize = 13;
app.Z_error_plot.FontWeight = 'bold';
app.Z_error_plot.XMinorTick = 'on';
app.Z_error_plot.YMinorTick = 'on';
app.Z_error_plot.NextPlot = 'add';
app.Z_error_plot.XMinorGrid = 'on';
app.Z_error_plot.YMinorGrid = 'on';
app.Z_error_plot.BackgroundColor = [0.749 0.9294 0.651];
app.Z_error_plot.Position = [588 1 283 290];

% Create RELEVANTPLOTTab
```

```
app.RELEVANTPLOTTab = uitab(app.TabGroup2);
app.RELEVANTPLOTTab.Title = 'RELEVANT PLOT';
app.RELEVANTPLOTTab.BackgroundColor = [0.9412 0.9412 0.4902];

% Create RELEVANT_VALUES_PLOT
app.RELEVANT_VALUES_PLOT = uiaxes(app.RELEVANTPLOTTab);
title(app.RELEVANT_VALUES_PLOT, 'RELEVANT VALUES PLOT')
xlabel(app.RELEVANT_VALUES_PLOT, 'X')
ylabel(app.RELEVANT_VALUES_PLOT, 'Y')
app.RELEVANT_VALUES_PLOT.FontWeight = 'bold';
app.RELEVANT_VALUES_PLOT.XMinorTick = 'on';
app.RELEVANT_VALUES_PLOT.YMinorTick = 'on';
app.RELEVANT_VALUES_PLOT.XGrid = 'on';
app.RELEVANT_VALUES_PLOT.XMinorGrid = 'on';
app.RELEVANT_VALUES_PLOT.YGrid = 'on';
app.RELEVANT_VALUES_PLOT.YMinorGrid = 'on';
app.RELEVANT_VALUES_PLOT.Position = [10 9 737 757];

% Create Y_GraphButtonGroup
app.Y_GraphButtonGroup = uibuttongroup(app.RELEVANTPLOTTab);
app.Y_GraphButtonGroup.TitlePosition = 'centertop';
app.Y_GraphButtonGroup.Title = 'Y_Graph';
app.Y_GraphButtonGroup.FontWeight = 'bold';
app.Y_GraphButtonGroup.Position = [754 471 123 289];

% Create Nom_XButton
app.Nom_XButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Nom_XButton.Text = 'Nom_X';
app.Nom_XButton.Position = [11 243 62 22];
app.Nom_XButton.Value = true;

% Create Nom_YButton
app.Nom_YButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Nom_YButton.Text = 'Nom_Y';
app.Nom_YButton.Position = [11 221 65 22];

% Create Nom_ZButton
app.Nom_ZButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Nom_ZButton.Text = 'Nom_Z';
app.Nom_ZButton.Position = [11 199 65 22];

% Create Red_XButton
app.Red_XButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Red_XButton.Text = 'Red_X';
app.Red_XButton.Position = [11 178 66 22];

% Create Red_YButton
app.Red_YButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Red_YButton.Text = 'Red_Y';
app.Red_YButton.Position = [11 157 67 22];

% Create Red_ZButton
app.Red_ZButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Red_ZButton.Text = 'Red_Z';
app.Red_ZButton.Position = [11 135 67 22];
```

```

% Create Delta_Nom_XButton
app.Delta_Nom_XButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Delta_Nom_XButton.Text = 'Delta_Nom_X';
app.Delta_Nom_XButton.Position = [11 114 97 22];

% Create Delta_Nom_YButton
app.Delta_Nom_YButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Delta_Nom_YButton.Text = 'Delta_Nom_Y';
app.Delta_Nom_YButton.Position = [11 91 97 22];

% Create Delta_Nom_ZButton
app.Delta_Nom_ZButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Delta_Nom_ZButton.Text = 'Delta_Nom_Z';
app.Delta_Nom_ZButton.Position = [11 70 96 22];

% Create Delta_Red_XButton
app.Delta_Red_XButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Delta_Red_XButton.Text = 'Delta_Red_X';
app.Delta_Red_XButton.Position = [11 49 93 22];

% Create Delta_Red_YButton
app.Delta_Red_YButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Delta_Red_YButton.Text = 'Delta_Red_Y';
app.Delta_Red_YButton.Position = [11 28 93 22];

% Create Delta_Red_ZButton
app.Delta_Red_ZButton = uiradiobutton(app.Y_GraphButtonGroup);
app.Delta_Red_ZButton.Text = 'Delta_Red_Z';
app.Delta_Red_ZButton.Position = [11 7 93 22];

% Create PLOTButton
app.PLOTButton = uibutton(app.RELEVANTPLOTTab, 'push');
app.PLOTButton.ButtonPushedFcn = createCallbackFcn(app, @PLOTButtonPushed,
true);

app.PLOTButton.BackgroundColor = [0.6824 1 0.4706];
app.PLOTButton.FontSize = 16;
app.PLOTButton.FontWeight = 'bold';
app.PLOTButton.Enable = 'off';
app.PLOTButton.Position = [754 353 123 44];
app.PLOTButton.Text = 'PLOT';

% Create PathCSVEditField_Relevant
app.PathCSVEditField_Relevant = uieditfield(app.RELEVANTPLOTTab, 'text');
app.PathCSVEditField_Relevant.Editable = 'off';
app.PathCSVEditField_Relevant.Position = [752 407 123 22];
app.PathCSVEditField_Relevant.Value = 'NaN';

% Create SelectpathCSVButton_2
app.SelectpathCSVButton_2 = uibutton(app.RELEVANTPLOTTab, 'push');
app.SelectpathCSVButton_2.ButtonPushedFcn = createCallbackFcn(app,
@SelectpathCSVButton_2Pushed, true);
app.SelectpathCSVButton_2.BackgroundColor = [0.6745 0.8039 0.8902];
app.SelectpathCSVButton_2.FontWeight = 'bold';
app.SelectpathCSVButton_2.Position = [754 441 121 22];

```

```
        app.SelectpathCSVButton_2.Text = 'Select path CSV';

        % Show the figure after all components are created
        app.SIMULATIONPROBA3UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = SIMULATION_PROBA_3_exported

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.SIMULATIONPROBA3UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.SIMULATIONPROBA3UIFigure)
    end
end
end
```