

**MEGARA Control System. User Manual.**

**Fiber MOS Positioning Tool**

**GTC Code:**

**Code:** TEC/MEG/171

**Issue:** 1.F

**Date:** 22/06/2017

**Pages: 29**

|  |  |
| --- | --- |
| **Authors:** | Isaac Morales Durán |
| **Revised by:** | África Castillo Morales  Ana Pérez Calpena  Armando Gil de Paz  Jorge Iglesias Páramo |
| **Approved by:** | Armando Gil de Paz  Maria Luisa García Vargas |
|  |  |

**Distribution List:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Affiliation** | **Date** |
| GRANTECAN | GRANTECAN | 22/06/2017 |
| MEGARA Team | All Partners | 22/06/2017 |
|  |  |  |

**Acronyms:**

CB Configuration Block

CBS Configuration Block Set

DDS Disjoint Disperse Subsets

EA Exclusion Area

FMPT Fiber MOS Positioning Tool

FMAT Fiber MOS Assignment Tool

FMOSA Fiber MOS Assignment file

GFCCS GTC Folded-Cassegrain Coordinate System

MCS MEGARA Control System

PPA Pair of Positions Angles

PP Positioning Program

DP Depositioning Program

SP Sky Point

SPM Security Perimetral Margin

S/W Software

TBD To Be Defined**Change Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue** | **Date** | **Section** | **Page** | **Change description** |
| 1.A | 26/01/15 | All | All | New Issue |
| 1.B | 23/06/15 | 3  4  5  6 | 9  9  11  14 | Description on how to get the help of the tool changed.  Description of the input file for the FMPT rewritten.  Description of the output files from the FMPT rewritten.  Description of the new functionalities of the FMPT rewritten. |
| 1.C | 21/02/16 | -  All  1  All  2  2  2  3  4  5  5.1  5.2  5.2  5.3  6.1  6.2  6.3 | 7-8  All  9  All  10  10  10  10  11-12  13  13  13  14-15  15  16  16-17  17 | Index updated.  Corrected use of abbreviation S/W.  Implemented functionality for generate parking program.  Version number updated to 3.0.7.  Execute ‘ldconfig’ at the end of installation.  Installation tested in Ubuntu 14.04.4  Application functioning in Solaris 10.  Output command help updated.  Corrected description of format file FMOSA.  Modified example of FMOSA file.  Modified description of content of log file.  PP and DP are stored in a single file jointly the content of FMOSA input file.  Modified description of format of the output file.  Modified description of file  ‘outputs-from-<inputfile>’.  Generating a pair (PP, DP) is generated a file called ‘PPDPandFMOSA-from-<inputfile>’.  Motion program for recovery now is called parking program instead depositioning program.  Furthermore has been described the syntax of the function for generate pairs (PP, DP).  Described the syntax of the function for generate parking programs. |
| 1.D | 09/08/16 | All  3  4  5.1  5.3  6.1  7.1.2  7.1.3  7.6  7.7  7.8  7.9  7.10  7.11  7.12  7.13 | All  10-11  11  13-14  17  18  20  21  23  24  24  24  24  24  25  25 | Strings changed:  “<inputfile>” 🡪 “<inputfilename>”  “<inputfile>.log” 🡪 “fmpt\_saa.log”  “PPDPandFMOSA” 🡪 “outputs”  “outputs” 🡪”other\_outputs”  “FMOSA file” 🡪 “file type FMOSA”  “3.0.6” 🡪 “3.3.4”  “3.0.7” 🡪 “3.3.4”  Updated output of command $ fmpt\_saa help  Added phrase:  “These parameters are ignored by the FMPT.”  Updated content of log file ‘fmpt\_saa.log’.  Updated content of file ‘other\_outputs...txt’.  Changed strings:  generatePPDP 🡪 generatePairPPDP\_offline  <path> 🡪 <path\_FMOSA>  Replaced figure due to error:  Figure 1 (error in labels P0 and P1).  Figure 2 (error in label P3).  Updated content of files of the FMM instance:  Instance.txt  ExclusionArea<Id>/Contour\_.txt  ExclusionArea<Id>/Instance.txt  RoboticPositioner<Id>/Contour\_\_\_\_.txt  RoboticPositioner<Id>/Contour\_.txt  RoboticPositioner<Id>/F1.txt  RoboticPositioner<Id>/F2.txt  RoboticPositioner<Id>/Instance.txt |
| 1.E | 16/01/16 | All | All | The document has been updated to reply the comments in LAR-INT-007 to LAR-INT-011 |
| 1.F | 22/06/17 | 4  4  6.2  6.3  7.11  7.13  All  7.6-7.13 | 13  15  21  22  26  27  All  27-29 | Added field “Angle (deg)” to file type FMOSA.  Example type FMOSA updated.  Function “generateParkProg\_online” return structure outputs instead ParkProg.  Function “generatePPDP\_online” return structure outputs instead PP and DP.  Redefined F1 for SB1 = 194086.302.  Removed SB1 and SB2 from ‘Instance.txt’.  String changed: “S2” 🡪 “S0”.  String changed: “3.3.4” 🡪 “4.8.0”.  Updated content of files of the FMM instance. |

**Reference Documents**

|  |  |  |
| --- | --- | --- |
| **Nº** | **Document Name** | **Code** |
| R.1 | MEGARA CS. Fiber MOS Assignment Tool (FMAT). User Manual | TEC/MEG/174 |
| R.2 | FMPT/FMAT tools compatibility report. | TEC/MEG/184 |
|  |  |  |
|  |  |  |
|  |  |  |

**Reference Documents (GTC codes)**

|  |  |  |
| --- | --- | --- |
| **Nº** | **Document Name** | **Code** |
| R.1 | MEGARA CS. Fiber MOS Assignment Tool (FMAT). User Manual | N/A |
| R.2 | FMPT/FMAT tools compatibility report. | N/A |
|  |  |  |
|  |  |  |
|  |  |  |

**INDEX**

1. Introduction 10

2. How To install the FMPT 10

3. Getting help running the FMPT SAA 11

4. Inputs of the FMPT SAA 12

5. OUTputs of the FMPT SAA 16

5.1 Log file 16

5.2 File containing the pair (PP, DP) 16

5.3 File containing other output data 18

6. EXECUTION OF THE FMPT 19

6.1 Generating a pair (PP, DP) 19

6.2 Generating a parking program 21

6.3 Regenerate a pair (PP, DP) 22

6.4 Getting (p\_1s, p\_\_\_3s) from (Ids, Xs, Ys) 23

7. APPENDIX: instance of the Fiber MOS Model 24

7.1 FMPT coordinate systems 24

7.1.1 FMPT S0 coordinate system 24

7.1.2 FMPT S1 coordinate system 25

7.1.3 FMPT S2 coordinate system 25

7.1.4 FMPT S3 coordinate system 26

7.2 Set of files composing the instance of the Fiber MOS Model 26

7.3 Fiber MOS Model - RP Origins Table (RoboticPositionerOriginsTable.txt) 27

7.4 Fiber MOS Model - EA Origins Table (ExclusionAreaOriginsTable.txt) 27

7.5 Fiber MOS Model - RP Map (InstanceMap.txt) 27

7.6 Fiber MOS Model - Instance properties (Instance.txt) 28

7.7 EA - Barrier Contour\_ (ExclusionArea<Id>/Countour\_.txt) 28

7.8 EA - Instance properties (ExclusionArea<Id>/Instance.txt) 28

7.9 RP - Arm Contour\_\_\_\_ (RoboticPositioner<Id>/Contour\_\_\_\_.txt) 28

7.10 RP - Barrier Contour\_ (RoboticPositioner<Id>/Contour\_.txt) 28

7.11 RP – Rotor 1 compression function (RoboticPositioner<Id>/F1.txt) 29

7.12 RP – Rotor 2 compression function (RoboticPositioner<Id>/F2.txt) 29

7.13 RP - Instance properties (RoboticPositioner<Id>/Instance.txt) 30

# Introduction

This document contains the User Manual of the MEGARA Fiber MOS Positioning Tool (FMPT). The manual allows the user to install the FMPT in his computer and to understand the FMPT functionality.

The S/W to configure the Fiber MOS has been split in two tools: Fiber MOS Assignment Tool (FMAT) [R.1] and Fiber MOS Positioning Tool (FMPT). The FMAT allows the user to prepare a Configuration Block Set (CBS) composed by several Configuration Blocks for the MEGARA MOS mode by assisting the user in the assignment of their sources to the different in-use robotic positioners of the MEGARA MOS (up to 92 positioners). Each file containing the information of a particular Configuration Block will be the input for the Fiber MOS Positioning Tool. This tool computes the sequence of movements that will be commanded before starting the exposure for reaching the source positions (positioning program, PP) as well as the sequence of movements that will be commanded after the exposure is finished for coming back to the parking position (depositioning program, DP). The Fiber MOS Positioning Tool also allows to compute a depositioning program to come back from an anomalous situation (during the observation) receiving, in this case, as input the current positions of the robotic positioners that are not at the parking position.

This manual describes only the installation and use of the FMPT, which is composed by the FMPT library, containing all necessary functionalities, and the application FMPT SAA, which uses the dynamic library ‘libfmpt.so’. The functionalities of the FMPT will be the following:

1. The GRANTECAN staff will use the FMPT as a standalone tool to generate pairs (PP, DP) for the Configuration Blocks of a CBS generated and previously validated by the FMAT. The FMPT also includes the validation of the source assignments to ensure there is no risk of collisions between the RPs.
2. MEGARA Control System (MCS) will make use of some FMPT functionalities to:

* Generate a depositioning program to come back from an anomalous situation (during the observation) knowing the disabled RPs and the positions of all RPs.
* Regenerate the pair (PP, DP) from the information about the positions of the disabled RPs. The rest of the RPs are in parking positions after executing the depositioning program previously computed.

# How To install the FMPT

The process to install this S/W can be found in the file INSTALL. The process is summarized below:

1. Decompress tar file 'megara-fmpt-4.8.0.tar.gz' (or .xz).
2. Change to directory './megara-fmpt-4.8.0'.
3. Execute './configure; make;' to configure and build this package.
4. Grant superuser privileges to have access to '/usr/local/...'.
5. Execute 'make install' (as superuser) to install the library and the application.
6. Execute 'ldconfig' (as superuser) to update the linkage with the libraries in case of Linux systems.

Both dynamic and static libraries will be installed in the usual directories of the operative system, according to the user distribution and version.

As an example, in Ubuntu 14.04.4 and Fedora 20:

* Static and dynamic libraries will be installed in '/usr/local/lib';
* The executable file fmpt\_saa will be installed in '/usr/local/bin';
* The instance of the Fiber MOS Model will be installed in:

'/usr/local/share/megara-fmpt/Models/MEGARA\_FiberMOSModel\_Instance';

* The sample input files will be installed in: 'usr/local/bin/share/megara-fmpt/Samples'.

These paths can vary depending on the operating system and distribution where this package is installed. In this document the mentioned paths assume that the installation has been done in Ubuntu 14.04.4.

The current version of the FMPT allows installation in Linux systems.

# Getting help running the FMPT SAA

The user can obtain help about the arguments of the FMPT SAA by typing:

$ fmpt\_saa help

And the terminal shows:

$ fmpt\_saa help

Print this help.

$ fmpt\_saa help\_generatePairPPDP\_offline

Print the help for function generatePairPPDP\_offline.

$ fmpt\_saa help\_generateParkProg\_offline

Print the help for function generateParkProg\_offline.

$ fmpt\_saa help\_others

Print the help about other commands of the FMPT.

$ fmpt\_saa aboutOf

Print the legend about of...

$ fmpt\_saa generateParkProg\_offline <FMOSA\_path>

Generate a valid parking program offline.

The generated parking program will be saved, even if there are either collided or obstructed RPs.

Note that motion programs generated with this function (without the argument outputs),

is guaranteed to be valid, because they pass a validation process considered unerring.

<FMOSA\_path>: absolute or relative path to the input file type FMOSA.

This command generates the files:

ParkProg\_outputs\_from\_<filename>: the parking program in format MCS,

comments about the collided RPs, if any,

comments about the obstructed RPs, if any,

comments about the validity of the generated MP,

and comments about the RPs whose radial motion is more close to 1 mm.

Where <filename> is the name of the input file.

$ fmpt\_saa generatePairPPDP\_offline <FMOSA\_path>

Generate a valid pair (PP, DP) offline.

If there are either collided or obstructed RPs, the generated pair (PP, DP) will not be saved.

Note that motion programs generated with this function (without the argument outputs),

is guaranteed to be valid, because they pass a validation process considered unerring.

<FMOSA\_path>: absolute or relative path to the input file type FMOSA.

This command generates the file:

PairPPDP\_outputs\_from\_<filename>: the pair (PP, DP) in format MCS,

comments about the collided RPs, if any,

comments about the obstructed RPs, if any,

comments about the validity of the generated MPs,

and comments about the RPs whose radial motion is more close to 1 mm.

the content of the input file type FMOSA,

Where <filename> is the name of the input file.

# Inputs of the FMPT SAA

In the following sections we describe in detail the input and output files of the FMPT SAA. In the Appendix of this document, the files that constitute the instance of the Fiber MOS Model, which is the major defining parameter for the program, are described.

For each Configuration Block the user needs to execute the FMPT to generate the positioning and depositioning programs (PP and DP). The input file for the FMPT is the Fiber MOS Assignment file (FMOSA), which should be loaded by the astronomer during the Phase-2. This input file is comprised by different sections: the header, the configuration block (CB) and the source list (OS).

The header contains the following information:

* Version of the FMAT used
* Generation date
* A list of errors, if any[[1]](#footnote-1)

The file starts with a comment row (ignored by the S/W) with the column description of the CB:

# Id| Ra| Dec| Pos

These field names correspond to the Identification number (Id) for the Block, Right Ascension and Declination where the center of the MEGARA MOS FOV (which coincides with the FC optical axis) is pointing along the FC rotator Position Angle (between 0º and 360º). All the angles are in degrees with six decimal positions. These parameters are ignored by the FMPT.

A row indicating that the information about the configuration block starts:

@@SOB@@

A line representing a CB, for example:

0| 15.01564| 45.004671| 0.050512

A row indicating that the information about the CB ends:

@@EOB@@

A comment row (ignored by the S/W) with the brief indication of the meaning of the different columns for each source included in this input file:

# Name Ra Dec Mag Type Pr Bid Pid X(mm) Y(mm) Enabled Comment

These correspond to the Source Name, Right Ascension and Declination of the source, Magnitude, Type of source, Priority, Block Id, Positioner Id, X, Y position of the positioner and whether the positioner is enabled/disabled and a Comment field. All the angles are in degrees with six decimal positions.

A row indicating that the list of sources starts.

@@SOS@@

A number of lines, each one representing a source, for example:

s:116|15.011244|45.022607|15.04|SOURCE|1|0|1|-9.171006|53.226932|0|1|foo comment

s:596|14.986557|45.032844|18.66|SOURCE|0|0|2|-60.909235|83.658767|0|1|foo comment

…

Note that both sources are assigned to the block 0 to positioners 1 and 2 and with priority 1 and 0 respectively.

A row indicating that the list of sources ends.

@@EOS@@

For each source, in each row, we can find 12 fields separated by vertical bars corresponding to parameters of a given Sky Point (SP). The fields are the following ones:

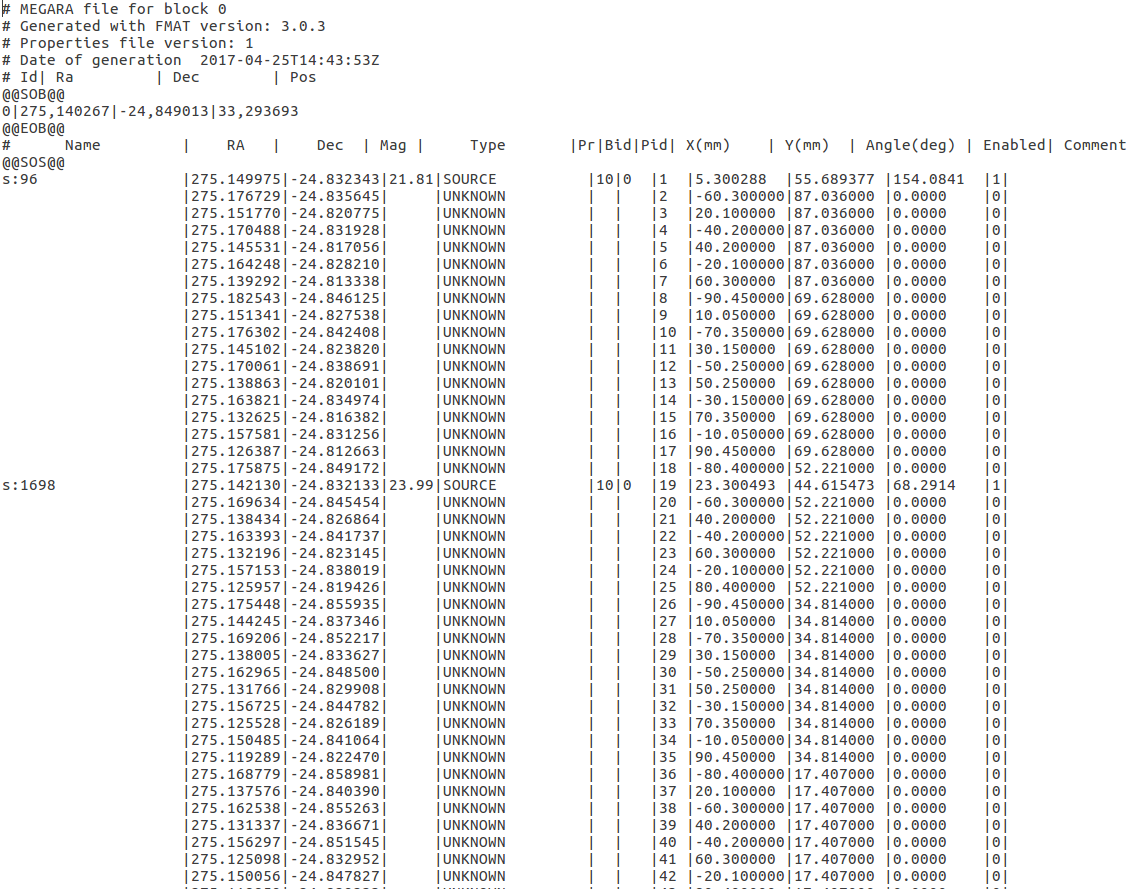
1. **Name**: Name of the source (catalog name, e.g. Mk348, NGC4151, etc.). Can be empty for unallocated sources.
2. **Ra:** Right Ascension in J2000.0. The format is degrees with six decimal positions.
3. **Dec:** Declination in J2000.0. The format is degrees with six decimal positions.
4. **Mag:** Magnitude. In this release this field interprets that magnitudes are all in the same band and photometrical system. Can be empty for unallocated sources.
5. **Type:** There are 4 types of SPs in the input preselected-sky-point list for the FMAT:

* **SOURCE**: target of scientific interest. It could be allocated or not to a RP.
* **REFERENCE**: target to be used as reference. It could have scientific interest or not. Reference-source SP corresponds to a source (star) that can be used to validate the correct position of the telescope and PA. It is generally recommended that every observation (at least for long exposures) contains, at the very least, three reference-source SPs.
* **BLANK**: sky background source. A blank SP is a point in a blank-sky region of the field. It is generally recommended one blank SP at least in every observation with the Fiber MOS. The blank SP observation will be used in the DFP/DRP and quick-look tools for sky subtraction.
* **UNKNOWN**: unknown type point (type has not been previously identified). This is the type for unallocated sources.

1. **Pr:** level of priority assigned to each source. The highest priority is “0” and the lowest priority is “10”. Can be empty for unallocated sources.
2. **Bid:** Block Id. **Will be empty for unallocated sources.**
3. **Pid:** Positioner Id. Can be empty for unallocated sources.
4. **X (mm):** Positioner x co-ordinate (mm with six decimals) in the Fiber MOS co-ordinate system.
5. **Y (mm):** Positioner y co-ordinate (mm with six decimals) in the Fiber MOS co-ordinate system.
6. **Angle (deg):** Position angle of the microlens in sexagesimal degrees.
7. **Enabled:** Booleanfield to indicate if the RP identified by the Pid is enabled (1 value) or disabled (0 value).
8. **Comment:** Field for comment.

The FMPT reads from this input file the values of the properties (Pid, X, Y) for each assigned RP and also the Priority and Enabled columns information. These properties represent a Projection Point allocated to the RP whose identifier matches with this particular Id.

Example of an input file for the FMPT:



# OUTputs of the FMPT SAA

After generating a pair (PP, DP) the output files from the FMPT, located at the working directory, would be the following:

## Log file

This file, named fmpt\_saa**.log**, contains the same output written in the terminal when the application is executed:

user@lenovo-W520:~$ fmpt\_saa generatePairPPDP\_offline /usr/local/share/megara-fmpt/Samples/megara-cb0.txt

FMPT SAA 4.8.0 is running…

Arguments with has called the program:

argv[0]: fmpt\_saa

argv[1]: generatePairPPDP\_offline

argv[2]: /usr/local/share/megara-fmpt/Samples/megara-cb0.txt

Loading FMM instance from: '/usr/local/share/megara-fmpt/Models/MEGARA\_FiberMOSModel\_Instance'

'/home/user/../data/Models/MEGARA\_FiberMOSModel\_Instance'

Fiber MOS Model instance loaded from '/usr/local/share/megara-fmpt/Models/MEGARA\_FiberMOSModel\_Instance'.

FMOSA table loaded from '/usr/local/share/megara-fmpt/Samples/megara-cb0.txt'.

Allocations got from the FMOSA table in MPG.

RPs moved to observing positions.

Generating pair (PP, DP)...

Generated pair (PP, DP) is valid.

Pair (PP, DP) saved in './outputs-from-megara-cb0.txt'.

Other outputs saved in './other\_outputs-from-megara-cb0.txt'.

## File containing the pair (PP, DP)

This file, named **PairPPDP\_outputs\_from\_<inputfilename>**, contains three blocks of data (the positioning program (PP), the depositioning program (DP) and by the end of the file the same content of the input FMOSA file). In this way all the information about the motion sequences (PP and DP) together with the input source assignments are located in only one file for each Configuration Block.

Each motion program is composed by a list of groups, where each group is a list of sentences representing each an instruction to be sent to the controller of a RP rotor, before starting a jointly movement.

The general description of this file is the following:

|  |
| --- |
| # Positioning program  @@SPP@@  pos\_<cb\_string\_id> {  group\_01 {  rp01 r1 <p\_1>  rp01 r2 <p\_\_\_3>  rp17 r1 <p\_1>  rp17 r2 <p\_\_\_3>  ...  }  group\_02 {  rp03 r1 <p\_1>  rp03 r2 <p\_\_\_3>  rp22 r1 <p\_1>  rp22 r2 <p\_\_\_3>  ...  }  ...  }  @@EPP@  # Depositioning program  @@SDP@@  depos\_<cb\_string\_id> {  group\_01 {  rp01 r1 <p\_1>  rp01 r2 <p\_\_\_3>  rp17 r1 <p\_1>  rp17 r2 <p\_\_\_3>  ...  }  group\_02 {  rp03 r1 <p\_1>  rp03 r2 <p\_\_\_3>  rp22 r1 <p\_1>  rp22 r2 <p\_\_\_3>  ...  }  ...  }  @@EPP@  # Id| Ra| Dec| Pos  @@SOB@@  0| 15.027879| 45.000422| 0.060858  @@EOB@@  # Name RA Dec Mag Type Pr Bid Pid X(mm) Y(mm) Enabled Comment  @@SOS@@  s:751 |15.027787|45.017925|20.67|SOURCE |6 |0 |1 |-0.137909 |51.932865 |1|foo comment  s:404 |15.003376|45.028110|21.74|SOURCE |0 |0 |2 |-51.296480|82.217116 |1|foo comment  …  |15.066176|45.029778| |UNKNOWN | | |97 |80.400000 |87.035553 |1|  @@EOS@@ |

Where the <cb\_string\_id> is the identifier of the CB for which a motion program (the PP or the DP) has been computed.

Each motion sentence must be written in two lines (corresponding to the two rotors of a RP), and has the following format:

rp<Id> r1 <p\_1>

rp<Id> r2 <p\_\_\_3>

where:

* <Id> represents the Id of the RP, completed with zeros on the left to get the length of two figures.
* <p\_1>: is the position (in steps) where the rotor 1 of the RP is going to be moved.
* <p\_\_\_3>: is the position (in steps) where the rotor 2 of the RP is going to be moved.

We note that (<p\_1>, <p\_\_\_3>) are the positions where the rotors shall be moved.

Therefore, in the case of computing a depositioning program, the final instructions for each rotor of each RP will be:

rpid r1 0

rpid r2 0

In the case of a positioning program, the final instructions for each rotor of each RP are not necessarily (0, 0) since the final positions must be the ones corresponding to the coordinates of the corresponding source.

## File containing other output data

This file, named **PairPPDP\_other-outputs\_from\_<inputfilename>**, is intended to contain simple data generated by the application.

Example: ‘other\_outputs-from-megara-cb0.txt’

|  |
| --- |
| # Pair (PP, DP) outputs generated from FMOSA megara\_2p0e5\_000003.txt  # Generated with FMPT version 4.8.0  # Date of generation: 2017-05-31T09:04:50  # This pair (PP, DP) has been generated without enabled-not-operative RPs.  # WARNING! the FMOSA contains collided items (either EAs or RPs):  # Collided items: {{RP19, RP27}, {RP20, RP28}, {RP48, RP54}, {RP69, RP92}}  # Collided RP ids: {19, 20, 27, 28, 48, 54, 69, 92}  # The initial positions of the collided or obstructed RPs match with their observing positions.  # A pair (PP, DP) is suitable to be executed when it is valid (avoid collisions) and there aren't  # dangerous RPs in the Fiber MOS (enabled-not-operative RPs with fault type dynamic or unknowledge),  # and there aren't neither collided nor obstructed RPs.  # According the actual status of the FMPT, this pair (PP, DP) is not suitable to be executed.  # Positioning program  # ERROR! This PP produces a collision when it is executed starting from the initial positions.  # All radial movements of this PP keep a security distance upper or equal to 1 mm.  # Depositioning program  # This DP avoids collisions when it is executed starting from the observing positions.  # All radial movements of this DP keep a security distance upper or equal to 1 mm.  PPvalid: False  DPvalid: True  Collided: {19, 20, 27, 28, 48, 54, 69, 92}  Obstructed: {}  Collided (including EAs): {{RP19, RP27}, {RP20, RP28}, {RP48, RP54}, {RP69, RP92}} |

Where:

* “PPvalid”: indicates if the positioning program has been successfully generated or conversely the (PP, DP) could not be obtained.
* “DPvalid”: indicates if the depositioning program has been successfully generated or conversely the (PP, DP) could not be obtained.
* “Collided”: Two RPs are considered as "collided" when the minimum distance between them is lower than the sum of the Security Perimetral Margins (SPMs) of the individual RPs (see Appendix). In the case that all RPs are enabled there will be no collided RPs and this field will contain an empty list. But in the case that some RPs are blocked in insecurity positions there could be some collided RPs and this field will contain the list of those RPs.
* “Obstructed”: A RP is obstructed when there is one or more adjacent disabled RPs blocked in insecurity positions that prevent the retraction of this RP to a security position. In the case that all the RPs are enabled there will be no obstructed RPs and this field will contain an empty list. But in the case that some RPs are blocked in insecurity positions there could be some obstructed RPs and this field will contain the list of those RPs.

# EXECUTION OF THE FMPT

## Generating a pair (PP, DP)

The GRANTECAN staff will make use the FMPT as a stand-alone tool to generate pairs (PP, DP) for the Configuration Blocks of a CBS generated, and previously validated by the FMAT. Although the compatibility of both tools, FMPT and FMAT has been checked in [R.2], the FMPT also includes a validation to ensure there is no risk of collisions between the RPs. When the FMPT detects RPs with risk of collision it computes the positioning and depositioning programs without considering those RPs.

The syntax for the PP and DP generation is as follows:

$ fmpt\_saa generatePairPPDP\_offline <path\_FMOSA>

<path\_FMOSA>: is the absolute or relative path to FMOSA file.

Setting parameters:

A directory containing the Fiber MOS Model Instance in the path: /usr/local/share/megara-fmpt/Models.

This parameter file is loaded by default from ‘MEGARA\_FiberMOSModel\_Instance’.

Inputs:

File <path\_FMOSA>[[2]](#footnote-2)contains a table with the projection points and their allocations to RPs.

The following output files will be saved in the same directory where the program is executed: fmpt\_saa.log, PairPPDP\_outputs\_from\_<filename>, PairPPDP\_other-outputs\_from\_<filename>, where <filename> is the name of the input file FMOSA located in <path\_FMOSA>.

## Generating a parking program

It could be the case that during the execution of a movement program, a RP fails and some of the rest of RPs are out of their parking positions. In this situation, the failed RP and the six adjacent to this one, must be disabled (using the FiberMOS CS panel by the support astronomer) and MCS will recover the original positions of the non-failed RPs using the FMPT functionality to generate a parking program as described in this section.

To generate the parking program, the FMPT would need as input the list of the disabled RPs and the list of the current positions of all the RPs to compute the parking program. This information will be provided by MCS as arguments when this function is called. The syntax to execute this functionality in MCS is the following:

bool generateParkingProgram\_online(OutputsParkProg& outputs,

TFiberMOSModel& FMM,

const vector<double>& p\_1s, const vector<double>& p\_\_\_3s,

const vector<int>& Ids);

Where:

* outputs: the structure containing the parking program to be generated, and all information for their execution.
* FMM: is the Fiber MOS Model.
* (p\_1s, p\_\_\_3s): are the starting positions (in steps), of all RPs in the FMM.
* Ids: is the list of identifiers of the RPs to be disabled.

This function is internally documented in the file ‘MotionProgramGenerator.h’ (provided with the release 4.8.0 of the FMPT).

## Regenerate a pair (PP, DP)

When the pair (PP, DP) of a particular CB is going to be executed by MCS, a malfunction or anomaly related with the rotors of the RPs could happen. This could be the case before or during the execution of the RP movement programs.

It is recommendable to execute the FMPT (to compute positioning and depositioning programs) just before the corresponding observing run. This is to avoid situations where there are sources assigned to RPs which are temporarily non-available and were considered as available at the moment of creating the PP and DP programs.

It could be the case that during the execution of a movement program, a RP fails and some of the rest of RPs are out of their parking positions. In this situation, the failed RP is disabled and MCS will recover the original positions of the non-failed RPs using the FMPT functionality to generate a parking program as described in last section.

As the RP that has failed could be disabled in an insecure position the pair PP and DP computed for that configuration block is not valid because its execution could produce collisions. To follow with the observing run, MCS would call the FMPT functionality to regenerate the pair (PP, DP), calling the function whose syntax is briefly described below:

bool generatePairPPDP\_online(OutputsPairPPDP& outputs,

TFiberMOSModel& FMM,

const vector<double>& p\_1s, const vector<double>& p\_\_\_3s,

const vector<int>& Ids);

Where:

* outputs: the estructure containing the pair to be generated, and all information for their execution.
* FMM: the Fiber MOS Model.
* (p\_1s, p\_\_\_3s): are the initial positions in steps, of all RPs in the FMM.
* Ids: is the list of identifiers of the RPs to be disabled.

This function is similar to the function to generate the original pair, but including the parameters (p\_1s, p\_\_\_3s) and Ids.

This function is internally documented in the file ‘MotionProgramGenerator.h’ (provided with the release 4.8.0 of the FMPT).

## Getting (p\_1s, p\_\_\_3s) from (Ids, Xs, Ys)

It is possible that going to regenerate a pair (PP, DP), the final positions of the RPs to be moved, are knowledge in Cartesian coordinates. For example, you have the vectors (Ids, Xs, Ys):

vector<int> Ids; //the Ids of the enabled RPs to be moved

vector<double> Xs; //the abscissa of the points to observe

vector<double> Ys; //the ordinate of the points to observe

But regeneration’s functions hope the parameters (p\_1s, p\_\_\_3s). These vectors contain the positions angles in steps of the rotors 1 and 2 respectively. Them, assuming that the status of the FMM (Fiber MOS Model) correspond to the status of the Fiber MOS real, you can get the vectors (p\_1s, p\_\_\_3s) by the following procedure:

//get the Initial Position List from the FMM

TPairPositionAnglesList IPL;

FMM.RPL.getPositions(IPL);

//build an allocation list from (Ids, Xs, Ys)

TAllocationList AL;

for(unsigned int i=0; i<Ids.size(); i++) {

//search the RP

int j = FMM.RPL.searchId(Ids[i]);

if(j >= FMM.RPL.getCount())

throw EImproperArgument("lateral effect");

//get the RP

TRoboticPositioner \*RP = FMM.RPL[j];

//check the enabling status of the RP

if(!RP->getActuator()->Disabled)

throw EImproperArgument("lateral effect");

//add the allocation

TAllocation \*A = new TAllocation(RP, Xs[i], Y[i]);

AL.Add(A);

}

//get the Final Position List from the (IPL, AL)

TPairPositionAnglesList FPL;

AL.getFinalPositionList(FPL, IPL);

//get (p\_1s, p\_\_\_3s) from the FPL

vector<double> p\_1s, p\_\_\_3s;

for(int i=0; i<FPL.getCount(); i++) {

TPairPositionAngles PPA = FPL[i];

p\_1s.push\_back(PPA.p\_1);

p\_\_\_3s.push\_back(PPA.p\_\_\_3);

}

# APPENDIX: instance of the Fiber MOS Model

## FMPT coordinate systems

The Fiber MOS Positioning Tool (FMPT) uses internally four different coordinate systems, which are described in the following subsections.

### FMPT S0 coordinate system

The FMPT S0 coordinate system coincides with the Fiber MOS focal frame coordinate system.

The Fiber MOS focal frame coordinate system is defined as follows:

* The origin is in the center of the frame.
* The Z-axis runs in direction of the Z-axis of the GFCCS against the incoming light.
* The Y-axis runs in direction of the Y-axis of the GFCCS, positive sense of Y-axis going away from the base plate.
* The X-axis forms a right handed system with the two previous.

The Fiber MOS focal frame shall be used to define the position of the each positioner and, therefore, locate the origin of each individual Fiber MOS positioner coordinate system.

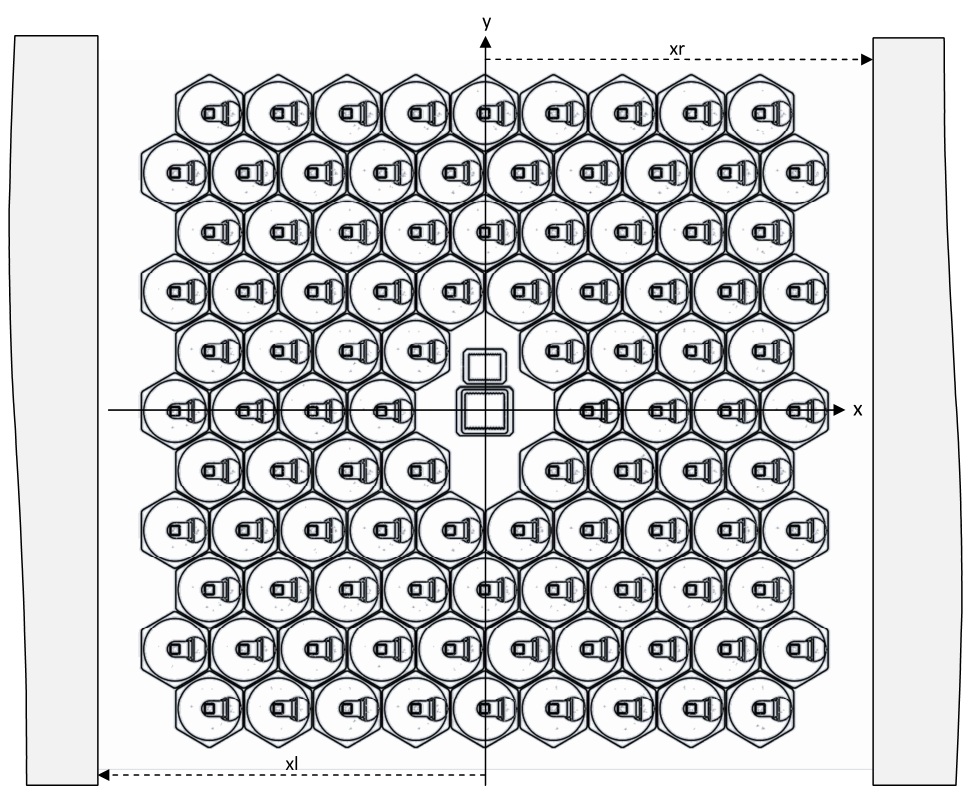


Figure 1: FMPT S0 coordinate system.

### FMPT S1 coordinate system

The FMPT S1 coordinate system is defined as follows:

* This is dextrorotatory.
* The origin of the system is at the center of rotation 1.
* The x-axis is oriented passing through rotation 1 and rotation 2 centers.

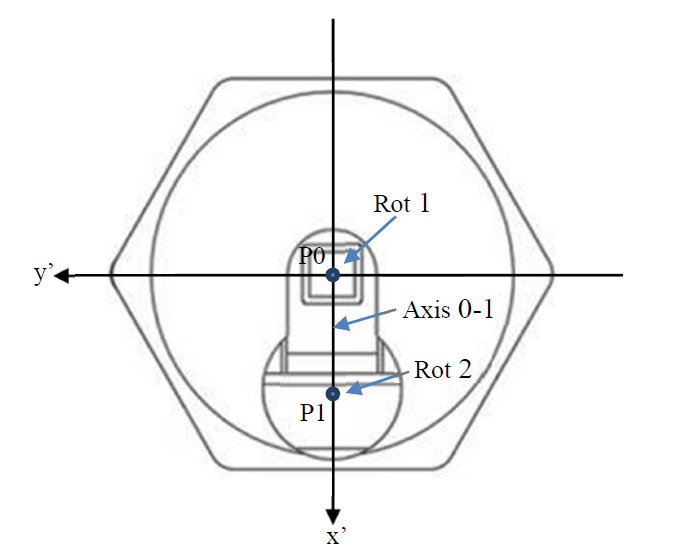


Figure 2: FMPT S1 coordinate system

### FMPT S2 coordinate system

The FMPT S2 coordinate system is defined as follows:

* This is dextrorotatory.
* The origin of the system is at the center of rotation 2.
* The x-axis is orientated in such way that it works as symmetrical axis to the arm’s contour.

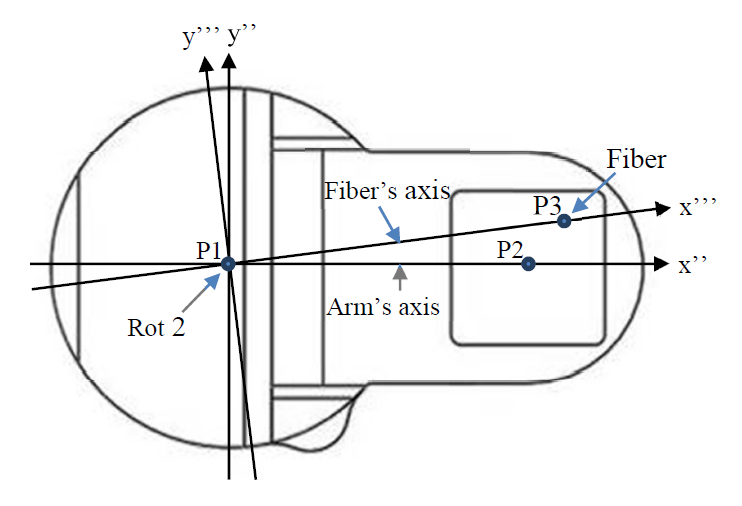


Figure 3: FMPT S2 and S3 coordinate systems

### FMPT S3 coordinate system

The FMPT S3 coordinate system is defined as follows:

* This is dextrorotatory.
* The origin of the system is at the center of rotation 2.
* The x-axis is orientated in such way that passes through rotation 1 and the center of the fiber minibundle.

## Set of files composing the instance of the Fiber MOS Model

The instance of the Fiber MOS Model is composed by a directory containing the following sets of data:

* **RP Origins Table**: containing a table to indicate the position and orientation of each RP in the Fiber MOS Model, containing the following fields: (Id, x0, y0, thetaS1).
* **EA Origins Table**: containing a table to indicate the position and orientation of each exclusion area (EA) in the Fiber MOS Model, containing the following fields: (Id, x0, y0, thetaS1).
* **RP Map**: containing a matrix to transform each point (x, y) given in Cartesian coordinates in S0, in a list of identifiers of RPs {Id} in whose observing domain could be the point (x, y).
* **Fiber MOS Additional Properties**: containing the additional properties of the Fiber MOS Model instance.
* **A subdirectory for each RP containing the following text files**:
  + **RP Arm Contour**\_\_\_\_: containing a list of segments and arcs to describe the template of the contour of the RP Arm, in S4 (in mm).
  + **RP Frontier Contour\_**: containing a list of segments and arcs to describe the template of the contour of the RP Frontier, in S1 (in mm).
  + **RP F1**: containing the compression function of the quantifier of rot 1 of the RP (in step/rad). Shall be defined in a domain that contains rot 1 domain [theta\_1min, theta\_1max].
  + **RP F2**: containing the compression function of the quantifier of rot 2 of the RP in (step/rad). Shall be defined in a domain that contains rot 2 domain [theta\_\_\_3min, theta\_\_\_3max].
  + **RP Additional Properties**: containing the additional properties of the RP Model instance. **WARNING:** this file must be updated by the GRANTECAN staff each time a RP is disabled in the real Fiber MOS. In case a RP is disabled the status property should be Disabled = True for the corresponding robotic positioner.
* **A subdirectory for each EA containing the following text files**:
  + **EA Barrier Contour\_**: containing a list of segments and arcs to describe the template of the contour of the EA Barrier, in S1 (in mm).
  + **EA Additional Properties**: containing the additional properties of the EA Model instance.

Each set of these sets of data is stored in a file. These files are stored in the directory of the instance and are root proprietary, because the user usually shall not modify the instance.

The names of these files and the default values of the data are shown below, with comments added to facilitate understanding.

## Fiber MOS Model - RP Origins Table (RoboticPositionerOriginsTable.txt)

|  |
| --- |
| # A table for indicate the position and orientation of each RP (Robotic Positioner):  # Id: identifier of the RP (a nonnegative integer number)  # x0: abscissa of the point P0 of the RP in S0 (in mm)  # y0: ordinate of the point P0 of the RP in S0 (in mm)  # thetaO1: orientation of S1 in S0 (in rad)  Id x0 y0 thetaO1  1 -0.045259375 52.2962875 0.0137000127470387  2 -60.4789828125 87.1743703125 0.00511258188666815  … … … …  100 80.4 -87.035553 0 |

## Fiber MOS Model - EA Origins Table (ExclusionAreaOriginsTable.txt)

|  |
| --- |
| # A table for indicate the position and orientation of each EA (Exclusion Area):  # Id: identifier of the EA (a nonnegative integer number)  # x0: abscissa of the point P0 of the EA in S0 (in mm)  # y0: ordinate of the point P0 of the EA in S0 (in mm)  # thetaO1: orientation of S1 in S0 (in rad)  Id x0 y0 thetaO1  1 0 0 0 |

## Fiber MOS Model - RP Map (InstanceMap.txt)

|  |
| --- |
| # A matrix to transform each point (x, y) given in Cartesian coordinates in S0,  # in a list of identifiers of RPs {Id} in whose observing domain could be the point (x, y):  xmin = -102.055 # lower limit in x-axis of the matrix of cells (in mm)  xmax = 102.055 # upper limit in x-axis of the matrix of cells (in mm)  ymin = -98.640553 # lower limit in y-axis of the matrix of cells (in mm)  ymax = 98.640553 # upper limit in y-axis of the matrix of cells (in mm)  I = 49 # number of rows of the cell matrix  J = 50 # number of columns of the cell matrix  # Cells[i][j]: list of RPs whose observing domain is in the square cell (i, j)  Cells[0][0] = {}  Cells[0][1] = {}  …  Cells[1][0] = {}  Cells[1][1] = {}  …  Cells[50][49] = {}  Cells[50][50] = {} |

## Fiber MOS Model - Instance properties (Instance.txt)

|  |
| --- |
| # Instance properties of the FMM (Fiber MOS Model):  Tolerances: # Tolerances of the Fiber MOS:  PAem = 0 # orientation error margin of the system S0 (in rad)  Pem = 0 # pointing error margin of the system S0 (in mm) |

## EA - Barrier Contour\_ (ExclusionArea<Id>/Countour\_.txt)

|  |
| --- |
| # A list of segments (Pa, Pb) and arcs (Pa, Pb, Pc, R) to describe  # the template of EA.Barrier.Contour\_, in S1 (in mm):  ({-9, -9.95}, {-14.2, -4.75}, {-9, -4.75}, 5.2)  ({-14.2, -4.75}, {-14.2, 6.721909})  ({-14.2, 6.721909}, {-13.0999999516436, 8.62716491624432}, {-12, 6.721909}, 2.2)  ({-13.0999999516436, 8.62716491624432}, {-10.6000001182939, 10.070541204891})  ({-9.7, 11.629387}, {-10.6000001182939, 10.070541204891}, {-11.5, 11.629387}, 1.8)  ({-9.7, 11.629387}, {-9.7, 21.2})  ({-9.7, 21.2}, {-7.5, 23.4}, {-7.5, 21.2}, 2.2)  ({-7.5, 23.4}, {7.5, 23.4})  ({7.5, 23.4}, {9.7, 21.2}, {7.5, 21.2}, 2.2)  ({9.7, 21.2}, {9.7, 11.629387})  ({10.6000001182939, 10.070541204891}, {9.7, 11.629387}, {11.5, 11.629387}, 1.8)  ({10.6000001182939, 10.070541204891}, {13.0999999516436, 8.62716491624432})  ({13.0999999516436, 8.62716491624432}, {14.2, 6.721909}, {12, 6.721909}, 2.2)  ({14.2, 6.721909}, {14.2, -4.75})  ({14.2, -4.75}, {9, -9.95}, {9, -4.75}, 5.2)  ({9, -9.95}, {-9, -9.95}) |

## EA - Instance properties (ExclusionArea<Id>/Instance.txt)

|  |
| --- |
| # Instance properties of the EA (Exclusion Area):  Tolerances: # Tolerance properties of the EA:  Eo = 0 # error margin in theta\_ orientation (in rad)  Ep = 0.25 # error margin in P0 position (in mm) |

## RP - Arm Contour\_\_\_\_ (RoboticPositioner<Id>/Contour\_\_\_\_.txt)

|  |
| --- |
| # A list of segments (Pa, Pb) and arcs (Pa, Pb, Pc) for describe  # the template of RP.Actuator.Arm.Contour\_\_\_\_ in S4 (in mm):  ({0.807111, 2.35}, {5.807111, 2.35})  ({5.807111, 2.35}, {5.807111, -2.35}, {5.807111, 0}, 2.35)  ({5.807111, -2.35}, {0.807111, -2.35})  ({0.807111, -2.35}, {0.807111, 2.35}) |

## RP - Barrier Contour\_ (RoboticPositioner<Id>/Contour\_.txt)

|  |
| --- |
| # A list of segments (Pa, Pb) and arcs (Pa, Pb, Pc, R) to describe  # the template of the RP.Barrier.Contour\_, in S1 (in mm):  ({13.955, 0}, {-13.955, 0}, {0, 0}, 13.955)  ({-13.955, 0}, {13.955, 0}, {0, 0}, 13.955) |

## RP – Rotor 1 compression function (RoboticPositioner<Id>/F1.txt)

|  |
| --- |
| # The compression-function of the quantifier of rot 1 of the RP in step/rad.  # Must be defined almost in the rot 1 domain [theta\_1min, theta\_1max]:  # theta\_1: position of rot 1 (in rad)  # p\_1: position of rot 1 (in step)  -6.28318530717959 -194086.302  0 0  6.28318530717959 194086.302  12.5663706143592 388172.604 |

## RP – Rotor 2 compression function (RoboticPositioner<Id>/F2.txt)

|  |
| --- |
| # The compression-function of the quantifier of rot 2 of the RP in step/rad.  # Must be defined in the rot 2 domain [theta\_\_\_3min, theta\_\_\_3max]:  # theta\_\_\_3: position of rot 2 (in rad)  # p\_\_\_3: position of rot 2 (in step)  -6.28318530717959 -30720  0 0  6.28318530717959 30720  12.5663706143592 61440 |

## RP - Instance properties (RoboticPositioner<Id>/Instance.txt)

|  |
| --- |
| # Instance properties of the RP (Robotic Positioner):  ActuatorInstance: # Instance properties of the RP.Actuator (sizing, orientation and others):  L01 = 5.9930890625 # distance between P0 and P1 (in mm)  theta\_1min = -0.043633231 # position angleâ€™s lower limit of the axis 0-1 respect S1 (in rad)  theta\_1max = 6.326818538 # position angleâ€™s upper limit of the axis 0-1 respect S1 (in rad)  theta\_1 = 0 # position angle of axis 0-1 respect S1 (in rad)  theta\_O3o = 3.1489737525635 # orientation of S3 respect S1 when theta\_1 = 0 (in rad)    ArmInstance: # Instance properties of the RP.Actuator.Arm (siz., ori. and quant.):  L12 = 5.8025 # distance between P1 and P2 (in mm)  L13 = 5.9437703125 # distance between P1 and P3 (in mm)  theta\_\_O3 = 0 # orientation of S3 respect S2 (in rad)  R3 = 0.75 # radio of representative circle of the microlens (in mm)  theta\_\_\_3min = -0.043633231 # position angleâ€™s lower limit of axis 1-3 respect S3 (in rad)  theta\_\_\_3max = 3.185225885 # position angleâ€™s upper limit of axis 1-3 respect S3 (in rad)  theta\_\_\_3 = 0 # position angle of axis 1-3 respect S3 (in rad)    SPMmin = 0.005 # SPM minimum: is the SPM due to the minimum jump during gen. (in mm)  SPMsim = 0.00893643794946 # SPM of simulation: is the max. deviation in the rad. traj. (in mm)  PAkd = Pre # position angles knowledge degree [Pre | App | Unk]  CMFInstance: # Instance properties of the RP.CMF:  SF1.vmaxabs = 3 # absolute maximum velocity of rot 1 when MFT = mftSquare (in step/ms)  SF2.vmaxabs = 0.945 # absolute maximum velocity of rot 2 when MFT = mftSquare (in step/ms)  RF1.vmaxabs = 3 # absolute maximum velocity of rot 1 when MFT = mftRamp (in step/ms)  RF2.vmaxabs = 0.945 # absolute maximum velocity of rot 2 when MFT = mftRamp (in step/ms)  RF1.amaxabs = 4294.967295 # absolute maximum accel. of rot 1 when MFT = mftRamp (in step/ms^2)  RF2.amaxabs = 4294.967295 # absolute maximum accel. of rot 2 when MFT = mftRamp (in step/ms^2)  MFM = Square # motion function type [Square | Ramp]  SSM = Free # square synchronous mode [Free, Tmin, MaxTmin]  RSM = Free # ramp synchronous mode [Free | Tmin | MaxTmin | Tv | MaxTv]  Id1 = 0 # CAN identifier of the rot 1 controller (a nonneg. integer number)  Id2 = 0 # CAN identifier of the rot 2 controller (a nonneg. integer number)  Tolerances: # Tolerances of the RP:  Eo = 0.000872665 # error margin in theta\_ orientation (in rad)  Ep = 0.1 # error margin in P0 position (in mm)  Tstop = 0 # margin time from last pos. stored in memory, to stop. rotors (in ms)  Tshiff = 1 # margin time to shift of all rotors of RPs in Fiber MOS (in ms)  SPMadd = 0.32 # SPM additional: is a component of SPM added once (in mm)  Status: # Qualitative status of the RP:  Disabled = False # disabling status [False | True]  FaultProbability = 0 # probability of fault status (a real number in [0, 1])  FaultType = Unk # type of fault [Unk | Sta | Dyn] |

Note that SPM values depend on (SPMmin, SPMsim, Tshiff, SMPadd) and the value assigned to the SPM is a function of (PAkd, Purpose). The calculus of SPM is a complex process which is described in the file:

‘megara-fmpt-4.8.0/data/Manuals/calculus\_of\_SPM.txt’

1. Errors could only be present if the property megara.fmat.strict is set to false in FMAT. The FMPT has is own validation routine to avoid sequences of movements where RPs could collide. [↑](#footnote-ref-1)
2. This file is the output of the FMAT whose name is ‘megara-cb0.txt’ or similar (cb1, cb2, etc.). [↑](#footnote-ref-2)