|  |  |
| --- | --- |
| moons_logo.jpg | |
| Document Title | MOONS Verification Process Functional Specification |
| Document Number | VLT-TRE-MON-XXXX-XXXX |
| Issue | 1.0 |
| Date | 18 March 2019 |

|  |  |  |  |
| --- | --- | --- | --- |
| Prepared By | S Watson | Signature | S Watson |
| MOONS FPU WPM | Date | 18/03/19 |
| Approved By | S. Beard | Signature |  |
| MOONS software WPM | Date |  |
| J. Nix | Signature |  |
| MOONS software engineer | Date |  |
| A. O’Brien | Signature |  |
| MOONS software engineer | Date |  |
| Released By |  | Signature |  |
|  | Date |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Contributors | | | |
| S Watson | S Beard | J Nix | A O’Brien |
| P Rees |  |  |  |
|  |  |  |  |

**Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Sections Affected** | **Change Description** |
| 0.1 | 15-02-2019 | All | First draft |
| 0.2 | 26-02-2019 | All | General improvements |
| 0.3 | 27-02-2019 | All | Styling changes, outputs added, parameter algorithms added, pre-requisites section removed |
| 0.4 | 13-03-2019 | 2.3, 2.4, 2.5, 2.9 | Sequence changes and various other minor changes. |
| 1.0 | 18-03-2018 | 2.10, 2.11 | Added explanatory notes to data dependencies |

**TABLE OF CONTENTS**

[1 Introduction 1](#_Toc3807130)

[2 Verification process 2](#_Toc3807131)

[2.1 General rules 2](#_Toc3807132)

[2.2 Initialisation 2](#_Toc3807133)

[2.3 Verification rig setup 3](#_Toc3807134)

[2.4 Collision detection 4](#_Toc3807135)

[2.5 Limit characterisation 4](#_Toc3807136)

[2.6 Pupil alignment 5](#_Toc3807137)

[2.7 Metrology calibration 6](#_Toc3807138)

[2.8 Metrology height 7](#_Toc3807139)

[2.9 Datum repeatability 7](#_Toc3807140)

[2.10 Positional repeatability and gearbox calibration 8](#_Toc3807141)

[2.11 Positional verification 9](#_Toc3807142)

[3 Hardware control specification 10](#_Toc3807143)

[3.1 FPU control 10](#_Toc3807144)

[3.2 Rotary stage 10](#_Toc3807145)

[3.3 Linear stage 10](#_Toc3807146)

[3.4 Lamps and control DAQ 11](#_Toc3807147)

[3.5 Camera control 11](#_Toc3807148)

[4 Image analysis specification 12](#_Toc3807149)

[4.1 Find metrology targets from posrep camera 12](#_Toc3807150)

[4.2 Find metrology targets from metcal camera 12](#_Toc3807151)

[4.3 Find backlit fibre from metcal camera 13](#_Toc3807152)

[4.4 Find metrology target heights from metht camera 13](#_Toc3807153)

[4.5 Find projected spot from pupaln camera 14](#_Toc3807154)

[5 Performance parameter calculations 15](#_Toc3807155)

[5.1 Metrology target calibration 15](#_Toc3807156)

[5.2 Pupil alignment 15](#_Toc3807157)

[5.3 Datum repeatability 16](#_Toc3807158)

[5.4 Positional repeatability 16](#_Toc3807159)

[5.5 Gearbox correction calibration 16](#_Toc3807160)

[5.6 Positional accuracy 17](#_Toc3807161)

[6 Verification process outputs 18](#_Toc3807162)

[6.1 Images 18](#_Toc3807163)

[6.2 Pass/fail results 18](#_Toc3807164)

[6.3 Critical test data 18](#_Toc3807165)

[6.4 Diagnostic test data 19](#_Toc3807166)

**LIST OF FIGURES**

[Figure 1: FPU Verification Process Flowchart 1](#_Toc3362518)

**LIST OF TABLES**

|  |  |
| --- | --- |
| TABLE | PAGE |

**No table of figures entries found.**

# Introduction

This document analyses the “FPU Verification Process Flow Chart”, below, and breaks it into smaller tasks, which are described in detail.

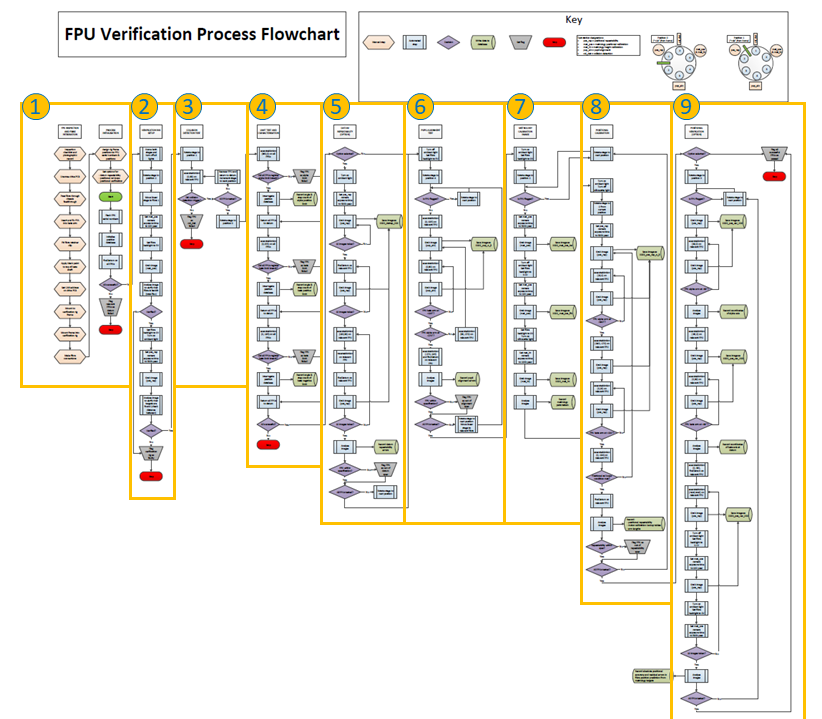


Figure 1: FPU Verification Process Flowchart

# Verification process

## General rules

* It is expected that the process will, for the most part, not be carried out in its entirety. The user should thus have options to pick certain tests to carry out, subject to data dependencies noted in the sequence descriptions.
* The user should be able to extract data from any test previously carried out and display it.
* When moving the rotary stage, the last successful operation from all FPUs must have been a findDatum. The only exception to this is prior to running the Collision Detection Test.
* Images taken should not be deleted or overwritten.
* Where red dots are indicated ● the intention is that the test will stop with an error and the operator will take immediate action.
* Otherwise, unless otherwise stated, it should be assumed that reasonably anticipated errors are handled by the software and the process should continue. For example, this could refer to one-off anomalous image analysis results or a failed positional repeatability test. Unexpected errors will result in a fatal exception.
* The verification process should be highly reliable as long as the environment is constant. This means that if parameters such as physical camera setup, illumination and temperature remain stable, the system should be robust and have a high probability of success.
* Where a process is manual (i.e. not controlled by software), it is stated in *italics.*
* Unless otherwise stated, all FPU and rotary stage angles are defined in absolute terms.
  + For descriptions of some loops, relative movements are used. These are indicated as (a,b) [relative]. However, the software should convert these to absolute position to avoid rounding errors.
  + For the FPU, positive angles are anticlockwise
  + For the rotary stage, positive angles are clockwise

## Initialisation

Purpose – this is the process where a user loads FPUs into the verification rig, flashes the electronics with the correct serial number and ensures that the FPU is responding to commands by running a findDatum operation.

*For each fibre positioner received*

*Note the unique serial number assigned to that fibre positioner (etched onto chassis)*

*If FPU position is not at or very close to datum then*

*Manually initialise FPU to the point where it can be automatically datumed*

*Fix to a position on the verification rig frame*

*Assign a CAN ID commensurate with that position*

*Make an association between the position and serial number in the verification software interface*

*Manually reprogram each FPU with the latest firmware*

*User sets configurable parameters and sets tests to be carried out*

For each of 6 FPUs

Flash the FPU PCB with the correct serial number

Initialise the FPU position database with (-180,0)

Find beta datum

Find alpha datum

Find datum for all FPUs simultaneously

If all FPUs successful

Set **INITIALISE = OK**

Start the automated verification tests (as below) 

Else

For each failed FPU

Set **INITIALISE = FAIL**

Stop with an error – FPU or electronics failure ●.

*If any FPUs have failed*

*If datum is confirmed non-functional*

*Remove FPU from verification rig*

*Choose a new FPU to fill this slot*

*Update the serial number to turntable slot to CAN bus mapping*

*Else*

*Manually correct the non-datum issue*

*Flag FPU for retest and leave it on the rig*

*Restart the initialisation process until all FPUs can be datumed and the automated tests can begin.*

## Verification rig setup

Purpose – this is a self-check of the verification rig critical functions, whereby sample images are taken from all test stations and the image analysis functions are run to verify that the images resulting are as expected.

Note – this sequence should adapt to the tests chosen by the user in the initialisation phase. The user may not request pupil alignment or metrology calibration tests because the FPU does not have a fibre, thus the self-check should skip the respective stations.

Configurable parameters

* **PUPALN\_POS\_N** – the rotary stage angle required to place each FPU under the first pupil alignment fold mirror, degrees
* **PUPALN\_LIN\_POS\_N** – the linear stage position required to illuminate each FPU fibre, mm
* **PUPALN\_EXPOSURE** – the exposure time for a correctly exposed image, microseconds
* **METCAL\_POS\_N** - the rotary stage angle required to place each FPU under the metcal camera, degrees
* **METCAL\_EXPOSURE\_TARGET** – the exposure time for a correctly exposed image of the targets, microseconds
* **METHT\_POS\_N** - the rotary stage angle required to place each FPU in front of the metht camera, degrees
* **METHT\_EXPOSURE** – the exposure time for a correctly exposed image, microseconds
* **POSREP\_POS\_N** – the rotary stage angle required to place each FPU under the positional repeatability camera, degrees
* **POSREP\_EXPOSURE** – the exposure time for a correctly exposed image, microseconds

Home the turntable and the linear stage

Turn off all lights

Move rotary stage to **PUPALN\_POS\_N** *(*3.2*)*

Move linear stage to **PUPALN\_LIN\_POS\_N** *(*3.3*)*

Configure pupaln camera with **PUPALN\_EXPOSURE** *(*3.5*)*

Configure lighting for pupil alignment test (ambient/silhouette off, backlight 5V) *(*3.4*)*

Take image with pupaln camera *(*3.5*)*

Pass image to pupil alignment image analysis script *(*4.5*)*

If exception returned then

Stop with an error – pupil alignment equipment failure ●

Move rotary stage to **METCAL\_POS\_N** *(*3.2*)*

Configure metcal camera with **METCAL\_EXPOSURE\_TARGET** *(*3.5*)*

Configure lighting for metrology calibration target image (backlight/silhouette off, ambient on) *(*3.4*)*

Take image with metcal camera *(*3.5*)*

Pass image to metrology calibration target image analysis script *(*4.2*)*

If exception returned then

Stop with an error – metrology calibration equipment failure ●

Move rotary stage to **METHT\_POS\_N** *(*3.2*)*

Configure metht camera with **METHT\_EXPOSURE** *(*3.5*)*

Configure lighting for metrology height image (backlight/ambient off, silhouette on) *(*3.4*)*

Take image with metht camera *(*3.5*)*

Pass image to metrology height image analysis script *(*4.4*)*

If error returned then

Stop with an error – metrology height equipment failure ●

Move rotary stage to **POSREP\_POS\_N** *(*3.2*)*

Configure posrep camera for **POSREP\_EXPOSURE** *(*3.5*)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(*3.4*)*

Take image with posrep camera *(*3.5*)*

Pass image to positional repeatability image analysis script *(*4.1*)*

If error returned then

Stop with an error – positional repeatability equipment failure ●

Home the turntable and the linear stage

Turn off all lights

## Collision detection

Purpose – to functionally test the collision detection circuit using a soft contact with the beta arm. This will give confidence that the system is functioning without putting the FPU at risk of unnecessary damage. The test is carried out by moving the rotary stage to the test position, then driving the FPU into a soft grounded contact. The collision is recovered using the standard FPU driver recovery commands.

Configurable parameters

* **COLDET\_POS\_N –** the rotary stage angle at which the FPU can reach the contact
* **COLDET\_ALPHA –** the alpha angle at which the FPU the FPU can reach the contact
* **COLDET\_BETA –** the beta angle at which the FPU is expected to make contact and cause a collision detection

Data dependencies

* None

For each of 6 FPUs

Move rotary stage to **COLDET\_POS\_N** *(*3.2*)*

Configure and execute waveform to bring FPU to (**COLDET\_ALPHA, COLDET\_BETA** - 5)

Configure and execute waveform to bring FPU to (**COLDET\_ALPHA**, **COLDET\_BETA** + 5)

If FPU registers collision then

Se **COLDET\_TEST = OK**

Use FPU drive escape commands until beta collision protection can be re-enabled

Return alpha to datum (and move rotary stage to safe position if necessary)

Return beta to datum

Run findDatum

Else

Set **COLDET\_TEST = FAIL**

Stop with an error – FPU collision detection circuit non-functional ●

## Limit characterisation

Purpose – to determine where the end stop limits of the FPU are with respect to the datum position. This will enable the software safety limits to be set for each FPU. This will be achieved by safely activating the limit switch of each stage in both directions.

For the beta arm, this means driving to a position slightly beyond the expected position of the end stop switch in both positive and negative directions. Both tests will require the software to successfully handle exceptions from the FPU driver and then for the beta stop recovery process to be followed.

For the alpha arm, the datum position represents the limit in the negative direction. The positive limit will be found by driving to a positive slightly beyond the expected position of the datum switch. This will require the software to successful handle an exception from the FPU driver and then for the alpha limit breach recovery process to be followed.

Configurable parameters

* **LIMIT\_ALPHA\_POS\_EXPECT –** the expected positive limit of the alpha arm, plus a small margin
* **LIMIT\_BETA\_NEG\_EXPECT –** the expected negative limit of the beta arm, plus a small margin
* **LIMIT\_BETA\_POS\_EXPECT –** the expected positive limit of the beta arm, plus a small margin

Data dependencies

* None

For each of 6 FPUs

Configure and execute waveform to bring FPU to (**LIMIT\_ALPHA\_POS\_EXPECT, 0**)

If FPU registers alpha limit breach then

Set **LIMIT\_ALPHA** to position at limit breach

Write **LIMIT\_ALPHA** to database

Use FPU driver commands to clear limit breach state

Return alpha to datum

Else

Set **LIMIT\_ALPHA** = **NOT FOUND**

Configure and execute waveform to bring FPU to (**0, LIMIT\_BETA\_NEG\_EXPECT**)

If FPU registers collision then

Set **LIMIT\_BETA\_NEG** to position at collision

Write **LIMIT\_BETA\_NEG** to database

Use FPU driver commands to clear collision state

Return beta to datum

Else

Set **LIMIT\_BETA\_NEG** = **NOT FOUND**

Configure and execute waveform to bring FPU to (**0, LIMIT\_BETA\_POS\_EXPECT**)

If FPU registers collision then

Set **LIMIT\_BETA\_POS** to position at collision

Write **LIMIT\_BETA\_POS** to database

Use FPU driver commands to clear collision state

Return beta to datum

Else

Set **LIMIT\_BETA\_POS** = **NOT FOUND**

If **LIMIT\_ALPHA** = **NOT FOUND** or **LIMIT\_BETA\_NEG** == **NOT FOUND** or **LIMIT\_BETA\_POS** == **NOT FOUND**

**LIMIT\_TEST = FAIL**

Stop with an error – limit not found, collision may have occurred ●

Else

**LIMIT\_TEST** = **OK**

## Pupil alignment

Purpose - to determine precession of the FPU optical axis as it rotates through 16 positions representing all combinations of the major compass directions on both alpha and beta arms. A high-power LED backlights the fibre, which is projected onto a screen via fold mirrors and imaged by a camera. Software finds the centre of the projected spot in the image and, from this, the errors in the chassis, alpha and beta axes can be derived.

Configurable parameters

* **PUPALN\_POS\_N** – the rotary stage angle required to place each FPU under the first pupil alignment fold mirror, degrees
* **PUPALN\_LIN\_POS\_N**– the linear stage position required to illuminate each FPU fibre, mm
* **PUPALN\_EXPOSURE** – the exposure time for a correctly exposed image, microseconds
* **PUPALN\_PASS** – the maximum total deviation from the calibrated centre point which represents an acceptable FPU, degrees

Data dependencies

* None

Read configurable parameters

Configure pupaln camera with **PUPALN\_EXPOSURE** *(*3.5*)*

Configure lighting for pupil alignment test (ambient/silhouette off, backlight 5V) *(*3.4*)*

For each of 6 FPUs

Move rotary stage to **PUPALN\_POS\_N** *(*3.2*)*

Move linear stage to **PUPALN\_LIN\_POS\_N** *(*3.3*)*

Configure and execute waveform to reach (-170, -170)

For 4 iterations

Take image with pupaln camera *(*3.5*)*

For 3 iterations

Configure and execute waveform to reach (0,90) [relative]

Take image with pupaln camera *(*3.5*)*

If Not the final iteration then configure and execute waveform to reach (90,-270) [relative]

Configure and execute waveform to each (-179,1)

Run findDatum

For 16 saved images

Pass image to pupil alignment image analysis script *(*4.5*)*

Pass coordinates to pupil alignment parameter script *(*5.2*)*

Write **PUPALN\_CHASSIS\_ERR, PUPALN\_ALPHA\_ERR**, **PUPALN\_BETA\_ERR** and **PUPALN\_TOTAL\_ERR** to database

If **PUPALN\_TOTAL\_ERR** is less than **PUPALN\_PASS** then

Set **PUPALN\_TEST = OK**

Else

Set **PUPALN\_TEST = FAIL**

## Metrology calibration

Purpose – to determine the static relationship between the metrology targets and the fibre aperture for each FPU, allowing the MOONS metrology position to use the targets to infer the position of the fibre. The required parameters are the distance between the fibre aperture and targets in mm, to a precision of +/- 0.001 mm. An image will be taken of the FPU with an ambient LED illuminating the targets. A second image will be taken with the fibre backlit. Software will find the positions of the targets using a centre of mass technique, then the position of the fibre using a centroiding algorithm.

Configurable parameters

* **METCAL\_POS\_N**- the rotary stage angle required to place each FPU under the metcal camera, degrees
* **METCAL\_EXPOSURE\_TARGET**– the exposure time for a correctly exposed image of the targets, microseconds
* **METCAL\_EXPOSURE\_FIBRE** – the exposure time for a correctly exposed image of the backlit fibre, microseconds
* **METCAL\_BACKLIGHT\_VOLTAGE\_LOW** – the voltage setting at which the fibre can be correctly exposed from the metcal camera

Data dependencies

* None

Read configurable parameters

For each of 6 FPUs

Move rotary stage to **METCAL\_POS\_N** *(*3.2*)*

Configure metcal camera with **METCAL\_EXPOSURE\_TARGET** *(*3.5*)*

Configure lighting for metrology calibration target image (backlight/silhouette off, ambient on) *(*3.4*)*

Take image with metcal camera *(*3.5*)*

Pass image to metrology calibration target image analysis script *(*4.2*)*

Configure metcal camera with **METCAL\_EXPOSURE\_FIBRE** *(*3.5*)*

Configure lighting for metrology calibration fibre image (ambient/silhouette off, backlight to **METCAL\_BACKLIGHT\_VOLTAGE\_LOW**) *(*3.4*)*

Take image with metcal camera *(*3.5*)*

Pass image to metrology calibration fibre image analysis script *(*4.3*)*

Pass coordinates to metrology target calibration parameter script *(*5.1*)*

Write **METCAL\_FIBRE\_LARGE\_TARGET\_DISTANCE**, **METCAL\_FIBRE\_SMALL\_TARGET\_DISTANCE** and **METCAL\_TARGET\_VECTOR\_ANGLE** to database

## Metrology height

Purpose – to determine the height of the metrology targets above the beta arm surface, allowing a correction to be made within the MOONS metrology software pipeline. An LED lights a white surface behind the beta arm such that the arm and targets are silhouetted and the edges can be clearly defined. The silhouette is imaged by a camera and software finds the surface of the beta arm, then the surfaces of the targets, and calculates the height.

Configurable parameters

* **METHT\_POS\_N**- the rotary stage angle required to place each FPU in front of the metht camera, degrees
* **METHT\_EXPOSURE**– the exposure time for a correctly exposed image, microseconds
* **METHT\_HEIGHT\_TOLERANCE** – maximum allowable height of both targets, mm

Data dependencies

* None

Read configurable parameters

Configure metht camera with **METHT\_EXPOSURE** *(*3.5*)*

Configure lighting for metrology height image (backlight/ambient off, silhouette on) *(*3.4*)*

For each of 6 FPUs

Move rotary stage to **METHT\_POS\_N** *(*3.1*)*

Take image with metht camera *(*3.5*)*

Pass image to metrology height image analysis script *(*4.4*)*

Write **METHT\_SMALL\_TARGET\_HEIGHT** and **METHT\_LARGE\_TARGET\_HEIGHT** to database

If **METHT\_SMALL\_TARGET\_HEIGHT** or **METHT\_LARGE\_TARGET\_HEIGHT** greater than zero and less than **METHT\_HEIGHT\_TOLERANCE** then

Set **METHT\_TEST = OK**

Else

Set **METHT\_TEST = FAIL**

## Datum repeatability

Purpose – to determine the repeatability of the FPU datum position under repeated datum operations, both with and without a prior FPU motion. An ambient LED illuminates the metrology targets and a camera images the targets after each FPU datum. Software finds the target coordinates using a centre of mass technique, then calculates the deviation from the average position. The maximum and standard deviation of the errors found represent the datum repeatability acceptance criteria.

Configurable constants

* **METCAL\_POS\_N** – the rotary stage angle required to place each FPU under the metcal camera, degrees
* **METCAL\_EXPOSURE** – the exposure time for a correctly exposed image, microseconds
* **DATREP\_ITERATIONS** – the number of datum operations made for each test
* **DATREP\_MAX\_PASS** – the acceptable maximum single error from the average position, mm
* **DATREP\_STD\_PASS** – the acceptable standard deviation of errors from the average position, mm

Data dependencies

* None

Read configurable parameters

Configure metcal camera for **METCAL\_EXPOSURE** *(*3.5*)*

Configure lighting for metrology calibration test (silhouette/backlight off, ambient on) *(*3.4*)*

For each of 6 FPUs

Move rotary stage to **METCAL\_POS\_N** *(*3.2*)*

For **DATREP\_ITERATIONS** iterations (datum)

Run findDatum on FPU

Take image with metcal camera *(*3.5*)*

Pass image to metrology calibration image analysis script *(*4.1*)*

For **DATREP\_ITERATIONS** iterations (move-then-datum)

Configure and execute waveform to reach (-150,30)

Run reverseMotion

Run findDatum on FPU

Take image with metcal camera *(*3.5*)*

Pass image to metrology calibration image analysis script *(*4.1*)*

Pass coordinates to datum repeatability parameter script *(*5.2*)*

Write **DATREP\_DAT\_ONLY\_MAX, DATREP\_DAT\_ONLY\_STD, DATREP\_MOVE\_DAT\_MAX** and **DATREP\_MOVE\_DAT\_STD** to database

If **DATREP\_DAT\_ONLY\_MAX** is less than **DATREP\_MAX\_PASS**

and **DATREP\_DAT\_ONLY\_STD** is less than **DATREP\_STD\_PASS**

and **DATREP\_MOVE\_DAT\_MAX** is less than **DATREP\_MAX\_PASS**

and **DATREP\_MOVE\_DAT\_STD** is less than **DATREP\_STD\_PASS** then

Set **DATREP\_TEST = OK**

Else

Set **DATREP\_TEST = FAIL**

## Positional repeatability and gearbox calibration

Purpose - to run each FPU stage through a sequence of equally spaced movements, then back, and repeats several times. An ambient LED illuminates the metrology targets and a camera images the targets after each movement. Software finds the target coordinates using a centre of mass technique, then calculates the deviation from the nominal position, as well as the spread of points at that nominal position. From the spread of points at each nominal position, the positional repeatability can be derived, which represents a pass/fail parameter. Also, given an acceptable repeatability, the error from the nominal position can be converted into a correction function or lookup table. This can be used to correct the FPU motion such that the absolute accuracy requirement is verified.

Data dependencies

* Limit characterisation test (required)
  + LIMIT\_ALPHA
  + LIMIT\_BETA\_NEG
  + LIMIT\_BETA\_POS
* Pupil alignment test (if previously carried out)
  + PUPALN\_TEST
* Datum repeatability test (if previous carried out)
  + DATREP\_TEST

Configurable parameters

* **POSREP\_POS\_N** – the rotary stage angle required to place each FPU under the positional repeatability camera, degrees
* **POSREP\_EXPOSURE** – the exposure time for a correctly exposed image, microseconds
* **POSREP\_PASS** – the maximum acceptable deviation from an average position of a grouping of measured points at a given nominal position, mm
* **POSREP\_INCREMENTS** – the number of movements made within each positive sweep from the starting position
* **POSREP\_ITERATIONS** – the number of times each FPU sweeps back and forth

Read configurable parameters

Configure posrep camera for **POSREP\_EXPOSURE** *(*3.5*)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(*3.4*)*

For each of 6 FPUs

If **DATREP == FAIL** or **PUPALN == FAIL** then

Skip FPU

Move rotary stage to **POSREP\_POS\_N** *(*3.2*)*

Configure and execute waveform to reach (-170, -170)

Take image with posrep camera *(*3.5*)*

For **POSREP\_ITERATIONS** iterations

For **POSREP\_INCREMENTS** iterations

Configure and execute waveform to reach (+ [{180 + **LIMIT\_ALPHA**} / **POSREP\_INCREMENTS**], 0) [relative]

Take image with posrep camera *(*3.5*)*

For **POSREP\_INCREMENTS** iterations

Configure and execute waveform to reach (- [{180 + **LIMIT\_ALPHA**} / **POSREP\_INCREMENTS**], 0) [relative]

Take image with posrep camera *(*3.5*)*

For **POSREP\_INCREMENTS** iterations

Configure and execute waveform to reach (0, + [{**LIMIT\_BETA\_POS** – **LIMIT\_BETA\_NEG**} / **POSREP\_INCREMENTS**]) [relative]

Take image with posrep camera *(*3.5*)*

For **POSREP\_INCREMENTS** iterations

Configure and execute waveform to reach (0, - [{**LIMIT\_BETA\_POS** – **LIMIT\_BETA\_NEG**} / **POSREP\_INCREMENTS**]) [relative]

Take image with posrep camera *(*3.5*)*

Run findDatum on FPU

For all images

Pass image to positional repeatability image analysis script *(*4.1*)*

Pass coordinates to positional repeatability parameter script *(*5.4*)*

Write **POSREP\_RSS, POSREP\_ALPHA\_MAX, POSREP\_BETA\_MAX** and all **POSREP\_ALPHA\_MAX\_AT\_ANGLE\_Φ** / **POSREP\_BETA\_MAX\_AT\_ANGLE\_Φ** values to database

If **POSREP\_RSS** is less than **POSREP\_PASS** then

Set **POSREP = OK**

Else

Set **POSREP = FAIL**

Pass coordinates to gearbox calibration parameter script *(*5.5*)*

Write **GEARCOR\_ALPHA** and **GEARCOR\_BETA** to database

## Positional verification

Purpose – to verify that the gearbox calibrations derived in the previous step result in absolute accuracy within specification. Note, this procedure uses random positions. However, it is not essential to use completely random positions – the test must merely include some element of randomness to mitigate the effects of systematic errors in the measurement setup.

Configurable parameters

* **POSREP\_POS\_N** – the rotary stage angle required to place each FPU under the positional repeatability camera
* **POSREP\_EXPOSURE** – the exposure time in microseconds for a correctly exposed image
* **POSVER\_PASS** – the maximum deviation from an average position of a grouping of measured points at a given nominal position which represents an acceptable FPU
* **POSVER\_ITERATIONS** – the number of random positions where real position is tested against nominal position

Data dependencies

* Positional repeatability test (required)
  + POSREP\_TEST
  + GEARCOR\_ALPHA
  + GEARCOR\_BETA
* Pupil alignment test (if previously carried out)
  + PUPALN\_TEST
* Datum repeatability test (if previously carried out)
  + DATREP\_TEST

Read configurable parameters

Configure posrep camera for **POSREP\_EXPOSURE** *(*3.5*)*

Configure lighting for positional repeatability test (silhouette/backlight off, ambient on) *(*3.4*)*

For each of 6 FPUs

If **DATREP == FAIL** or **PUPALN == FAIL** or **POSREP == FAIL** then

Skip FPU

Move rotary stage to **POSREP\_POS\_N** *(*3.2*)*

Apply **GEARCOR\_ALPHA** and **GEARCOR\_BETA** corrections

Configure and execute waveform to reach (-170, -170)

Take image with posrep camera *(*3.5*)*

For 7 iterations

Configure and execute waveform to reach (45,0) [relative]

Take image with posrep camera *(*3.5*)*

For **POSVER\_ITERATIONS** iterations

Generate new **POSVER\_ALPHA\_RAND\_N** and **POSVER\_BETA\_RAND\_N**

Write **POSVER\_ALPHA\_RAND\_N** and **POSVER\_BETA\_RAND\_N** to database

Configure and execute waveform to reach (**POSVER\_ALPHA\_RAND\_N**, **POSVER\_BETA\_RAND\_N**)

Take image with posrep camera *(*3.5*)*

Reverse motion

Run findDatum on FPU

For all images

Pass image to positional repeatability image analysis script *(*4.1*)*

Pass coordinates to positional accuracy parameter script *(*5.6*)*

Write**POSVER\_ERROR\_MAX** and all **POSVER\_ERROR\_N** valuesto database

If **POSVER\_ERROR\_MAX** is less than **POSVER\_PASS** then

Set **POSVER\_TEST = OK**

Else

Set **POSVER\_TEST = FAIL**

# Hardware control specification

## FPU control

* See FPU protocol 2 documentation

## Rotary stage

* Purpose – the rotary stage rotates the verification frame such that FPUs can be positioned under various test stations.
* Rotary stage - Thorlabs NR360S/M - <https://www.thorlabs.de/thorproduct.cfm?partnumber=NR360S/M>
* Controller - Thorlabs BSC201 – <https://www.thorlabs.de/thorproduct.cfm?partnumber=BSC201>
* Required operations
  + Home
  + Forward absolute movement
  + Reverse absolute movement
* Adjustable parameters
  + Speed

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stage angles | Frame position | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| metcal | 251 | 311 | 11 | 71 | 131 | 191 |
| metht | 268 | 328 | 28 | 88 | 148 | 208 |
| posrep | 132 | 192 | 252 | 312 | 12 | 72 |
| pupaln | TBD | TBD | TBD | TBD | TBD | TBD |

## Linear stage

* Purpose – the linear stage moves a backlight LED and lens such that light is shone into one of the 6 fibre patch cables coupled with the FPU fibres. This allows imaging of the fibre centre and back projection of the fibre aperture such that pupil alignment can be measured.
* Linear stage - Thorlabs MTS50/M-Z8 - [https://www.thorlabs.de/newgrouppage9.cfm?objectgroup\_id=3002&pn=MTS50/M-Z8#3006](https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=3002&pn=MTS50/M-Z8" \l "3006)
* Controller - Thorlabs BSC201 – <https://www.thorlabs.de/thorproduct.cfm?partnumber=BSC201>
* Required operations
  + Home
  + Forward absolute movement
  + Reverse absolute movement
* Adjustable parameters
  + Speed

## Lamps and control DAQ

* Purpose – three LED lamps allow different setups of ambient and directional illumination within the verification enclosure, as well as backlighting of the fibres.
* Ambient illumination - Thorlabs LED cluster LIU365A - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2853>
* Metrology target silhouetting - Thorlabs LED cluster LIU365A - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2853>
* Fibre backlight - Thorlabs mounted LED MCWHLP1 - <https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=2692>
* LED driver – Thorlabs LEDD1B - <https://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=2616>
* DAQ - RedLab1208LS - [https://www.meilhaus.org/downloadserver/redlab/manual/RedLab%201208LS\_en.pdf](https://www.meilhaus.org/downloadserver/redlab/manual/RedLab 1208LS_en.pdf)
* Required operations
  + On-off for ambient illumination and target silhouetting lamps
  + Voltage control (0-5V) for fibre backlight
  + Delay to allow for warmup

## Camera control

* Purpose – four cameras take images of the FPU and fibre projection during the various tests
* Positional repeatability – Basler ace acA3800-10gm GigE - <https://www.edmundoptics.com/p/Basler-ace-acA3800-10gm-Monochrome-GigE-Camera/32412/>
* Metrology calibration – Basler ace acA3800-10gm GigE - <https://www.edmundoptics.com/p/Basler-ace-acA3800-10gm-Monochrome-GigE-Camera/32412/>
* Metrology height – Basler ace acA3800-10gm GigE - <https://www.edmundoptics.com/p/Basler-ace-acA3800-10gm-Monochrome-GigE-Camera/32412/>
* Pupil alignment – Basler ace acA1920-40gm GigE - <https://www.edmundoptics.com/p/basler-ace-aca1920-40gm-monochrome-gige-camera/3429/>
* Required operations
  + Connect to camera and load configuration using given parameters
  + Save image from a single camera given its IP address
* Adjustable parameters
  + Exposure time

|  |  |  |
| --- | --- | --- |
| Camera | ID number | IP address |
| metcal | 22390460 | 169.254.189.121 |
| metht | 22390461 | 169.254.190.121 |
| posrep | 22390458 | 169.254.187.121 |
| pupaln | 22584939 | 169.254.108.113 |

# Image analysis specification

## Find metrology targets from posrep camera

* Purpose – to find coordinates for the metrology targets in real space using the posrep camera. Required measurement uncertainty is < 0.005 mm (~1/5 pixel).
* Used in the following tests:
  + Datum repeatability *(*2.9*)*
  + Positional repeatability and gearbox correction *(*2.10*)*
  + Positional verification *(*2.11*)*
* Inputs
  + Path - bitmap image on file from posrep camera
* Configurable parameters
  + **POSREP\_SMALL\_DIAMETER** – expected diameter of small target, mm
  + **POSREP\_LARGE\_DIAMETER** – expected diameter of large target, mm
  + **POSREP\_DIAMETER\_TOLERANCE** – allowable tolerance of both targets, mm
  + **POSREP\_QUALITY\_METRIC** – minimum circularity for target identification, dimensionless
  + **POSREP\_THRESHOLD** – pixel intensity for image thresholding, 0-255
  + **POSREP\_PLATESCALE** – image platescale, mm/pixel
  + **POSREP\_DISTORTION\_MATRIX** – camera calibration matrix
* Return
  + **POSREP\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **POSREP\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **POSREP\_SMALL\_TARGET\_QUALITY** – circularity of small target contour, dimensionless
  + **POSREP\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **POSREP\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
  + **POSREP\_LARGE\_TARGET\_QUALITY** – circularity of large target contour, dimensionless
* Exceptions
  + Multiple small targets found
  + Multiple large targets found
  + No small targets found
  + No large targets found
  + Incorrect targets found (distance between out of tolerance)
* Algorithm
  + Process image (distortion correction, grayscale, Gaussian blur, threshold)
  + Find contours
  + Filter contours on size and circularity to detect targets and reject contamination
  + Find centre of targeted contours using Centre of Mass

## Find metrology targets from metcal camera

* Purpose – to find coordinates for the metrology targets in real space using the metcal camera. Required measurement uncertainty is < 0.001 mm (~1/5 pixel).
* Used in the following tests:
  + Metrology calibration *(*2.7*)*
* Inputs
  + Path - bitmap image on file from metcal camera
* Configurable parameters
  + **METCAL\_SMALL\_DIAMETER** – expected diameter of small target, mm
  + **METCAL\_LARGE\_DIAMETER** – expected diameter of large target, mm
  + **METCAL\_DIAMETER\_TOLRANCE** – allowable tolerance of both targets, mm
  + **METCAL\_QUALITY\_METRIC** – minimum circularity for target identification, dimensionless
  + **METCAL\_THRESHOLD** – pixel intensity for image thresholding, 0-255
  + **METCAL\_PLATESCALE** – image platescale, mm/pixel
* Return
  + **METCAL\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **METCAL\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **METCAL\_SMALL\_TARGET\_QUALITY** – circularity of small target contour, dimensionless
  + **METCAL\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **METCAL\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
  + **METCAL\_LARGE\_TARGET\_QUALITY** – circularity of large target contour, dimensionless
* Exceptions
  + Multiple small targets found
  + Multiple large targets found
  + No small targets found
  + No large targets found
  + Incorrect targets found (distance between out of tolerance)
* Algorithm
  + Process image (grayscale, Gaussian blur, threshold)
  + Find contours
  + Filter contours on size and circularity to detect targets and reject contamination
  + Find centre of targeted contours using Centre of Mass

## Find backlit fibre from metcal camera

* Purpose – to find coordinates for the metrology targets in real space using the metcal camera. Required measurement uncertainty is < 0.001 mm (~1/5 pixel).
* Used in the following tests:
  + Metrology calibration *(*2.7*)*
* Inputs
  + Path - bitmap image on file from metcal camera
* Configurable parameters
  + **METCAL\_PLATESCALE** – image platescale, mm/pixel
  + **METCAL\_TBD** – unknown parameters relating to Gaussian fit
* Return
  + **METCAL\_FIBRE\_X** – X coordinate of fibre, mm
  + **METCAL\_FIBRE\_Y** – Y coordinate of fibre, mm
  + **METCAL\_TBD** – unknown parameter relating to quality of Gaussian fit
* Exceptions
  + Unable to find maxima
  + Unable to fit Gaussian distribution
  + Unacceptable quality of fit
* Algorithm
  + Find fibre location (find maxima)
  + Find Gaussian fit around point

## Find metrology target heights from metht camera

* Purpose – to measure the height of the metrology targets above the beta arm surface to allow correction for elongation in the metrology system. Required measurement uncertainty is < 0.001 mm (~1/5 pixel).
* Used in the following tests:
  + Metrology height *(*2.8*)*
* Inputs
  + Path - bitmap image on file from metht camera
* Configurable parameters
  + **METHT\_PLATESCALE** – image platescale, mm/pixel
  + **METHT\_THRESHOLD** – pixel intensity for image thresholding, 0-255
  + **METHT\_SCAN\_HEIGHT** – height at which image is scanned to find side of beta arm, pixels
  + **METHT\_GAUSS\_BLUR** – Gaussian blur parameter, pixels (must be odd number)
  + **METHT\_STANDARD\_DEV** – acceptable standard deviation of target points, mm
  + **METHT\_NOISE\_METRIC** – acceptable image noise metric, dimensionless
* Return
  + **METHT\_SMALL\_TARGET\_HEIGHT** – height of small target above beta arm, mm
  + **METHT\_LARGE\_TARGET\_HEIGHT** – height of large target above beta arm, mm
  + **METHT\_SMALL\_TARGET\_QUALITY** – standard deviation of small target points, mm
  + **METHT\_LARGE\_TARGET\_QUALITY** – standard deviation of large target points, mm
* Exceptions
  + Standard deviation of small target points exceeds limit
  + Standard deviation of large target points exceeds limit
  + Excessive image noise
* Algorithm
  + Process image (Gaussian blur, grayscale and threshold)
  + Check image noise
  + Find side of beta arm
  + At fixed offsets from arm side, find transition pixel representing top edge of arm and targets
  + Fit line through arm points
  + Find normal from line to target points
  + Average normals to give target height

## Find projected spot from pupaln camera

* Purpose – to measure misalignment of the projected axis from a backlit fibre as the FPU rotates and thus determine the angular errors in each of the three mechanism axes. . Required measurement uncertainty is < 1 mm (~TBD pixel).
* Used in the following tests:
  + Pupil alignment *(*2.6*)*
* Inputs
  + Path - bitmap image on file from pupaln camera
* Configurable parameters
  + **PUPALN\_CIRCULARITY\_THRESH** – minimum circularity for target identification, dimensionless
  + **PUPALN\_THRESHOLD** – pixel intensity for image thresholding, 0-255
  + **PUPALN\_NOISE\_METRIC** – acceptable image noise metric, dimensionless
  + **PUPALN\_DISTORTION\_MATRIX** – camera calibration matrix
  + **PUPALN\_PLATESCALE** – image platescale, mm/pixel
* Return
  + **PUPALN\_SPOT\_X** – X coordinate of projected spot centre of mass, mm
  + **PUPALN\_SPOT\_Y** – Y coordinate of projected spot centre of mass, mm
  + **PUPALN\_CIRCULARITY** – circularity of projected spot, dimensionless
* Exceptions
  + Unable to find spot
  + Unacceptable spot quality
  + Unacceptable image noise
* Algorithm
  + Process image (distortion correct, convert, de-noise and threshold)
  + Find contours
  + Filter contours on size and circularity to detect targets and reject contamination
  + Find centre of targeted contour using Centre of Mass

# Performance parameter calculations

## Metrology target calibration

* Input
  + **METCAL\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **METCAL\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **METCAL\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **METCAL\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
  + **METCAL\_FIBRE\_X** – X coordinate of fibre, mm
  + **METCAL\_FIBRE\_Y** – Y coordinate of fibre, mm
* Return
  + **METCAL\_FIBRE\_LARGE\_TARGET\_DISTANCE** – distance from fibre to large target, mm
  + **METCAL\_FIBRE\_SMALL\_TARGET\_DISTANCE** – distance from fibre to small target, mm
  + **METCAL\_TARGET\_VECTOR\_ANGLE** – angle subtended between vectors from fibre to large & small targets
* Algorithm
  + **METCAL\_FIBRE\_LARGE\_TARGET\_DISTANCE** = magnitude of vector from fibre coordinate to large target coordinate
  + **METCAL\_FIBRE\_SMALL\_TARGET\_DISTANCE** = magnitude of vector from fibre coordinate to small target coordinate
  + **METCAL\_TARGET\_VECTOR\_ANGLE** = angle subtended between **METCAL\_FIBRE\_LARGE\_TARGET\_DISTANCE** and **METCAL\_FIBRE\_SMALL\_TARGET\_DISTANCE**
    - x = sin-1 [ a X b / ( |a| |b| ) ]

## Pupil alignment

* Configurable parameters
  + **PUPALN\_CALIBRATED\_CENTRE\_X** – X coordinate of calibrated central coordinate, mm
  + **PUPALN\_CALIBRATED\_CENTRE\_Y** – Y coordinate of calibrated central coordinate, mm
* Input
  + **PUPALN\_SPOT\_X** – X coordinate of projected spot centre of mass, mm
  + **PUPALN\_SPOT\_Y** – Y coordinate of projected spot centre of mass, mm
* Return
  + **PUPALN\_CHASSIS\_ERR** – chassis angular error, degrees
  + **PUPALN\_ALPHA\_ERR** – alpha angular error, degrees
  + **PUPALN\_BETA\_ERR** – beta angular error, degrees
  + **PUPALN\_TOTAL\_ERR** – total angular error, degrees
  + **PUPALN\_ERROR\_BARS** – error bars TBD, probably using st dev of magnitudes
* Algorithm
  + Given **PUPALN\_SPOT\_X** and **PUPALN\_SPOT\_Y** datasets equivalent to (αN, β1-4) for N = 1-4, total 16 points
  + Beta error
    - For N = 1-4
      * Average (αN, β1-4) to give βcircleN coordinate
      * Find magnitude of vector from βcircleN to each (αN, β1-4)
      * Average all magnitudes to give beta error at αN
    - **PUPALN\_BETA\_ERR** = average ofall beta errors
  + Alpha error
    - Average all βcircleN coordinates to give αcircle coordinate
    - Find magnitude of vector from αcircle to each βcircleN
    - **PUPALN\_ALPHA\_ERR** = average all magnitudes
  + Chassis error
    - Average all coordinates
    - Subtract calibrated centre coordinates from average = xc
    - **PUPALN\_CHASSIS\_ERR** = tan-1(xc / 4101.4)
  + Total error
    - **PUPALN\_TOTAL\_ERR** = sum(**PUPALN\_CHASSIS\_ERR**, **PUPALN\_ALPHA\_ERR, PUPALN\_BETA\_ERR**)

## Datum repeatability

* Input
  + **METCAL\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **METCAL\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **METCAL\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **METCAL\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
* Return
  + **DATREP\_DAT\_ONLY\_MAS** – maximum error of ‘datum-only’ error, mm
  + **DATREP\_DAT\_ONLY\_STD** – standard deviation of ‘datum-only’ errors, mm
  + **DATREP\_MOVE\_DAT\_MAX** – maximum error of ‘move-then-datum’ error, mm
  + **DATREP\_MOVE\_DAT\_STD** – standard deviation of ‘move-then-datum’ errors, mm
* Algorithm
  + Average all coordinates to create baseline coordinate
  + For each ‘datum-only’ coordinate
    - Subtract baseline coordinate from coordinate
  + For each move-then-datum coordinate
    - Subtract baseline coordinate from move-then-datum coordinate
  + Find standard deviation and maximum ‘error – datum only’ values
  + Find standard deviation and maximum ‘error – move-then-datum’ values

## Positional repeatability

* Input
  + **POSREP\_INCREMENTS** – the number of movements made within each positive sweep from the starting position
  + **POSREP\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **POSREP\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **POSREP\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **POSREP\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
* Return
  + **POSREP\_ALPHA\_MAX\_AT\_ANGLE\_Φ** – maximum positional error at average of all points at alpha angle Φ, where Φ depends on the configurable parameters in 2.10
  + **POSREP\_BETA\_MAX\_AT\_ANGLE\_Φ** – maximum positional error at average of all points at beta angle Φ, where Φ depends on the configurable parameters in 2.10
  + **POSREP\_ALPHA\_MAX** – maximum alpha positional error at any angle
  + **POSREP\_BETA\_MAX** – maximum beta positional error at any angle
  + **POSREP\_RSS** – RSS of **POSREP\_ALPHA\_MAX** and **POSREP\_BETA\_MAX**
* Algorithm
  + For each value of Φ
    - Average all X and Y coordinates to create baseline coordinate
    - Subtract baseline coordinate from each coordinate
    - **POSREP\_ALPHA\_MAX\_AT\_ANGLE\_Φ / POSREP\_BETA\_MAX\_AT\_ANGLE\_Φ** = max value
  + **POSREP\_ALPHA\_MAX / POSREP\_BETA\_MAX** = max value
  + **POSREP\_RSS** = RSS of **POSREP\_ALPHA\_MAX** and **POSREP\_BETA\_MAX**

## Gearbox correction calibration

* Input
  + **POSREP\_INCREMENTS** – the number of movements made within each positive sweep from the starting position
  + **POSREP\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **POSREP\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **POSREP\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **POSREP\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
* Return
  + **GEARCOR\_ALPHA** – alpha gearbox correction lookup table / function
  + **GEARCOR\_BETA** – beta gearbox correction lookup table / function
* Algorithm
  + TBD

## Positional accuracy

* Input
  + **POSVER\_ALPHA\_RAND\_N** – Randomized alpha positions
  + **POSVER\_BETA\_RAND\_N** – Randomized beta positions
  + **POSREP\_SMALL\_TARGET\_X** – X coordinate of small target, mm
  + **POSREP\_SMALL\_TARGET\_Y** – Y coordinate of small target, mm
  + **POSREP\_LARGE\_TARGET\_X** – X coordinate of large target, mm
  + **POSREP\_LARGE\_TARGET\_Y** – Y coordinate of large target, mm
* Return
  + **POSVER\_ERROR\_N** – Positional error against nominal position, mm
  + **POSVER\_ERROR\_MAX** – Maximum positional error against nominal position, mm
* Algorithm
  + TBD

# Verification process outputs

## Images

* Folder structure */moonsdata/Verification/FPUID/*
  + FPUID – four digit FPU serial number
* Image filename convention *FPUID\_TEST-ID\_TEST-C\_DATE\_IMAGE.bmp*
  + TEST-ID – the test identifier e.g. posrep, metcal etc
  + TEST-C – the test cycle. For most FPUs this will only ever by 1 but if a test is repeated, this should increment to 2.
  + DATE – format *yymmdd*
  + IMAGE – three digit image identifier, resetting to 000 for each test

## Pass/fail results

* INITIALISE = {N/A, OK, FAIL}
* COLDET\_TEST = {N/A, OK, FAIL}
* LIMIT\_TEST = {N/A, OK, FAIL}
* PUPALN\_TEST = {N/A, OK, FAIL}
* METHT\_TEST = {N/A, OK, FAIL}
* DATREP\_TEST = {N/A, OK, FAIL}
* POSREP\_TEST = {N/A, OK, FAIL}
* POSVER\_TEST = {N/A, OK, FAIL}

## Critical test data

These values represent the minimum dataset required from the verification process.

* LIMIT\_CHARACTERISATION\_RESULTS { }
  + LIMIT\_ALPHA
  + LIMIT\_BETA\_NEG
  + LIMIT\_BETA\_POS
* PUPIL\_ALIGNMENT\_RESULTS { }
  + PUPALN\_CHASSIS\_ERR
  + PUPALN\_ALPHA\_ERR
  + PUPALN\_BETA\_ERR
  + PUPALN\_TOTAL\_ERR
* METROLOGY\_CALIBRATION\_RESULTS { }
  + METCAL\_FIBRE\_LARGE\_TARGET\_DISTANCE
  + METCAL\_FIBRE\_SMALL\_TARGET\_DISTANCE
  + METCAL\_TARGET\_VECTOR\_ANGLE
* METROLOGY\_HEIGHT\_RESULTS { }
  + METHT\_SMALL\_TARGET\_HEIGHT
  + METHT\_LARGE\_TARGET\_HEIGHT
* DATUM\_REPEATABILITY\_RESULTS { }
  + DATREP\_DAT\_ONLY\_MAS
  + DATREP\_DAT\_ONLY\_STD
  + DATREP\_MOVE\_DAT\_MAX
  + DATREP\_MOVE\_DAT\_STD
* POSITIONAL\_REPEATABILITY\_RESULTS { }
  + POSREP\_ALPHA\_MAX
  + POSREP\_BETA\_MAX
  + POSREP\_RSS
* GEARBOX\_CALIBRATION\_RESULTS { }
  + GEARCOR\_ALPHA
  + GEARCOR\_BETA
* POSITIONAL\_VERIFICATION\_RESULTS { }
  + POSVER\_ERROR\_MAX

## Diagnostic test data

These values are unnecessary if an FPU clearly passes all the pass/fail tests but in borderline or failed cases, will be useful to diagnose where the problems originate. These datasets will be much larger than the critical test datasets and may need to be categorised by arm angle or test iteration.

* PUPIL\_ALIGNMENT\_RESULTS { }
  + PUPALN\_ERROR\_BARS
  + PUPALN\_SPOT\_X
  + PUPALN\_SPOT\_Y
  + PUPALN\_CIRCULARITY
* METROLOGY\_CALIBRATION\_RESULTS { }
  + METCAL\_SMALL\_TARGET\_X
  + METCAL\_SMALL\_TARGET\_Y
  + METCAL\_SMALL\_TARGET\_QUALITY
  + METCAL\_LARGE\_TARGET\_X
  + METCAL\_LARGE\_TARGET\_Y
  + METCAL\_LARGE\_TARGET\_QUALITY
* METROLOGY\_HEIGHT\_RESULTS { }
  + METHT\_SMALL\_TARGET\_QUALITY
  + METHT\_LARGE\_TARGET\_QUALITY
* DATUM\_REPEATABILITY\_RESULTS { }
  + POSREP\_SMALL\_TARGET\_X
  + POSREP\_SMALL\_TARGET\_Y
  + POSREP\_SMALL\_TARGET\_QUALITY
  + POSREP\_LARGE\_TARGET\_X
  + POSREP\_LARGE\_TARGET\_Y
  + POSREP\_LARGE\_TARGET\_QUALITY
* POSITIONAL\_REPEATABILITY\_RESULTS { }
  + POSREP\_ALPHA\_MAX\_AT\_ANGLE\_Φ
  + POSREP\_BETA\_MAX\_AT\_ANGLE\_Φ
  + POSREP\_SMALL\_TARGET\_X
  + POSREP\_SMALL\_TARGET\_Y
  + POSREP\_SMALL\_TARGET\_QUALITY
  + POSREP\_LARGE\_TARGET\_X
  + POSREP\_LARGE\_TARGET\_Y
  + POSREP\_LARGE\_TARGET\_QUALITY
* POSITIONAL\_VERIFICATION\_RESULTS { }
  + POSVER\_ERROR\_N