LINC-NIRVANA

The **L**BT **IN**terferometric **C**amera and
Near-**I**nfra**R**ed / **V**isible **A**daptive
i**N**terferometer for **A**stronomy

A collaborative project of the MPIA Heidelberg, INAF-Arcetri,
Universität zu Köln, and MPIfR Bonn

http://www.mpia.de/LINC

LINC-NIRVANA

—

Fringe and Flexure Tracking System software requirements

Doc. No.	LN-KOELN-FDR-ICS-001
Short Title	FFTS Software
Issue	1.1
Date	06. June 2005

	Name	Date	Signature
Prepared	T. Bertram	06. June 2005	
Approved	C. Straubmeier	06. June 2005	
Released	M. Kürster	07. June 2005	

Document Change Record

Issue	Date	Section/Paragraph Affected	Reasons / Remarks
0.1	9. May 2005	new document	....
1.0	20. May 2005	all	approver remarks
1.1	06. June 2005	all	PI remarks

Contents

List of Figures

1 Scope

This document outlines the operation of the FFTS and provides an overview of the software requirements and tasks.

2 Applicable Documents

No.	Title	Number and Issue
AD1	Calibration Unit	LN-MPIA-FDR-AIT-007

3 Acronyms and Abbreviations

DPU	Detector Positioning Unit
ICS	Instrument control software module
LFFTS	Fringe and Flexure Tracker software module
LTCS	LN Telescope control wrapper system (cf. LN-MPIA-FDR-ICS-003-ICS)
NTP	Network time protocol
UC	Use case

4 Introduction

The design of the LFFTS software module is a work in progress. Software development should always start with a thorough analysis of the system requirements, infrastructure and interaction with subsystems, other modules and hardware.
This document describes the results of this analysis in a standard approach as a set of use cases. First the main tasks are outlined in the operations section 5.

5 Operations

In this section we introduce the main tasks of the FFTS and the sequence in which they will be executed. The following flowcharts give a coarse outline of the main tasks, rather than a complete representation of the whole system. This overview is intended as a starting point for the requirement analysis and the software design.

Figure 1 shows the task flow of a regular observing night. Certain calibration steps are required for fringe tracking. Since the source and the stability of the calibration of the various parameters is not yet known, the outlined sequence is only preliminary. Both the position of the optical axis and the plate scale have to be known prior to the observation, in order to be able to calculate the trajectory the PSF of a reference star within the focal plane. These parameters may both be obtained by the FFTS itself.

Once calibrated and all required parameters are provided, the FFTS is ready to operate. For each pointing, the trajectory of the reference star PSF has to be calculated and the FFTS detector moved in a way that it follows the PSF in the focal plane (denoted as 'PSF following' in the following). The alignment of the optical axes of the two single eye telescopes has to be checked before the fringe

Figure 1: Flowchart: main sequence of FFTS tasks. The encircled tasks are described in more detail in a following flowchart. Slanted boxes indicate input communication with other systems.

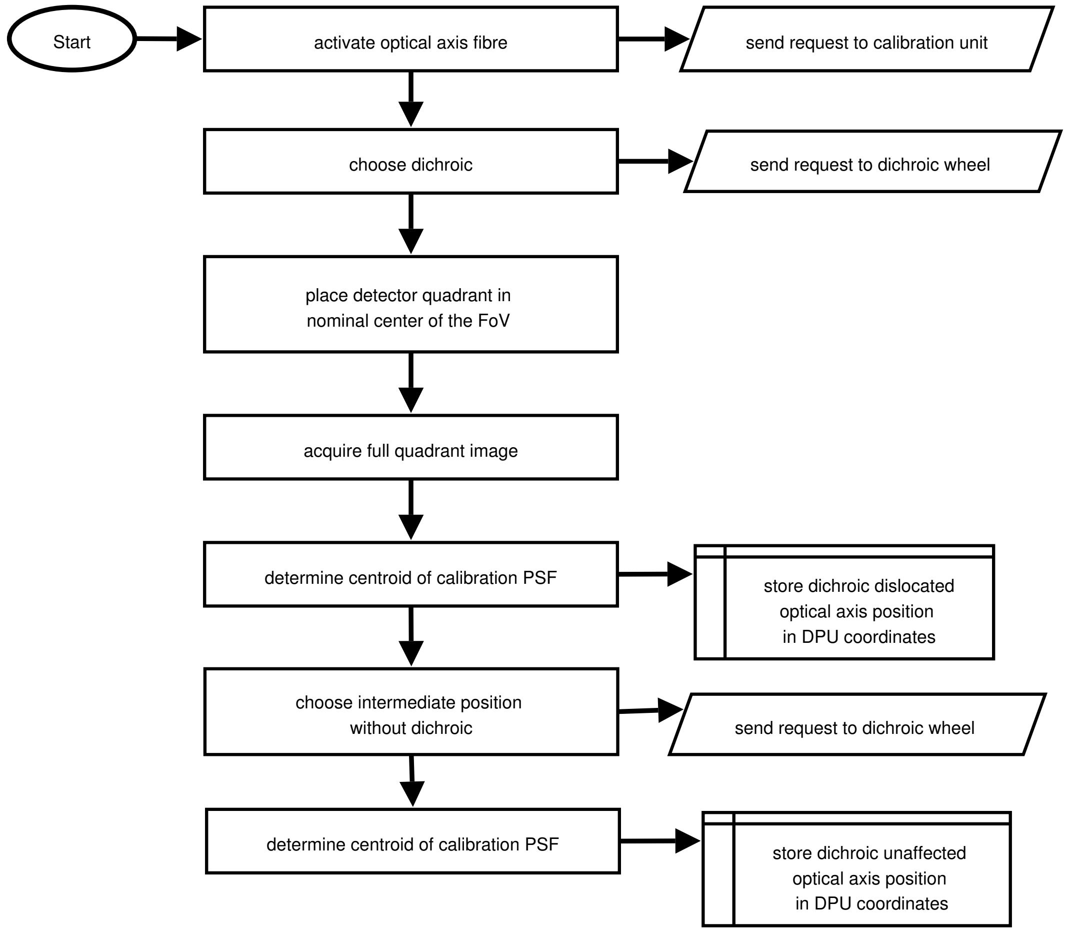

Figure 2: Task flowchart: determine optical axis position.

tracking loop is closed. In addition, for the high frequency fringe tracking analysis, the frames obtained by the FFTS detector will be used in a flexure monitoring loop.

5.1 Optical axis position

Whenever the telescope is pointed at an object, the optical axis defines the center of the circular path that the PSF of an off axis reference star describes in the focal plane. Therefore, the knowledge of this position is essential for PSF following. The optical axis of LINC-NIRVANA can be identified with the calibration unit (cf. AD 1).

Part of the focal plane will be located below the cryogenic dichroic, which reflects part of the light to the science detector. The tilted dichroic introduces an offset of the optical axis. This offset position corresponds to the center of the trajectories of reference star PSFs that have a small angular distance to the science object. All PSFs with larger distances to the optical axis will not be affected by the dichroic and therefore will have a different center of rotation.

Figure 2 outlines the tasks that must be performed to determine the locations of the two centers.

5.2 Plate scale

The absolute plate scale is the second important optical parameter for the calculation of the PSF trajectory. It has to be identical for both single-eye telescopes. Figure 3 demonstrates, how the plate scale could be obtained by the FFTS. Information about the plate scale can also come from a variety of additional sources, including the GWS or MHWS or the infrared camera system LIRS) It is not yet clear, how stable the plate scale is and how often it needs to be adjusted or confirmed by systems other than the single eye AO systems.
The accurate positions of two stars within the usable FoV of the FFTS, preferably with a large angular separation, are required for the plate scale determination with the FFTS. The plate scale has to be known beforehand to an extent that allows us to position the detector quadrant such that the PSF can be detected, i.e. the uncertainty of the PSF position has to be smaller than 2.5".

5.3 Following the reference star PSF

figure 5 shows the steps that will be performed to follow the PSF of the reference star in the focal plane. The calculated trajectory will have to be adjusted to the real trajectory of the PSF. Deviations from the adjusted trajectory will be registered by the flexure loop (cf. figure 8).

5.4 Optical axes alignment

The optical axes of the two single-eye telescopes have to be exactly aligned to provide maximum fringe contrast and highest angular resolution for the science detector. The alignment of the two axes may be checked by either the science detector or the FFTS detector. A confirmation of the alignment with the FFTS has the advantage of the guaranteed presence of a bright point source PSF. In that case, the telescope does not need to be re-pointed. Figure 6 outlines how this can be achieved by the FFTS and AO. In the outer parts of the FoV that is accessible by FFTS, static optical errors prevent a perfect overlap of the single-eye PSFs. This permanent displacement in not a result of misaligned optical axes; therefore it has to be known and taken into account when correcting for a possible misalignment.

5.5 Fringe tracking loop

Once all necessary parameters are available and the fringe tracking reference star PSF can be sampled with the FFTS detector (PSF tracing is active), the zeroth, "white" fringe will have to be identified and the fringe tracking loop can be closed. Figure 7 shows the steps that have to be taken to close the loop. The zeroth fringe will be searched by scanning the OPD range of the piston mirror. The OPD range in which the zeroth fringe has to be located will be narrowed by successively increasing the bandwidth of the FFTS filter.
Once a fringe pattern is identified, the fringe tracking loop can be activated and locked on the fringes that are seen in the OPD-affected PSF. At this stage atmospheric OPD variation can be compensated by the FFTS, leaving an OPD which is fully controlable. By varying the controlled OPD, the fringe pattern is moved within the Airy envelope of the PSF. As long as the zeroth fringe is not in the center, the fringe contrast is asymmetric. This asymmetry provides the direction in which the zeroth fringe can be found. For narrow-band filters with a large coherence length, the asymmetry is not as

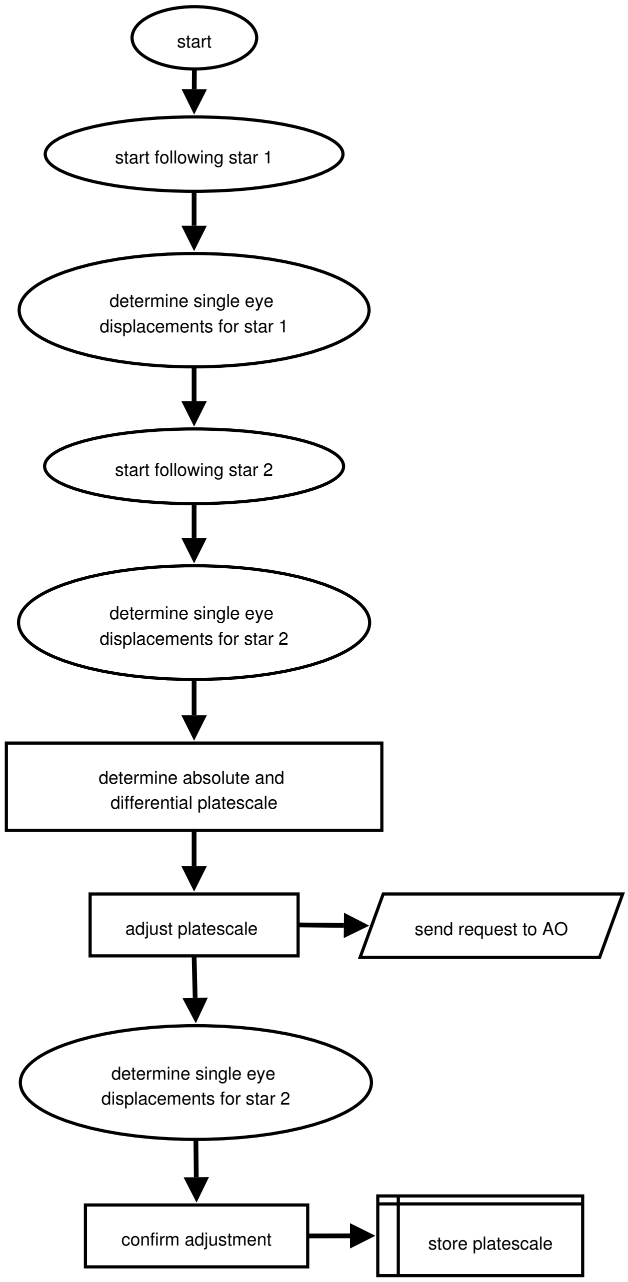

Figure 3: Task flowchart: determine plate scale. The determination of the single-eye displacement is outlined in figure 4, the PSF tracing task in figure 5.

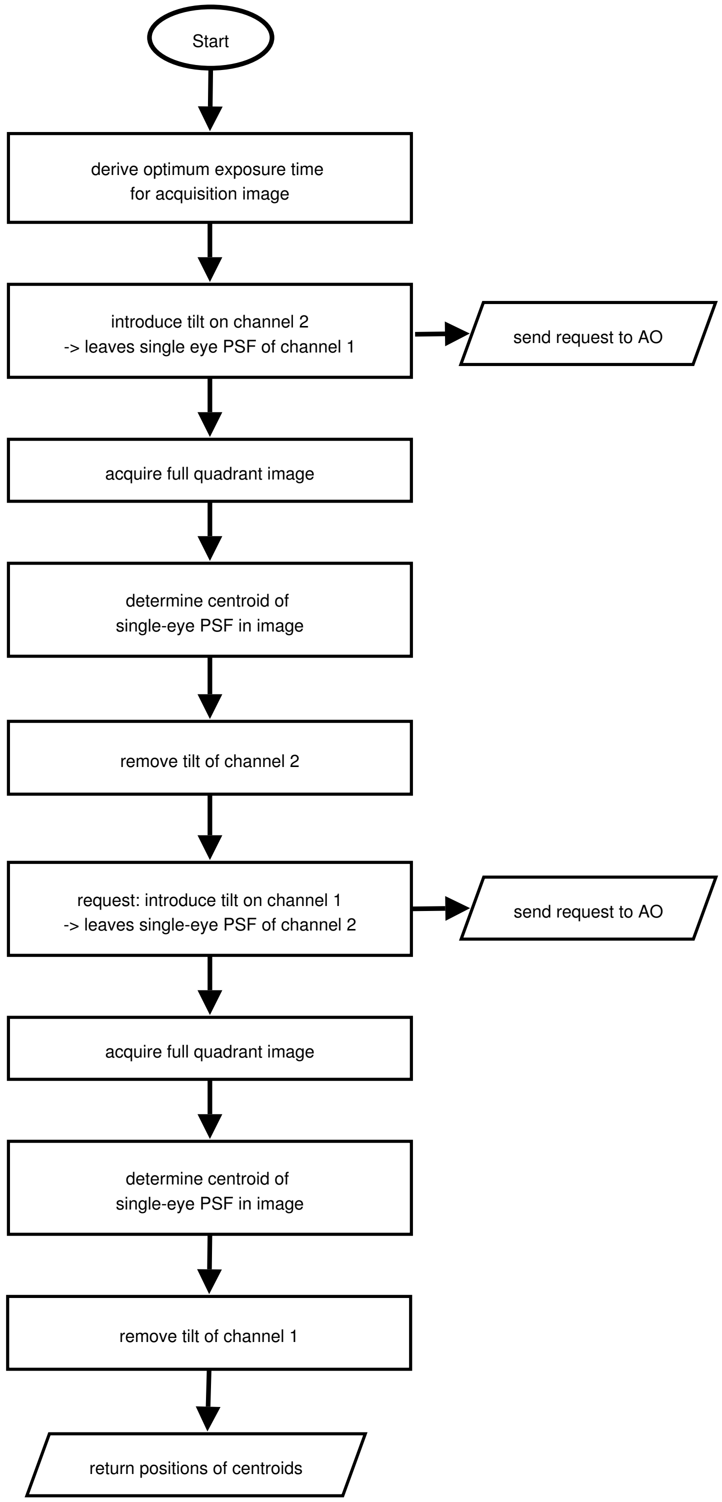

Figure 4: Task flowchart: determine single eye displacement, i.e. 2 individual measurements for single eye with artificial tilt introduced for the other eye.

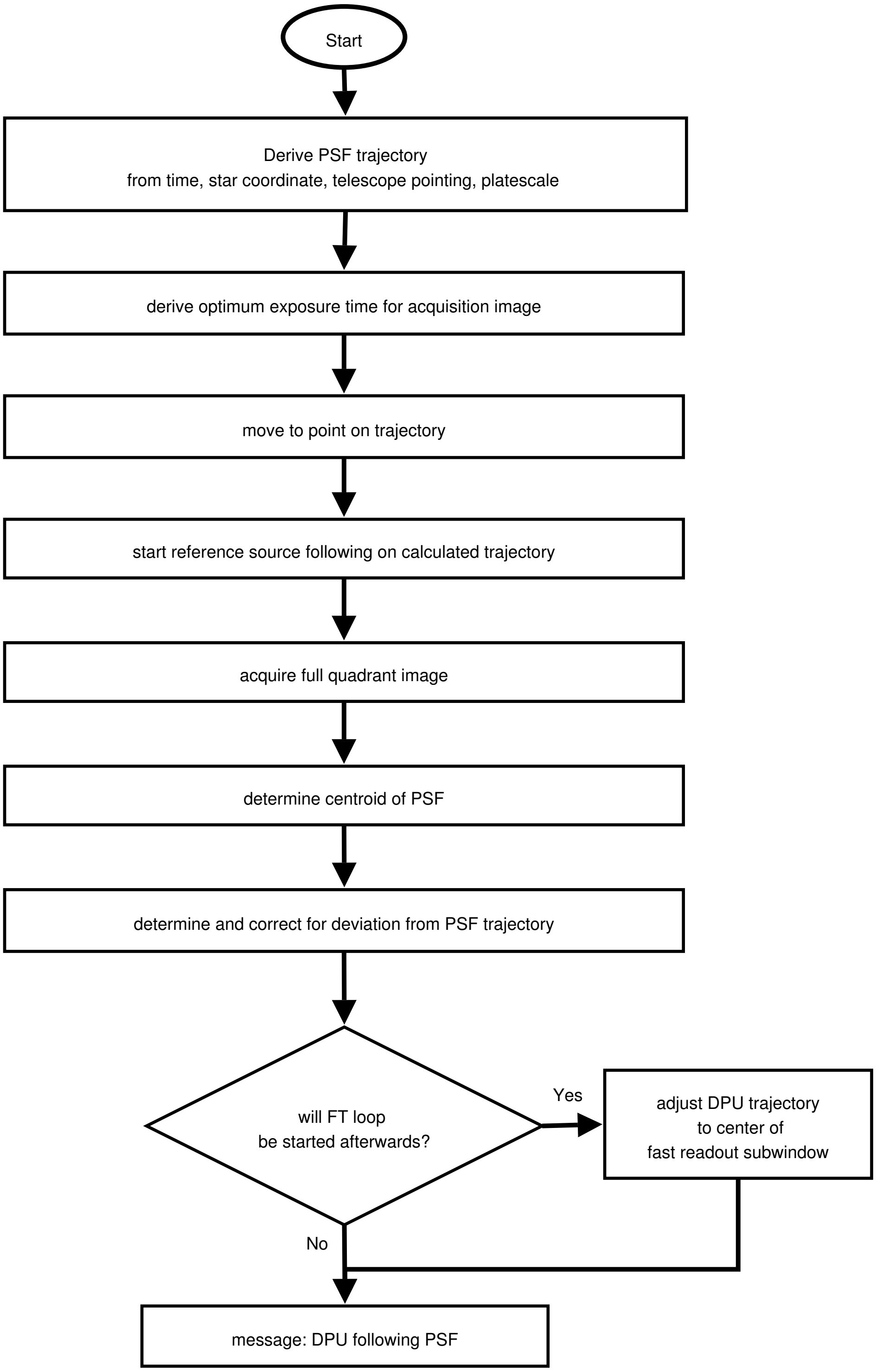

Figure 5: Task flowchart: follow reference star PSF

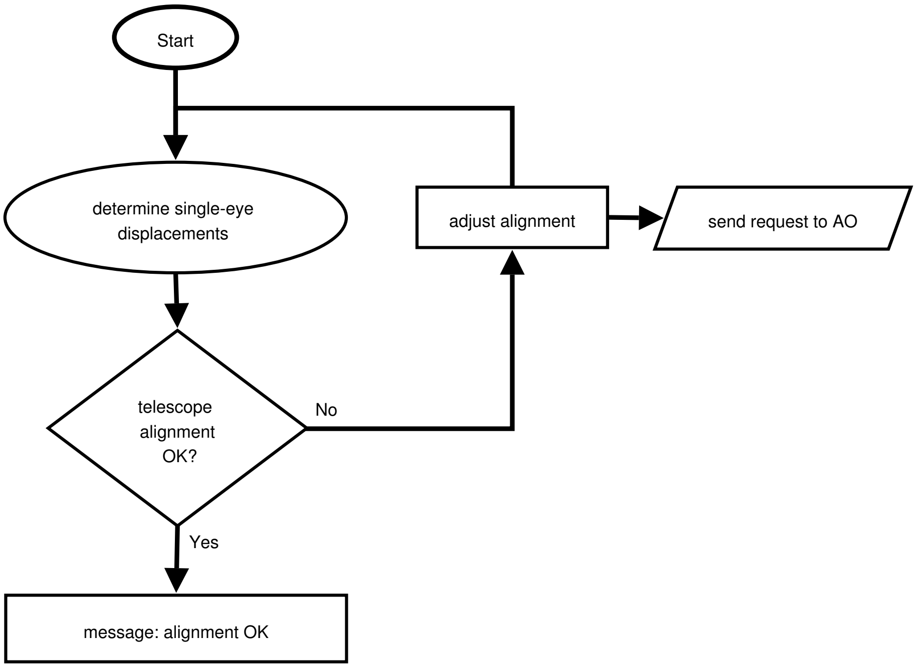

Figure 6: Task flowchart: determine optical axes alignment

clear as for the broad-band filters. In the narrow-band case, the approximate center of the fringe modulation function can be found by scanning the full range in which fringes can be identified and choosing the center of this range.

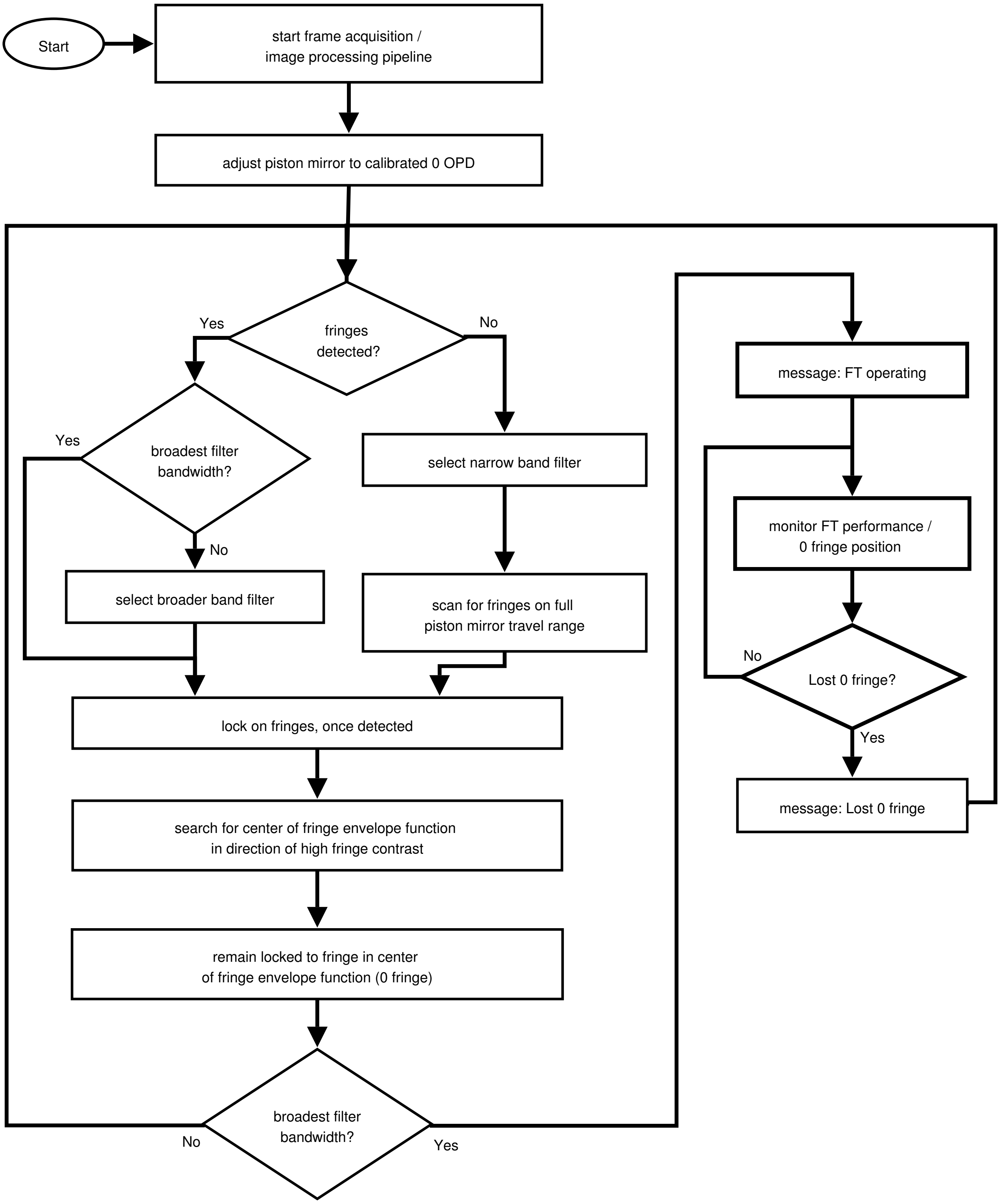

Figure 7: Task flowchart: fringe tracking loop

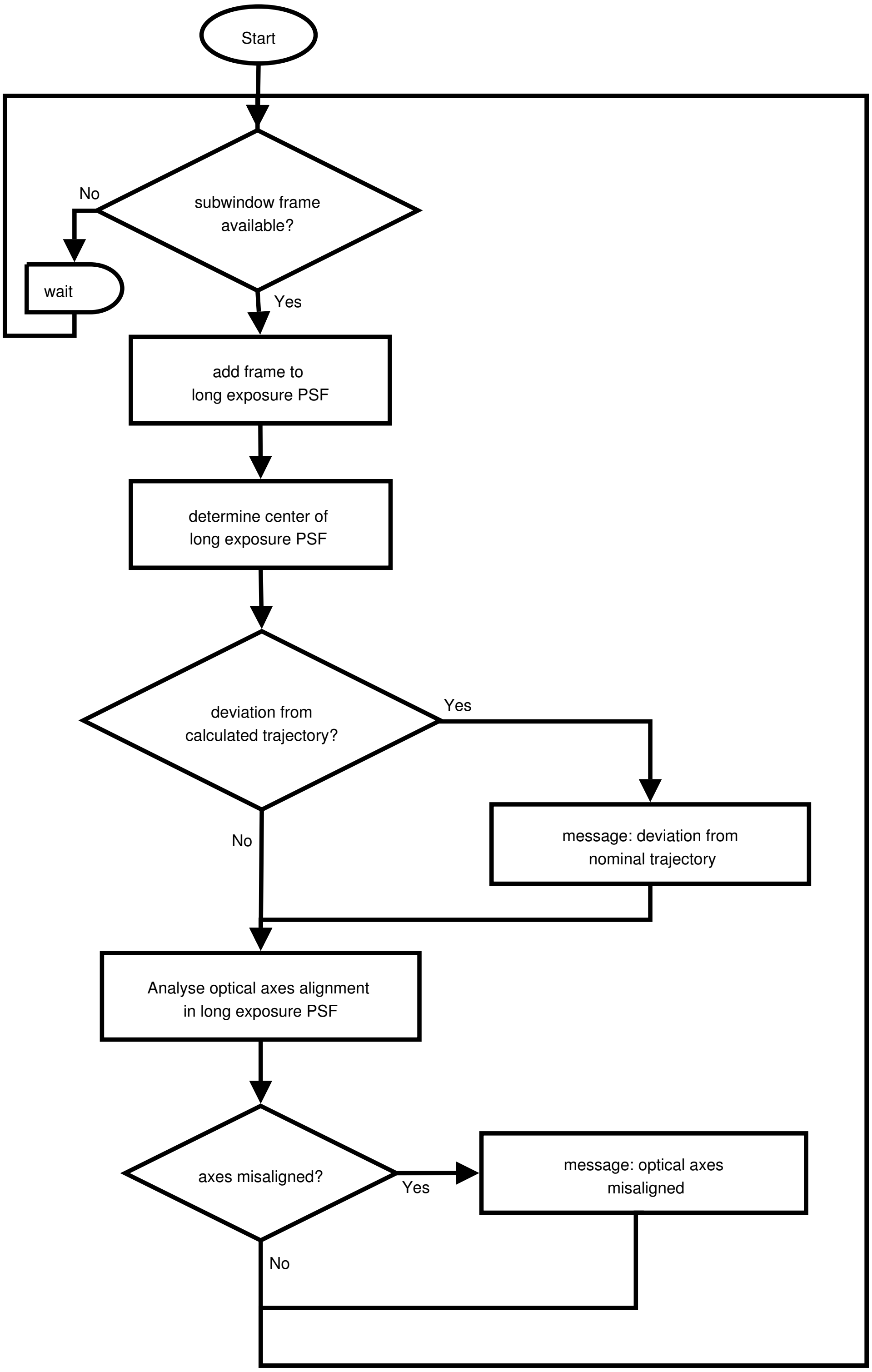

Figure 8: Task flowchart: flexure loop

6 Requirement analysis

In the following, the main actors, requirements and constraints for the LFFTS module are collected and the main use cases listed. Figure 9 shows the main packages within the LFFTS software module.

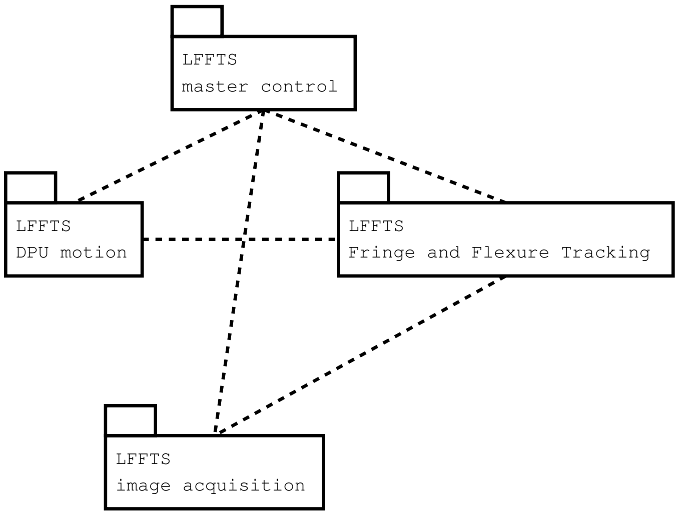

Figure 9: Package diagram: main packages

6.1 LFFTS DPU motion

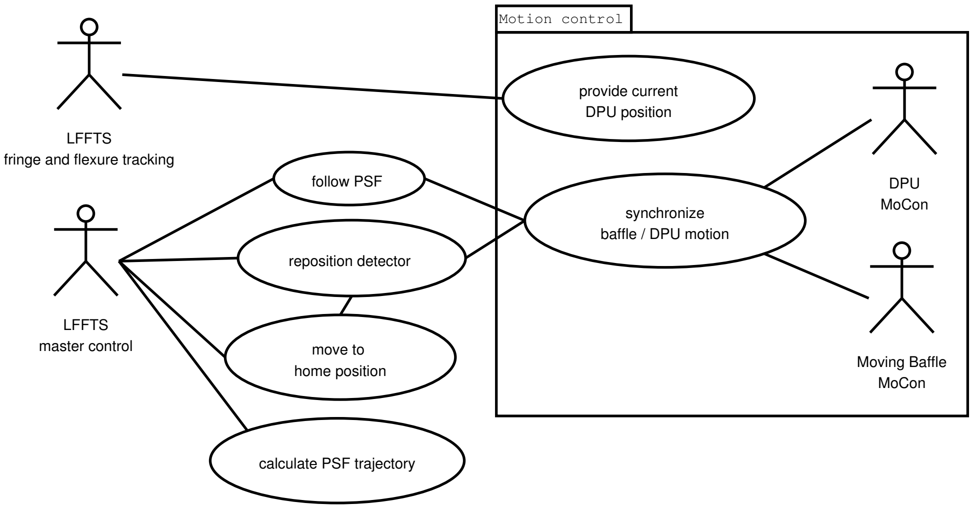

Figure 10: Use case diagram: DPU motion

6.1.1 Use case: reposition detector

Description:
Position the detector at a specified point on the focal plane. The point may be expressed in sky coordinates of a reference star or directly in DPU coordinates.

Postconditions:
Detector is moved to requested position.
Requirements:

R1: Prevention of repositioning through the center of the Moving Baffle (singularity in baffle rotation velocity).

R2: NTP operating. Synchronization with instrument / telescope time.

R3: Knowledge of vignetted region by dichroic.

R4: Knowledge of focal plane curvature.

Basic path: Position at reference star PSF

1: LFFTS master control provides PSF trajectory.

2: Check if PSF trajectory is accessible by DPU. Consider R3. If not accessible, continue with B.1.

3: Determine constant Z Position of PSF trajectory in DPU coordinates. Consider R4.

4: Find earliest point on PSF trajectory which is accessible by DPU. For this, determine the time required to move the DPU into the region of the PSF trajectory.

5: Determine DPU trajectory to reach this point. Consider R1.

6: Move DPU on determined DPU trajectory to this point, as described in UC 6.1.4.

7: Send message once position is reached.

8: Done.

Alternate path A: Position at DPU coordinate

A.1: LFFTS master control provides position in DPU coordinates.

A.2: Continue with 5.

Alternate path B:

B.1: Send failure message with reason.

B.2: Done.

6.1.2 Use case: follow PSF

Description:
Moves the DPU such that the detector follows the trajectory of the reference star PSF.
Preconditions:
Detector is located at or close to PSF trajectory.
Postconditions:
Detector continuously follows reference star trajectory.
Requirements:

R4: Knowledge of focal plane curvature.

R5: Knowledge of motor control parameters of DPU stages.

R7: Resolution of trajectory linearization.

Basic path:

1: LFFTS master control provides PSF trajectory.

2: Calculate corresponding DPU trajectory. Consider R4, R5 and R7.

3: Adjust current position to get detector to coincide with PSF trajectory.

4: Start moving DPU on determined DPU trajectory as described in UC 6.1.4

5: Send message: PSF following active.

6: Wait for updated PSF trajectory or for cancellation of PSF following.

7: If updated PSF trajectory is available, continue with 1.

8: If cancellation of PSF tracing requested by LFFTS master control: Done.

6.1.3 Use case: calculate PSF trajectory

Postconditions:
Returns the PSF trajectory in the focal plane in DPU coordinates.
Requirements:

R6: Sky coordinate epoch correction parameters.

R7: Resolution of trajectory linearization.

R12: Different position of optical axis, with or without dichroic in the optical path.

Basic path:

1: LFFTS master control provides pointing of the telescope.

2: LFFTS master control provides sky coordinates of reference star.

3: LFFTS master control provides plate scale.

4: LFFTS master control provides optical axis position.

5: Optional: LFFTS master control provides eventual PSF position offset from acquisition image.

6: LFFTS master control provides reference time.

7: Determine distance and parallactic angle at reference time of reference star with respect to the telescope pointing sky coordinate. Consider R6.

8: Determine parallactic angle velocity as a function of zenith distance.

9: Calculate PSF trajectory in DPU coordinates from reference time on. Consider R7 and R12.

10: Return PSF trajectory.

11: Done.

6.1.4 Use case: synchronize baffle / DPU motion

Description:
Controls motion of DPU. Assures that the Moving Baffle and DPU move synchronously.
Postconditions:
Synchronous motion of Baffle and DPU according to provided trajectory.
Requirements:

R2: NTP operating. Synchronization with instrument / telescope time.

Basic path:

1: Caller provides DPU trajectory.

2: Decompose DPU trajectory into linear stage motion profiles.

3: Determine Moving Baffle motion profile from DPU trajectory.

4: Upload profiles to DPU motor controller and Moving Baffle motor controller

5: Start synchronous motion

6: Periodically determine actual positions of DPU and Moving Baffle.

7: Calculate deviation from DPU nominal trajectory.

8: Calculate correction command sequence.

9: Upload correction command sequence into motor controller buffer.

10: If not cancelled, continue with step 6.

11: If cancelled, stop motors.

12: Done.

6.1.5 Use case: provide current DPU position

Postconditions:
Provides current DPU position
Basic path:

1: Fringe and Flexure Tracking requests current DPU position.

2: Request actual position of DPU from motor controller.

3: Provide current DPU position with time stamp.

4: Done.

6.1.6 Use case: move to home position

Postconditions:
Moves DPU to the center position and re-initializes the incremental encoder reference.
Basic path:

1: LFFTS master control provides DPU coordinate origin as DPU coordinates.

2: reposition as described in use case 6.1.1.

3: Submit 'MoveFindHome' command to DPU motor controller for each of the axes.

4: Submit 'MoveFindHome' command to Moving Baffle motor controller.

5: Done.

6.2 LFFTS fringe and flexure tracking

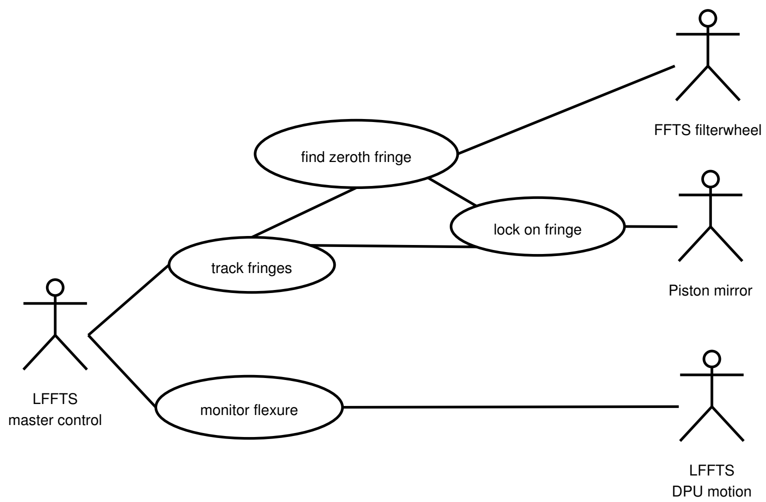

Figure 11: Use case diagram: fringe and flexure tracking

6.2.1 Use case: track fringes

Description:
The main fringe tracking and OPD control loop.
Preconditions:

- PSF tracing is active.
- frame acquisition / image processing pipeline is active.

Basic path:

1: LFFTS master control provides threshold value and tolerable number of frames without zero OPD.

2: Activate 'lock on fringe' as described in UC 6.2.3.

3: Find zeroth fringe as described in UC 6.2.2.

4: send message: 'Fringe tracking active'.

5: Get differential piston, S/N, fringe contrast from piston analysis of next frame, as described in UC 6.2.3.

6: Calculate average. fringe contrast / performance parameters.

7: Store parameters.

8: If differential piston is larger than the threshold value for more than n frames: send message 'lost zeroth fringe' and continue with 3.

9: Else continue with 5.

Alternate path A:

A.1: If LFFTS master control orders to stop fringe tracking, stop 'lock on fringe', described in UC 6.2.3.

A.2: Done.

6.2.2 Use case: find zeroth fringe

Description:
Scans for the zeroth fringe.
Postconditions:
'lock on fringe' locks on zeroth fringe.
Requirements:

R8: Calibrated point of zero OPD for piston mirror.

Basic path:

1: Continuously receive 'lock on fringe' (UC 6.2.3)fitting parameters.

2: If for several frames (TBD) 'lock on fringe' was not successful, continue with A.1.

3: Adjust user specified piston offset in sub-λ steps in positive direction.

4: if fringe contrast decreases, adjust user specified piston offset in sub-λ steps in negative direction.

5: Determine center of fringe envelope function.

6: Adjust user specified piston offset to this center.

7: Have FFTS filter wheel change the filter to broader bandwidth.

8: Continue with 2 until broadest filter is reached.

9: Done.

Alternate path A:

A.1: Have FFTS filter wheel change the filter to narrow bandwidth.

A.2: Position piston mirror at calibrated point of zero OPD (R8).

A.3: Scan with piston mirror in positive direction.

A.4: If 'lock on fringe' (UC 6.2.3) provides successful fitting results, continue with 3.

A.5: Position piston mirror at calibrated point of zero OPD (R8).

A.6: Scan with piston mirror in negative direction.

A.7: If 'lock on fringe' (UC 6.2.3) provides successful fitting results, continue with 3.

A.8: If 'lock on fringe' (UC 6.2.3) was not successful in fitting, send message: 'Cannot find fringes'.

A.9: Done.

6.2.3 Use case: lock on fringe

Description:
Analyzes incoming PSF frames and controls OPD by moving the piston mirror. Determines differential piston and diagnostic parameters.
Requirements:

R9: Lookup table of static misalignment of single eye PSFs due to optical errors.

Basic path:

1: Wait for message 'new frame available' from frame acquisition / image processing pipeline.

2: Copy frame out of shared memory.

3: Apply PSF fitting routine. Retrieve differential piston and fringe contrast. Consider R9.

4: Determine S/N.

5: Provide parameters to caller.

6: If PSF fitting fails, send message: 'fit failed' and continue with 1.

7: If differential piston with a value greater than λ cannot clearly be identified (narrow bandwidth), calculate differential piston modulo λ.

8: Get gain.

9: Get user specified piston offset.

10: Calculate nominal differential piston modulo λ, which is expected from user specified piston offset.

11: Calculate correction signal to be sent to piston mirror from gain, user specified piston offset and determined differential piston.

12: Send correction signal to piston mirror.

13: Continue with 1.

Alternate path A:

A.1: If caller requests change of gain, adjust gain.

A.2: continue with 1.

Alternate path B:

A.1: If caller requests change of user specified piston offset, adjust this value.

A.2: continue with 1.

Alternate path C:

A.1: If caller requests to stop: Done.

6.2.4 Use case: monitor flexure

Description:
Obtains the same frames as 'lock on fringe' UC 6.2.3. Averages them to increase S/N. Analyzes centroid of PSF and shape in terms of flexure effects.

Requirements:

R9: Lookup table of static misalignment of single eye PSFs due to optical errors.

Basic path:

1: LFFTS master control provides PSF trajectory.

2: Wait for message 'new frame available' from frame acquisition / image processing pipeline.

3: Copy frame out of shared memory.

4: Add frame to running average to increase S/N.

5: Determine S/N.

6: If S/N is not sufficient (TBD) increase number of frames to be averaged over.

7: Apply centroid fit to running average.

8: Compare centroid position with position expected from PSF trajectory. Consider R9.

9: If the deviation is larger than a threshold value, send message 'Deviation from nominal PSF trajectory'.

10: Provide deviation vector.

11: Apply 2D fit to PSF. Analyze alignment of single eye PSFs. Consider R9.

12: If the deviation is larger than a threshold value, send message 'Misalignment of single eye PSFs'.

13: Provide misalignment vector.

14: Continue with 2

Alternate path A:

A.1: If LFFTS master control requests stop: Done.

6.3 LFFTS image acquisition

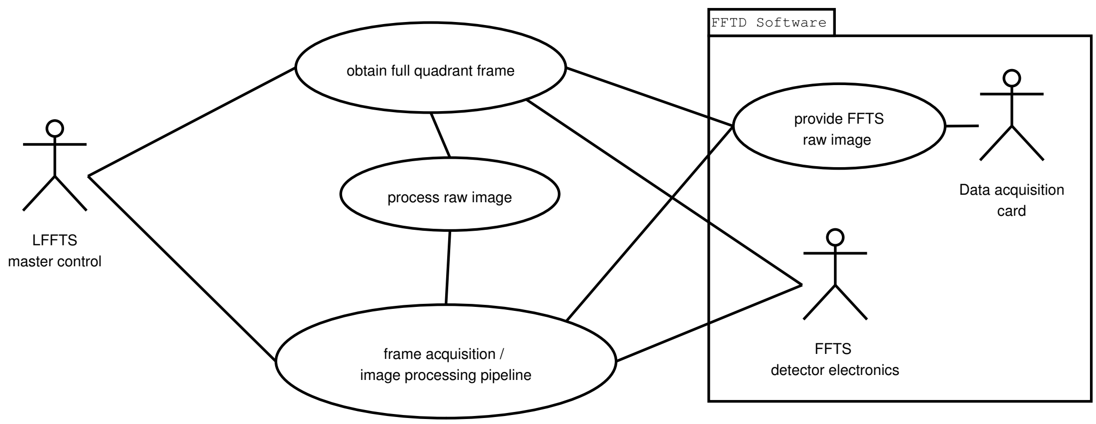

Figure 12: Use case diagram: image acquisition

6.3.1 Use case: acquire full quadrant image

Description:
Controls camera to obtain one full quadrant image. Dark and flat field are applied and the resulting frame is returned.
Basic path:

1: LFFTS master control provides exposure time.

2: Submit frame size and exposure time to FFTS detector electronics.

3: Start camera (submit ‘start’ to FFTS detector electronics).

4: Wait for message ‘new raw frame available’ (cf. UC 6.3.3).

5: Stop camera.

6: Get semaphore to access shared memory.

7: Get frame out of shared memory.

8: Release semaphore.

9: Process raw image as described in UC 6.3.4.

10: Provide processed image.

11: Done.

6.3.2 Use case: frame acquisition / image processing pipeline

Description:
Starts camera with subwindow readout at high frame rate. Continuously processes subwindow frames and provides result.
Basic path:

1: LFFTS provides exposure time.

2: LFFTS provides subwindow on detector, in which PSF is sampled.

3: Submit frame position, size and exposure time to FFTS detector electronics.

4: Start camera (submit ‘start’ to FFTS detector electronics).

5: Wait for message ‘new raw frame available’ (cf. UC 6.3.3).

6: Get semaphore to access shared memory.

7: Get frame out of shared memory.

8: Release semaphore.

9: Process raw image as described in UC 6.3.4.

10: Provide processed image in a different shared memory area.

11: Send message: ‘new frame available’.

12: Continue with 5.

Alternate path A:

A.1: LFFTS master control orders to stop.

A.2: Stop camera.

A.3: Done.

6.3.3 Use case: provide FFTS raw image

Description:
This use case is related to the FFTS detector software package. It will allocate shared memory, communicate the address of this memory, provide incoming raw frames from the FFTS detector and notify, once a new frame is available.
Basic path:

1: Allocate shared memory for N full quadrant images.

2: Communicate shared memory address to frame receiving tasks.

3: Read incoming frame from Data acquisition card.

4: Reserve semaphore to access shared memory.

5: Write raw frame to i'th position in the shared memory.

6: Send message: 'new raw frame available'.

7: Release semaphore.

8: Increase i. If shared memory is full, start again with i=0.

9: Continue with 3.

6.3.4 Use case: process raw image

Postconditions:
Flat dark and flat field corrected image.
Requirements:

R10: Detector dark

R11: Detector flat

Basic path:

1: Caller provides frame.

2: Subtract detector dark (R10).

3: Apply flat field (R11).

4: Provide processed frame.

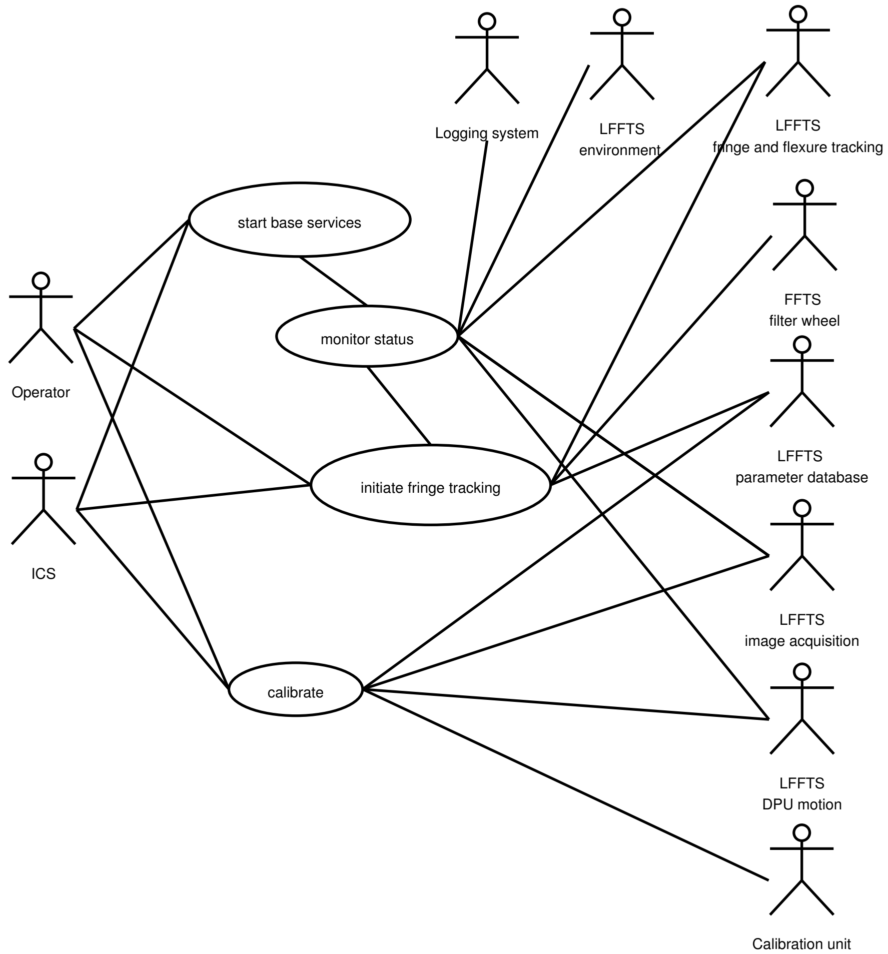

Figure 13: Use case diagram: master control

6.4 LFFTS master control

6.4.1 Use case: start base services

Description:
Contains all basic tasks that must be available as soon as LFFTS is started, such as the logging system, monitoring system, LFFTS parameter database, and alarm system.
Basic path:

1: ICS or operator starts LFFTS module.

2: Start alarm system.

3: Start logging system.

4: Initialize LFFTS parameter database.

5: Start monitoring as described in UC 6.4.2

6.4.2 Use case: monitor status

Description:
Provides status of each LFFTS subsytem, monitors all important parameters, communicates status to operator and ICS.

Basic path:

1: Get status of each LFFTS subsystem.

2: If active, catch messages of LFFTS fringe and flexure tracking.

3: If message 'fringe tracking active' or 'lost zeroth fringe' is received, notify ICS.

4: If fringe tracking is active get FT quality parameters (fringe contrast, S/N, correction variability).

5: If flexure monitoring is active, get current deviations.

6: Get temperature of FFTS mechanics and FFTS detector from LFFTS environment.

7: If parameters exceed threshold values, send warning to ICS.

8: Update user interface with current status and parameters.

9: Write all messages, warnings, errors to logging system.

10: Stop fringe tracking, if required subsystems fail to operate.

11: Continue with 1.

6.4.3 Use case: initiate fringe tracking

Description:
Assures that all required parameters are available and that the DPU follows the PSF. Chooses fringe tracking filter assembly. Initiates fringe tracking
Basic path:

1: Operator or ICS provides fringe tracking reference star coordinate.

2: Operator or ICS provides fringe tracking reference star magnitude.

3: Operator or ICS provides fringe tracking central wavelength.

4: Request pointing of telescope from LTCS. Alternatively Operator or ICS provides future pointing of telescope (to start repositioning DPU while telescope is moving to final position).

5: Make sure that all calibration parameters (plate scale, optical axis position, single eye optical axes alignment) are available. Either obtain them from Operator or ICS or calibrate them yourself (cf. UC 6.4.4).

6: Get fast readout subwindow position and size from parameter file.

7: Calculate PSF trajectory (cf. UC 6.1.3).

8: Reposition detector (cf. UC 6.1.1).

9: Execute 'follow PSF' as described in UC 6.1.2, even if PSF is not actually there (e.g. telescope not at final position).

10: Move FFTS filter wheel to filter assembly that corresponds to the fringe tracking central wavelength.

11: Operator or ICS request fringe tracking (once telescope is in place and other (AO) subsystems are operating).

12: Calculate exposure time for full quadrant image.

13: Acquire full quadrant image (cf. UC 6.3.1)

14: Determine centroid of PSF.

15: Calculate deviation from calculated PSF trajectory.

16: Adjust PSF trajectory, given the actual PSF position on the detector and the central position of the detector fast readout subwindow.

17: Update PSF trajectory for UC 6.1.2.

18: Calculate exposure time for frame acquisition / image processing pipeline.

19: Start frame acquisition / image processing pipeline (cf. 6.3.2).

20: Track fringes (cf. UC 6.2.1).

21: Monitor flexure (cf. 6.2.4).

22: Wait for cancellation by operator, ICS or status monitoring.

23: Stop monitoring flexure.

24: Stop tracking fringes

25: If a new pointing is expected, continue with 1. Otherwise, move to home position (cf. 6.1.6)

26: Done.

6.4.4 Use case: calibrate

Description:
Performs various calibration steps. This use case subsumes several calibration use cases. See figure 2, figure 3 and figure 4 for an outline of the calibration steps.