Upgrade Adaptive Optics (UAO) development plan

Prepared by: Marco Xompero, Alfio Puglisi, Enrico Pinna, Armando Riccardi, Simone Esposito

Approved by: 

Released by: 

ABSTRACT

The UAO project is intended to improve the reliability of the FLAO adaptive systems of the LBT telescope. The project has two main tasks: (a) the refurbishment of software and firmware of the ASM to reduce all the overheads caused by the failure events and by skip frame events and therefore minimize the system downtime (b) the upgrade of the finite state machine of the FLAO system to improve flexibility and efficiency of AO system operations on sky.

This document is based on the outputs coming from various meeting between LBTO and the Arcetri AO group and is intended to suggest a possible roadmap for both tasks listed above. In particular, the ASM work is derived from a quantitative and qualitative analysis of failure events occurred to the ASM DX side.
## Modification Record

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Section/Paragraph affected</th>
<th>Reason/Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>03 Jul 2015</td>
<td>Marco Xompero</td>
<td>All</td>
<td>First release of the document</td>
</tr>
<tr>
<td>1.1</td>
<td>01 Dec 2015</td>
<td>Marco Xompero</td>
<td>All</td>
<td>Update after ASM unit recovery and Microgate firmware schedule</td>
</tr>
<tr>
<td>1.2</td>
<td>20 Dec 2015</td>
<td>M. Xompero, A. Puglisi</td>
<td>All</td>
<td>Add section on FSM development</td>
</tr>
<tr>
<td>1.3</td>
<td>12 Jan 2016</td>
<td>All</td>
<td>All</td>
<td>Review of document consistency</td>
</tr>
<tr>
<td>1.4</td>
<td>19 Feb 2016</td>
<td>A. Puglisi</td>
<td>All</td>
<td>Revised deadlines</td>
</tr>
<tr>
<td>1.5</td>
<td>25 Mar 2016</td>
<td>M. Xompero, A. Puglisi</td>
<td>All</td>
<td>Revised deadlines. Expanded task description</td>
</tr>
</tbody>
</table>
## Abbreviations, acronyms and symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBT</td>
<td>Large Binocular Telescope</td>
</tr>
<tr>
<td>ICC</td>
<td>Control current leakage integrator</td>
</tr>
<tr>
<td>ASM</td>
<td>Adaptive secondary mirror</td>
</tr>
<tr>
<td>UAO</td>
<td>Upgrade adaptive optics</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Machine</td>
</tr>
</tbody>
</table>
Contents

1 ASM robustness improvement 5
   1.1 Summary.......................................................................................................................... 5
   1.2 Recovery from failure procedure (Task A)........................................................................ 6
   1.3 Skip frame algorithm analysis and modifications (Task B)................................................. 6
   1.4 ASM Failure sources.......................................................................................................... 6
   1.4.1 RIPTimeoutTwiceEvent (35.8%).................................................................................... 7
   1.4.2 RIPEmergencyStopDistEvent (19.7%) (Task C and CL)................................................ 7
   1.4.3 Over-temperature events (5.9%) (Task E)......................................................................... 8
   1.4.4 Unknown events (28.0%) (Task D).................................................................................. 8
   1.4.5 RIPSwingArmEvent (4.0%)........................................................................................... 8
   1.4.6 RIPSwingArmEvent (2.2%) (Task F).............................................................................. 8
   1.5 Test for validation of Task B, C and CL............................................................................ 8
   1.5.1 Test ID#01SL: capacitive sensors jump and recovery (Task C)..................................... 8
   1.5.2 Test ID#01CL: capacitive sensors jump and recovery during closed loop (Task CL)..... 9
   1.5.3 Test ID#02: skip frame recovery from position (Task B).............................................. 9

2 State machine (FSM) update 11
   2.1 Automatic and Intervention mode...................................................................................... 11
   2.2 FSM diagram..................................................................................................................... 11
   2.3 Snapshots.......................................................................................................................... 12
   2.4 PSF calibrator mode.......................................................................................................... 12
   2.5 Skip frame / shell RIP....................................................................................................... 13
   2.6 FSM tasks.......................................................................................................................... 13
   2.6.1 New sequence definition (Task G).................................................................................. 13
   2.6.2 Review and update of existing commands (Task H)...................................................... 13
   2.6.3 AOS upgrade (Task I)................................................................................................... 13
   2.6.4 AO Arbitrator implementation (Task J).......................................................................... 13
   2.6.5 New operator GUI (Task K).......................................................................................... 13

3 Current status on tasks and schedule 13

4 APPENDIX 17
   4.1 Stress analysis.................................................................................................................... 17
   4.2 ASM Recovery procedure.................................................................................................. 18
   4.3 FSM Action descriptions.................................................................................................... 20
   4.3.1 Preset AO...................................................................................................................... 20
   4.3.2 Center star...................................................................................................................... 20
   4.3.3 Center pupils.................................................................................................................. 20
   4.3.4 Check flux....................................................................................................................... 20
   4.3.5 Close loop...................................................................................................................... 20
   4.3.6 Optimize gain................................................................................................................. 20
   4.3.7 Apply optical gain/NCPA.............................................................................................. 20
   4.4 Minutes of meeting 3 Jul 2015........................................................................................ 21

1. Main discussion topics 21
   1.1 Removal of FASTDIAGNOSTICS checks......................................................................... 21
   1.1.1 Introduction and Rationale............................................................................................ 21
   1.1.2 Proposal....................................................................................................................... 21
   1.1.3 Notes............................................................................................................................. 21
   1.2 Provide ASM fault condition directly from ASM electronics........................................ 22
   1.2.1 Introduction and Rationale............................................................................................ 22
1.2.2. Proposal....................................................................................................................................................... 22

1.2.3. Notes........................................................................................................................................................... 22

1.3. Safety hardware................................................................................................................................................... 22

1.3.1. Introduction and Rationale........................................................................................................................... 22

1.3.2. Proposal........................................................................................................................................................... 22

1.4. Automatic detection of faulty actuators ............................................................................................................. 23

1.4.1. Introduction and Rationale........................................................................................................................... 23

1.4.2. Proposal........................................................................................................................................................... 23

1.5. Optimized reload and restart after RIP............................................................................................................. 23

1.5.1. Introduction and Rationale........................................................................................................................... 23

1.6. Skip frame recovery............................................................................................................................................ 23

1.6.1. Introduction and Rationale........................................................................................................................... 23

1.6.2. Notes............................................................................................................................................................... 24

1.6.3. Proposal........................................................................................................................................................... 24

2. Additional discussion topics .................................................................................................................................... 24

3. Action items .......................................................................................................................................................... 24

4. Proposed firmware modifications (MG).................................................................................................................. 25

5. Proposed software modifications (Arcetri) ........................................................................................................... 25

5 References ............................................................................................................................................................. 26

1 ASM robustness improvement

1.1 Summary

During the meeting held in Arcetri the 03th Jul 2015 between LBTO and INAF, the general issue of the ASM failures source identification has been addressed, for both closed loop and seeing limited operations. The contributors to the discussion were LBTO (G. Brusa), MICROGATE (R. Biasi, M. Andrighettoni), INAF (A. Riccardi, S. Esposito, M. Xompero, R. Briguglio, A. Puglisì, E. Pinna)

The output was a list of actions to be taken from Arcetri, Microgate and LBTO and part of them are reported in the document as working steps.

We analyzed also the behavior of the ASM when the unit reaches the coil current limits or shell position limits during closed loop operations (triggering a “skip frame” interrupt) and we identified a possible upgrade.

1.2 Recovery from failure procedure (Task A)

The ASM recovery procedure is a mandatory process that occurs every time the system detects a failure and it should put the system back in operating as fast as possible. In the past the procedure was not tolerant about the concurrent failures and the result was the system hanged up for long time before correctly back up to an operating state.

The new procedure agreed with LBTO (G. Brusa) should speed up the analysis and guarantee a sufficient set of diagnostics data that will drive the procedure behavior. See section 4.2 for details.

In particular we have split the activity in the following subtasks:

1. Test on the hardware of the new recovery procedure scheme.

2. Night time test validation is required into a technical night to check for bugs during telescope operations. It can be added as a generic contingency during other test operations.
1.3 Skip frame algorithm analysis and modifications (Task B)

According to the observing run experience, the skip frame algorithm is weak in the recovering from position or current boundary conditions. The main reason is supposed to be held in the control current leakage integrator inside the firmware actuator control loop so a new algorithm has been suggested. The practical effect is an observation stop and a long overhead time is needed to resume back because the mirror “freezes” itself in skip frame conditions and becomes unresponsive. The steps identified to address this issue are:

- Microgate firmware. Implement new skip algorithm by moving the contribution of the Integrator of Control Currents (ICC) outside of the force skip frame checks and apply a digital saturation to the ICC values themselves in order to avoid force divergence. The check will be done only taking into account the static forces and the feed-forward forces.
- Arcetri. Analyze, if possible, the maximum ICC contribution in terms of current (nominally set around 0.2 N) for quasi-static tracking and wind correction. On AOSupervisor, limit fast diagnostics process actions

In particular we have split the activity in the following subtasks:

1. Development and test of firmware to separate control current integrator from skip frame algorithm
2. FastDiagnostics process update to dealt with the new ICC saturation: monitor the values and log the activity.
3. Update of IDL software to cope with firmware modifications. Set up new skip frame thresholds.
4. Analyze the telemetry data to quantify the threshold value of ICC due to the wind contribution.
5. Functionality test of software and firmware. (see Test ID#02)
6. Night time testing

1.4 ASM Failure sources

Preliminary analyses on the failure events occurred during night time were summarized in a table. It has been performed only for the DX unit, taking into account one year of ASM use (June 2014 – May 2015).

The columns states are the three main states of the ASM plus one “Uncategorized”. The events of this last column were not associated to any telemetry dump so the state for them is still unknown.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>REST</th>
<th>SET</th>
<th>CLOOP</th>
<th>UNCAT</th>
<th>%</th>
<th>% (no REST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIPAdamTCSPowerFaultEvent</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>RIPAdamWatchdogExpEvent</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>RIPElevationEvent</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>31</td>
<td>2.9</td>
<td>4.0</td>
</tr>
<tr>
<td>RIPEmergencyStopDistEvent</td>
<td>96</td>
<td>50</td>
<td>89</td>
<td>30</td>
<td>22.3</td>
<td>19.7</td>
</tr>
<tr>
<td>RIPSwingArmEvent</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>1.6</td>
<td>2.2</td>
</tr>
<tr>
<td>RIPTimeoutTwiceEvent</td>
<td>31</td>
<td>2</td>
<td>9</td>
<td>296</td>
<td>28.5</td>
<td>35.8</td>
</tr>
<tr>
<td>Over-temperature</td>
<td>22</td>
<td>25</td>
<td>26</td>
<td>0</td>
<td>6.1</td>
<td>5.9</td>
</tr>
<tr>
<td>Pie Shape</td>
<td>126</td>
<td>24</td>
<td>8</td>
<td>0</td>
<td>13.3</td>
<td>3.7</td>
</tr>
<tr>
<td>Unknown</td>
<td>19</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>12.4</td>
<td>14.9</td>
</tr>
<tr>
<td>Unknown – RIPEmergencyStopCurrEvent</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>0.9</td>
<td>1.3</td>
</tr>
<tr>
<td>Unknown - CNT_NOT_SYNC</td>
<td>22</td>
<td>0</td>
<td>36</td>
<td>0</td>
<td>4.9</td>
<td>4.2</td>
</tr>
<tr>
<td>Unknown – POSCHECK</td>
<td>11</td>
<td>64</td>
<td>1</td>
<td>0</td>
<td>6.4</td>
<td>7.6</td>
</tr>
</tbody>
</table>
We started the analysis under some hypothesis:

- We exclude the “PIE Shape” event, which was generated by hardware failures and that LBTO addressed in the past
- We consider the “Unknown” events all together
- We not address the event RIPAdamTCSPowerFaultEvent and RIPAdamWatchdogExpEvent because not relevant in quantity
- Moreover, instead of focus on the overall percentage, we are focusing on the events that were explicitly stopping an observation during the nighttime

We took all the relevant parameters left and we scheduled a set of action in order to directly address the source each failure.

As improvement for the future analysis, LBTO (G. Brusa) suggested to continue the failure analysis software considering a monthly time slicing. This activity should be continued by LBTO. This possible improvement is not part of the UAO framework.

### 1.4.1 RIPTimeoutTwiceEvent (35.8%)

This kind of event is a shell rest forced by the AOSupervisor software. The source at the software level is the lack of diagnostics frame coming from the system in a large time span (currently 2 seconds).

The system firmware deadlocks itself in some internal step. The effect, detected by the supervisor software, is the cited lack of diagnostics frame.

This stop of diagnostics frame flow has been noticed to be strongly correlated with particular timeout events on Ethernet packets and it has been observed on the Switch BCU and on the Crate BCU #5 (so called “CA timeout event”)

Roberto and Mario are strongly convinced that the source is due to the power supply of the Switch BCU and agree with LBTO (G. Brusa) to address the issue during the summer shutdown. No action was required on Arcetri side.

### 1.4.2 RIPEmergencyStopDistEvent (19.7%) (Task C-SL and C-CL)

This event is corresponding to a jump in the capacitive sensor reading and both AOSupervisor software and the firmware reacts forcing the shell to rest state. This event is triggered currently even if a single actuator is giving bad readings in a single occurrence and it has to be avoided.

In order to increase the system robustness, action will be taken to all software and firmware level:

- Microgate firmware. Currently a single actuator capacitive sensor jump can cause the shell rest. The suggested modification is to increase this number to 4 actuators of the same DSP in order to maintain the safety of the shell and in the meanwhile to avoid sporadic events to stop the observation. See appendix 4.4 from G. Brusa for approval of this modifications for LBTO. See also the appendix 4.1 for a stress analysis confirming the safety of this implementation.
- Arcetri software. Remove the thresholds on the force checks performed by FastDiagnostics process. Design and implement an algorithm to deal with jumping actuators during seeing limited (C-SL) and during closed loop (C-CL) operations. Fix the Zernike on FastDiagnostics computation and limit the number of data dump bunches.

The number of connections between firmware and software is very large and not yet deeply clarified, mainly in the AO closed loop state. The implementation of the algorithm for the automatic disabling of bad actuators while the ASM is fully working in optical closed loop is considered as a goal task.

In particular we have split the activity of C task in the following subtasks:

1. Development and test of firmware to modify the automatic mirror rest for capacitive sensors reading jumps.
2. Update FastDiagnostics to skip the bad capacitive sensor readings on ofload computations.
3. Development and test the procedure for automatic bad actuator removal without mirror rest during seeing limited operations. See TestID#01SL.
4. Update the Arbitrator to trigger the procedure of bad actuator removal.

5. Night time functionality check and behavior monitoring

The task CL has a similar activity partition:

1. Development and test the procedure for automatic bad actuator removal without mirror rest during closed loop operations. See TestID#01CL.
2. Update the Arbitrator to trigger the procedure of bad actuator removal
3. Night time functionality check and behavior monitoring

1.4.3 Unknown events (28.0%) (Task D)

A large fraction of events couldn’t be assigned to a specific source. The main issue is inside the complexity of the system and inside the complexity of the interactions between hardware and software. The steps identified to address this issue are:

- Arcetri software: upgrade telemetry for saving continuously even the bad actuator values. Limit the number of master Diagnostic data dump.
- Arcetri doc. Deliver a table with all the possible events, which force a shell rest.
- Microgate doc. Update the design document inserting a table with all the firmware source of failure and if they are latched or not. Analyze the feasibility of firmware modification on this side.

Microgate found hardware issues in the extension of the diagnostics data. In particular some component is limiting the possibility to latch events and transfer them to the master diagnostics packets. No firmware modification will be implemented on this side.

In particular we have split the activity in the following subtasks:

1. Deliver the table of hardware faults.
2. Modify FastDiagnostics telemetry in order to continuously save even the bad actuator values. Limit the number of master Diagnostic data dump.

1.4.4 Over-temperature events (5.9%) (Task E)

A not negligible amount of events are related to over-temperature detections.

So to address the problem we agree in the following action list:

- Microgate. Verify and if needed modify temperature thresholds
- Arcetri. Collect the current temperature thresholds set and update software configuration once received Microgate feedback
- LBTO. Since after a quick analysis it seems the cooling is not properly circulating in terms of flux. Providing the nominal flux to the unit is a task for LBTO

In particular we have split the activity in the following subtasks:

1. Send new housekeeper threshold table to LBTO
2. Support LBTO to update housekeeper thresholds

1.4.5 RIPElevationEvent (4.0%)

The elevation value is critical for the safety of the mirror so any lack of elevation or any detection under 30 elevation degrees is causing a rest of the shell. To address at once all the data-link issues that can cause false positive behaviors, LBTO started an hardware upgrade in order to link the elevation check to a value coming from an inclinometer directly mounted on the telescope.

LBTO (G. Brusa) will take care about investigating the current status of the inclinometer upgrade. This task therefore is not considered part of UAO framework.

INAF – Osservatorio Astrofisico di Arcetri
Largo E. Fermi, 5 - 50125 Firenze - ITALY
http://adopt.arcetri.astro.it
1.4.6 RIPSwingArmEvent (2.2%) (Task F)

This event has not been discussed but a bug was found in the AOS code and currently is fixed and ready for test in the next TCS build (IT#5625). The main source of the problem was a broken data link between AOS and the AOSupervisor software after an AOSupervisor full stop and start.

1.5 Test for validation of Task B, C-CS and C-CL

Hereafter the list of test to be performed accordingly the described items. All the tests can be performed daytime.

1.5.1 Test ID#01SL: capacitive sensors jump and recovery (Task C)

The aim of the test is verify the behavior of the mirror and of the software in the current configuration and test the main steps of the automatic recovery procedure that will be implemented.

Prerequisites: Telescope at zenith

Procedure:

1. Select an actuator ACT with high gains and ICC enabled
2. Set the mirror
3. Update the internal threshold counter of ACT (\textit{pos\_check\_cnt\_threshold}) to 2^{31}-1 to avoid automatic rest
4. Set FastDiagnostics thresholds of ACT to plus and minus Infinity (i.e. no thresholds)
5. Simulate the single actuator jump behavior on the final algorithm by modifying the \textit{C_{easy}} parameter
6. Detect ACT using the \textit{pos\_check\_counter\_cnt} value
7. Open the control loop on ACT
8. Reset the ICC value on ACT

Expected results:

- No shell sudden reset
- Easy actuator jump detection

1.5.2 Test ID#01CL: capacitive sensors jump and recovery during closed loop (Task CL)

The aim of the test is verify the behavior of the mirror and of the pyramid WFS software during the steps of the automatic recovery procedure that will be implemented.

Prerequisites: FLAO WFS powered up; retro-reflector holder; Telescope at zenith

Procedure:

1. Select an actuator ACT with high gains and ICC enabled
2. Set the mirror
3. Update the internal threshold counter of ACT (\textit{pos\_check\_cnt\_threshold}) to 2^{31}-1 to avoid automatic rest
4. Set FastDiagnostics thresholds of ACT to plus and minus Infinity (i.e. no thresholds)
5. Configure the WFS and close the loop with 400 modes
6. Simulate the single actuator jump behavior on the final algorithm by modifying the \textit{C_{easy}} parameter
7. Detect ACT using the \textit{pos\_check\_counter\_cnt} value
8. Null the proportional gain and the derivative gain on ACT
9. Reset the ICC value on ACT

Expected results:
- No shell sudden reset
- Stable steady state closed loop without crash

1.5.3 Test ID#02: skip frame recovery from position (Task B)

**Prerequisites:** FLAO WFS powered up; Telescope at zenith

**Procedure:**

1. Set the shell
2. Disable automatic loop aperture on skip frames
3. Put gain 0.001 and 10 modes reconstructor
4. Change black level of CCD quadrants to simulate a tilt
5. Go to a position bound and verify the skip frame due to position
6. Go back to a good position
7. Repeat the test with ICC disabled

**Expected results:**

The mirror should not be able to come back to the previous good position and should be “frozen” in skip frame state. After the firmware upgrade it should be able to recover and start to be responsive again after the large tilt condition is relaxed.
2 State machine (FSM) update

2.1 Automatic and Intervention mode

The new FSM GUI will have two main operating modes:

1. Automatic: everything goes on automatically as it happens now at the telescope. A preset either completes successfully, or fails. If a preset fail, or is cancelled before completing, the operator has to send another preset.

2. Intervention (manual). This mode has two sub-modes:
   a. Run-to-finish: (waiting for a better name from a native English speaker...) The preset executes automatically until an error occurs, or until the user presses the “Pause” button on the FSM GUI. In both cases, the FSM GUI will wait for operator intervention and will not send an error reply to the TCS unless the user presses the appropriate button.
   b. Step-by-step: the preset sequence is executed one step at a time, and at each step the software pauses and waits for operator intervention in the FSM GUI. The operator has to choose whether to go execute the next step, skip to a later step if allowed, change parameters, etc.

Switching between operating modes is possible at any time on the FSM GUI, even during the execution of a preset.

A “Stop” button is always available to stop the current step at the earlier possible opportunity. Also an “Abort preset” is available to abort the current preset and send an error reply back to TCS.

2.2 FSM diagram

This is the FSM diagram that we are going to implement. Shown on the diagram is the main sequence path, that is, the sequence of steps followed in Automatic mode, or in Intervention (Manual) mode if the user makes no changes.

Actions marked as “skippable” are those that are possible to skip in Intervention step-by-step mode: each action has a button, and to skip an action the user simply avoids pressing that button and presses instead the one for a later action. It should be stressed that, if any action is skipped, it is the user’s responsibility to make sure that all parameters are correct for the loop.

Not all actions are skippable, due to the need of synchronizing AO state with TCS state.

It is also possible to “go back”, that is, to select a button earlier in the sequence than the current one. In this case, after warning the user, the FSM GUI will roll the system state back to the selected point and resume execution. The FSM GUI will take care of proper TCS communication in order to avoid TCS errors.
2.3 Snapshots

After each step completes, a complete system snapshot is saved on disk. When in Intervention mode, the user can choose at any time to load a previously saved snapshot (in Run-to-finish mode, this requires to pause the preset sequence). This will restore all parameters saved in the snapshot, including the AO system state. The GUI will reflect the new sequence status and allow resuming the preset sequence. User can also save a snapshot at any time if desired.

This item has been de-scoped.
2.4 PSF calibrator mode

It is useful to repeat an AO loop keeping all parameters intact, except that a different target is pointed. In order to do this, an option is provided on the PresetAO step, where a previous snapshot can select as a reference. The sequence will go on as usual, but all parameters except stage positions and camera lens positions will be taken from the snapshot rather than optimized on sky.

This item has been de-scoped.

2.5 Skip frame / shell RIP

In Intervention mode, when the loop opens because of a “maximum skip reached” error, the AO state becomes ReadyForCloop and the FSM GUI allows the user to re-close the loop right away clicking on the “Close Loop” button. Re-optimizing the gain is optional, since the previous gains will still be loaded. As long as the user does not press the “Abort preset” button, TCS will not be informed that the loop has been opened.

In case of shell RIP, the same procedure can be followed, but this requires setting the shell. Currently (by choice) this is done by the T/O. Whether to allow an automatic re-setting of the shell by the FSM GUI is a policy decision to take by LBTO.

2.6 FSM tasks

The new FSM implementation has been divided into several sub-tasks.

2.6.1 New sequence definition (Task G)

The new sequence has to be defined in terms of defined states and possible actions for each state. Care should be taken in order not to disrupt the current sequence of operations as seen from the TCS/AOS.

2.6.2 Review and update of existing commands (Task H)

The new sequence will require some update to the commands made available by the AdSec and WFS subsystems, especially in the acquisition sequence. Currently multiple interfaces exist (the old “Arbitrator library” and the ICE interfaces used by ARGOS and LN), that overlap partially but not completely. The task goal is to identify the complete list of current commands and produce a single command set that can satisfy all functionality requirements in a new ICE interface.

2.6.3 AOS upgrade (Task I)

While no functionality is implemented in the AOS from the operational point of view, the software interface is changed from the old Arbitrator library to a new ICE interface.

In particular we have split the activity in the following subtasks:

1. Test of the new ICE interface in simulated environment
2. Daytime functionality tests

2.6.4 AO Arbitrator implementation (Task J)

The new FSM will require a complete rewrite of the AO Arbitrator, which is the top-level software layer coordinating the various AO subsystems. A python implementation is foreseen, using publicly-available tools for code generation whenever possible (e.g. State Machine Compiler, or SMC).

In particular we have split the activity in the following subtasks:

1. State machine implementation with simulated environment tests
2. Daytime functionality tests
3. On-sky functionality tests

2.6.5 New operator GUI (Task K)

A new operator GUI will be available on the AO workstation in order to enable Intervention mode. This GUI will use the same ICE interface that the AOS is using to communicate with the new AO Arbitrator.
In particular we have split the activity in the following subtasks:

1. Implementation of new GUI, with simulated environment tests
2. Daytime functionality tests
3. On-sky functionality tests.

### 3 Current status on tasks and schedule

In the following table we have the summary of the mentioned tasks.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>People</th>
<th>Completion percentage</th>
<th>Missing part</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Recovery procedure: implement new recovery procedure and notify the operator in case of fault</td>
<td>Marco, Alfio</td>
<td>Arcetri 70% Microgate n/a</td>
<td><strong>Arcetri</strong>: debug and test daytime; test nighttime</td>
</tr>
<tr>
<td>B</td>
<td>Skip frame test, firmware upgrade and software configurations</td>
<td>Marco, Alfio, Lorenzo Mario</td>
<td>Arcetri 70% Microgate 80%</td>
<td><strong>Arcetri</strong>: analysis of wind contribution; FastDiagnostics behaviors, test of Microgate firmware</td>
</tr>
<tr>
<td>C-SL</td>
<td>Capacitive sensor jump new algorithm: firmware and software modification for SL operations</td>
<td>Marco, Alfio, Luca, Mario</td>
<td>Arcetri 50% Microgate 80%</td>
<td><strong>Arcetri</strong>: debug and test daytime; test night time. FastDiagnostics update. CL algorithm to be confirmed. Arbitrator update for monitoring. Test of Microgate firmware</td>
</tr>
<tr>
<td>C-CL</td>
<td>Capacitive sensor jump new algorithm: firmware and software modification for CL operations</td>
<td>Marco, Alfio, Luca, Mario</td>
<td>Arcetri 50% Microgate 80%</td>
<td><strong>Arcetri</strong>: debug and test daytime; test night time. FastDiagnostics update. CL algorithm to be confirmed. Arbitrator update for monitoring. Test of Microgate firmware</td>
</tr>
<tr>
<td>D</td>
<td>Investigation of unknown source of failures</td>
<td>Marco, Lorenzo Gianluca Alfio</td>
<td>Arcetri 50% Microgate 70%</td>
<td><strong>Arcetri</strong>: modify telemetry of FastDiagnostics process. Continue improvements on faults notifications. <strong>Microgate</strong>: deliver the table with hardware faults</td>
</tr>
<tr>
<td>E</td>
<td>Update temperature thresholds</td>
<td>Marco</td>
<td>Arcetri 90% Microgate 0%</td>
<td><strong>Arcetri</strong>: support LBTO to update housekeeper thresholds. <strong>Microgate</strong>: send new thresholds parameters</td>
</tr>
<tr>
<td>F</td>
<td>Swing arm bug fixing</td>
<td>Luca</td>
<td>Arcetri 100% Microgate n/a</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>FSM New sequence definition</td>
<td>Alfio, Enrico</td>
<td>Arcetri 100%</td>
<td>Done</td>
</tr>
<tr>
<td>H</td>
<td>Review and update of existing commands</td>
<td>Alfio</td>
<td>Arcetri 80%</td>
<td>Split of the WFS “AcquireRef” command into multiple steps.</td>
</tr>
<tr>
<td>I</td>
<td>AOS upgrade</td>
<td>Luca</td>
<td>Arcetri 100%</td>
<td>Test of the new ICE interface</td>
</tr>
<tr>
<td>J</td>
<td>AO Arbitrator implementation</td>
<td>Luca, Gianluca, Alfio</td>
<td>Arcetri 90%</td>
<td>Implementation of new state machine, test with simulated environment.</td>
</tr>
<tr>
<td>K</td>
<td>New operator GUI</td>
<td>Francesco, Luca</td>
<td>Arcetri 80%</td>
<td>Implement GUI from mock-up</td>
</tr>
</tbody>
</table>
The first Microgate time slot available for the firmware development is set to early January 2016. We agreed that a first release of firmware ready-to-test will be delivered by 15th January 2016. Daytime test session (preferably split into multiple days) is required to assess the goodness of software and firmware modification before a night time test.

The tasks still pending have been then split into single activities covering the work still not completed to track the progresses.

<table>
<thead>
<tr>
<th>Task</th>
<th>Activity left</th>
<th>Who</th>
<th>Telescope request</th>
<th>Proposed deadlines Jan 12th</th>
<th>New deadlines Jan 27th</th>
<th>New deadlines Mar 23th</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A2</td>
<td>Arcetri</td>
<td>nighttime</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>A night time test validation is required into a technical night to check for bugs during telescope operations. It can be added as a generic contingency during other test operations.</td>
</tr>
<tr>
<td>B</td>
<td>B1</td>
<td>Microgate</td>
<td>-</td>
<td>15 Jan 2016</td>
<td>Done</td>
<td>Done</td>
<td>Development and test of firmware to separate control current integrator from skip frame algorithm</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>Arcetri</td>
<td>-</td>
<td>29 Jan 2016</td>
<td>26 Feb 2016</td>
<td>30 Apr 2016</td>
<td>FastDiagnostics process configuration and behavior modifications on variable alarms</td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>Arcetri</td>
<td>-</td>
<td>29 Jan 2016</td>
<td>Done</td>
<td>Done</td>
<td>Firmware driven IDL software modifications</td>
</tr>
<tr>
<td></td>
<td>B4</td>
<td>Arcetri</td>
<td>-</td>
<td>26 Feb 2016</td>
<td>26 Feb 2016</td>
<td>30 Apr 2016</td>
<td>Wind contribution analysis</td>
</tr>
<tr>
<td></td>
<td>B5</td>
<td>Arcetri</td>
<td>daytime</td>
<td>19 Feb 2016</td>
<td>18 Mar 2016</td>
<td>11 Apr 2016</td>
<td>Functionality test of software and firmware. See Test ID#02</td>
</tr>
<tr>
<td></td>
<td>B6</td>
<td>Arcetri</td>
<td>nighttime</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>C-SL</td>
<td>SL1</td>
<td>Microgate</td>
<td>-</td>
<td>15 Jan 2016</td>
<td>Done</td>
<td>Done</td>
<td>Development and test of firmware to modify the automatic mirror rest for capacitive sensors reading jumps.</td>
</tr>
<tr>
<td></td>
<td>SL2</td>
<td>Arcetri</td>
<td>-</td>
<td>29 Jan 2016</td>
<td>26 Feb 2016</td>
<td>30 Apr 2016</td>
<td>Update FastDiagnostics behaviors to manage bad capacitive sensor readings on offload computations.</td>
</tr>
<tr>
<td></td>
<td>SL3</td>
<td>Arcetri</td>
<td>daytime</td>
<td>19 Feb 2016</td>
<td>18 Mar 2016</td>
<td>11 Apr 2016</td>
<td>Test procedure for automatic bad actuator removal without mirror rest during seeing limited operations. See TestID#01SL.</td>
</tr>
<tr>
<td></td>
<td>SL4</td>
<td>Arcetri</td>
<td>-</td>
<td>26 Feb 2016</td>
<td>26 Feb 2016</td>
<td>30 Apr 2016</td>
<td>Arbitrator update and final implementation of bad actuator</td>
</tr>
<tr>
<td>C-CL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL5</td>
<td>Arcetri</td>
<td>nighttime</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Removal procedure without mirror rest. Integration in the AOSupervisor framework.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Night time functionality check and behavior monitoring.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL1</td>
<td>Arcetri</td>
<td>daytime</td>
<td>19 Feb 2016</td>
<td>18 Mar 2016</td>
<td>11 Apr 2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test procedure for automatic bad actuator removal without mirror rest during closed loop operations. See TestID#01CL.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL2</td>
<td>Arcetri</td>
<td>-</td>
<td>26 Feb 2016</td>
<td>26 Feb 2016</td>
<td>30 Apr 2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arbitrator update and final implementation of bad actuator removal procedure without mirror rest. Integration in the AOSupervisor framework.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL3</td>
<td>Arcetri</td>
<td>nighttime</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Night time functionality check and behavior monitoring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deliver the table of hardware faults</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>Arcetri</td>
<td>-</td>
<td>29 Jan 2016</td>
<td>26 Feb 2016</td>
<td>30 Apr 2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify FastDiagnostics telemetry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send new housekeeper threshold table to LBTO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E2</td>
<td>Arcetri</td>
<td>-</td>
<td>29 Jan 2016</td>
<td>26 Feb 2016</td>
<td>10 May 2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support LBTO to update housekeeper thresholds</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Arcetri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Arcetri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Arcetri</td>
<td>-</td>
<td>29 Jan 2016</td>
<td>26 Feb 2016</td>
<td>Done</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Split “AcquireRef” WFS command into multiple, smaller steps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Arcetri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test of the new ICE Interface in simulated environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1</td>
<td>Arcetri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daytime functionality tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>Arcetri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>State machine implementation with simulated environment tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daytime functionality tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>On-sky functionality tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daytime functionality tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation of new GUI, with simulated environment tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

INAF – Osservatorio Astrofisico di Arcetri
Largo E. Fermi, 5 - 50125 Firenze - ITALY
http://adopt.arcetri.astro.it
Note: daytime tasks I2, J2 and K2 are to be performed together, since most AO functionality requires a complete system to operate on.

Schedule Gantt diagram in the next page.
4 APPENDIX

4.1 Stress analysis

The new firmware modification allows three new conditions that can be potentially dangerous for the shell safety if we consider the total amount of mechanical stress which will be induced to the shell.

We set a force threshold to 0.8 [N] (the current saturation value) and we simulated the effect of all possible conditions in which can fail one or two or three actuators of the same DSP board.

The worst case for each analysis is reported in Figure 1.

![Maximum stress value and pattern for a failure of one, two or three actuators in a DSP board.](image)

With this threshold, we are about a factor three below the conservative value of 3MPa so in principle the threshold can be tuned if needed.

4.2 ASM Recovery procedure

In the following is detailed the recovery procedure as agreed with LBTO (G. Brusa). The mirror failures are classified from to recoverable or unrecoverable according to the source of the failure. In particular, after any mirror failure event that causes a sudden mirror rest, we will have:

- Save the last 1000 master diagnostics frames
• Force a housekeeper readings of slow diagnostics variable
• Perform the master diagnostics and housekeeper data analysis and correlation
• Check if the dump was caused by:
  o Power loss
  o Components over temperature
  o Communication error
  o Set of not working actuators

If positive, then mark the error as unrecoverable. Else mark it as recoverable and try to restore the system.

Appendix C Meeting minutes
4.3 FSM Action descriptions

NOTE: The precise definition of each step is still TBD. The following is a preliminary description of each step.

4.3.1 Preset AO

Very similar to the current preset:
- Filter wheels and CCDs are configured using lookup tables
- Backgrounds are taken on both ccd39 and ccd47
- XYZ stages are moved to the expected star position
- Pupil rotator and ADC tracking are started

4.3.2 Center star

- One or more frames are averaged on the ccd47 and the star located
- XYZ stages are moved to center the star on the ccd47 hot-spot

4.3.3 Center pupils

- AdSec is configured for centering loop
- All offloads are enabled
- Loop is closed and camera lens tracking is enabled
- Wait until camera lens is centered
- Loop is opened.

4.3.4 Check flux

- Flux on the ccd39 is measured
- System is reconfigured (frequency, binning, etc.) if the flux is different from the expected one
- If binning has changed, the previous step to center the camera lens is repeated

4.3.5 Close loop

- AdSec is configured for AO loop with low gain (0.1 on 10 modes)
- All offloads are enabled
- Loop is closed
- Wait until all offloads are under a threshold

4.3.6 Optimize gain

- Gain optimization routine is executed.

4.3.7 Apply optical gain/NCPA

- Continuous optical gain measurement is started
- NCPA correction is started.
 Minutes of meeting 3 Jul 2015

Discussion on Adaptive Secondary hardware, firmware and software upgrades

G. Brusa August 26, 2015 – v1.1

This short document reports the discussion held on July 3, 2015 in Arcetri concerning upgrades to the LBT Adaptive Secondaries to improve robustness, reliability and easiness of operation. Here is the list of participants: Mario Andrighettoni, Roberto Blasi, Runa Briguglio, Guido Brusa, Enrico Pinna, Alfio Puglisi, Armando Riccardi and Marco Xompero.

1. Main discussion topics

1.1. Removal of FASTDIAGNOSTICS checks

1.1.1. Introduction and Rationale

Currently there are two position-out-of-bounds checks working simultaneously: a fast (70 kHz) mechanism implemented in the DSP code and a slow (~100 Hz) mechanism implemented by one of the housekeeping process of the AdSec supervisor software (**fastdiagnostics**). This second mechanism implements a force out-of-bounds check as well. It is apparent that the second (slow) mechanism serves virtually no purpose while at the same time causing un-necessary RIP commands. We should retain the first (fast) mechanism only. The fast mechanism should also be modified to avoid un-necessary RIPs. At the moment in fact the mechanism is triggered by single -actuator position-out-of-bound events, which are often the result of position reading ‘jumps’ and not dangerous situations like large mirror stroke or large applied deformations.

1.1.2. Proposal

We propose to eliminate the **fastdiagnostics** check (~100Hz) for position and force and change the DSP code from checking one single actuator out of range (currently >135um) to all the four actuators managed by the same DSP. This change will provide protection against large strokes of the mirror while preventing a RIP from single actuator jump. To provide checks of forces (currently provided by the **fastdiagnostics**) we can add a check on all four actuators forces out of range (|F| >0.8N TBC – anyway it will be a configurable parameter) however, before proceeding with this additional modification we should verify that it is necessary.
1.1.3. Notes

- Armando found a plot of stress as a function of modal deformation that seems to show that the maximum stress on the thin shell is either limited by the maximum stroke (low order modes) or maximum force (high order modes), this seems to indicate that there is not need for force checks (AI1)

- we should keep in mind that after removing the use of fastdiagnostics we will only be using the housekeeping, which has a refresh rate of one second; this will become the maximum rate for feedback on the status of the system (safety will not be affected however)

- eliminating the fastdiagnostics check will have the additional benefit of allowing the saving of the telemetry of all the actuators, i.e. even the disabled one, which at the moment is not done.

1.2. Provide ASM fault condition directly from ASM electronics

1.2.1. Introduction and Rationale

At the moment the analysis of a RIP is done ‘post-facto’. To perform this analysis, telemetry of the system is saved and later used to infer the most likely cause of the event. The current situation has the drawback of not providing and immediate feedback to the operator; moreover because the information saved is incomplete in many cases it is not possible to determine the cause of the RIP event. The general idea for this task is to fix the current situation by: first organizing the knowledge on the system and then providing the necessary hardware, firmware or software modifications to be able to extract all the necessary information for the correct diagnosing of the RIP event.

1.2.2. Proposal

We will start by providing a list of faults conditions at the ASM level (AI2a) and a similar list of faults detected by the AO supervisor (AI2b), the next step will be to come up with a proposal of required modifications needed to achieve full detectability of the system’s status (AI2c). These modifications may be hardware (for instance if a fault conditions needs to be latched in the systems logic), firmware (logic modification) or software (Arcetri side).

1.2.3. Notes
• Can we make the overcurrent signal available thru the logic (see Al2c)?
• As a general reference document use 641a006e.

1.3. Safety hardware

1.3.1. Introduction and Rationale

We discussed several hardware issues still to be addressed.

1.3.2. Proposal

• add a protection against cooling temperature below dew point using a shut-off valve (see ‘freezing’ accident)
• replace the faulty flowmeter on DX using a model that is less prone to getting clogged, after adequate time to test the new model replace the SX flowmeter as well (AI13)
• increase the cooling flow coming from the instrument cooling system
• set in use the inclinometer
• make use of the cooling flow data from telemetry to alert the operator about a potential cooling issue (low flow).

See AI3.

1.4. Automatic detection of faulty actuators

1.4.1. Introduction and Rationale

With the elimination of the position and forces out-of-bounds checks from the supervisor side (see section 1.1.2) we will need to add a monitoring software process that keeps track of faulty actuators, mostly ‘jumping’ (position) actuators. This process will detect anomalous behavior and disable the actuators, first the control current integrator will be disabled (small jumps) then, if needed (large jumps), the local position loop will be opened. The disabled actuators will be kept under monitoring and will be re-enabled if the fault is recovered.

1.4.2. Proposal

There is no detailed proposal at this point, we discussed a possible scheme: use the data from fastdiagnostics (~100 Hz) to detect discrepancies between position and position command. In order to tweak the behavior of the automatic detection process we will have to ‘tweak’ the amount of delta that triggers the disabling (~100nm for control current integrator, ~1um for open loop) and decide the amount of time that this condition has to persist (or has disappeared) before the disabling (enabling) is triggered (see...
1.5. Optimized reload and restart after RIP

1.5.1. Introduction and Rationale

System restart (supervisor side) should be limited to the modules that were affected by the RIP, how can we identify what needs to be re-loaded on the firmware side, to avoid re-loading everything every-time? One suggestion is to first time-profile the system restart in order to identify the initializations requiring longest time. See AI5.

1.6. Skip frame recovery

1.6.1. Introduction and Rationale

How can we recover from skip frame? It seems that during skip frame the current keeps growing and the new predicted force is out of bounds so that no recovery is possible.

1.6.2. Notes

We can check the theory about the control current integrator being the culprit of increasing forces during skip frames by running the AdSec with control current integrator disabled. To check we can use a non-zero slope null vector and no optical feedback (see AI6).

1.6.3. Proposal

After verification as per AI6, MIC proposes to determine the skip condition only on base of the static forces + feed-forward.

2. Additional discussion topics

Marco shows a table with events of RIP, this is the beginning of a very detailed investigation, considering all the causes for RIPS and computing a statistics.

Alfio requests to have one day a month to perform tests on the AO system, it should be part of the Engineering tasks.

Enrico presented SOUL, the need to separate the work on robustness from necessary upgrades for SOUL was discussed.
3. Action items

AI1 – (Open) Armando will update the stress plot (as a function of modal deformation) shown during the meeting (Arcetri, end of July)

AI2a – (Open) make a table of the causes of sys_fault and with a description of how we can detect these causes, add the case of causes currently not detectable (Microgate, mid August)

AI2b - (Open) Marco will make table of conditions detected by the supervisor, causing sys_fault (Arcetri, end of July)

AI2c – (Open) to produce a proposal of required modifications needed to achieve full detectability of the system's status (Microgate, end of September)

AI3 - (Open) LBTO Engineering can implement all these changes. Guido will prepare a schedule (LBTO, end of July)

AI4a – (Closed) check what is implemented by ESO. MG reports that: w.r.t. the distance checking, the current implementation on VLT DSM is identical to the LBT one. Also, there is no force check (apart from the saturation of the individual actuators that is also implemented on LBT) (Microgate, end of July).

AI4b (Open) develop a software module capable of monitoring the actuator position and automatically enabling/disabling actuators based on position error values (Arcetri, LBTO, no deadline yet).

AI5 - (Open) Arcetri will investigate how we can determine the corrupted modules and what needs to be re-loaded after a RIP (Arcetri, no deadline yet)

AI6 – (Open) Guido will schedule a time at the telescope when the theory about control current integrator theory can be checked (LBTO, by end of July).

AI7 - (Open) Guido establish LBT financial commitment: use maintenance contract or issue a new contract (LBTO, end of July)?

AI8 - (Open) Marco analysis of wind to tune the ICC threshold (Arcetri, end of December)

AI9 - (Open) LBTO accepts the design that does not use the FASTDIAGNOSTICS mechanism (LBTO, no deadline yet)

AI10 - LBTO will send the spare mini-crate to MG, MG will implement the power supply modification and send spare back for testing on the telescope units (LBTO, ship ASAP and get back before Sep 1st)

AI11 – (Open) revise temperature limits for AdSec (Microgate, no deadline yet)

AI12 - (Open) Marco will write the test plan for the entire upgrade (Arcetri, no deadline yet)

AI13 – (Open) Microgate Find a solution to interface the flowmeter selected for the VLT unit to the LBT one. This new model, installed on VLT DSM after 'lessons learned' on LBT seems to work ok, so far. (Microgate, end of December).

<table>
<thead>
<tr>
<th>#</th>
<th>Action Item</th>
<th>Location</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI1</td>
<td>stress plot</td>
<td>Arcetri</td>
<td>End of July</td>
</tr>
<tr>
<td>AI2a</td>
<td>sys_fault table</td>
<td>Microgate</td>
<td>Mid August</td>
</tr>
<tr>
<td>AI2b</td>
<td>supervisor table</td>
<td>Arcetri</td>
<td>End of July</td>
</tr>
<tr>
<td>AI2c</td>
<td>proposal</td>
<td>Microgate</td>
<td>End of July</td>
</tr>
<tr>
<td>AI3</td>
<td>hardware mods</td>
<td>LBTO</td>
<td>Mid September</td>
</tr>
</tbody>
</table>
4. **Proposed firmware modifications (MG)**

**F1** - four actuators for DSPPOSCHERDNT and perhaps four for DSPCURRCHDNT

**F2** - change of skip frame criterion using the sum of bias, offload and ff only forces plus set a threshold to maximum icc

**F3** - latching of diagnostics variable, for instance the crate watchdog

**F4** - put the bus_sys_fault in the master diagnostics

5. **Proposed software modifications (Arcetri)**

**S1** - telemetry with all actuators but calculations with valid actuators only, in the interim, before the new implementation of the position out-of-bounds check

**S2** - detection of jumping actuators and disabling
5 References

2. MICROGATE, *Adaptive secondary control system: LBT672 design report electronics*, CAN 641a006e
Title: Upgrade Adaptive Optics (UAO) development plan
Document Type: Technical Report
Source: INAF-Osservatorio Astrofisico di Arcetri
Issued by: Marco Xompero, Alfio Puglisi
Date_of_Issue: 23 Mar 2016
Revised by:
Date_of_Revision:
Checked by:
Date_of_Check:
Accepted by:
Date_of_Acceptance:
Released by:
Date_of_Release:
File Type: MS-WORD
Local Name:
Category: WRITE THE CAN CATEGORY HERE
Sub-Category: WRITE THE CAN SUB-CATEGORY HERE
Assembly: WRITE THE CAN ASSEMBLY HERE
Sub-Assembly:
Part Name:
CAN designation: nnnfnnn
Revision: 1.5
Doc_info_end