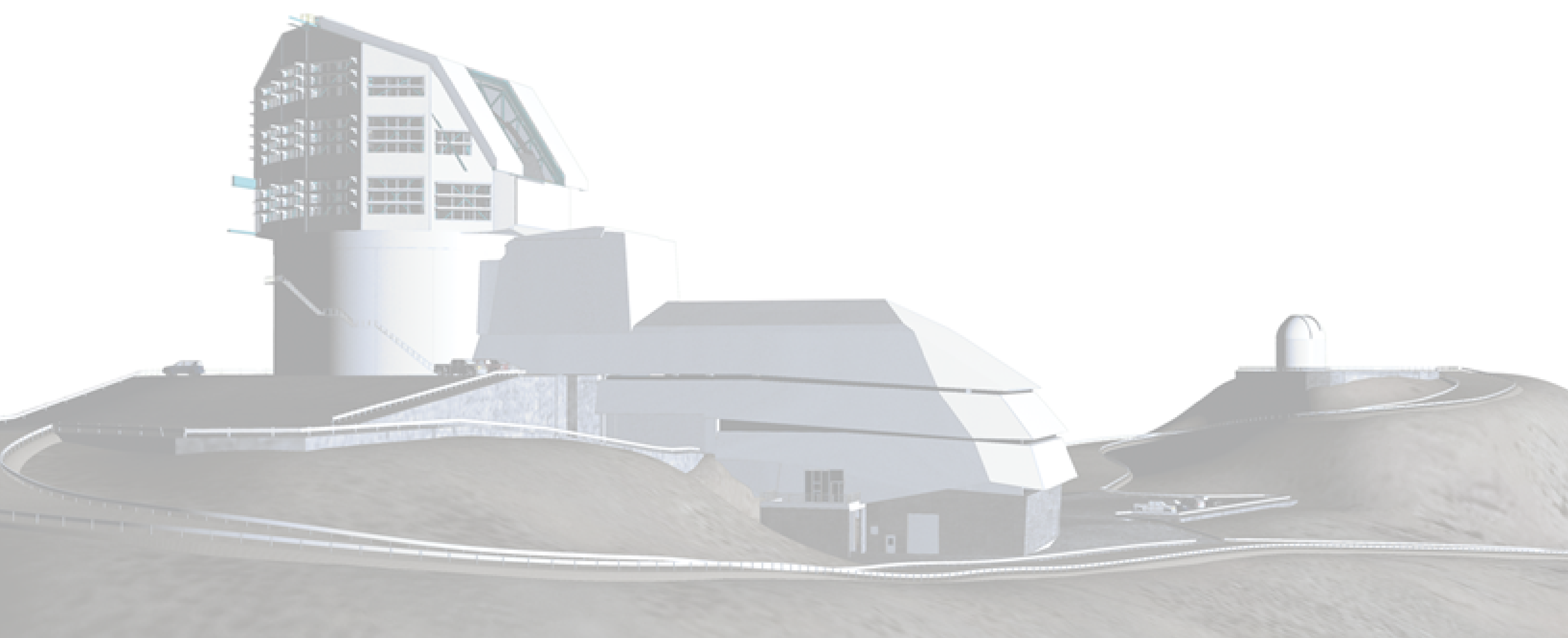
Vera C. Rubin Observatory
Software Test Report

LVV-P68: M2 Hexapod Functional Re-verification and Integration with SAL Test Plan and Report

Holger Drass

SCTR-21

Latest Revision: 2022-05-11



Abstract

This is the test plan and report for **M2 Hexapod Functional Re-verification and Integration with SAL**, an LSST milestone pertaining to the Project System Engineering and Commissioning.
This document is based on content automatically extracted from the Jira test database on 2022-05-11 . The most recent change to the document repository was on 2022-05-11.

Change Record

Version	Date	Description	Owner name
	2020-02-20	First Draft	Kevin Siruno
1.0	2020-03-09	LVV-P68 Approved SE-1372.	Kevin Siruno
3.0	2022-035-04	M2 hexapod verification on level 3 report SITCOM-204.	Holger Drass
3.1	2022-035-04	M2 hexapod verification on level 3 report after SE revisionSITCOM-204.	Holger Drass

Document curator: Holger Drass
Document source location: `https://github.com/lsst-dm/SCTR-21`
Version from source repository: c821860

Contents

LVV-P68: M2 Hexapod Functional Re-verification and Integration with SAL Test Plan and Report

1 Introduction

1.1 Objectives

The objective of this test plan is to re-verify

- the hardware and software functional requirements of the M2 hexapod without SAL
- the software functional requirements of the M2 hexapod integrated with SAL 4.0 or higher.

This re-verification is done after shipment from the vendor's facility to the Summit, as defined in LTS-206 and LTS-160.
This test campaign will exercise the functionality of the hardware and software that was executed previously and meets the following criteria:

- Requires a laser tracker, mechanical gauges, induction current probe, temperature sensors
- Requires the vendor's EUI software and hardware via local control
- Requires control via SAL 4.0 or higher

The hardware and software requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by LSST during the Factory Acceptance Test review.

1.2 System Overview

The purpose of the M2 hexapod is to maintain proper orientation of the M2 Cell Assembly. It is attached to the spider spindle of the Top End Assembly of the TMA. Although the mass of the M2 mirror cell assembly is greater than the camera, the actuators of the M2 hexapod are identical to the Camera Hexapod's actuators. For this reason, the M2 Hexapod and Camera hexapod have the same operator's manual and similar test procedures.

1.3 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P68 Jira Test Plan and related Test Cycles (LVV-C147 LVV-C203).

Section 1 provides an overview of the test campaign, the system under test (SIT-COM Integration), the applicable documentation, and explains how this document is organized. Section 2 provides additional information about the test plan, like for example the configuration used for this test or related documentation. Section 3 describes the necessary roles and lists the individuals assigned to them.

Section 4 provides a summary of the test results, including an overview in Table 3, an overall assessment statement and suggestions for possible improvements. Section 5 provides detailed results for each step in each test case.

The current status of test plan LVV-P68 in Jira is **Approved** .

1.4 References

[1] **[LSE-209]**, Lotz, P., 2016, *Software Component to OCS Interface*, LSE-209, URL https://ls.st/LSE-209

[2] **[LTS-206]**, Neill, D., Sebag, J., Gressler, W., 2017, *Hexapods and Rotator Specifications Document*, LTS-206, URL https://ls.st/LTS-206

[3] **[LTS-160]**, Schumacher, G., 2018, *TCS to Hexapods and Rotator Interface Control Document*, LTS-160, URL ls.st/LTS-160

2 Test Plan Details

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

This test plan is going to be executed in two verification environments:

- The M2 Hexapod will be verified in a climate-controlled environment **on the 3rd floor of the Summit Facility** on the TEA structure with **the M2 mass surrogate installed and facing downwards.** The TEA is mounted on its shipping mount.
- The M2 Hexapod will be verified **on the TMA in the Dome of the Summit Facility** on the TEA structure with **the M2 mass surrogate installed and following the elevation suggested in the vendor's documentation.**

2.3 Entry Criteria

In order to test the M2 Hexapod functionality, the following criteria must be met first:

- All the test setup for the Data Acquisition system must be completed and ready to record data for the laser tracker
- The laser tracker and 3 SMRs are installed and setup
- All utilities and electrical connections are hooked up and allow the M2 Hexapod to be powered on and controlled
- The EFD must be set up to be able to store events and telemetry data
- The temperature measurement system is operational and the EFD is able to record temperature

2.4 Exit Criteria

In order for this event to be considered complete, the following criteria must be met:

- Raw test data, events, and telemetry have been saved for the M2 Hexapod in the EFD.
- All test data has been analyzed and post-processed.
- All test steps have been statused in the Jira Test Cases within this Test Plan and actual results populated as required.
- A summary of the results of the test campaign has been captured in the Overall Assessment and Recommended Improvements fields of this Test Plan
- A link to the verification artifacts used to produce the summary of results has been populated in the Verification Artifacts field of this Test Plan
- For tests producing quantitative results reporting of the analysis shall include traceability to the raw data of the test and estimates for the statistical significance of the result(s).
- Any failures have been captured in the FRACAS project

2.5 Related Documentation

Jira Attachments

To LVV-C147 results	M2hexapod_thetestmoves_02-25Feb2021_Results_updateRT27042021_comment_hd.xlsx
To LVV-C147 results	M2Hexapodverification-tests_2021.check4.docx

All documents provided as attachments in Jira are downloaded to Github and linked here for convenience. However, since they are not properly versioned, they should be considered informal and therefore not be part of the verification baseline.

2.6 PMCS Activity

Primavera milestones related to the test campaign:

- None

3 Personnel

The personnel involved in the test campaign is shown in the following table.

	T. Plan LVV-P68 owner:	**Holger Drass**	
	T. Cycle LVV-C147 owner:	**Holger Drass**	
Test Cases	**Assigned to**	**Executed by**	**Additional Test Personnel**
LVV-T1804	Holger Drass	Holger Drass	(1) Software Engineer (1) Hardware Engineer
LVV-T1800	Holger Drass	Holger Drass	(1) Software Engineer (1) Mechanical Engineer (1) Systems Engineer
LVV-T1802	Holger Drass	Holger Drass	(1) Software Engineer (1) Hardware Engineer
	T. Cycle LVV-C203 owner:	**Holger Drass**	
Test Cases	**Assigned to**	**Executed by**	**Additional Test Personnel**
LVV-T1804	Holger Drass		(1) Software Engineer (1) Hardware Engineer
LVV-T1800	Holger Drass		(1) Software Engineer (1) Mechanical Engineer (1) Systems Engineer
LVV-T1802	Holger Drass		(1) Software Engineer (1) Hardware Engineer

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P68:		**M2 Hexapod Functional Re-verification and Integration with SAL**		Approved
T. Cycle LVV-C147:		**M2 Hexapod Re-verification on Level 3**		Done
Test Cases	**Ver.**	**Status**	**Comment**	**Issues**
LVV-T1804	1	Initial Pass	The start-up procedure was executed in a different test case before and did not need to be executed again.	
LVV-T1800	1	Fail	Since this execution, this (M2) hexapod was lifted to the TMA. In the meantime, the **camera** hexapod software and hardware were extensively tested and improved.	LVV-19511 LVV-19511 LVV-19511 LVV-19511
LVV-T1802	2	Fail		LVV-19512 LVV-19512 LVV-19512 LVV-19512 LVV-19512 LVV-19512 LVV-19512 LVV-19512
T. Cycle LVV-C203:		**M2 Hexapod Re-verification on the TMA**		Not Executed
Test Cases	**Ver.**	**Status**	**Comment**	**Issues**
LVV-T1804	1	Not Executed		
LVV-T1800	1	Not Executed		
LVV-T1802	2	Not Executed		

Table 3: Test Campaign Summary

4.2 Overall Assessment

The following results are for the first test-cycle LVV-C147:

During the February 2020 verification, the M2 hexapod was still on level three, and the M2 hexapod software components low-level controller, EUI, and CSC were still under develop-

ment.
Since the hexapod has been received from the vendor the sets of commands for the EUI and the CSC have diverged. The "positionSet" command is only supported by the EUI and is retired for the CSC. At the CSC level, this functionality is now included directly in the commands to move the hexapod. Other commands were either modified themselves or their payload has changed. For an up-to-date list of all supported commands and their permitted payloads see MT Hexapod Sal interfaces.
Some events were not created in the EFD as expected. In detail, the "inPosition" and "controlledStop" events were not generated in the EFD. The "compensationMode" was not activating correctly and not reported to the EFD as expected. For details see the Execution of test case LVV-T1804 and the execution of test case LVV-T1802.
Apart from this, the CSC is going to be updated to a standbyState-entry state machine. This implies that the start-up procedure will change. Various other changes were necessary to keep up with the development of the SAL/DDS itself and to accommodate changes in the network infrastructure of the observatory.
All of these changes have been tested with positive results using the camera hexapod since the M2 hexapod was lifted on top of the TMA and is not accessible for a long time.

On the hardware side, the M2 hexapod presented random faults in Drive 0 and Drive 1. The failure is randomly happening over a time range between a few minutes and at least 11 hours. The fault appears in the EFD and on the EUI. On the EUI the fault is presented as "Feedback Error Drive0". A detailed investigation on this issue was performed by first switching the cables to a different actuator and later by changing the actuator itself. Both attempts are excluding that an actuator as the source of failure but did not solve the problem (for details see FRACAS-53). This issue was also observed on the camera hexapod and was solved by changing the cables to the actuators.
The second hardware-related issue concerns the requirements LTS-206-REQ-0164-V-02: 3.5.12_1 Positioning - LSST Re-verification and LTS-206-REQ-0178-V-02: 3.5.24_1 Hexapod Absolute Accuracy - LSST Re-verification. Tests on the positioning in X,Y,Z translation combined with a rotation failed to reach the required precision. The same issue was observed for the hexapod's absolute accuracy. The reasons are lying most likely in the measurement setup itself. The laser tracker measurements are at the limit of the laser tracker's precision and the Mitutoyo gauges mounts were a preliminary solution. Testing the camera hexapod with an improved Mitutoyo setup has shown that the same requirements are fulfilled for the camera hexapod. The hardware tests are executed and documented as part of the execution of test case LVV-

T1800.
In addition, all the aforementioned tests and discoveries were done shortly before the M2 hexapod needed to be installed at the TMA. Therefore, some of the hardware tests could not be executed.

The following results are for the second test cycle:

4.3 Recommended Improvements

The following recommendations are derived from the first test-cycle
As a general comment, enough time should be allocated to test the EUI and CSC commands as well as to execute the rest of the hardware tests.
To improve the situation before the tests on the TMA the recommendation is to update the M2 hexapod low-level controller, the EUI, and the CSC to the latest versions.
The software tests for the EUI should be executed when the state machine is updated to the standbyState-entry state machine. The test case should be updated to account for state machine tests.
The configuration part of the test cases should be updated to account for the change in accordance with LSE-209 and need to be tested for the first time.
For the hardware aspects of the M2 hexapod, a new measurement setup that is similar to what was used for the camera hexapod should be used. With the new measurement setup in place, the failing tests should be repeated to confirm the positioning in X,Y,Z in combination with rotation and the hexapod's absolute accuracy are within their requirements (LTS-206-REQ-0164-V-02: 3.5.12_1 Positioning - LSST Re-verification and LTS-206-REQ-0178-V-02: 3.5.24_1 Hexapod Absolute Accuracy - LSST Re-verification). In case new faults are encountered, especially blind spots on the encoders, cleaning of the actuators similar to the camera hexapod actuators should be performed. To address the random fault in drive zero and to ensure that is not happening in another drive all the cables for the M2 hexapod should be updated similar to the camera hexapod and the 32 MOOG test moves (Moog test protocol paragraph 3.5.12), as well as the endurance test corresponding to one night of operation, should be executed. The FRACAS-53 ticket should be closed out.

Test cases with an "initial pass" or "fail" need to be executed in the next test cycle. The test plan needs to be revised, and make a new test cycle with the M2 hexapod on the TMA. Update the test cases to include the updated software and updated wiring and the improved gauge measurement setup. All of the "Not executed" steps need to be revised as required and executed because we did not have enough time to execute the tests.

The following recommendations are derived from the second test-cycle:

5 Detailed Test Results

5.1 Test Cycle LVV-C147

Open test cycle *M2 Hexapod Re-verification on Level 3* in Jira.

Test Cycle name: M2 Hexapod Re-verification on Level 3
Status: Done

Re-verify the hardware and software for the M2 Hexapod that was previously tested by MOOG and verify the integrated M2 hexapod with SAL 4.0 or higher.

5.1.1 Software Version/Baseline

1. M2 Hexapod Control Software with SAL v4.0 or higher
2. EFD with SAL v4.0 or higher

5.1.2 Configuration

No varying configuration between test cycles.

5.1.3 Test Cases in LVV-C147 Test Cycle

5.1.3.1 LVV-T1804 - M2 Hexapod Software Functional Re-verification

Version **1**. Open *LVV-T1804* test case in Jira.

The objective of this test case is to re-verify the functional requirements of the M2 hexapod's software, after shipment of the hardware from the vendor's facility to the Summit, as defined in LTS-206 and LTS-160. This test case will only exercise the functionality that was executed previously and meets the following criteria:

- Only requires the M2 hexapod to be operable

- Only requires testing the synchronous mode
  - **Asynchronous mode is not a standard mode of operation**
- Only requires the vendor's EUI software and hardware via local control
  - Does **NOT** require integration with SAL
- Does **NOT** require the M2 hexapod to be rotated to various elevation angles.
- This test case can be executed with a **simulated M2 mass or actual M2 hardware**

The software functional requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by Rubin Observatory during the Factory Acceptance Test review. The test procedure used during the vendor's acceptance testing is the *LSST Hexapods-Rotator Software Acceptance Test Procedure* which is attached to this test case. The test steps of this test case are taken directly from that document on how to perform the test in a similar way as was performed previously and the test steps include changes noted by the vendor.

See the attached *LSST Hexapod Operator's Manual* for more information on how to operate the hexapod.

Preconditions:
Prior to the execution of this test case to re-verify the M2 Hexapod hardware functional requirements, the following Summit tasks must be completed:

- The measurement equipment has been set up for testing
  - `https://jira.lsstcorp.org/browse/SUMMIT-1943`

Execution status: **Initial Pass**

Final comment:
The start-up procedure was executed in a different test case before and did not need to be executed again.

Detailed steps results:

Step 1 Step Execution Status: **Pass**

Description

STARTING THE EUI

Double click the Hexapod GUI Viewer desktop icon on the computer.

- This can be done on the Dell Management PC or another computer on the same network

Expected Result

A prompt to enter the password is shown.

Actual Result

Deviation:

- Access is now through X2GO.

Step 2 Step Execution Status: **Pass**

Description

Enter the password "lsst-vnc"

- If the EUI isn't automatically up and running when the VNC opens, double click on the Hexapod-eGUI icon on the VNC viewer

Expected Result

The EUI is in the Offline State/PublishOnly substate and is able to publish through SAL but cannot receive commands.

Actual Result

Deviation:

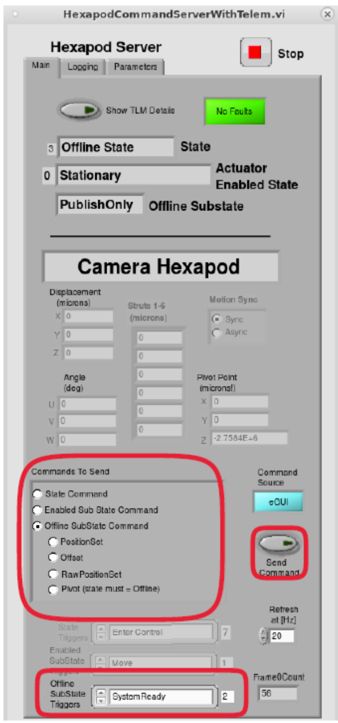
- Access is now through X2GO.
- Use the command line to start the latest version of the EUI.
- Starts in Offline State/PublishOnly.

Step 3 Step Execution Status: **Initial Pass**

Description

OFFLINESTATE/AVAILABLESTATE

On the Main tab, select the "Offline SubState Cmd" field in the Commands to Send section, set the Offline SubState Triggers to "System Ready" and click on the Send Command button.



Expected Result

The system transitions from the OfflineState/PublishOnly substate to the OfflineState/AvailableState substate and the Command Source says eGUI.

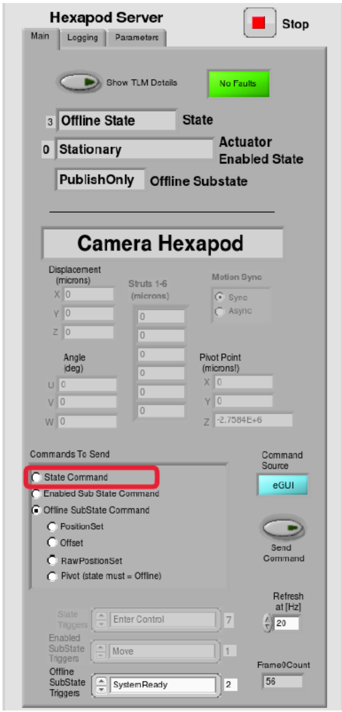
Actual Result

Step 4 Step Execution Status: **Initial Pass**

Description

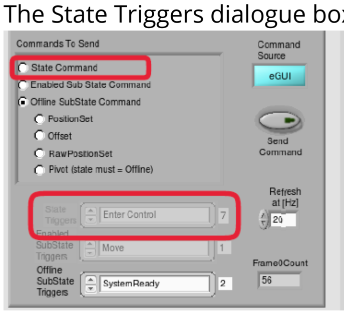
OFFLINESTATE -> STANDBYSTATE

Click on the State Command field in the Commands to Send section.



Expected Result

The State Triggers dialogue box shown below becomes visible.



Actual Result

Step 5 Step Execution Status: **Initial Pass**

Description

Scroll through the available trigger options to select "Enter Control" and click the Send Command button.

Expected Result

The system transitions to the Standby state and the primary state display box at the top of the Main says Standby

State.

Actual Result

Step 6 Step Execution Status: **Initial Pass**

Description

STANDBYSTATE -> DISABLEDSTATE

From the StandbyState, send a Start State command.

Expected Result

The system transitions into DisabledState and the current configuration parameters are maintained from the default parameters or from the previous DDS start command.

Actual Result

Step 7 Step Execution Status: **Initial Pass**

Description

DISABLEDSTATE -> ENABLEDSTATE

From the DisabledState, send an Enable State Command.

Expected Result

The system transitions into the EnabledState/Stationary substate, the motor drives are enabled and and motion can be commanded.

Actual Result

Step 8 Step Execution Status: **Not Executed**

Description

<conditional state>

FAULTSTATE

If a Fault occurs in any of the other states, the system will automatically transition to the Fault State. While in the Fault state, send a clearError.

Note: If the fault that occurs goes through the interlock system, reset the safety relay switch and send a clearError command.

Expected Result

The system transitions back to the OfflineState/PublishOnly substate. (Go back to Step 3)

Actual Result

Step 9	Step Execution Status: **Initial Pass**

Description

Section 3.3.1 EUI Tests of the attached Software Acceptance Test Procedure

At startup, confirm that the system starts in the Offline/PublishOnly state.

Expected Result

The hexapod starts in the Offline/PublishOnly state.

Actual Result

The state machine in general allowed to transition the M2 hexapod in the enabledState/Stationary and to clear faults when necessary. The state machine was under development during this testing period not further investigated.

Step 10	Step Execution Status: **Not Executed**

Description

Send an offline substate trigger of systemReady.

Expected Result

The system transitions into the Offline/Available substate.

Actual Result

Step 11	Step Execution Status: **Not Executed**

Description

Send an EnterControl trigger.

Expected Result
The system transitions from Offline/Available to Standby state.

Actual Result

Step 12 Step Execution Status: **Not Executed**

Description
Send a Start trigger.

Expected Result
The system transitions from Standby to Disabled state.

Actual Result

Step 13 Step Execution Status: **Not Executed**

Description
Send an Enable trigger.

Expected Result
The system transitions from Disabled to Enabled state.

Actual Result

Step 14 Step Execution Status: **Not Executed**

Description
Send a Disable trigger.

Expected Result
The system transitions from Enabled to Disabled state.

Actual Result

Step 15 Step Execution Status: **Not Executed**

Description

Send a Standby trigger.

Expected Result

The system transitions from Disabled state to Standby state.

Actual Result

Step 16 Step Execution Status: **Not Executed**

Description

Send a exitControl trigger.

Expected Result

The system transitions from Standby state to Offline state.

Actual Result

Step 17 Step Execution Status: **Not Executed**

Description

Return to the Enabled state and trip the safety interlock switch.

Expected Result

The system transitions to Fault state.

Actual Result

Step 18 Step Execution Status: **Initial Pass**

Description

Reset the safety interlock and send a ClearError trigger.

Expected Result

The CSC, upon receiving the ¨clearError¨ trigger, transitions from FaultState to OfflineState/PublishOnly when the system was in any of the OfflineStates before the error occurred. The CSC, upon receiving the "clearError" trigger, transitions to StandbyState when it was in EnableState or DisableState before the error occurred.

Actual Result

Step 19 Step Execution Status: **Not Executed**

Description

Section 4.1 Hexapod Events of the attached Software Acceptance Test Procedure

In the Enabled/Stationary state, unplug a motor encoder cable for one of the actuators.

Test Data

Deviation: Perform the following set of steps using the EUI instead of the DDS and verify the events are displayed on the EUI.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

Not tested

Step 20 Step Execution Status: **Not Executed**

Description

Send the "clearError" trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 21	Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, unplug a linear encoder cable for one of the actuators.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

Not tested

Step 22	Step Execution Status: **Not Executed**

Description

Send the “clearError” trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 23	Step Execution Status: **Not Executed**

Description

Unplug a motor power cable from one of the actuators and command a PositionSet/Move.

Expected Result

A Following Error event is created and the system transitions to Fault state.

Actual Result

not tested

Step 24	Step Execution Status: **Not Executed**

Description

Send the “clearError” trigger and bring the system to the Enabled/Stationary state.

Expected Result
The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 25 Step Execution Status: **Initial Pass**

Description
Activate an extension limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result
An Extended Limit Switch error event is created and the system transitions into Fault state.

Actual Result
Tested by Felipe. Upon manually triggering an actuator limit switch, an Extended Limit Switch error event is created and the system transitions into Fault state.

Step 26 Step Execution Status: **Initial Pass**

Description
Send the "clearError" trigger and bring the system to the Enabled/Stationary state.

Expected Result
The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 27 Step Execution Status: **Initial Pass**

Description
Activate a retraction limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result
A Retracted Limit Switch error event is created and the system transitions into Fault state.

Actual Result

Tested by Felipe. Upon manually triggering an actuator limit switch, an Retracted Limit Switch error event is created and the system transitions into Fault state.

Step 28 Step Execution Status: **Initial Pass**

Description

Send the “clearError” trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 29 Step Execution Status: **Not Executed**

Description

Unplug the Ethercat cable between the control PC and the first Copley XE2 drive.

Expected Result

An Ethercat Lost event is created and the system transitions to Fault state.

Actual Result

Not tested

Step 30 Step Execution Status: **Not Executed**

Description

Send the “clearError” trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 31 Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #1 - Synchronous PositionSet and Move Commands

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (0um, 0um, 200um, 0 deg, 0 deg, 0 deg) using the EUI.

Expected Result

The hexapod doesn't move.

Actual Result

Due to time constraints, commands were not tested from the EUI. See the tests in LVV-T1802 "Integration of M2 Hexapod with SAL".

Step 32 Step Execution Status: **Not Executed**

Description

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg) using the EUI.

Expected Result

The hexapod doesn't move.

Actual Result

Step 33 Step Execution Status: **Not Executed**

Description

Send a move command using the EUI.

Test Data

Pivot position is shown in the GUI. Please mention in the results. Use the MOOG pivot point for comparability with the previous results.

Expected Result

The hexapod moves to the last commanded position of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg). Since the test is done in synchronous mode the actuators are expected to complete the move at nearly the same time as seen on the motion complete lights on the telemetry screen.

Actual Result

Step 34 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 35 Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #2 - Pivot, PositionSet and Move Commands

In enabled/stationary state and at the last commanded position of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg), change the pivot point from the default location to (0,0,0) using the EUI.

Expected Result

The actuator positions do not change, but the hexapod position is (-407um, -3982um, 199um, 0.01deg, -0.05deg, 0.002deg)

Actual Result

Step 36 Step Execution Status: **Not Executed**

Description

In the enabled/stationary state, send a positionSet command of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg) using the EUI.

Expected Result

The hexapod doesn't move.

Actual Result

Step 37 Step Execution Status: **Not Executed**

Description

Send a move command using the EUI.

Expected Result

The hexapod moves to the commanded position of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg) and the actuators change position to account for the new pivot point.

Actual Result

Step 38 Step Execution Status: **Not Executed**

Description

Wait 39s

Expected Result

Actual Result

Step 39 Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #4 - Synchronous Offset and Move Commands

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (500um, 800um, 200um, 0 deg, 0 deg, 0 deg).

Expected Result
The hexapod doesn't move.

Actual Result

Step 40 Step Execution Status: **Not Executed**

Description
With the synchronous button enabled and in enabled/stationary state, send an offset command of (0um, 0um, 2000um, 0 deg, 0 deg, 0 deg).

Expected Result
The hexapod doesn't move.

Actual Result

Step 41 Step Execution Status: **Not Executed**

Description
Send a move command.

Expected Result
The hexapod moves only 2000um in Z from the previous position. Since the test is done in synchronous mode the actuators are expected to complete the move at nearly the same time as seen on the motion complete lights on the telemetry screen.

Actual Result

Step 42	Step Execution Status: **Not Executed**

Description

Wait 39s

Expected Result

Actual Result

Step 43	Step Execution Status: **Not Executed**

Description

Instead of Asynchronous Test

With the synchronous button enabled and in enabled/stationary state, send a position set command of (0um, 0um, 0um, 0.1deg, 0deg, 0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Step 44	Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod moves to the commanded position of (0um, 0um, 0um, 0.1deg, 0deg, 0deg)

Actual Result

Step 45	Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 46 Step Execution Status: **Not Executed**

Description

With the synchronous button enabled and in enabled/stationary state, send a position set command of (0um, 0um, 0um, 0deg, 0.1deg, 0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Step 47 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod moves to the commanded position of (0um, 0um, 0um, 0deg, 0.1deg, 0deg)

Actual Result

Step 48 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 49 Step Execution Status: **Not Executed**

Description

With the synchronous button enabled and in enabled/stationary state, send a position set command of (0um, 0um, 0um, 0.1deg, 0.1deg, 0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Step 50 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod moves to the commanded position of (0um, 0um, 0um, 0.1deg, 0.1deg, 0deg)

Actual Result

Step 51 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 52	Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #5 - Stop Commands

In enabled/stationary state, send a position set command of (0um, 0um, 5000um, 0 deg, 0 deg, 0 deg).

Expected Result

The hexapod doesn't move.

Actual Result

Step 53	Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod starts to move to the commanded position.

Actual Result

Step 54	Step Execution Status: **Not Executed**

Description

Wait 3s.

Expected Result

Actual Result

Step 55	Step Execution Status: **Not Executed**

Description

Send a stop command.

Expected Result

The hexapod quickly comes to a stop prior to reaching the commanded position.

Actual Result

5.1.3.2 LVV-T1800 - M2 Hexapod Hardware Functional Re-verification

Version **1**. Open *LVV-T1800* test case in Jira.

The objective of this test case is to re-verify the functional requirements of the M2 hexapod's hardware, after shipment from the vendor's facility to the Summit, as defined in LTS-206. This test case will only exercise the functionality that was executed previously and meets the following criteria:

- Only requires the M2 hexapod to be operable
- Only requires the EUI software and hardware via local control
- Requires a laser tracker, mechanical gauges, inductive current probe
- Does require the M2 hexapod temperature sensors to be operating
- Does **NOT** require the M2 hexapod to be rotated to various elevation angles
- Does **NOT** require the M2 hexapod to be in a climate-controlled environment

The hardware functional requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by Rubin Observatory during the Factory Acceptance Test review. The test procedure used during the vendor's acceptance testing is the *LSST Hexapods-Rotator Acceptance Test Procedure* which is attached to this test case. The test steps of this test case are taken directly from that document on how to perform the test in a similar way as was performed previously and the test steps include changes noted by the vendor.

See the attached *LSST Hexapod Operator's Manual* for more information on how to operate the hexapod.

Preconditions:
Prior to the execution of this test case to re-verify the M2 Hexapod hardware functional requirements, the following Summit tasks must be completed:

- The measurement equipment has been set up for testing
  - `https://jira.lsstcorp.org/browse/SUMMIT-1943`
- The laser tracker has been set up for measurements
  - `https://jira.lsstcorp.org/browse/SUMMIT-3951`

Execution status: **Fail**

Final comment:
Since this execution, this (M2) hexapod was lifted to the TMA. In the meantime, the **camera** hexapod software and hardware were extensively tested and improved.

Issues found during the execution of LVV-T1800 test case:

- LVV-19511 M2 Hexapod absolute XYZ value accuracy not reached
- LVV-19511 M2 Hexapod absolute XYZ value accuracy not reached

Detailed steps results:

Step 1	Step Execution Status: **Pass**

Description
STARTING THE EUI

Double click the Hexapod GUI Viewer desktop icon on the computer.

- This can be done on the Dell Management PC or another computer on the same network

Expected Result
A prompt to enter the password is shown.

Actual Result
Deviation:

- Access is now through X2GO.

Step 2 Step Execution Status: **Pass**

Description
Enter the password "lsst-vnc"

- If the EUI isn't automatically up and running when the VNC opens, double click on the Hexapod-eGUI icon on the VNC viewer

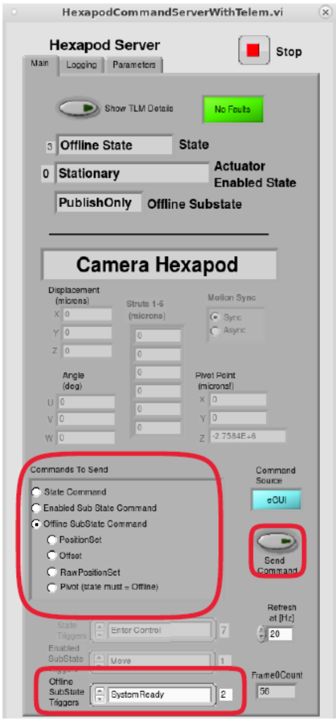
Expected Result
The EUI is in the Offline State/PublishOnly substate and is able to publish through SAL but cannot receive commands.

Actual Result
Deviation:

- Access is now through X2GO.
- Use the command line to start the latest version of the EUI.
- Starts in Offline State/PublishOnly.

Step 3 Step Execution Status: **Initial Pass**

Description
OFFLINESTATE/AVAILABLESTATE
On the Main tab, select the "Offline SubState Cmd" field in the Commands to Send section, set the Offline SubState Triggers to "System Ready" and click on the Send Command button.



Expected Result

The system transitions from the OfflineState/PublishOnly substate to the OfflineState/AvailableState substate and the Command Source says eGUI.

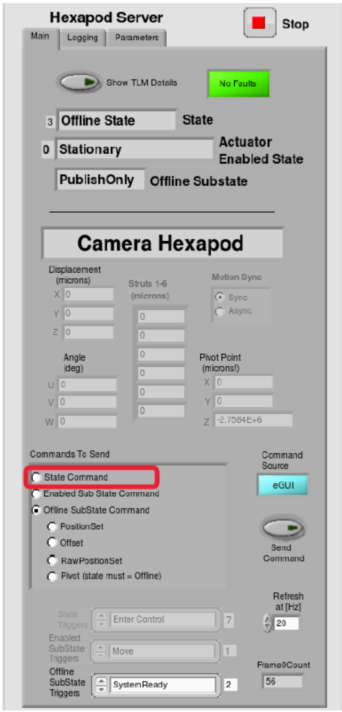
Actual Result

Step 4 Step Execution Status: **Initial Pass**

Description

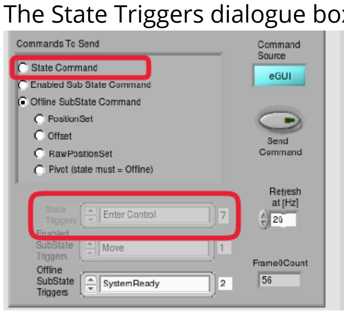
OFFLINESTATE -> STANDBYSTATE

Click on the State Command field in the Commands to Send section.



Expected Result

The State Triggers dialogue box shown below becomes visible.



Actual Result

Step 5	Step Execution Status: **Initial Pass**

Description

Scroll through the available trigger options to select “Enter Control” and click the Send Command button.

Expected Result

The system transitions to the Standby state and the primary state display box at the top of the Main says Standby

State.

Actual Result

Step 6	Step Execution Status: **Initial Pass**

Description

STANDBYSTATE -> DISABLEDSTATE

From the StandbyState, send a Start State command.

Expected Result

The system transitions into DisabledState and the current configuration parameters are maintained from the default parameters or from the previous DDS start command.

Actual Result

Step 7	Step Execution Status: **Initial Pass**

Description

DISABLEDSTATE -> ENABLEDSTATE

From the DisabledState, send an Enable State Command.

Expected Result

The system transitions into the EnabledState/Stationary substate, the motor drives are enabled and and motion can be commanded.

Actual Result

Step 8	Step Execution Status: **Initial Pass**

Description

<conditional state>

FAULTSTATE

If a Fault occurs in any of the other states, the system will automatically transition to the Fault State. While in the Fault state, send a clearError.

Note: If the fault that occurs goes through the interlock system, reset the safety relay switch and send a clearError command.

Expected Result

The system transitions back to the OfflineState/PublishOnly substate. (Go back to Step 3)

Actual Result

Step 9 Step Execution Status: **Fail**

Description

Follow *3.5.12 Positioning* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 57-58.

Test Data

Deviation: Test at a single elevation angle.

Expected Result

The position of the hexapod is able to reach the commanded positions within the absolute accuracy specifications of 25um in Z, 125um in XY, 83x10-5deg in RXRY, and 750x10-5deg in RZ.

Actual Result

The errors in XY are considered via sqrt(Xerr**2+Yerr**2) and individually for each measurement.
The results for the test steps 25, 26, 27, 28, 30, and 32 are out of specs.
See results in M2 hexapod_the test moves_02-25 Feb2021_Results_updateRT27042021_comment_hd column AB and AC from line 34 on.

Issues found executing this step:

- LVV-19511 M2 Hexapod absolute XYZ value accuracy not reached

---

Step 10 Step Execution Status: **Pass**

Description

Follow *3.5.15 Radial (X and Y) Translation Range* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 59.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result

The hexapod is capable of moving to the positions in the XY plane listed in the Acceptance Test Procedure.

Actual Result

Deviation: In this test, the X and Y moves are up to a maximum of 10mm not 10.05mm.
Hexapod reached the positions within the allowed tolerances and.
See lines 13-19, M2 hexapod_the test moves_02-25 Feb2021_Results_updateRT27042021_comment_hd

---

Step 11 Step Execution Status: **Initial Pass**

Description

Follow *3.5.13 Centers of Rotation* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 58-59.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement. The spherically mounted retroreflector (SMR) will be mounted on the ring holding the M2 mass surrogate or the M2 mass simulator

Expected Result

The center of rotation is able to be moved.

Actual Result

initial pass. Details see page 6, M2 Hexapod verification-tests_2021.check
This was with the hexapod on level 3 in the "zenith" position. Test again with hexapod at 0 elevations.

Step 12 Step Execution Status: **Initial Pass**

Description

Follow *3.5.17 Axial (Z) Translation Range* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 60.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result
The hexapod is capable of moving to the positions in the Z plane listed in the Acceptance Test Procedure.

Actual Result
Initial pass. Hexapod moved to Z=8.9mm within the allowed tolerances. See page 5, M2 Hexapod verification-tests_2021.check4.

Step 13	Step Execution Status: **Initial Pass**

Description
Follow *3.5.19 Rotational Range Around X-Axis (Tip) and Y-Axis (Tilt)* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 61.

Test Data
Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result
The hexapod is capable of moving to the positions in the RXRY plane listed in the Acceptance Test Procedure.

Actual Result
Initial pass. Hexapod moved to RX, RY = 0.175deg within the allowed tolerances. See page 5, M2 Hexapod verification-tests_2021.check4.

Step 14	Step Execution Status: **Initial Pass**

Description
Follow *3.5.21 Rotation Range Around Z-Axis (Twist)* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 62.

Test Data
Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result
The hexapod is capable of moving to the positions in the RZ-axis listed in the Acceptance Test Procedure.

Actual Result

Initial pass. Hexapod moved to the expected position within the allowed tolerances. See page 5, M2 Hexapod verification-tests_2021.check4.

Step 15 Step Execution Status: **Not Executed**

Description

Follow *3.5.23 Hexapod Repeatability* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 63-70.

Test Data

Deviation: Allow a minimum of 30 seconds between moves.

Expected Result

The repeatability of the hexapod is likely better than can be determined by the test equipment. This test will likely falsely show a deficiency in the hexapod performance as a result of test equipment accuracy/ repeatability limitation.

Actual Result

See page 7, M2 Hexapod verification-tests_2021.check4: Hexapod repeatability tests...not done, no time

---

Step 16 Step Execution Status: **Fail**

Description

Follow *3.5.24 Hexapod Absolute Accuracy* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 70-74.

Test Data

Deviation: Test at a single elevation angle.

Expected Result

The accuracy of the hexapod is at least the following:

Axis	Required Accuracy (um, deg)
X	125
Y	125
Z	25

Axis	Required Accuracy (um, deg)
RX	0.00083
RY	0.00083
RZ	0.0075

NOTE: The accuracy of the hexapod may be better than can be determined by the test equipment. This may falsely show a deficiency in the hexapod performance as a result of test equipment accuracy/ repeatability limitation.

Actual Result

Deviation: Only Z was tested.

Test failed the required accuracy. See page 8, M2 Hexapod verification-tests_2021.check4.

See lines 140-150, M2 hexapod_the test moves_02-25 Feb2021_Results_updateRT27042021_comment_hd

Issues found executing this step:

- LVV-19511 M2 Hexapod absolute XYZ value accuracy not reached

Step 17 Step Execution Status: **Not Executed**

Description

Follow *3.5.26 Hexapod Radial (X and Y) and Axial (Z) Velocity Range* and *3.5.27 Hexapod Rotational Velocity* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 75.

Test Data

Deviation: Only test this using synchronous mode. Wait for 39s between each movement.

Expected Result

The hexapod velocity exceeds the 106um/s in XY and 0.0062deg/s in RXYRY and RZ requirements.

Actual Result

Not tested

Step 18 Step Execution Status: **Not Executed**

Description

Follow 3.5.28 *Hexapod Heat Dissipation* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 75-76.

Test Data

Deviation: Calculate the power by having an amp meter on the legs. This test can be done simultaneously with the other test steps.

Expected Result

The current measured by the inductive current probes is calculated to meet the heat dissipation requirement.

Actual Result

Not measured.

Step 19 Step Execution Status: **Not Executed**

Description

Follow *3.5.14 Cross Talk Motion* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 59.

Test Data

Deviation: Analyze data from 3.5.15, 3.5.17, and 3.5.19 test steps after testing to verify cross talk.

Expected Result

There is no cross-talk observed.

Actual Result

5.1.3.3 LVV-T1802 - Integration of M2 Hexapod with SAL

Version **2**. Open *LVV-T1802* test case in Jira.

The objective of this test case is to re-verify the functional requirements of the M2 hexapod's software, after shipment of the hardware from the vendor's facility to the Summit, as defined in LTS-206 and LTS-160. This test case will only exercise the functionality that was executed previously and meets the following criteria:

- Only requires the use of Rubin Observatory code to replace MOOG's middleware code
- Only requires the M2 hexapod to be operable
- Only requires command through the CSC after the PXI real-time controller is switched from GUI mode to DDS mode
- Only requires testing the synchronous mode
  - **Asynchronous mode is not a standard mode of operation**
- Does require the M2 hexapod temperature sensors to be operating
- Does **NOT** require the M2 hexapod to be rotated to various elevation angles.
- Does **NOT** require the M2 hexapod to be in a climate-controlled environment

The software functional requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by Rubin Observatory during the Factory Acceptance Test review. The test procedure used during the vendor's acceptance testing is the *LSST Hexapods-Rotator Software Acceptance Test Procedure* which is attached to this test case. The test steps of this test case are the same steps from the procedure for the testing of the Camera Hexapod. The order of the steps was changed to reflect the *Proposal of Hexapod Test on the Dec. 2019* Confluence page which can be found linked in the Traceability tab.

See the attached *LSST Hexapod Operator's Manual* for more information on how to operate the hexapod.

Preconditions:
Prior to the execution of this test case to re-verify the M2 Hexapod hardware functional requirements, the following Summit tasks must be completed:

- The measurement equipment has been set-up for testing
  - `https://jira.lsstcorp.org/browse/SUMMIT-1943`

Execution status: **Fail**

Final comment:

Issues found during the execution of LVV-T1802 test case:

- LVV-19512 M2 Hexapod SAL verification failures
- LVV-19512 M2 Hexapod SAL verification failures
- LVV-19512 M2 Hexapod SAL verification failures
- LVV-19512 M2 Hexapod SAL verification failures

Detailed steps results:

Step 1 Step Execution Status: **Pass**

Description

STARTING THE EUI

Double click the Hexapod GUI Viewer desktop icon on the computer.

- This can be done on the Dell Management PC or another computer on the same network

Expected Result

A prompt to enter a password is shown.

Actual Result

See: LVV-E1022

Step 2 Step Execution Status: **Pass**

Description

Enter the password "lsst-vnc"

- If the EUI isn't automatically up and running when the VNC opens, double click on the Hexapod-eGUI icon on the VNC viewer

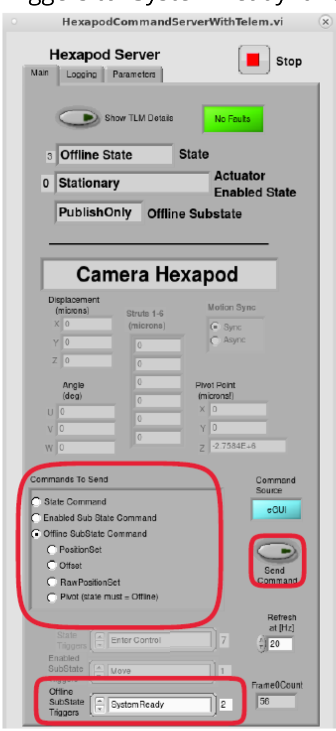
---

Expected Result
The EUI is in the Offline State/PublishOnly substate and is able to publish through SAL but cannot receive commands.

Actual Result
See: LVV-E1022

Step 3 Step Execution Status: **Initial Pass**

Description
OFFLINESTATE/PUBLISHONLY -> OFFLINESTATE/AVAILABLESTATE
On the Main tab, select the "Offline SubState Cmd" field in the Commands to Send section, set the Offline SubState Triggers to "System Ready" and click on the Send Command button.



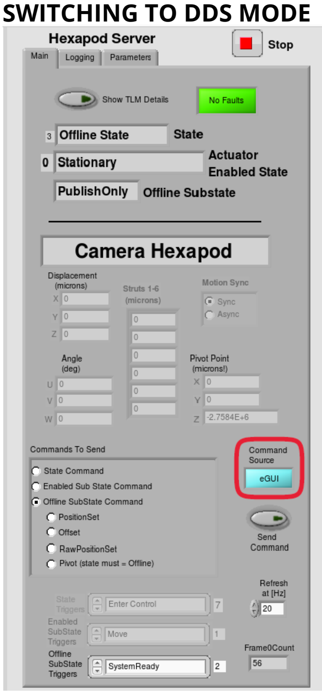
Expected Result
The system transitions from the OfflineState/PublishOnly substate to the OfflineState/AvailableState substate.

Actual Result
See: LVV-E1022

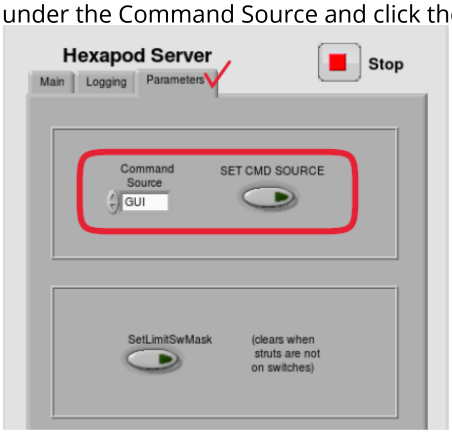
Step 4 Step Execution Status: **Initial Pass**

Description

SWITCHING TO DDS MODE



If the Command Source does not show DDS, go to the Parameters tab, select DDS under the Command Source and click the Set Cmd Source button.



Note: If the GUI is used after being set to DDS mode, the system will switch back the Command Source to GUI and ignore any DDS commands. The Command Source must show DDS in order to receive DDS commands.

Expected Result

The system is capable of receiving/responding to DDS commands.

Actual Result

See: LVV-E1022

Step 5 Step Execution Status: **Initial Pass**

Description

OFFLINESTATE -> STANDBYSTATE

The system receives an enterControl State Transition command through DDS.

Expected Result

The system transitions into the StandbyState and is capable of receiving/responding to DDS commands.

Actual Result

See: LVV-E1022

Step 6 Step Execution Status: **Initial Pass**

Description

STANDBYSTATE -> DISABLEDSTATE

From the StandbyState, send a start command through the DDS.

Expected Result

The system transitions into DisabledState after receiving/responding to DDS command and the wrapper in the PXI real time controller looks for the configuration file.

If the configuration file is invalid or out of range, the system will transition into a Fault State

Actual Result

See: LVV-E1022

Step 7 Step Execution Status: **Initial Pass**

Description

DISABLEDSTATE -> ENABLEDSTATE

From the DisabledState, send an enable state command through the DDS.

Expected Result

The system transitions into the EnabledState/Stationary substate, the motor drives are enabled, motor brakes are released and the system is capable of receiving/responding to DDS commands.

Actual Result

See: LVV-E1022

Step 8 Step Execution Status: **Initial Pass**

Description

FAULTSTATE

If a Fault occurs in any of the other states, the system will automatically transition to the Fault State. While in the Fault state, send a clearError command through the DDS.

Note: If the fault that occurs goes through the interlock system, reset the safety relay switch and send a clearError command.

Expected Result

The system transitions back to the OfflineState/PublishOnly substate and is not capable of receiving/responding to DDS commands. (Go back to Step 3)

Actual Result

See: LVV-E1022

Step 9 Step Execution Status: **Initial Pass**

Description

Verify that the thermal sensors are connected and producing telemetry into the EFD.

Expected Result

All actuator temperatures are published to the EFD.

Actual Result

Deviation: Thermal sensors were not available during this test. Instead, a waiting time of at least 39sec between each move was used to ensure that the hexapod does not overheat.

Step 10 Step Execution Status: **Initial Pass**

Description

The following steps define what the Jupyter Notebook for this test case implements. Executing the Jupyter notebook is the only actual command and control step that needs to be executed.

Expected Result

The Jupyter notebook controls the system to run through the steps below.

Actual Result

Yes, a Jupiter notebook can be used to control the hexapod. Notebook is named lvv-t1802.ipynb

---

Step 11 Step Execution Status: **Initial Pass**

Description

Verify all the telemetry is being ingested into the EFD.

Expected Result

All telemetry defined in the script is being ingested into the EFD.

Actual Result

Deviation:

- CSC and KAFKA producer needed to be restarted

Step 12 Step Execution Status: **Pass**

Description

MOVE TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #1 - Synchronous PositionSet and Move Commands

In enabled/stationary state, send a positionSet command of (0um, 0um, 200um, 0 deg, 0 deg, 0 deg, s).

Test Data

Deviation: Skip this step. positionSet and move command replaced by new move command. Now, the hexapod starts movement directly after receiving the command.

Expected Result

The hexapod does not move.

Actual Result

Step 13	Step Execution Status: **Pass**

Description

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (500um, -500um, 200um, 0.01deg, -0.015deg, 0deg).

Test Data

Deviation: Skip this step. positionSet and move command replaced by new move command. Now, the hexapod starts movement directly after receiving the command.

Expected Result

The hexapod does not move

Actual Result

Step 14	Step Execution Status: **Initial Pass**

Description

With the hexapod in in enabled/stationary state sync=True and send the move command of (x= 500um,y= -500um, z=200um, u=0.01deg, v=-0.015deg, w=0deg).

Expected Result

- The hexapod moves to (x= 500um,y= -500um, z=200um, u=0.01deg, v=-0.015deg, w=0deg)

- Since the Hexapod is in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result
The M2 hexapod moved as expected to the commanded position (x=500,y=-500,z=200, u=0.01,v=-0.015,w=0,sync=True) all actuators reached the position nearly at the same time.

Results from the EFD from before and after the movement:

hex position
0.05 0.30 -0.36 -0.00 0.00 0.00

hex position
500.24 -500.36 199.90 0.01 -0.02 0.00

Step 15	Step Execution Status: **Fail**

Description
Record the corresponding DDS events that were generated.

Expected Result

- The controllerState.enabledSubstate goes to MOVING_POINT_TO_POINT when the move begins and STATIONARY when the move ends.
- An inPosition event is generated when the move is complete

Actual Result

- Expected results from the EFD:
  1. EnabledSubstate.MOVING_POINT_TO_POINT
  2. EnabledSubstate.STATIONARY
- Fail: The *inPosition* events were not generated.

Issues found executing this step:

- LVV-19512 M2 Hexapod SAL verification failures

Step 16 Step Execution Status: **Pass**

Description

Wait 39 seconds.

Expected Result

Actual Result

Step 17 Step Execution Status: **Initial Pass**

Description

Record the corresponding thermal sensors and verify they are below 19 deg C. If they are above 19 deg C, wait until they are below 19 deg C to perform the following steps.

Expected Result

All actuators are below 19 deg C.

Actual Result

Sensors not available.

Step 18 Step Execution Status: **Initial Pass**

Description

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #5 - Stop Commands

In the enabled/stationary state, send a move command of (x=0um, y=0um, z=5000um, u=0deg, v=0deg, w=0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Hexapod started to move.

Step 19	Step Execution Status: **Initial Pass**

Description

Wait 3s.

Expected Result

Actual Result

This is a wait statement in the code.

Step 20	Step Execution Status: **Initial Pass**

Description

Send a stop command.

Expected Result

- The hexapod stops before reaching the previously commanded position

Actual Result

Hexapod stopped

Step 21	Step Execution Status: **Fail**

Description

Record the corresponding DDS events that were generated.

Expected Result

- The controllerState.enabledSubstate goes to CONTROLLED_STOPPING when the stop is requested, then STATIONARY when the hexapod has halted.
- No inPosition event is generated.

Actual Result

The hexapod moved and stop before finishing the movement.

FAIL: controllerState.enabledSubstate goes to CONTROLLED_STOPPING : CONTROLLED_STOPPING not reported.

Results from the EFD from before and after the movement:

```
INFO:Script:START- M2 Hexapod Integration Test -- LVV-T1802 Test Step 11

hex position

      0.16        0.45       -0.34       -0.00        0.00        0.00
```

```
EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

INFO:Script:STOP- M2 Hexapod Integration Test -- LVV-T1802 Test Step 11

hex position

    -9.43       17.09     1241.17        0.00        -0.00        0.00
```

Issues found executing this step:

- LVV-19512 M2 Hexapod SAL verification failures

Step 22	Step Execution Status: **Pass**

Description

Wait 39 seconds.

Expected Result

Actual Result

Step 23	Step Execution Status: **Initial Pass**

Description

Record the corresponding thermal sensors and verify they are below 19 deg C. If they are above 19 deg C, wait until they are below 19 deg C to perform the following steps.

Expected Result

All actuators are below 19 deg C.

Actual Result

see above.

Step 24	Step Execution Status: **Initial Pass**

Description

Section 3.1.2 of the attached Software Acceptance Test Procedure
Test Sequence #9 - positionSet and moveLUT

Update: Test the "setCompensationMode" command.

In enabled/stationary state, send a move command of (x=0um, y=0um, z=800um, u=0deg, v=0deg, w=0deg)

Test Data

Deviation: There is no "positionSet" and no "moveLUT" command anymore. "positionSet" and "move" command replaced by new "move" command. Now, the hexapod starts movement directly after receiving the command. moveLUT is replaced by a "setCompensationMode".

Expected Result

The hexapod moves to the position (x=0um, y=0um, z=800um, u=0deg, v=0deg, w=0deg) and, since we are moving in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result

Hexapod moved back to (x=0,y=0,z=0, u=0,v=0,w=0,sync=True).

Wait 39sec.

Move to (x=0um, y=0um, z=800um, u=0deg, v=0deg, w=0deg) worked.

Results from the EFD from after the movement:

```
INFO:Script:START- M2 Hexapod Integration Test -- LVV-T1802 Test Step 17



EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY



hex position

     -0.09        0.00      799.94       -0.00       -0.00        0.00
```

Step 25 Step Execution Status: **Initial Pass**

Description

Ensure that MTMount publishes the telescope elevation angle and MTRotator publishes the rotation angle of the rotator. Either as real components or through controllers simulating the components.

Expected Result

Published telescope elevation and rotator angle.

Actual Result

Step 26	Step Execution Status: **Initial Pass**

Description

In enabled/stationary state, set "setCompensationMode" command to enable=True.

Expected Result

The hexapod does not move and the MTHexapod.command_setCompensationMode appears as true in the EFD.

logevent_compensatedPosition is sent to the EFD.

Actual Result

Compensation mode switched activated on demand.

Results from the EFD from after the movement:

```
compsensation mode enabled? True

hex position

     -0.09        0.00       799.94       -0.00        -0.00        0.00

EnabledSubstate.MOVING_POINT_TO_POINT
```

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

`EnabledSubstate.MOVING_POINT_TO_POINT`

`EnabledSubstate.STATIONARY`

`EnabledSubstate.MOVING_POINT_TO_POINT`

`EnabledSubstate.STATIONARY`

`EnabledSubstate.MOVING_POINT_TO_POINT`

`EnabledSubstate.STATIONARY`

`EnabledSubstate.MOVING_POINT_TO_POINT`

`EnabledSubstate.STATIONARY`

`EnabledSubstate.MOVING_POINT_TO_POINT`

Step 27 Step Execution Status: **Fail**

Description

In enabled/stationary state, send a move command of (0um, 0um, 800um, 0deg, 0deg, 0deg)

Expected Result

The hexapod moves to a slightly different position than (0um, 0um, 800um, 0deg, 0deg, 0deg) and, since we are moving in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result

- Fail: Sending a move command while in compensation mode it failed. Command arrives probably during:

  `EnabledSubstate.MOVING_POINT_TO_POINT`

In this substate, no move command can be received?

- Also, the code does not execute the next lines to stop the compensation mode. It can be stopped with a separately executed setCompensationMode=False command.

Issues found executing this step:

- LVV-19512 M2 Hexapod SAL verification failures

Step 28 Step Execution Status: **Fail**

Description

Check if there are any different events between move with and without setCompensationMode=True. Check the movement in the EFD use:
Compare logevent_compensatedPosition to logevent_uncompensatedPosition

Expected Result

The changes are expected according to this table:

zenith angle		M2 Hexapod motions					
		um	um	um	deg	deg	deg
deg	rads	dx	dy	dz	rx	ry	rz
90	**1.570796**	2.942346	556.6612	-656.9706	0.006705	-2.2133E-05	-9.264E-05
85	**1.48353**	2.133244	556.057	-567.0034	0.006638	-1.8487E-05	-7.4613E-05
80	**1.396263**	1.366087	546.5259	-478.2827	0.006471	-1.4965E-05	-5.74668E-05
75	**1.308997**	0.646713	528.1404	-391.4837	0.006206	-1.1593E-05	-4.13318E-05
70	**1.22173**	-0.019403	501.0403	-307.2671	0.005845	-8.3957E-06	-2.63309E-05
65	**1.134464**	-0.62719	465.4319	-226.2737	0.00539	-5.3987E-06	-1.25782E-05
60	**1.047198**	-1.172025	421.5862	-149.1199	0.004845	-2.6245E-06	-1.78402E-07
55	**0.959931**	-1.649759	369.837	-76.39305	0.004214	-9.4085E-08	1.07741E-05
50	**0.872665**	-2.056758	310.5781	-8.646518	0.003502	2.1732E-06	2.0196E-05
45	**0.785398**	-2.389924	244.2603	53.60408	0.002713	4.1601E-06	2.80156E-05
40	**0.698132**	-2.646721	171.3886	109.885	0.001856	5.8515E-06	3.41734E-05
35	**0.610865**	-2.825195	92.51743	159.7678	0.000934	7.2345E-06	3.86224E-05
30	**0.523599**	-2.923987	8.247089	202.873	-4.29E-05	8.2987E-06	4.13289E-05
25	**0.436332**	-2.942346	-80.78108	238.8724	-0.001069	9.0359E-06	4.22722E-05
20	**0.349066**	-2.880131	-173.8895	267.4922	-0.002136	9.4405E-06	4.14452E-05
15	**0.261799**	-2.737817	-270.3696	288.5144	-0.003236	9.5094E-06	3.88542E-05
10	**0.174533**	-2.516487	-369.4871	301.7791	-0.004361	9.2421E-06	3.45188E-05
5	**0.087266**	-2.217825	-470.4876	307.1853	-0.005501	8.6406E-06	2.84721E-05
0	**0**	-1.844103	-572.6024	304.692	-0.006649	7.7096E-06	2.076E-05

Actual Result

- Fail???: No compensation was applied. Results from EFD in the notebook and in Chronograph are the same.

BUT this might be a consequence of a missing "*inPosition*" event. See step 8.

- INFO:Script:START- M2 Hexapod Integration Test – LVV-T1802 Test Step 17
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- hex position
- -0.09 0.00 799.94 -0.00 -0.00 0.00
-
- compsensation mode enabled? True
- hex position
- -0.09 0.00 799.94 -0.00 -0.00 0.00
-
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- EnabledSubstate.STATIONARY
- EnabledSubstate.MOVING_POINT_TO_POINT
- INFO:Script:STOP- M2 Hexapod Integration Test – LVV-T1802 Test Step 17
- hex position
- -0.09 0.00 799.94 -0.00 -0.00 0.00

Step 29	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send again the same move command of (0um, 0um, 800um, 0deg, 0deg, 0deg)

Expected Result

The hexapod does not move since it stayed in compensationMode.

Actual Result

No tested

Step 30	Step Execution Status: **Pass**

Description

Wait 39 seconds.

Expected Result

Actual Result

Step 31	Step Execution Status: **Initial Pass**

Description

Record the corresponding thermal sensors and verify they are below 19 deg C. If they are above 19 deg C, wait until they are below 19 deg C to perform the following steps.

Expected Result

All actuators are below 19 deg C.

Actual Result

see above

Step 32	Step Execution Status: **Initial Pass**

Description

OFFSET TEST
Section 3.1.2 of the attached Software Acceptance Test Procedure
Test Sequence #4 - Synchronous Offset and Move Commands
In enabled/stationary state, send a move command of (x=500um, y=800um, z=200um, u=0deg, v=0deg, w=0deg)

Test Data
Deviation: There is no positionSet command anymore. positionSet and move command replaced by new move command. Now, the hexapod starts movement directly after receiving the command.

Expected Result

- The hexapod moves to (x=500um, y=800um, z=200um, u=0deg, v=0deg, w=0deg)
- Since the Hexapod is in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result
Movement successful to position (x=500um, y=800um, z=200um, u=0deg, v=0deg, w=0deg):

```
INFO:Script:START- M2 Hexapod Integration Test -- LVV-T1802 Test Step 24

hex position

     -0.03       -0.21       -0.22        0.00        0.00        0.00

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY
```

```
hex position

    500.05      800.33      199.87      -0.00       -0.00       0.00
```

Step 33 Step Execution Status: **Initial Pass**

Description

In enabled/stationary state, send an offset command of (0um, 0um, 500um, 0deg, 0deg, 0deg).

Expected Result

- The hexapod moves only 500um in Z from the previous position
- The actuators complete the move at nearly the same time.

Actual Result

Offset successfully applied of (0um, 0um, 500um, 0deg, 0deg, 0deg):

```
EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

INFO:Script:STOP- M2 Hexapod Integration Test -- LVV-T1802 Test Step 24

hex position

    500.37      799.63      699.90      -0.00       -0.00       -0.00
```

Step 34 Step Execution Status: **Initial Pass**

Description

Send a move command.

Test Data

Deviation: Skip this step. The Hexapod has already moved.

Expected Result

- The hexapod moves only 500um in Z from the previous position
- The actuators complete the move at nearly the same time.

Actual Result

Step 35	Step Execution Status: **Pass**

Description

Wait 39 s

Expected Result

Actual Result

Step 36	Step Execution Status: **Fail**

Description

Record the corresponding DDS events that were generated.

Expected Result

- The controllerState.enabledSubstate goes to MOVING_POINT_TO_POINT when the move begins and STATIONARY when the move ends
- The inPosition event is True when the move finishes
- The inPosition event is False when the enabledSubstate goes back to STATIONARY.

Actual Result

- Results from the EFD:

1. EnabledSubstate.MOVING_POINT_TO_POINT
2. EnabledSubstate.STATIONARY

- Fail: The inPosition events were not generated.

Issues found executing this step:

- LVV-19512 M2 Hexapod SAL verification failures

Step 37 Step Execution Status: **Initial Pass**

Description

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #2 -Pivot, PositionSet and Move Commands

In enabled/stationary state, send a move command of (x=2000um,y=-3500um,z=200um,u=0.01deg,v=-0.05deg, w=0.002deg,sync=true)

Test Data

Deviation: Record any offset commands necessary to test before sending the move command.

Expected Result

The hexapod moves to the commanded position

Actual Result

Hexapod reach the position

Step 38 Step Execution Status: **Initial Pass**

Description

In the enabled/stationary state, send the setPivot command of (0,0,0).

Expected Result

The actuator positions do not change but the hexapod position changes to account for the new pivot point.

Actual Result

The EUI showed the new hexapod position accounting for the new pivot point.

Step 39	Step Execution Status: **Fail**

Description

In the enabled/stationary state, send again the move command of (x=2000um, y=-3500um, z=200um, u=0.01deg, v=-0.05deg, w=0.002deg,sync=true)

Test Data

Deviation: Record any offset commands necessary to test before sending the move command.

Expected Result

The hexapod doesn't move. Position values in the EFD appear different.

Actual Result

Fail: ??? Change of Pivot point does not seem to have any effect on values:

```
INFO:Script:START- M2 Hexapod Integration Test -- LVV-T1802 Test Step 29

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY

hex position

   2000.33    -3500.21      199.97        0.01       -0.05        0.00



Change Pivot point and move

EnabledSubstate.MOVING_POINT_TO_POINT

EnabledSubstate.STATIONARY
```

```
hex position

   2000.33    -3500.18      199.96        0.01        -0.05         0.00



INFO:Script:STOP- M2 Hexapod Integration Test -- LVV-T1802 Test Step 29

hex position

   2000.33    -3500.18      199.96        0.01        -0.05         0.00



But: Is this a consequence from failing inPosition? See Step 8.
```

Step 40 Step Execution Status: **Pass**

Description

Send a move command.

Test Data

Deviation: This step is obsolete. Hexapod already moved.

Expected Result

Confirm the hexapod moves to the commanded position and the actuators change position to account for the new pivot point.

Actual Result

Step 41 Step Execution Status: **Pass**

Description

Wait 39s.

Expected Result

Actual Result

Step 42	Step Execution Status: **Not Executed**

Description

CONFIGURE LIMITS TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #6 - configureLimits Command

In enabled/stationary state, send a configureLimits command of (12000um, -1000um, 1000um, 0.1, -0.1, 0.05)

Test Data

Deviation: Skip complete test. This test uses an obsolete command. The configuration is now done before and should not be changed this state

Expected Result

The command is rejected for being outside acceptable limits.

Actual Result

Skipped: Commands and mechanism obsolete

Step 43	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a configureLimits command of (1000um, -1000um, 1000um, 0.1, -0.1, 0.05)

Expected Result

The command is accepted.

Actual Result

Step 44	Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 45 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a positionSet command of (850um, 0um, 500um, 0deg, 0deg, 0deg)

Test Data

Deviation: This command can be any valid positionSet command within the newly configured limits.

Expected Result

The command is accepted.

Actual Result

Step 46 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 47 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a positionSet command of (1200um, 0um, 200um, 0deg, 0deg, 0deg)

Expected Result

The command is rejected for being outside of range limits

Actual Result

Step 48 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The Hexapod doesn't move.

Actual Result

Step 49 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a positionSet command of (990um, 990um, 200um, 0deg, 0deg, 0deg)

Expected Result

The command is rejected for being outside of range limits.

Actual Result

Step 50 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a positionSet command of (500um, 500um, 200um, 0deg, 0.1 deg, 0.01deg)

Expected Result

The command is accepted.

Actual Result

Step 51	Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The previously accepted command is executed.

Actual Result

Step 52	Step Execution Status: **Not Executed**

Description

Wait 39s

Expected Result

Actual Result

Step 53	Step Execution Status: **Not Executed**

Description

Record the DDS events that were generated.

Expected Result

The change is reflected in the settingsApplied event and the EUI.

Actual Result

Step 54	Step Execution Status: **Not Executed**

Description

CONFIGURE ACCELERATION TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #7 - configureAcceleration Command

In enabled/stationary state, at a position of (0, 0, 0, 0, 0, 0) with the velocity and acceleration values set to their nominal values, send a positionSet command of (0um, 0um, 4900um, 0 deg, 0 deg, 0 deg, s).

Test Data

Deviation: Skip complete test. This test uses an obsolete command. The configuration is now done before and should not be changed this state

Expected Result

The hexapod doesn't move.

Actual Result

Skipped: Commands and mechanism obsolete

Step 55 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The move takes approximately 9 seconds to complete.

Actual Result

Step 56 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 57 Step Execution Status: **Not Executed**

Description

Send a configureAcceleration command of 1000.

Expected Result

Confirm command is rejected for being outside of acceptable limits.

Actual Result

Step 58 Step Execution Status: **Not Executed**

Description

Send a configureAcceleration command of 100.

Expected Result

The command is accepted.

Actual Result

Step 59 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a postionSet command of (0um, 0um, 0um, 0 deg, 0 deg, 0 deg, s).

Expected Result

The hexapod doesn't move.

Actual Result

Step 60 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

It takes approximately 13 seconds to complete the commanded move with the reduced acceleration value.

Actual Result

Step 61	Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 62	Step Execution Status: **Not Executed**

Description

Send a configureAcceleration command of 500 to return the acceleration limit to its nominal value.

Expected Result

The command is accepted.

Actual Result

Step 63	Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated.

Expected Result
The change is reflected in the settingsApplied event and the EUI.

Actual Result

Step 64 Step Execution Status: **Not Executed**

Description
CONFIGURE VELOCITY TEST
Section 3.1.2 of the attached Software Acceptance Test Procedure
Test Sequence #8 - configureVelocity Command
In enabled/stationary state, at a position of (0, 0, 0, 0, 0, 0), send a configureVelocity command of (10000, .01, 100, .01).

Test Data
Deviation: Skip complete test. This test uses an obsolete command. The configuration is now done before and should not be changed this state

Expected Result
This command is rejected for being outside of acceptable limits.

Actual Result
Skipped: Commands and mechanism obsolete

Step 65 Step Execution Status: **Not Executed**

Description
In enabled/stationary state, send a configureVelocity command of (100, .01, 200, .01).

Expected Result
This command is accepted.

Actual Result

Step 66 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a positionSet command of (0, 0um, 2000um, 0 deg, 0 deg, 0 deg, s).

Expected Result

The command is accepted

Actual Result

Step 67 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

It takes approximately 20 seconds to complete the commanded move.

Actual Result

Step 68 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 69 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a configureVelocity command of (100, .01, 100, .01).

Expected Result

This command is accepted.

Actual Result

Step 70	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send an offset command of (0, 0um, 2000um, 0 deg, 0 deg, 0 deg).

Expected Result

This command is accepted

Actual Result

Step 71	Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

It takes approximately 40 seconds to complete the commanded move.

Actual Result

Step 72	Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 73 Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated:

Expected Result

The change is reflected in the settingsApplied event and the EUI.

Actual Result

Step 74 Step Execution Status: **Not Executed**

Description

Section 3.3.2 of the attached Software Acceptance Test Procedure Hexapod Action on State Commands

In the Offline/PublishOnly state, send all commands

Expected Result

There is no change and command is rejected.

Actual Result

: State machine not yet conform with the OfflineState entry state machine. Test after update.

Step 75 Step Execution Status: **Not Executed**

Description

In the Offline/Available state, send an enterControl command

Expected Result

The system enters the Standby state.

Actual Result

Step 76	Step Execution Status: **Not Executed**

Description

In the Standby state, send any command except start or exitControl

Expected Result

There is no change and command is rejected.

Actual Result

Step 77	Step Execution Status: **Not Executed**

Description

In the Standby state, send an exitControl command.

Expected Result

The system transitions into the Offline/Available state.

Actual Result

Step 78	Step Execution Status: **Not Executed**

Description

In the Standby state, send a start command.

Expected Result

The system transitions into the Disabled state.

Actual Result

Step 79	Step Execution Status: **Not Executed**

Description

In the Disabled state, send any command except for the enabled or standby command.

Expected Result
There is no change and the command is rejected.

Actual Result

Step 80	Step Execution Status: **Not Executed**

Description
In the Disabled state, send the standby command.

Expected Result
The system transitions into the Standby state.

Actual Result

Step 81	Step Execution Status: **Not Executed**

Description
In the Disabled state, send the enable command.

Expected Result
The system transitions into the Enabled/Stationary state.

Actual Result

Step 82	Step Execution Status: **Not Executed**

Description
In the Enabled/Stationary state, send either the enterControl command, exitControl command, start command, clearError command, or enable command.

Expected Result
There is no change and command is rejected.

Actual Result

Step 83 Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, send a disable command.

Expected Result

The system transitions into Disabled state.

Actual Result

Step 84 Step Execution Status: **Not Executed**

Description

In the Fault state, send any command except the clearError command.

Expected Result

There is no change and command is rejected.

Actual Result

Step 85 Step Execution Status: **Not Executed**

Description

In the Fault state, send the clearError command.

Expected Result

The system transitions from Faultstate to Offlinestate only when the system was in Offlinestate originally. Otherwise, it transitions to standby.

The system, receiving a ClearError trigger, transitions to Standbystate when it was in Enablestate or Disablestate bevor.

Actual Result

Step 86 Step Execution Status: **Not Executed**

Description

Section 4 of the attached Software Acceptance Test Procedure

In the Enabled/Stationary state, unplug a motor encoder cable for one of the actuators.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

not tested

Step 87 Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, unplug a linear encoder cable for one of the actuators.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

not tested

Step 88 Step Execution Status: **Not Executed**

Description

Unplug a motor power cable from one of the actuators and command a Move.

Expected Result

A Following Error event is created and the system transitions to Fault state.

Actual Result

not tested

Step 89 Step Execution Status: **Pass**

Description

Activate an extension limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result

An Extended Limit Switch error event is created and the system transitions into Fault state.

Actual Result

- Done by Felipe when changing the limit switch logic to normally closed. The extended Limit Switch error event was created and the system transitioned into a Fault state.

Step 90 Step Execution Status: **Pass**

Description

Activate a retraction limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result

A Retracted Limit Switch error event is created and the system transitions into Fault state.

Actual Result

- Done by Felipe when changing the limit switch logic to normally closed. The extended Limit Switch error event was created and the system transitioned into a Fault state.

Step 91 Step Execution Status: **Not Executed**

Description

Unplug the Ethercat cable between the control PC and the first Copley XE2 drive.

Expected Result

An Ethercat Lost event is created and the system transitions to Fault state.

Actual Result

not tested

5.2 Test Cycle LVV-C203

Open test cycle *M2 Hexapod Re-verification on the TMA* in Jira.

Test Cycle name: M2 Hexapod Re-verification on the TMA
Status: Not Executed

M2 hexapod location has changed for this installed on the TMA.
New cabling for the hexapod actuators is installed.
Low-level controller, EUI, and CSC Software are updated.

5.2.1 Software Version/Baseline

1. M2 Hexapod Control Software with SAL v4.0 or higher
2. EFD with SAL v4.0 or higher

5.2.2 Configuration

No varying configuration for the hexapod and the mass simulator between test cycles.

5.2.3 Test Cases in LVV-C203 Test Cycle

5.2.3.1 LVV-T1804 - M2 Hexapod Software Functional Re-verification

Version **1**. Open *LVV-T1804* test case in Jira.

The objective of this test case is to re-verify the functional requirements of the M2 hexapod's software, after shipment of the hardware from the vendor's facility to the Summit, as defined in LTS-206 and LTS-160. This test case will only exercise the functionality that was executed previously and meets the following criteria:

- Only requires the M2 hexapod to be operable
- Only requires testing the synchronous mode
  - **Asynchronous mode is not a standard mode of operation**
- Only requires the vendor's EUI software and hardware via local control
  - Does **NOT** require integration with SAL
- Does **NOT** require the M2 hexapod to be rotated to various elevation angles.
- This test case can be executed with a **simulated M2 mass or actual M2 hardware**

The software functional requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by Rubin Observatory during the Factory Acceptance Test review. The test procedure used during the vendor's acceptance testing is the *LSST Hexapods-Rotator Software Acceptance Test Procedure* which is attached to this test case. The test steps of this test case are taken directly from that document on how to perform the test in a similar way as was performed previously and the test steps include changes noted by the vendor.

See the attached *LSST Hexapod Operator's Manual* for more information on how to operate the hexapod.

Preconditions:
Prior to the execution of this test case to re-verify the M2 Hexapod hardware functional requirements, the following Summit tasks must be completed:

- The measurement equipment has been set up for testing
  - `https://jira.lsstcorp.org/browse/SUMMIT-1943`

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description

Release the Lock-Out-Tag-Out (LOTO) for the Rotator Circuit and the Hexapod Circuit using the keys stored in Cabinet 2 at Level 3.

Expected Result

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Access the Netbooter for the Rotator and Hexapod and turn on the Drivers and the CPU for each component in this order.
Here are the addresses for the netbooters:
rot-netbooter.cp.lsst.org
camhex-netbooter.cp.lsst.org
The user/password is admin/admin.

Expected Result

Actual Result

Step 3 Step Execution Status: **Not Executed**

Description

Raise (disengage) the E-Stops for the CCW and HexRot.
Announce on #summit-announce Slack Channel.

Expected Result

Actual Result

Step 4 Step Execution Status: **Not Executed**

Description

Use the little black knob in the Hexapod Cabinet to reset the power of its controller.

Expected Result

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

STARTING THE EUI

Connect to the hexapod management computer using X2Go using your IPA/VPN credentials.
Open a terminal.
Change to the folder

cd /rubin/hexapod/build/

Start the EUI with the command:

./runCamHexEui

for the camera and the M2 hexapod, respectively.

Example Code

```
# Check if CamHex EUI is running
ps -aux | grep runCamHex


cd /rubin/rotator/build
./runRotEui
```

Check if M2Hex EUI is running
ps -aux | grep runM2Hex

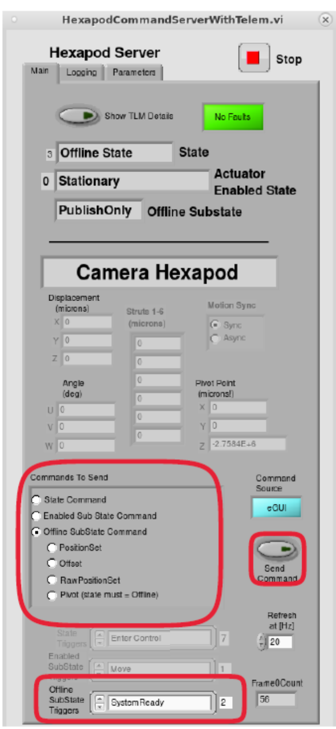
Expected Result
The EUI is in the Offline State/PublishOnly substate and is able to publish through SAL but cannot receive commands.

Actual Result

Step 6	Step Execution Status: **Not Executed**

Description
Transition from OFFLINE State/Publish Only Substate to OFFLINE State/Available Substate.

On the Main tab, select the **Offline SubState Command** radio button field in the **Commands to Send** section.
Then, set the **Offline SubState Triggers** to **System Ready**.
Finally, click on the **Send Command** button.



Expected Result

The system transitions from the OfflineState/PublishOnly substate to the OfflineState/AvailableState substate and the Command Source says eGUI.

Actual Result

Step 7	Step Execution Status: **Not Executed**

Description

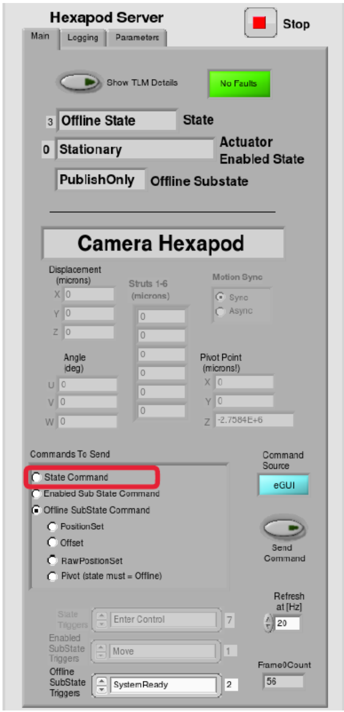
Transition from OFFLINE State to STANDBY State.

On **Commands to Send**, select the **State Commands** radio button**.**

On **State Triggers**, scroll and select **Enter Control**.

Finally, click on the **Send Command** button.

The Hexapod should go to **STANDBY** state.



Expected Result

The system transitions to the Standby state and the primary state display box at the top of the Main says Standby state.

Actual Result

Step 8	Step Execution Status: **Not Executed**

Description

Transition from STANDBY State to DISABLED State.

On **Commands to Send**, select the **State Commands** radio button.

On **State Triggers**, scroll and select the **Start** option.

Click on the **Send Command** button.

The Hexapod should go to DISABLED state.

Expected Result
The system transitions into DISABLED State and the current configuration parameters are maintained from the default parameters or from the previous DDS start command.

Actual Result

Step 9 Step Execution Status: **Not Executed**

Description
Transition from DISABLED State to ENABLED State.

On **Commands to Send**, select the **State Commands** radio button.

On **State Triggers**, scroll and select the **Enabled** option.

Click on the **Send Command** button.

The Hexapod should go to ENABLED state.

Expected Result
The system transitions into the ENABLED State/Stationary sub-state, the motor drives are enabled and motion can be commanded.

Actual Result

Step 10 Step Execution Status: **Not Executed**

Description
<conditional state>

Recovering from a FAULT State

If a fault occurs in any of the other states, the system will automatically transition to the FAULT State.
While in the Fault state, select the **clearError** option in the **State Triggers** option list. Then, click on the **Send Command** button**.**

Note: If the fault that occurs goes through the interlock system, reset the safety relay switch and send a **clearError** command.

Expected Result
The system transitions back to the STANDBY state. (Go back to Step 8)

Actual Result

Step 11	Step Execution Status: **Not Executed**

Description
Section 3.3.1 EUI Tests of the attached Software Acceptance Test Procedure
At startup, confirm that the system starts in the Offline/PublishOnly state.

Expected Result
The hexapod starts in the Offline/PublishOnly state.

Actual Result

Step 12	Step Execution Status: **Not Executed**

Description
Send an offline substate trigger of systemReady.

Expected Result
The system transitions into the Offline/Available substate.

Actual Result

Step 13	Step Execution Status: **Not Executed**

Description
Send an EnterControl trigger.

Expected Result
The system transitions from Offline/Available to Standby state.

Actual Result

Step 14	Step Execution Status: **Not Executed**

Description
Send a Start trigger.

Expected Result
The system transitions from Standby to Disabled state.

Actual Result

Step 15	Step Execution Status: **Not Executed**

Description
Send an Enable trigger.

Expected Result
The system transitions from Disabled to Enabled state.

Actual Result

Step 16	Step Execution Status: **Not Executed**

Description
Send a Disable trigger.

Expected Result
The system transitions from Enabled to Disabled state.

Actual Result

Step 17 Step Execution Status: **Not Executed**

Description

Send a Standby trigger.

Expected Result

The system transitions from Disabled state to Standby state.

Actual Result

Step 18 Step Execution Status: **Not Executed**

Description

Send a exitControl trigger.

Expected Result

The system transitions from Standby state to Offline state.

Actual Result

Step 19 Step Execution Status: **Not Executed**

Description

Return to the Enabled state and trip the safety interlock switch.

Expected Result

The system transitions to Fault state.

Actual Result

Step 20 Step Execution Status: **Not Executed**

Description

Reset the safety interlock and send a ClearError trigger.

Expected Result

The CSC, upon receiving the ¨clearError¨ trigger, transitions from FaultState to OfflineState/PublishOnly when the system was in any of the OfflineStates before the error occurred. The CSC, upon receiving the "clearError" trigger, transitions to StandbyState when it was in EnableState or DisableState before the error occurred.

Actual Result

Step 21 Step Execution Status: **Not Executed**

Description

Section 4.1 Hexapod Events of the attached Software Acceptance Test Procedure

In the Enabled/Stationary state, unplug a motor encoder cable for one of the actuators.

Test Data

Deviation: Perform the following set of steps using the EUI instead of the DDS and verify the events are displayed on the EUI.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

Step 22 Step Execution Status: **Not Executed**

Description

Send the "clearError" trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 23 Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, unplug a linear encoder cable for one of the actuators.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

Step 24 Step Execution Status: **Not Executed**

Description

Send the “clearError” trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 25 Step Execution Status: **Not Executed**

Description

Unplug a motor power cable from one of the actuators and command a PositionSet/Move.

Expected Result

A Following Error event is created and the system transitions to Fault state.

Actual Result

Step 26 Step Execution Status: **Not Executed**

Description

Send the “clearError” trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 27 Step Execution Status: **Not Executed**

Description

Activate an extension limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result

An Extended Limit Switch error event is created and the system transitions into Fault state.

Actual Result

Step 28 Step Execution Status: **Not Executed**

Description

Send the "clearError" trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 29 Step Execution Status: **Not Executed**

Description

Activate a retraction limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result

A Retracted Limit Switch error event is created and the system transitions into Fault state.

Actual Result

Step 30 Step Execution Status: **Not Executed**

Description

Send the "clearError" trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 31 Step Execution Status: **Not Executed**

Description

Unplug the Ethercat cable between the control PC and the first Copley XE2 drive.

Expected Result

An Ethercat Lost event is created and the system transitions to Fault state.

Actual Result

Step 32 Step Execution Status: **Not Executed**

Description

Send the "clearError" trigger and bring the system to the Enabled/Stationary state.

Expected Result

The system is in the Enabled/Stationary state and ready to be commanded.

Actual Result

Step 33 Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #1 - Synchronous PositionSet and Move Commands

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (0um, 0um, 200um, 0 deg, 0 deg, 0 deg) using the EUI.

Expected Result

The hexapod doesn't move.

Actual Result

Step 34 Step Execution Status: **Not Executed**

Description

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg) using the EUI.

Expected Result

The hexapod doesn't move.

Actual Result

Step 35 Step Execution Status: **Not Executed**

Description

Send a move command using the EUI.

Test Data

Pivot position is shown in the GUI. Please mention in the results. Use the MOOG pivot point for comparability with the previous results.

Expected Result

The hexapod moves to the last commanded position of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg). Since the test is done in synchronous mode the actuators are expected to complete the move at nearly the same time as seen on the motion complete lights on the telemetry screen.

Actual Result

Step 36 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 37 Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #2 - Pivot, PositionSet and Move Commands

In enabled/stationary state and at the last commanded position of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg), change the pivot point from the default location to (0,0,0) using the EUI.

Expected Result

The actuator positions do not change, but the hexapod position is (-407um, -3982um, 199um, 0.01deg, -0.05deg, 0.002deg)

Actual Result

Step 38 Step Execution Status: **Not Executed**

Description

In the enabled/stationary state, send a positionSet command of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg) using the EUI.

Expected Result

The hexapod doesn't move.

Actual Result

Step 39	Step Execution Status: **Not Executed**

Description

Send a move command using the EUI.

Expected Result

The hexapod moves to the commanded position of (2000um, -3500um, 200um, .01 deg, -.05deg, .002deg) and the actuators change position to account for the new pivot point.

Actual Result

Step 40	Step Execution Status: **Not Executed**

Description

Wait 39s

Expected Result

Actual Result

Step 41	Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #4 - Synchronous Offset and Move Commands

With the synchronous button enabled and in enabled/stationary state, send a positionSet command of (500um, 800um, 200um, 0 deg, 0 deg, 0 deg).

Expected Result
The hexapod doesn't move.

Actual Result

Step 42	Step Execution Status: **Not Executed**

Description
With the synchronous button enabled and in enabled/stationary state, send an offset command of (0um, 0um, 2000um, 0 deg, 0 deg, 0 deg).

Expected Result
The hexapod doesn't move.

Actual Result

Step 43	Step Execution Status: **Not Executed**

Description
Send a move command.

Expected Result
The hexapod moves only 2000um in Z from the previous position. Since the test is done in synchronous mode the actuators are expected to complete the move at nearly the same time as seen on the motion complete lights on the telemetry screen.

Actual Result

Step 44	Step Execution Status: **Not Executed**

Description
Wait 39s

Expected Result

Actual Result

Step 45 Step Execution Status: **Not Executed**

Description

Instead of Asynchronous Test

With the synchronous button enabled and in enabled/stationary state, send a position set command of (0um, 0um, 0um, 0.1deg, 0deg, 0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Step 46 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod moves to the commanded position of (0um, 0um, 0um, 0.1deg, 0deg, 0deg)

Actual Result

Step 47 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 48 Step Execution Status: **Not Executed**

Description

With the synchronous button enabled and in enabled/stationary state, send a position set command of (0um, 0um, 0um, 0deg, 0.1deg, 0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Step 49 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod moves to the commanded position of (0um, 0um, 0um, 0deg, 0.1deg, 0deg)

Actual Result

Step 50 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 51 Step Execution Status: **Not Executed**

Description

With the synchronous button enabled and in enabled/stationary state, send a position set command of (0um, 0um, 0um, 0.1deg, 0.1deg, 0deg)

Expected Result

The hexapod doesn't move.

Actual Result

Step 52 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod moves to the commanded position of (0um, 0um, 0um, 0.1deg, 0.1deg, 0deg)

Actual Result

Step 53 Step Execution Status: **Not Executed**

Description

Wait 39s.

Expected Result

Actual Result

Step 54 Step Execution Status: **Not Executed**

Description

Section 3.1.1 of the attached Software Acceptance Test Procedure
Test Sequence #5 - Stop Commands

In enabled/stationary state, send a position set command of (0um, 0um, 5000um, 0 deg, 0 deg, 0 deg).

Expected Result

The hexapod doesn't move.

Actual Result

Step 55 Step Execution Status: **Not Executed**

Description

Send a move command.

Expected Result

The hexapod starts to move to the commanded position.

Actual Result

Step 56 Step Execution Status: **Not Executed**

Description

Wait 3s.

Expected Result

Actual Result

Step 57 Step Execution Status: **Not Executed**

Description

Send a stop command.

Expected Result

The hexapod quickly comes to a stop prior to reaching the commanded position.

Actual Result

5.2.3.2 LVV-T1800 - M2 Hexapod Hardware Functional Re-verification

Version **1**. Open *LVV-T1800* test case in Jira.

The objective of this test case is to re-verify the functional requirements of the M2 hexapod's hardware, after shipment from the vendor's facility to the Summit, as defined in LTS-206. This test case will only exercise the functionality that was executed previously and meets the following criteria:

- Only requires the M2 hexapod to be operable
- Only requires the EUI software and hardware via local control
- Requires a laser tracker, mechanical gauges, inductive current probe
- Does require the M2 hexapod temperature sensors to be operating
- Does **NOT** require the M2 hexapod to be rotated to various elevation angles
- Does **NOT** require the M2 hexapod to be in a climate-controlled environment

The hardware functional requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by Rubin Observatory during the Factory Acceptance Test review. The test procedure used during the vendor's acceptance testing is the *LSST Hexapods-Rotator Acceptance Test Procedure* which is attached to this test case. The test steps of this test case are taken directly from that document on how to perform the test in a similar way as was performed previously and the test steps include changes noted by the vendor.

See the attached *LSST Hexapod Operator's Manual* for more information on how to operate the hexapod.

Preconditions:

Prior to the execution of this test case to re-verify the M2 Hexapod hardware functional requirements, the following Summit tasks must be completed:

- The measurement equipment has been set up for testing
  - `https://jira.lsstcorp.org/browse/SUMMIT-1943`
- The laser tracker has been set up for measurements
  - `https://jira.lsstcorp.org/browse/SUMMIT-3951`

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description

Release the Lock-Out-Tag-Out (LOTO) for the Rotator Circuit and the Hexapod Circuit using the keys stored in Cabinet 2 at Level 3.

Expected Result

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Access the Netbooter for the Rotator and Hexapod and turn on the Drivers and the CPU for each component in this order.
Here are the addresses for the netbooters:
rot-netbooter.cp.lsst.org
camhex-netbooter.cp.lsst.org
The user/password is admin/admin.

Expected Result

Actual Result

Step 3 Step Execution Status: **Not Executed**

Description

Raise (disengage) the E-Stops for the CCW and HexRot.
Announce on #summit-announce Slack Channel.

Expected Result

Actual Result

Step 4 Step Execution Status: **Not Executed**

Description

Use the little black knob in the Hexapod Cabinet to reset the power of its controller.

Expected Result

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

STARTING THE EUI

Connect to the hexapod management computer using X2Go using your IPA/VPN credentials.
Open a terminal.
Change to the folder

cd /rubin/hexapod/build/

Start the EUI with the command:

./runCamHexEui

for the camera and the M2 hexapod, respectively.

Example Code

```
# Check if CamHex EUI is running
ps -aux | grep runCamHex


cd /rubin/rotator/build
./runRotEui


# Check if M2Hex EUI is running
ps -aux | grep runM2Hex
```

Expected Result

The EUI is in the Offline State/PublishOnly substate and is able to publish through SAL but cannot receive commands.

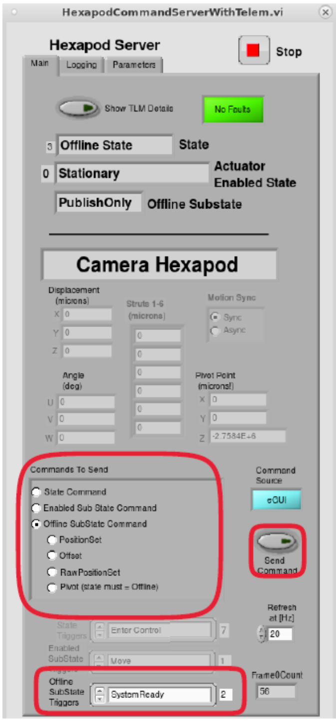
Actual Result

Step 6	Step Execution Status: **Not Executed**

Description

Transition from OFFLINE State/Publish Only Substate to OFFLINE State/Available Substate.

On the Main tab, select the **Offline SubState Command** radio button field in the **Commands to Send** section.
Then, set the **Offline SubState Triggers** to **System Ready**.
Finally, click on the **Send Command** button.



Expected Result

The system transitions from the OfflineState/PublishOnly substate to the OfflineState/AvailableState substate and the Command Source says eGUI.

Actual Result

Step 7 Step Execution Status: **Not Executed**

Description

Transition from OFFLINE State to STANDBY State.

On **Commands to Send**, select the **State Commands** radio button**.**

On **State Triggers**, scroll and select **Enter Control**.

Finally, click on the **Send Command** button.

The Hexapod should go to **STANDBY** state.

Expected Result

The system transitions to the Standby state and the primary state display box at the top of the Main says Standby state.

Actual Result

Step 8 Step Execution Status: **Not Executed**

Description

Transition from STANDBY State to DISABLED State.

On **Commands to Send**, select the **State Commands** radio button.

On **State Triggers**, scroll and select the **Start** option.

Click on the **Send Command** button.

The Hexapod should go to DISABLED state.

Expected Result
The system transitions into DISABLED State and the current configuration parameters are maintained from the default parameters or from the previous DDS start command.

Actual Result

Step 9	Step Execution Status: **Not Executed**

Description
Transition from DISABLED State to ENABLED State.

On **Commands to Send**, select the **State Commands** radio button.

On **State Triggers**, scroll and select the **Enabled** option.

Click on the **Send Command** button.

The Hexapod should go to ENABLED state.

Expected Result
The system transitions into the ENABLED State/Stationary sub-state, the motor drives are enabled and motion can be commanded.

Actual Result

Step 10	Step Execution Status: **Not Executed**

Description
<conditional state>

Recovering from a FAULT State

If a fault occurs in any of the other states, the system will automatically transition to the FAULT State.
While in the Fault state, select the **clearError** option in the **State Triggers** option list. Then, click on the **Send Command** button**.**

Note: If the fault that occurs goes through the interlock system, reset the safety relay switch and send a **clearError** command.

Expected Result

The system transitions back to the STANDBY state. (Go back to Step 8)

Actual Result

Step 11 Step Execution Status: **Not Executed**

Description

Follow *3.5.12 Positioning* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 57-58.

Test Data

Deviation: Test at a single elevation angle.

Expected Result

The position of the hexapod is able to reach the commanded positions within the absolute accuracy specifications of 25um in Z, 125um in XY, 83x10-5deg in RXRY, and 750x10-5deg in RZ.

Actual Result

Step 12 Step Execution Status: **Not Executed**

Description

Follow *3.5.15 Radial (X and Y) Translation Range* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 59.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result

The hexapod is capable of moving to the positions in the XY plane listed in the Acceptance Test Procedure.

Actual Result

Step 13 Step Execution Status: **Not Executed**

Description

Follow *3.5.13 Centers of Rotation* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 58-59.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement. The spherically mounted retroreflector (SMR) will be mounted on the ring holding the M2 mass surrogate or the M2 mass simulator

Expected Result

The center of rotation is able to be moved.

Actual Result

Step 14 Step Execution Status: **Not Executed**

Description

Follow *3.5.17 Axial (Z) Translation Range* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 60.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result

The hexapod is capable of moving to the positions in the Z plane listed in the Acceptance Test Procedure.

Actual Result

Step 15 Step Execution Status: **Not Executed**

Description

Follow *3.5.19 Rotational Range Around X-Axis (Tip) and Y-Axis (Tilt)* of the LSST Hexapods-Rotator Acceptance

Test Procedure, Sheet 61.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result

The hexapod is capable of moving to the positions in the RXRY plane listed in the Acceptance Test Procedure.

Actual Result

Step 16	Step Execution Status: **Not Executed**

Description

Follow *3.5.21 Rotation Range Around Z-Axis (Twist)* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 62.

Test Data

Deviation: Test at a single elevation angle. Wait for 39s between each movement.

Expected Result

The hexapod is capable of moving to the positions in the RZ-axis listed in the Acceptance Test Procedure.

Actual Result

Step 17	Step Execution Status: **Not Executed**

Description

Follow *3.5.23 Hexapod Repeatability* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 63-70.

Test Data

Deviation: Allow a minimum of 30 seconds between moves.

Expected Result

The repeatability of the hexapod is likely better than can be determined by the test equipment. This test will likely falsely show a deficiency in the hexapod performance as a result of test equipment accuracy/ repeatability limitation.

Actual Result

Step 18 Step Execution Status: **Not Executed**

Description

Follow *3.5.24 Hexapod Absolute Accuracy* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 70-74.

Test Data

Deviation: Test at a single elevation angle.

Expected Result

The accuracy of the hexapod is at least the following:

Axis	Required Accuracy (um, deg)
X	125
Y	125
Z	25
RX	0.00083
RY	0.00083
RZ	0.0075

NOTE: The accuracy of the hexapod may be better than can be determined by the test equipment. This may falsely show a deficiency in the hexapod performance as a result of test equipment accuracy/ repeatability limitation.

Actual Result

Step 19 Step Execution Status: **Not Executed**

Description

Follow *3.5.26 Hexapod Radial (X and Y) and Axial (Z) Velocity Range* and *3.5.27 Hexapod Rotational Velocity* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 75.

Test Data

Deviation: Only test this using synchronous mode. Wait for 39s between each movement.

Expected Result

The hexapod velocity exceeds the 106um/s in XY and 0.0062deg/s in RXYRY and RZ requirements.

Actual Result

Step 20 Step Execution Status: **Not Executed**

Description

Follow 3.5.28 *Hexapod Heat Dissipation* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 75-76.

Test Data

Deviation: Calculate the power by having an amp meter on the legs. This test can be done simultaneously with the other test steps.

Expected Result

The current measured by the inductive current probes is calculated to meet the heat dissipation requirement.

Actual Result

Step 21 Step Execution Status: **Not Executed**

Description

Follow *3.5.14 Cross Talk Motion* of the LSST Hexapods-Rotator Acceptance Test Procedure, Sheet 59.

Test Data

Deviation: Analyze data from 3.5.15, 3.5.17, and 3.5.19 test steps after testing to verify cross talk.

Expected Result
There is no cross-talk observed.

Actual Result

5.2.3.3 LVV-T1802 - Integration of M2 Hexapod with SAL

Version **2**. Open *LVV-T1802* test case in Jira.

The objective of this test case is to re-verify the functional requirements of the M2 hexapod's software, after shipment of the hardware from the vendor's facility to the Summit, as defined in LTS-206 and LTS-160. This test case will only exercise the functionality that was executed previously and meets the following criteria:

- Only requires the use of Rubin Observatory code to replace MOOG's middleware code
- Only requires the M2 hexapod to be operable
- Only requires command through the CSC after the PXI real-time controller is switched from GUI mode to DDS mode
- Only requires testing the synchronous mode
  - **Asynchronous mode is not a standard mode of operation**
- Does require the M2 hexapod temperature sensors to be operating
- Does **NOT** require the M2 hexapod to be rotated to various elevation angles.
- Does **NOT** require the M2 hexapod to be in a climate-controlled environment

The software functional requirements were previously verified during the test campaign by the vendor at the vendor's facility and accepted by Rubin Observatory during the Factory Acceptance Test review. The test procedure used during the vendor's acceptance testing is the *LSST Hexapods-Rotator Software Acceptance Test Procedure* which is attached to this test case.

The test steps of this test case are the same steps from the procedure for the testing of the Camera Hexapod. The order of the steps was changed to reflect the *Proposal of Hexapod Test on the Dec. 2019* Confluence page which can be found linked in the Traceability tab.

See the attached *LSST Hexapod Operator's Manual* for more information on how to operate the hexapod.

Preconditions:
Prior to the execution of this test case to re-verify the M2 Hexapod hardware functional requirements, the following Summit tasks must be completed:

- The measurement equipment has been set-up for testing
  - `https://jira.lsstcorp.org/browse/SUMMIT-1943`

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1	Step Execution Status: **Not Executed**

Description

Hexapod DDS Startup Procedure

Release the Lock-Out-Tag-Out (LOTO) for the Rotator Circuit and the Hexapod Circuit using the keys stored in Cabinet 2 at Level 3.

Expected Result

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Access the Netbooter for the Rotator and Hexapod and turn on the Drivers and the CPU for each component in this order.
Here are the addresses for the netbooters:
rot-netbooter.cp.lsst.org
camhex-netbooter.cp.lsst.org
Look for the credentials in the 1password MainTel vault.

Expected Result

Actual Result

Step 3 Step Execution Status: **Not Executed**

Description

Raise (disengage) the E-Stops for the CCW and HexRot.
Announce on #summit-announce Slack Channel.

Expected Result

Actual Result

Step 4 Step Execution Status: **Not Executed**

Description

Use the little black knob in the Hexapod Cabinet to reset the power of its controller.

Expected Result

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

STARTING THE EUI

Connect to the hexapod management computer using X2Go
Open a terminal
change to the folder

cd /rubin/hexapod/build/

Start the EUI with the command:

./runCamHexEui
./runM2HexEui

for the camera and the M2 hexapod, respectively.

Example Code

```
# Check if CamHex EUI is running
ps -aux | grep runCamHex

cd /rubin/hexapod/build
./runCamHexEUI

# Check if M2Hex EUI is running
ps -aux | grep runM2Hex
```

Expected Result

The EUI is in the Offline State/PublishOnly substate and is able to publish through SAL but cannot receive commands.

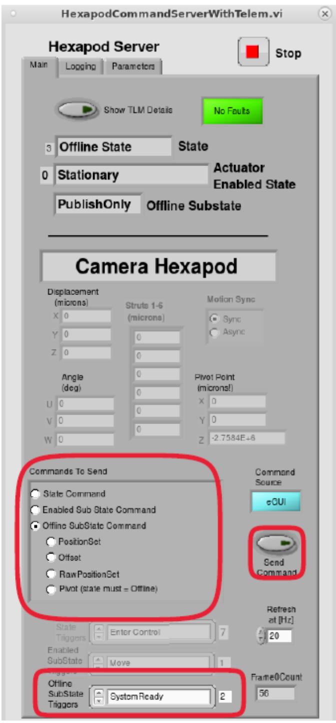
Actual Result

Step 6 Step Execution Status: **Not Executed**

Description

Transition from OFFLINE State/Publish Only Substate to OFFLINE State/Available Substate.

On the Main tab, select the **Offline SubState Command** radio button field in the **Commands to Send** section.
Then, set the **Offline SubState Triggers** to **System Ready**.
Finally, click on the **Send Command** button.



Expected Result

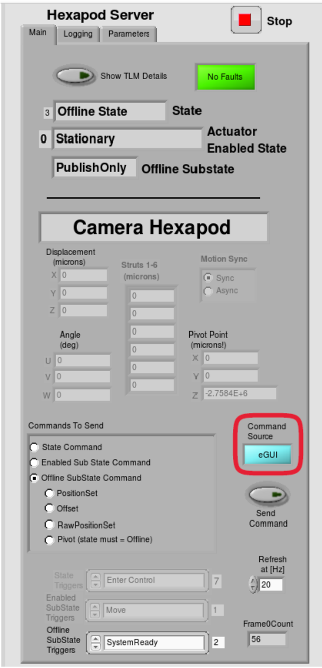
The system transitions from the OfflineState/PublishOnly substate to the OfflineState/AvailableState substate and the Command Source says eGUI.

Actual Result

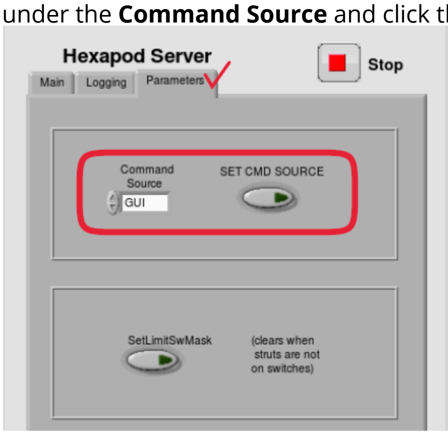
Step 7 Step Execution Status: **Not Executed**

Description

SWITCHING TO DDS MODE



If the Command Source does not show DDS, go to the **Parameters** tab, select **DDS** under the **Command Source** and click the **Set Cmd Source** button.



Note: If the GUI is used after being set to DDS mode, the system will switch back the Command Source to GUI and ignore any DDS commands. The Command Source must show DDS in order to receive DDS commands.

Expected Result

The system is capable of receiving/responding to DDS commands.

The **Command Source = DDS** green LED in the Telemetry Tab should turn on.

Actual Result

Step 8	Step Execution Status: **Not Executed**

Description

From any state send the CSC into OfflineState/AvailableState using the EUI. Change the control at the EUI from EUI control to CSC control.

Verify the Hexapod is commandable by DDS by checking the EFD.

Expected Result

The *MTHexapod_logevent_commandableByDDS* publishes true.

Actual Result

Step 9	Step Execution Status: **Not Executed**

Description

Verify the TCP/IP is connected to the low level controller.

Expected Result

The *MTHexapod_logevent_connected* publishes true.

Actual Result

Step 10	Step Execution Status: **Not Executed**

Description

Verify the *MTHexapod_logevent_configuration* event is publishing data to the EFD.

Test Data

Note: This step is to verify the *MTHexapod_logevent_configuration* event is publishing initial data. There will be steps later to verify that the event is published as a result of updating values.

Expected Result

The *MTHexapod_logevent_configuration* is publishing data.

Actual Result

Step 11 Step Execution Status: **Not Executed**

Description

Verify the *MTHexapod_logevent_interlock* event is unengaged and publishing data to the EFD.

Expected Result

The *MTHexapod_logevent_interlock* is publishing and shows no safety interlock is engaged.

Actual Result

Step 12 Step Execution Status: **Not Executed**

Description

Hit the E-stop.

Test Data

Note: This will mimic triggering an interlock signal from the GIS.

Expected Result

The *MTHexapod_logevent_interlock* publishes true.

Actual Result

Step 13 Step Execution Status: **Not Executed**

Description

Reset the E-stop and clear the error:

- With the CSC in the Fault state, issue a *standby* command via the CSC using the notebook.

Expected Result
The Hexapod and E-stop are reset and *MTHexapod_logevent_interlock* publishes false.

Actual Result

Step 14 Step Execution Status: **Not Executed**

Description
Verify that the thermal sensors are connected and producing telemetry into the EFD. "lsst.sal.ESS.temperature"

Expected Result
All actuator temperatures are published to the EFD.

Actual Result

Step 15 Step Execution Status: **Not Executed**

Description
The following steps define what the Jupyter Notebook for this test case implements. Executing the Jupyter notebook is the only actual command and control step that needs to be executed.
Transition the state machine into *disabledState* to publish telemetry

Expected Result
The Jupyter notebook controls the system to run through the steps below.

Actual Result

Step 16 Step Execution Status: **Not Executed**

Description
Verify the *MTHexapod_actuators* telemetry is being published to the EFD. Positions, velocities, accelerations.

Expected Result

The *MTHexapod_actuators* telemetry is being ingested into the EFD.

Actual Result

Step 17 Step Execution Status: **Not Executed**

Description

Verify the *MTHexapod_application* data is being published to the EFD.

Expected Result

The *MTHexapod_application* telemetry is being ingested into the EFD.

Actual Result

Step 18 Step Execution Status: **Not Executed**

Description

Verify the *MTHexapod_electrical* data is being published to the EFD.

Expected Result

The *MTHexapod_electrical* telemetry is being ingested into the EFD.

Actual Result

Step 19 Step Execution Status: **Not Executed**

Description

MOVE TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #1 - Synchronous Move Commands

With the synchronous button enabled and in enabled/stationary state, send a *move* command of (500um, -500um, 200um, 0.01deg, -0.015deg, 0deg).

Expected Result

- The hexapod moves to (x= 500um,y= -500um, z=200um, u=0.01deg, v=-0.015deg, w=0deg)
- Since the Hexapod is in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result

Step 20	Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated.

Expected Result

- The controllerState.enabledSubstate goes to MOVING_POINT_TO_POINT when the move begins and STATIONARY when the move ends.
- An *MTHexapod_logevent_inPosition* event is generated when the move is complete

Actual Result

Step 21	Step Execution Status: **Not Executed**

Description

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #5 - Stop Commands

In the enabled/stationary state, send a *move* command of (x=0um, y=0um, z=5000um, u=0deg, v=0deg, w=0deg)

Expected Result

The hexapod begins to move.

Actual Result

Step 22 Step Execution Status: **Not Executed**

Description

Wait 3s.

Expected Result

The hexapod is still moving.

Actual Result

Step 23 Step Execution Status: **Not Executed**

Description

Send a *stop* command.

Expected Result

- The hexapod stops before reaching the previously commanded position

Actual Result

Step 24 Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated.

Expected Result

- The controllerState.enabledSubstate goes to CONTROLLED_STOPPING when the stop is requested, then STATIONARY when the hexapod has halted.
- No *MTHexapod_logevent_inPosition* event is generated.

Actual Result

Step 25	Step Execution Status: **Not Executed**

Description

Test the "setCompensationMode" command.

In enabled/stationary state, send a *move* command of (x=0um, y=0um, z=800um, u=0deg, v=0deg, w=0deg)

Expected Result

The hexapod moves to the position (x=0um, y=0um, z=800um, u=0deg, v=0deg, w=0deg) and, since we are moving in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result

Step 26	Step Execution Status: **Not Executed**

Description

Ensure that MTMount publishes the telescope elevation angle and MTRotator publishes the rotation angle of the rotator. Either as real components or through controllers simulating the components.

Expected Result

Published telescope elevation and rotator angle.

Actual Result

Step 27	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, set *setCompensationMode* command to enable=True.

Expected Result

- The hexapod does not move and the *MTHexapod_logevent_compensationMode* event appears as true in the EFD.

- The *MTHexapod_logevent_compensatedPosition* is also sent to the EFD.

Actual Result

Step 28 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *move* command of (0um, 0um, 800um, 0deg, 0deg, 0deg)

Expected Result

The hexapod moves to a slightly different position than (0um, 0um, 800um, 0deg, 0deg, 0deg) and, since we are moving in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result

Step 29 Step Execution Status: **Not Executed**

Description

Check if there are any different events between move with and without setCompensationMode=True. Check the movement in the EFD use:

Compare *MTHexapod_logevent_compensatedPosition* to *MTHexapod_logevent_uncompensatedPosition*

Expected Result

The changes are expected according to this table:

		M2 Hexapod motions					
zenith angle		um	um	um	deg	deg	deg
deg	rads	dx	dy	dz	rx	ry	rz
90	**1.570796**	2.942346	556.6612	-656.9706	0.006705	-2.2133E-05	-9.264E-05
85	**1.48353**	2.133244	556.057	-567.0034	0.006638	-1.8487E-05	-7.4613E-05
80	**1.396263**	1.366087	546.5259	-478.2827	0.006471	-1.4965E-05	-5.74668E-05
75	**1.308997**	0.646713	528.1404	-391.4837	0.006206	-1.1593E-05	-4.13318E-05
70	**1.22173**	-0.019403	501.0403	-307.2671	0.005845	-8.3957E-06	-2.63309E-05
65	**1.134464**	-0.62719	465.4319	-226.2737	0.00539	-5.3987E-06	-1.25782E-05
60	**1.047198**	-1.172025	421.5862	-149.1199	0.004845	-2.6245E-06	-1.78402E-07
55	**0.959931**	-1.649759	369.837	-76.39305	0.004214	-9.4085E-08	1.07741E-05
50	**0.872665**	-2.056758	310.5781	-8.646518	0.003502	2.1732E-06	2.0196E-05
45	**0.785398**	-2.389924	244.2603	53.60408	0.002713	4.1601E-06	2.80156E-05
40	**0.698132**	-2.646721	171.3886	109.885	0.001856	5.8515E-06	3.41734E-05
35	**0.610865**	-2.825195	92.51743	159.7678	0.000934	7.2345E-06	3.86224E-05
30	**0.523599**	-2.923987	8.247089	202.873	-4.29E-05	8.2987E-06	4.13289E-05
25	**0.436332**	-2.942346	-80.78108	238.8724	-0.001069	9.0359E-06	4.22722E-05
20	**0.349066**	-2.880131	-173.8895	267.4922	-0.002136	9.4405E-06	4.14452E-05
15	**0.261799**	-2.737817	-270.3696	288.5144	-0.003236	9.5094E-06	3.88542E-05
10	**0.174533**	-2.516487	-369.4871	301.7791	-0.004361	9.2421E-06	3.45188E-05
5	**0.087266**	-2.217825	-470.4876	307.1853	-0.005501	8.6406E-06	2.84721E-05
0	**0**	-1.844103	-572.6024	304.692	-0.006649	7.7096E-06	2.076E-05

Actual Result

Step 30 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send again the same *move* command of (0um, 0um, 800um, 0deg, 0deg, 0deg)

Expected Result

The hexapod does not move since it stayed in compensationMode.

Actual Result

Step 31 Step Execution Status: **Not Executed**

Description

OFFSET TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #4 - Synchronous Offset and Move Commands

In enabled/stationary state, send a *move* command of (x=500um, y=800um, z=200um, u=0deg, v=0deg, w=0deg)

Test Data

Expected Result

- The hexapod moves to (x=500um, y=800um, z=200um, u=0deg, v=0deg, w=0deg)
- Since the Hexapod is in synchronous mode, the actuators complete the move at nearly the same time.

Actual Result

Step 32 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send an *offset* command of (0um, 0um, 500um, 0deg, 0deg, 0deg).

Expected Result

- The hexapod moves only 500um in Z from the previous position
- The actuators complete the move at nearly the same time.
- The *MTHexapod_logevent_compensationOffset* starts to publish data to the EFD based on the computed compensation offset and the input parameters provided.

Actual Result

Step 33 Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated.

Expected Result

- The controllerState.enabledSubstate goes to MOVING_POINT_TO_POINT when the move begins and STATIONARY when the move ends
- The *MTHexapod_logevent_inPosition* event is True when the move finishes
- The *MTHexapod_logevent_inPosition* event is False when the enabledSubstate goes back to STATIONARY.

Actual Result

Step 34 Step Execution Status: **Not Executed**

Description

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #2 - setPivot and Move Commands

In enabled/stationary state, send a *move* command of (x=2000um,y=-3500um,z=200um,u=0.01deg,v=-0.05deg, w=0.002deg,sync=true)

Test Data

Deviation: Determine where the original pivot point is before sending a *setPivot* command of (0, 0, 0). Record any offset commands necessary to test before sending the move command.

Expected Result

The hexapod moves to the commanded position

Actual Result

Step 35	Step Execution Status: **Not Executed**

Description

In the enabled/stationary state, send the *setPivot* command of (0,0,0).

Expected Result

- The actuator positions do not change but the hexapod position changes to account for the new pivot point.
- The *MTHexapod_logevent_configuration* is updated.

Actual Result

Step 36	Step Execution Status: **Not Executed**

Description

In the enabled/stationary state, send again the *move* command of (x=2000um, y=-3500um, z=200um, u=0.01deg, v=-0.05deg, w=0.002deg,sync=true)

Test Data

Deviation: Record any offset commands necessary to test before sending the move command.

Expected Result

The hexapod doesn't move. Position values in the EFD appear different.

Actual Result

Step 37	Step Execution Status: **Not Executed**

Description

CONFIGURE LIMITS TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #6 - configureLimits Command

In enabled/stationary state, send a configureLimits command of (12000um, -1000um, 1000um, 0.1, -0.1, 0.05)

Test Data

Deviation: Skip complete test. This test uses an obsolete command. The configuration is now done before and should not be changed this state - The obsolete command was the positionSet command which has been changed to move.

Expected Result

The command is rejected for being outside acceptable limits.

Actual Result

Step 38	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *configureLimits* command of (1000um, -1000um, 1000um, 0.1, -0.1, 0.05)

Expected Result

The command is accepted and *MTHexapod_logevent_configuration* event is updated.

Actual Result

Step 39	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *move* command of (850um, 0um, 500um, 0deg, 0deg, 0deg)

Test Data

Note: This command can be any valid *move* command within the newly configured limits.

Expected Result

The command is accepted and the hexapod moves to the commanded position.

Actual Result

Step 40	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *move* command of (1200um, 0um, 200um, 0deg, 0deg, 0deg)

Test Data

Note: This command can be any valid *move* command within the newly configured limits.

Expected Result

The command is rejected for being outside of range limits

Actual Result

Step 41	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *move* command of (990um, 990um, 200um, 0deg, 0deg, 0deg)

Expected Result

The command is rejected for being outside of range limits.

Actual Result

Step 42	Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *move* command of (500um, 500um, 200um, 0deg, 0.1 deg, 0.01deg)

Expected Result

The command is accepted and moves to the commanded position.

Actual Result

Step 43 Step Execution Status: **Not Executed**

Description

Record the DDS events that were generated.

Expected Result

The change is reflected in the *MTHexapod_logevent_configuration* event and the EUI.

Actual Result

Step 44 Step Execution Status: **Not Executed**

Description

CONFIGURE ACCELERATION TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #7 - configureAcceleration Command

In enabled/stationary state, at a position of (0, 0, 0, 0, 0, 0) with the velocity and acceleration values set to their nominal values, send a *move* command of (0um, 0um, 4900um, 0 deg, 0 deg, 0 deg, s).

Test Data

Deviation: Skip complete test. This test uses an obsolete command. The configuration is now done before and should not be changed this state - the configureAcceleration command is still valid but only through the CSC

Expected Result

The move takes approximately 9 seconds to complete.

Actual Result

Step 45 Step Execution Status: **Not Executed**

Description

Send a *configureAcceleration* command of 1000.

Expected Result
Confirm command is rejected for being outside of acceptable limits.

Actual Result

Step 46 Step Execution Status: **Not Executed**

Description
Send a *configureAcceleration* command of 100.

Expected Result
The command is accepted.

Actual Result

Step 47 Step Execution Status: **Not Executed**

Description
In enabled/stationary state, send a *move* command of (0um, 0um, 0um, 0 deg, 0 deg, 0 deg, s).

Expected Result
It takes approximately 13 seconds to complete the commanded move with the reduced acceleration value.

Actual Result

Step 48 Step Execution Status: **Not Executed**

Description
Send a *configureAcceleration* command of 500 to return the acceleration limit to its nominal value.

Expected Result
The command is accepted.

Actual Result

Step 49 Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated.

Expected Result

The change is reflected in the *MTHexapod_logevent_configuration* event and the EUI.

Actual Result

Step 50 Step Execution Status: **Not Executed**

Description

CONFIGURE VELOCITY TEST

Section 3.1.2 of the attached Software Acceptance Test Procedure

Test Sequence #8 - configureVelocity Command

In enabled/stationary state, at a position of (0, 0, 0, 0, 0, 0), send a *configureVelocity* command of (10000, .01, 100, .01).

Test Data

Deviation: Skip complete test. This test uses an obsolete command. The configuration is now done before and should not be changed - this state the configureVelocity command is still valid but only through the CSC

Expected Result

This command is rejected for being outside of acceptable limits.

Actual Result

Step 51 Step Execution Status: **Not Executed**

Description

In enabled/stationary state, send a *configureVelocity* command of (100, .01, 200, .01).

Expected Result
This command is accepted.

Actual Result

Step 52 Step Execution Status: **Not Executed**

Description
In enabled/stationary state, send a *move* command of (0, 0um, 2000um, 0 deg, 0 deg, 0 deg, s).

Expected Result
It takes approximately 20 seconds to complete the commanded move.

Actual Result

Step 53 Step Execution Status: **Not Executed**

Description
In enabled/stationary state, send a *configureVelocity* command of (100, .01, 100, .01).

Expected Result
This command is accepted.

Actual Result

Step 54 Step Execution Status: **Not Executed**

Description
In enabled/stationary state, send an *offset* command of (0, 0um, 2000um, 0 deg, 0 deg, 0 deg).

Expected Result
This command is accepted

Actual Result

Step 55	Step Execution Status: **Not Executed**

Description

Send a *move* command.

Expected Result

It takes approximately 40 seconds to complete the commanded move.

Actual Result

Step 56	Step Execution Status: **Not Executed**

Description

Record the corresponding DDS events that were generated:

Expected Result

The change is reflected in the *MTHexapod_logevent_configuration* event and the EUI.

Actual Result

Step 57	Step Execution Status: **Not Executed**

Description

Section 3.3.2 of the attached Software Acceptance Test Procedure Hexapod Action on State Commands

The state machine is a standbyState entry state machine.

Transition the state machine into offlineState/publishOnly state.

In the offlineState/publishOnly state, send all commands

Test Data

Note: This section utilizes the commands defined in LSE-209

Expected Result

There is no change and command is rejected.

Actual Result

Step 58 Step Execution Status: **Not Executed**

Description

Transition the state machine into offlineState/Available state.
In the offlineState/Available state, send an enterControl command.

Expected Result

The system enters the standbyState.

Actual Result

Step 59 Step Execution Status: **Not Executed**

Description

In the **standbyState,** send any command except start or exitControl

Expected Result

There is no change and command is rejected.

Actual Result

Step 60 Step Execution Status: **Not Executed**

Description

In the Standby state, send an exitControl command.

Expected Result

The system transitions into the Offline/Available state.

Actual Result

Step 61 Step Execution Status: **Not Executed**

Description

Transition the state machine into standbyState.
In the Standby state, send a start command.

Expected Result

The system transitions into the disabledState.

Actual Result

Step 62 Step Execution Status: **Not Executed**

Description

In the Disabled state, send any command except for the enabled or standby command.

Expected Result

There is no change and the command is rejected.

Actual Result

Step 63 Step Execution Status: **Not Executed**

Description

In the Disabled state, send the standby command.

Expected Result

The system transitions into the Standby state.

Actual Result

Step 64 Step Execution Status: **Not Executed**

Description

Transition the state machine into disabledstate
In the Disabled state, send the enable command.

Expected Result

The system transitions into the Enabled/Stationary state.

Actual Result

Step 65	Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, send either the enterControl command, exitControl command, start command, clearError command, or enable command.

Expected Result

There is no change and command is rejected.

Actual Result

Step 66	Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, send a disable command.

Expected Result

The system transitions into Disabled state.

Actual Result

Step 67	Step Execution Status: **Not Executed**

Description

In the Fault state, send any command.

Expected Result

The state machine transitions into the commanded state.

Actual Result

Step 68	Step Execution Status: **Not Executed**

Description

Section 4 of the attached Software Acceptance Test Procedure

In the Enabled/Stationary state, unplug a motor encoder cable for one of the actuators.

Test Data

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

Step 69	Step Execution Status: **Not Executed**

Description

In the Enabled/Stationary state, unplug a linear encoder cable for one of the actuators.

Expected Result

A Drive Fault error event is created and the system transitions to Fault state.

Actual Result

Step 70	Step Execution Status: **Not Executed**

Description

Unplug a motor power cable from one of the actuators and command a Move.

Expected Result

A Following Error event is created and the system transitions to Fault state.

Actual Result

Step 71	Step Execution Status: **Not Executed**

Description

Activate an extension limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result

An Extended Limit Switch error event is created and the system transitions into Fault state.

Actual Result

Step 72	Step Execution Status: **Not Executed**

Description

Activate a retraction limit switch on one of the actuators by removing the limit switch cover and manually tripping.

Expected Result

A Retracted Limit Switch error event is created and the system transitions into Fault state.

Actual Result

Step 73	Step Execution Status: **Not Executed**

Description

Unplug the Ethercat cable between the control PC and the first Copley XE2 drive.

Expected Result

An Ethercat Lost event is created and the system transitions to Fault state.

Actual Result

A Acronyms used in this document

Acronym	Description
AC	Alternating Current
CCW	Camera Cable Wrap
CPU	Central Processing Unit
CSC	Commandable SAL Component
DDS	Data Distribution System
EFD	Engineering and Facility Database
EUI	Engineering User Interface System
FRACAS	Failure Reporting Analysis and Corrective Action System
GIS	Global Interlock System
GUI	Graphical User Interface
IP	Internet Protocol
IPA	FreeIPA is an integrated security information management solution
LED	Light-Emitting Diode
LOTO	Lock Out Tag Out
LSE	LSST Systems Engineering (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LTS	LSST Telescope and Site (Document Handle)
LVV	LSST Verification and Validation
M2	Secondary Mirror
MT	Main Telescope
PMCS	Project Management Controls System
PSE	Project Systems Engineering
SAL	Service Abstraction Layer
SE	System Engineering
SITCOM	System Integration, Test and Commissioning
SMR	Spherically Mounted Retroreflector
TCP	Transmission Control Protocol
TEA	Top End Assembly
TMA	Telescope Mount Assembly
VPN	virtual private network