

# MTAOS handling of rejected commands

This notebook is used for the level 3 integration tests from test plan LVV-P81 (<https://jira.lsstcorp.org/secure/Tests.jspa#/testPlan/LVV-P81>) as part of test cycle LVV-C176 (<https://jira.lsstcorp.org/secure/Tests.jspa#/testCycle/LVV-C176>). The following tests are currently run as part of this notebook:

- LVV-T2193 (<https://jira.lsstcorp.org/secure/Tests.jspa#/testCase/LVV-T2193>)

Execution steps are separated by horizontal lines. Upon completion, save the notebook and its output as a pdf file to be attached to the test execution in JIRA.

Last updated by E. Dennihy 20211020

---

Load all the needed libraries. Get the remotes ready Code in the notebook including section: "Check the summary state of each CSC".

```
In [1]: %load_ext autoreload
        %autoreload 2
```

```
In [2]: import rubin_jupyter_utils.lab.notebook as nb
        nb.utils.get_node()
```

```
/tmp/ipykernel_36804/1665379685.py:2: DeprecationWarning: Call to deprecated f
unction (or staticmethod) get_node. (Please use lsst.rsp.get_node())
    nb.utils.get_node()
```

```
Out[2]: 'yagan07'
```

```
In [3]: import os
        import sys
        import asyncio
        import logging

        import pandas as pd
        import numpy as np

        from datetime import datetime
        from matplotlib import pyplot as plt

        from lsst.ts import salobj
        from lsst.ts.observatory.control.maintel import MTCS, ComCam
        from lsst.ts.observatory.control import RotType
```

```
| lsst.ts.utils.tai INFO: Update leap second table
| lsst.ts.utils.tai INFO: current_tai uses the system TAI clock
```

```
In [4]: logging.basicConfig(format="%(name)s: %(message)s", level=logging.DEBUG)
```

```
In [5]: log = logging.getLogger("setup")
```

```
log.level = logging.DEBUG
```

```
In [6]: domain = salobj.Domain()
```

```
In [7]: mtcs = MTCS(domain=domain, log=log)
        mtcs.set_rem_loglevel(40)
```

```
| setup.MTCS DEBUG: mtmount: Adding all resources.
| setup.MTCS DEBUG: mptg: Adding all resources.
| setup.MTCS DEBUG: mtaos: Adding all resources.
| setup.MTCS DEBUG: mtm1m3: Adding all resources.
| setup.MTCS DEBUG: mtm2: Adding all resources.
| setup.MTCS DEBUG: mthexapod_1: Adding all resources.
| setup.MTCS DEBUG: mthexapod_2: Adding all resources.
| setup.MTCS DEBUG: mtrotator: Adding all resources.
| setup.MTCS DEBUG: mtdome: Adding all resources.
| setup.MTCS DEBUG: mtdometrajectory: Adding all resources.
| MTHexapod INFO: Read historical data in 0.03 sec
| MTHexapod INFO: Read historical data in 0.04 sec
| MTHexapod.electrical WARNING: electrical DDS read queue is filling: 11 of 1
| 00 elements
| MTHexapod.application WARNING: application DDS read queue is filling: 12 of
| 100 elements
| MTHexapod.actuators WARNING: actuators DDS read queue is filling: 12 of 100
| elements
```

```
In [8]: await mtcs.start_task
```

```
Out[8]: [None, None, None, None, None, None, None, None, None, None]
```

```
In [9]: script = salobj.Controller("Script", index=42658886)
        await asyncio.sleep(10) # wait 10 second may help with DDS problems; closing a
| Script INFO: Read historical data in 0.00 sec
```

```
In [10]: script.log.info(f"LVV-T2193 - START -- {datetime.now()} UTC")
```

```
| Script INFO: LVV-T2193 - START -- 2022-04-08 15:25:49.344458 UTC
```

---

Ready M1M3: Raise mirror, turn on FB, clear forces

Need to have M1M3 LUT use its inclinometer.

---

Ready M2: Turn on FB, clear forces

Need to have M2 LUT use its inclinometer

---

Get camera hexapod ready: check config; make sure LUT is on, and has valid inputs; make

sure hex is at LUT position

---

Get M2 hexapod ready: check config; make sure LUT is on, and has valid inputs; make sure hex is at LUT position

---

Slew to the next target. Choose a target such that the rotator stays within a couple of degrees of its initial position. This is because the CCW is not running (MTmount in simulation mode).

```
In [13]: target = await mtcs.find_target(el=60, az=120, mag_limit=8)
         print(target)
```

```
HD 13444
```

```
In [14]: await mtcs.slew_object(target, rot_type=RotType.PhysicalSky, rot=1.9)
```

```
setup.MTCS INFO: Slewing to HD 13444: 02 10 23.6882 -40 21 20.199
setup.MTCS DEBUG: Setting rotator physical position to 1.9 deg. Rotator will track sky.
setup.MTCS DEBUG: Wait 5.0s for rotator to settle down.
setup.MTCS DEBUG: Workaround for rotator trajectory problem. Moving rotator to its current position: 1.80
setup.MTCS DEBUG: Wait for MTRotator in position event.
setup.MTCS DEBUG: MTRotator in position: False.
setup.MTCS INFO: MTRotator in position: True.
setup.MTCS DEBUG: MTRotator in position True. Waiting settle time 5.0s
setup.MTCS DEBUG: Sending slew command.
setup.MTCS DEBUG: Scheduling check coroutines
setup.MTCS DEBUG: process as completed...
setup.MTCS DEBUG: Monitor position started.
setup.MTCS DEBUG: Waiting for Target event from mtmount.
setup.MTCS DEBUG: mtmount: <State.ENABLED: 2>
setup.MTCS DEBUG: mtptg: <State.ENABLED: 2>
setup.MTCS DEBUG: mtaos: <State.ENABLED: 2>
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
setup.MTCS DEBUG: mtm2: <State.ENABLED: 2>
setup.MTCS DEBUG: mthexapod_1: <State.ENABLED: 2>
setup.MTCS DEBUG: mthexapod_2: <State.ENABLED: 2>
setup.MTCS DEBUG: mtrotator: <State.ENABLED: 2>
setup.MTCS DEBUG: mtdome: <State.ENABLED: 2>
setup.MTCS DEBUG: mtdometrajectory: <State.ENABLED: 2>
setup.MTCS DEBUG: Wait for mtmount in position events.
setup.MTCS DEBUG: Wait for dome in position event.
setup.MTCS DEBUG: Wait for MTRotator in position event.
setup.MTCS DEBUG: MTRotator in position: True.
setup.MTCS DEBUG: MTRotator already in position. Handling potential race condition.
```

```

| setup.MTCS DEBUG: Wait for MTMount elevation in position event.
| setup.MTCS DEBUG: MTMount elevation in position: True.
| setup.MTCS DEBUG: MTMount elevation already in position. Handling potential
| race condition.
| setup.MTCS DEBUG: Wait for MTMount azimuth in position event.
| setup.MTCS DEBUG: MTMount azimuth in position: True.
| setup.MTCS DEBUG: MTMount azimuth already in position. Handling potential r
| ace condition.
| setup.MTCS DEBUG: Mount target: private_revCode: bdc00ba, private_sndStam
| p: 1649431737.9592094, private_rcvStamp: 1649431737.965274, private_seqNum:
| 24064, private_identity: MTMount, private_origin: 263534, elevation: 60.245
| 575010381266, elevationVelocity: 0.003151124900074682, azimuth: 119.1569865
| 9072122, azimuthVelocity: 0.0009708654238739545, taiTime: 1649431738.01755
| 2, trackId: 1, tracksys: SIDERREAL, radesys: ICRS, priority: 0
| setup.MTCS INFO: MTMount elevation in position: False.
| setup.MTCS INFO: MTMount azimuth in position: False.
| setup.MTCS INFO: MTMount elevation in position: True.
| setup.MTCS DEBUG: MTMount elevation in position True. Waiting settle time
| 3.0s
| setup.MTCS INFO: MTMount azimuth in position: True.
| setup.MTCS DEBUG: MTMount azimuth in position True. Waiting settle time 3.0
| s
| setup.MTCS INFO: MTRotator in position: False.
| setup.MTCS DEBUG: [Tel]: Az = +119.355[ -0.2]; El = +060.182[ +0.1] [Ro
| t]: +001.800[ +0.0] [Dome] Az = +000.000; El = +000.000
| setup.MTCS DEBUG: Dome azimuth in position.
| setup.MTCS DEBUG: Dome elevation in position.
| setup.MTCS INFO: MTRotator in position: True.
| setup.MTCS DEBUG: MTRotator in position True. Waiting settle time 3.0s

```

---

clear all corrections using cmd\_resetCorrection

```
In [15]: await mtcs.rem.mtaos.cmd_resetCorrection.start()
```

```
Out[15]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa1f9446eb0>
```

```
In [16]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[16]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa18d267d90>
```

---

Add 1um of z7 to the system via OFC, issue the corrections.

Compare the corrections sent vs forces and position changes applied. This is currently done in a separate notebook or on Chronograf.

```
In [17]: wavefront_errors = np.zeros(19)
```

```
In [18]: wavefront_errors[3]=1.0
```

```
In [19]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1)
```

```
Out[19]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa1f941db80>
```

```
In [20]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[20]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa18d40d6d0>
```

---

Make plots using telemetry from each component to verify the changes in the DOFs. This step does not currently involve running any commands in this notebook. This step must be verified using a separate notebook.

---

Put M2 hexapod in DISABLED state (so that we can test command rejection).

```
In [21]: await mtcs.set_state(salobj.State.DISABLED, components=["mthexapod_2"])
```

```
| setup.MTCS DEBUG: [mthexapod_2]::[<State.ENABLED: 2>, <State.DISABLED: 1>]
```

```
| setup.MTCS INFO: All components in <State.DISABLED: 1>.
```

---

Add 1um of z7 to the system via OFC. Expect m2 hexapod corrections are rejected, and all other corrections applied, then undone.

```
In [22]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1)
```

```
Out[22]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa19e437580>
```

```
In [23]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```

-----
AckError                                Traceback (most recent call last)
Input In [23], in <cell line: 1>()
----> 1 await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.py:483, in RemoteCommand.start(self, data, timeout, wait_done)
    479 cmd_info = CommandInfo(
    480     remote_command=self, seq_num=seq_num, wait_done=wait_done
    481 )
    482 self.salinfo._running_cmds[seq_num] = cmd_info
--> 483 return await cmd_info.next_ackcmd(timeout=timeout)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.py:201, in CommandInfo.next_ackcmd(self, timeout)
    199 ackcmd = await self._wait_task
    200 if ackcmd.ack in self.failed_ack_codes:
--> 201     raise base.AckError(msg="Command failed", ackcmd=ackcmd)
    202 return ackcmd
    203 except asyncio.TimeoutError:

AckError: msg='Command failed', ackcmd=(ackcmd private_seqNum=640532057, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result="Failed: Failed to apply correction to: ['m2hex']. ")

```

---

Re-enable M2 hexapod Make it ready for AOS

```

In [24]: await mtcs.set_state(salobj.State.ENABLED, components=["mthexapod_2"])

| setup.MTCS DEBUG: [mthexapod_2]::[<State.DISABLED: 1>, <State.ENABLED: 2>]
| setup.MTCS INFO: All components in <State.ENABLED: 2>.

```

---

Re-issue the correction.

```

In [25]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1)

Out[25]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa184c65700>

In [26]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)

Out[26]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa19e6dd460>

```

---

Reject the latest corrections.

```

In [27]: await mtcs.rem.mtaos.cmd_rejectCorrection.start()

Out[27]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fa184c774c0>

In [28]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)

```

Out[28]: <ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7fa184e9a1f0>

---

Add 2um of z7 via OFC

In [29]: `wavefront_errors[3] = 2.0`

In [30]: `wavefront_errors`

Out[30]: `array([0., 0., 0., 2., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])`

In [31]: `await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1)`

Out[31]: <ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7fa184c240a0>

In [32]: `await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)`

Out[32]: <ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7fa184ec95e0>

---

Stop Tracking

In [33]: `await mtcs.stop_tracking()`

`setup.MTCS DEBUG: Stop tracking.`

In [34]: `script.log.info(f"LVV-T2193 - END -- {datetime.now()} UTC")`

`Script INFO: LVV-T2193 - END -- 2022-04-08 15:29:06.347210 UTC`

---

Wrap up. Put each component to the following states: mtaos --> standby m1m3 --> standby m2 --> standby camera hex --> standby m2 hex --> standby

In [ ]: `await mtcs.set_state(salobj.State.STANDBY, components=["mtaos"])`

In [ ]: `await mtcs.lower_m1m3()`

In [ ]: `await mtcs.set_state(salobj.State.STANDBY, components=["mtm1m3"])`

In [ ]: `await mtcs.set_state(salobj.State.STANDBY, components=["mtm2"])`

In [ ]: `await mtcs.set_state(salobj.State.STANDBY, components=["mthexapod_1"])`

In [ ]: `await mtcs.set_state(salobj.State.STANDBY, components=["mthexapod_2"])`

In [ ]: `await mtcs.standby()`