

# LVV-T2193

March 8, 2022

## 1 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”.

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

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

```
/tmp/ipykernel_52086/1665379685.py:2: DeprecationWarning: Call to deprecated
function (or staticmethod) get_node. (Please use lsst.rsp.get_node())
      nb.utils.get_node()
```

```
[2]: 'yagan06'
```

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

      import pandas as pd
      import numpy as np

      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
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

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

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

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

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

<IPython.core.display.HTML object>

---

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).

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

<IPython.core.display.HTML object>

HD 21222

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

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>



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

```
[11]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f7a26c0c550>
```

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

```
[12]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f79abc81e80>
```

---

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.

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

```
[14]: wavefront_errors[3]=1.0
```

```
[15]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors,
↳ timeout=10)
```

```
[15]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f7a24da7b80>
```

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

```
[16]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f799aa2d070>
```

---

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).

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

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

---

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

```
[18]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors,
↳ timeout=10)
```

[18]: <ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f79abc65130>

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

```
-----
AckError                                Traceback (most recent call last)
Input In [19], 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-2.0
  ↳ 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-2.0
  ↳ 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

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

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

---

Re-issue the correction.

```
[21]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors,
  ↳ timeout=10)
```

[21]: <ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f79b4025730>

```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f799ac3ee20>
```

```
await mtcs.rem.mtaos.cmd_rejectCorrection.start()
```

```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f79abc21580>
```

```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f79abb73280>
```

```
wavefront_errors[3] = 2.0
```

wavefront\_errors

[illegible]

```
await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors,
↳ timeout=10)
```

```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f7a2683b910>
```

```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f799a856190>
```

```
await mtcs.stop_tracking()
```

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

<IPython.core.display.HTML object>

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

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

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

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

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

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