

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_39847/1665379685.py:2: DeprecationWarning: Call to deprecated f
unction (or staticmethod) get_node. (Please use lsst.rsp.get_node())
    nb.utils.get_node()
```

```
Out[2]: 'yagan03'
```

```
In [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
```

```
| 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: mtptg: 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 0.04 sec
MTHexapod INFO: Read historical data in 0.08 sec
```

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

```
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 10
of 100 elements
```

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

```
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 12
of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 1
0 of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 2
2 of 100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 22 of
100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 11 of
100 elements
```

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

WARNING: AstropyDeprecationWarning: Transforming a frame instance to a frame class (as opposed to another frame instance) will not be supported in the future. Either explicitly instantiate the target frame, or first convert the source frame instance to a `astropy.coordinates.SkyCoord` and use its `transform_to()` method. [astropy.coordinates.baseframe]

astroquery WARNING: AstropyDeprecationWarning: Transforming a frame instance to a frame class (as opposed to another frame instance) will not be supported in the future. Either explicitly instantiate the target frame, or first convert the source frame instance to a `astropy.coordinates.SkyCoord` and use its `transform_to()` method.

MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 13 of 100 elements

MTHexapod.application WARNING: tel_application DDS read queue is filling: 13 of 100 elements

MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 13 of 100 elements

HD 80321

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

setup.MTCS INFO: Slewing to HD 80321: 09 17 43.3952 -41 00 41.867

setup.MTCS DEBUG: Setting rotator physical position to 1.9 deg. Rotator will track sky.

setup.MTCS WARNING: Camera cable wrap following disabled in MTMount.

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: 0.49

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 co
ndition.
| 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: 1654282444.835467, private_rcvStamp: 1654282444.8357236, private_seqNum:
32754, private_identity: MTMount, private_origin: 28805, elevation: 59.9523
7923673001, elevationVelocity: 0.0030995374245548764, azimuth: 120.79631054
11311, azimuthVelocity: 0.0010892539618935942, taiTime: 1654282444.8946857,
trackId: 1, tracksys: SIDEREAL, radesys: ICRS, priority: 0
| setup.MTCS INFO: MTMount elevation in position: False.
| setup.MTCS INFO: MTMount azimuth in position: False.
| setup.MTCS INFO: MTRotator in position: False.
| setup.MTCS INFO: MTRotator in position: True.
| setup.MTCS DEBUG: MTRotator in position True. Waiting settle time 3.0s
| setup.MTCS DEBUG: [Tel]: Az = +119.718[ +1.1]; El = +061.090[ -1.1] [Ro
t]: +000.488[ +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: MTMount azimuth in position: True.
| setup.MTCS DEBUG: MTMount azimuth in position True. Waiting settle time 3.0
s
| setup.MTCS INFO: MTMount elevation in position: True.
| setup.MTCS DEBUG: MTMount elevation in position True. Waiting settle time
3.0s

```

clear all corrections using cmd_resetCorrection

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

```
Out[11]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbe358d160>
```

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

Out[12]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efc2f588100>

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

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

In [15]: `await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1`

Out[15]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efc2f53b8b0>

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

Out[16]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbe345a7f0>

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 [17]: `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 [18]: `await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1`

Out[18]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbc0f2bca0>

In [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-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.py:485, in RemoteCommand.start(self, data, timeout, wait_done)
    481 cmd_info = CommandInfo(
    482     remote_command=self, seq_num=seq_num, wait_done=wait_done
    483 )
    484 self.salinfo._running_cmds[seq_num] = cmd_info
--> 485 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:195, in CommandInfo.next_ackcmd(self, timeout)
    193 ackcmd = await self._wait_task
    194 if ackcmd.ack in self.failed_ack_codes:
--> 195     raise base.AckError(msg="Command failed", ackcmd=ackcmd)
    196 return ackcmd
    197 except asyncio.TimeoutError:

AckError: msg='Command failed', ackcmd=(ackcmd private_seqNum=743583437, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result="Failed: Failed to apply correction to: ['m2hex']. ")
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 11
of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 1
1 of 100 elements

```

Re-enable M2 hexapod Make it ready for AOS

```

In [20]: 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 [21]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1)

Out[21]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbe345a970>

In [22]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)

Out[22]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbe36ad880>

```

Reject the latest corrections.

```

In [23]: await mtcs.rem.mtaos.cmd_rejectCorrection.start()

```

Out[23]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbe33aad30>

In [24]: `await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)`

Out[24]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbe358d460>

Add 2um of z7 via OFC

In [25]: `wavefront_errors[3] = 2.0`

In [26]: `wavefront_errors`

Out[26]: `array([0., 0., 0., 2., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])`

In [27]: `await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1)`

Out[27]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efbda2c3880>

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

Out[28]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7efc2f141f10>

Stop Tracking

In [29]: `await mtcs.stop_tracking()`

`setup.MTCS DEBUG: Stop tracking.`

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()`

In []:

