MTAOS add aberrations to M1M3+M2+hexapod

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 cylce LVV-C176 (https://jira.lsstcorp.org/secure/Tests.jspa#/testCycle/LVV-C176). The following tests are currently run as part of this notebook:

- LVV-T2190 (https://jira.lsstcorp.org/secure/Tests.jspa#/testCase/LVV-T2190)

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_37963/1665379685.py:2: DeprecationWarning: Call to deprecate
d function (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 astropy import time
from lsst.ts import utils

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.
```

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

```
MTHexapod INFO: Read historical data in 0.09 sec
MTHexapod INFO: Read historical data in 0.10 sec
Out[8]: [None, None, None, None, None, None, None, None, None, None]
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling:
16 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 fram
e class (as opposed to another frame instance) will not be supported in the
future.  Either explicitly instantiate the target frame, or first convert t
he source frame instance to a `astropy.coordinates.SkyCoord` and use its `t
ransform_to()` method. [astropy.coordinates.baseframe]
astroquery WARNING: AstropyDeprecationWarning: Transforming a frame inst
ance to a frame class (as opposed to another frame instance) will not be
supported in the future.  Either explicitly instantiate the target fram
e, or first convert the source frame instance to a `astropy.coordinates.
SkyCoord` and use its `transform_to()` method.
MTHexapod.application WARNING: tel_application DDS read queue is fillin
g: 16 of 100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 14
of 100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 29
of 100 elements
HD  46259
```

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

```
setup.MTCS INFO: Slewing to HD  46259: 06 30 29.5383 -40 34 20.771
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 rota
tor to its current position: 1.65
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 INFO: 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 race condition.
setup.MTCS DEBUG: Mount target: private_revCode: bdcb00ba, private_sndStamp: 1654013960.695753, private_rcvStamp: 1654013960.6961117, private_seqNum: 17958, private_identity: MTMount, private_origin: 28805, elevation: 60.11221937913184, elevationVelocity: 0.003131267998928629, azimuth: 119.79766442788664, azimuthVelocity: 0.0010154320392619434, taiTime: 1654013960.7548969, 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: MTMount azimuth in position: True.
setup.MTCS DEBUG: MTMount azimuth in position True. Waiting settle time 3.0s
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 INFO: MTMount elevation in position: True.
setup.MTCS DEBUG: MTMount elevation in position True. Waiting settle time 3.0s
setup.MTCS DEBUG: [Tel]: Az = +120.006[  -0.2]; El = +060.764[  -0.7] [Rot]: +001.653[  -0.0] [Dome] Az = +000.000; El = +000.000
setup.MTCS DEBUG: Dome azimuth in position.
setup.MTCS DEBUG: Dome elevation in position.
```

add 1um of z7 to the system via OFC

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

In [11]:
```
t = time.Time(utils.current_tai(), format="unix", scale="tai")
t.format = "isot"
print(t.utc)
```

```
2022-05-31T16:19:24.763
```

In [12]:
```
wavefront_errors = np.zeros(19)
```

In [13]:
```
wavefront_errors[3] += 1.0 # add1 um to z7
```

In [14]:
```
await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeou
```

Out[14]:
```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f880eecacd0>
```

This command primes the corrections, the issueCorrection command is needed to actually command them to be sent

In [15]:
```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

Out[15]:
```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f880ee91880>
```

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 noteboook.

---

reset the corrections using the resetCorrection command

Compare the corrections sent vs forces and position changes applied (these are all expected to be zero). This is currently done in a separate notebook or on Chronograf.

In [16]:
```
await mtcs.rem.mtaos.cmd_resetCorrection.start()
```

Out[16]:
```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f880f4e7f10>
```

In [17]:
```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

Out[17]:
```
<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f881fa955b0>
```

add 2um of z7 to the system via OFC

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

```
In [18]: wavefront_errors[3] = 2.0 # add 2.0 um of z7
```

```
In [19]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeou
Out[19]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f880ed0c9a0>
```

```
In [20]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
Out[20]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f880ed98130>
```

Stop Tracking

```
In [21]: await mtcs.stop_tracking()
setup.MTCS DEBUG: Stop tracking.
```

```
In [22]: t = time.Time(utils.current_tai(), format="unix", scale="tai")
         t.format = "isot"
         print(t.utc)
2022-05-31T16:20:54.337
```

Reset corrections

```
In [23]: await mtcs.rem.mtaos.cmd_resetCorrection.start()
Out[23]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f881fe84820>
```

```
In [24]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
Out[24]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7f880f528a60>
```

Check that the corrections in step 10 are twice of those in step 7. This step does not currently involve running any commands in this notebook. This step must be verified using a separate noteboook.

Wrap up. Put each component to the following states: mtaos --> standby m1m3 --> lower mirror --> 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()
```