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

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

```
In [19]:  import rubin_jupyter_utils.lab.notebook as nb
          nb.utils.get_node()
```

```
/tmp/ipykernel_6793/1665379685.py:2: DeprecationWarning: Call to deprecated fu
nction (or staticmethod) get_node. (Please use lsst.rsp.get_node())
  nb.utils.get_node()
```

```
Out[19]:  'yagan03'
```

```
In [20]:  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
```

In [21]:
```python
logging.basicConfig(format="%(name)s:%(message)s", level=logging.DEBUG)
```

In [22]:
```python
log = logging.getLogger("setup")
log.level = logging.DEBUG
```

In [23]:
```python
domain = salobj.Domain()
```

In [24]:
```python
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 [25]:
```python
await mtcs.start_task
```

```
MTHexapod INFO: Read historical data in 0.03 sec
MTHexapod INFO: Read historical data in 0.05 sec
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 24
 of 100 elements
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 11
 of 100 elements
```

Out[25]: [None, None, None, None, None, None, None, None, None, None]

```
MTHexapod.application WARNING: tel_application DDS read queue is filling: 1
5 of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 3
2 of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 1
3 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: 13 of
100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 31 of
100 elements
MTM1M3.logevent_appliedCylinderForces ERROR: evt_appliedCylinderForces DDS
 read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedBalanceForces ERROR: evt_appliedBalanceForces DDS re
ad queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedAzimuthForces ERROR: evt_appliedAzimuthForces DDS re
ad queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedActiveOpticForces ERROR: evt_appliedActiveOpticForce
s DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedAberrationForces ERROR: evt_appliedAberrationForces
 DDS read queue is full (100 elements); data may be lost
```

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 [26]:
```python
target = await mtcs.find_target(el=60, az=120, mag_limit=8)
print(target)
```

WARNING: AstropyDeprecationWarning: Transforming a frame instance to a frame c
lass (as opposed to another frame instance) will not be supported in the futur
e. Either explicitly instantiate the target frame, or first convert the sourc
e frame instance to a `astropy.coordinates.SkyCoord` and use its `transform_to
()` method. [astropy.coordinates.baseframe]
astroquery WARNING: AstropyDeprecationWarning: Transforming a frame instanc
e to a frame class (as opposed to another frame instance) will not be suppo
rted in the future. Either explicitly instantiate the target frame, or fir
st convert the source frame instance to a `astropy.coordinates.SkyCoord` an
d use its `transform_to()` method.
HD 68785

In [27]:
```python
await mtcs.slew_object(target, rot_type=RotType.PhysicalSky, rot=1.9)
```

setup.MTCS INFO: Slewing to HD 68785: 08 12 15.9693 −40 30 10.015
setup.MTCS DEBUG: Setting rotator physical position to 1.9 deg. Rotator wil
l 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: 1.51
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: bdcb00ba, private_sndStam
p: 1652899857.5759044, private_rcvStamp: 1652899857.5762122, private_seqNu
m: 72295, private_identity: MTMount, private_origin: 35669, elevation: 59.9
1362333466776, elevationVelocity: 0.0031343049718555007, azimuth: 119.70036
768826968, azimuthVelocity: 0.0009813651485491408, taiTime: 1652899857.6350
014, 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 3.0
s
setup.MTCS INFO: MTRotator in position: False.
setup.MTCS DEBUG: [Tel]: Az = +119.677[  +0.0]; El = +060.815[  -0.9] [Ro
t]: +001.514[  +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 elevation in position: True.
setup.MTCS DEBUG: MTMount elevation in position True. Waiting settle time
 3.0s
setup.MTCS INFO: MTRotator in position: True.
```

```
setup.MTCS DEBUG: MTRotator in position True. Waiting settle time 3.0s
```

---

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 [30]:
```
t = time.Time(utils.current_tai(), format="unix", scale="utc")
t.format = "isot"
print(t.utc)
```

```
2022-05-18T18:52:04.616
```

In [31]:
```
wavefront_errors = np.zeros(19)
```

In [32]:
```
wavefront_errors[3] += 1.0 # add1 um to z7
```

In [33]:
```
await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1
```

Out[33]:   `<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7faf3d9e24c0>`

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

In [34]:
```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

Out[34]:   `<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7faf4ebff2b0>`

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 [35]:
```
await mtcs.rem.mtaos.cmd_resetCorrection.start()
```

Out[35]:   `<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7faf2476bc70>`

In [36]:
```
await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

Out[36]:   `<ddsutil.MTAOS_ackcmd_fd03e870 at 0x7faf3538b880>`

---

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

```
In [38]:  await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=1
```

```
Out[38]:  <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7faed73702e0>
```

```
In [39]:  await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[39]:  <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7faf4ef3d280>
```

Stop Tracking

```
In [40]:  await mtcs.stop_tracking()
```

```
setup.MTCS DEBUG: Stop tracking.
```

```
In [41]:  t = time.Time(utils.current_tai(), format="unix", scale="tai")
          t.format = "isot"
          print(t.utc)
```

```
2022-05-18T18:59:11.160
```

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()
```