

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 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-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_19862/1665379685.py:2: DeprecationWarning: Call to deprecated function (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 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.
|MTHexapod INFO: Read historical data in 0.04 sec
|MTHexapod INFO: Read historical data in 0.07 sec
|MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling:
11 of 100 elements
|MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling:
10 of 100 elements
|MTHexapod.application WARNING: tel_application DDS read queue is fillin
g: 22 of 100 elements
|MTHexapod.application WARNING: tel_application DDS read queue is fillin
g: 10 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
```

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

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

```
In [9]: await mtcs.enable()
```

```
|setup.MTCS INFO: Enabling all components
|setup.MTCS DEBUG: Expand overrides None
|setup.MTCS DEBUG: Complete overrides: {'mtmount': '', 'mtptg': '', 'mtao
s': '', 'mtm1m3': '', 'mtm2': '', 'mthexapod_1': '', 'mthexapod_2': '',
'mtrotator': '', 'mtdome': '', 'mtdometrajectory': ''}
|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 INFO: All components in <State.ENABLED: 2>.

```

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 [10]: 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.

MTM1M3.powerSupplyData ERROR: tel_powerSupplyData DDS read queue is full (100 elements); data may be lost

MTMount.elevation ERROR: tel_elevation DDS read queue is full (100 elements); data may be lost

MTRotator.rotation ERROR: tel_rotation DDS read queue is full (100 elements); data may be lost

MTHexapod.electrical ERROR: tel_electrical DDS read queue is full (100 elements); data may be lost

MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 73 of 100 elements

MTRotator.motors ERROR: tel_motors DDS read queue is full (100 elements); data may be lost

MTMount.azimuth ERROR: tel_azimuth DDS read queue is full (100 elements); data may be lost

MTHexapod.application WARNING: tel_application DDS read queue is filling: 74 of 100 elements

MTM1M3.pidData ERROR: tel_pidData DDS read queue is full (100 elements); data may be lost

MTHexapod.application ERROR: tel_application DDS read queue is full (100 elements); data may be lost

MTRotator.electrical ERROR: tel_electrical DDS read queue is full (100 elements); data may be lost

MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 73 of 100 elements

MTPtg.mountPosition ERROR: tel_mountPosition DDS read queue is full (100 elements); data may be lost

MTHexapod.actuators ERROR: tel_actuators DDS read queue is full (100 elements); data may be lost

MTM1M3.outerLoopData ERROR: tel_outerLoopData DDS read queue is full (100 elements); data may be lost

MTRotator.ccwFollowingError ERROR: tel_ccwFollowingError DDS read queue is full (100 elements); data may be lost

MTM1M3.inclinometerData ERROR: tel_inclinometerData DDS read queue is full (100 elements); data may be lost

MTM1M3.imsData ERROR: tel_imsData DDS read queue is full (100 elements); data may be lost

MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue is full (100 elements); data may be lost

MTM1M3.hardpointActuatorData ERROR: tel_hardpointActuatorData DDS read queue is full (100 elements); data may be lost

MTM1M3.forceActuatorData ERROR: tel_forceActuatorData DDS read queue is full (100 elements); data may be lost

MTM1M3.accelerometerData ERROR: tel_accelerometerData DDS read queue is full (100 elements); data may be lost

MTM1M3.logevent_forceActuatorWarning ERROR: evt_forceActuatorWarning DDS read queue is full (100 elements); data may be lost

MTM1M3.logevent_appliedThermalForces ERROR: evt_appliedThermalForces DDS read queue is full (100 elements); data may be lost

MTM1M3.logevent_appliedStaticForces ERROR: evt_appliedStaticForces DDS read queue is full (100 elements); data may be lost

MTM1M3.logevent_appliedForces ERROR: evt_appliedForces DDS read queue is full (100 elements); data may be lost

MTM1M3.logevent_appliedElevationForces ERROR: evt_appliedElevationForces DDS read queue is full (100 elements); data may be lost

```
MTM1M3.logevent_appliedCylinderForces ERROR: evt_appliedCylinderForces DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedBalanceForces ERROR: evt_appliedBalanceForces DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedAzimuthForces ERROR: evt_appliedAzimuthForces DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedActiveOpticForces ERROR: evt_appliedActiveOpticForces 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
HD 84235
```

```
In [1]: await mtcs.slew_object(target, rot_type=RotType.PhysicalSky, rot=1.9)
```

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

```
2022-06-21T18:53:40.215
```

```
In [60]: wavefront_errors = np.zeros(19)
```

```
In [61]: wavefront_errors[3] += 1.0 # add 1 um to z7
```

```
In [63]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=60.)
```

```
Out[63]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc1b26ef850>
```

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

```
In [64]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[64]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc1b09dd4f0>
```

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.

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.

For LVV-T2241, we add 1 microns to the previous commands and add them before we reset. The total should be 2.

```
In [65]: wavefront_errors[3] += 1.0 # add 1 um to z7
```

```
In [66]: print(wavefront_errors[3])
```

```
2.0
```

```
In [67]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=60.)
```

```
Out[67]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc1b0851d00>
```

```
In [68]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[68]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc162d95130>
```

```
In [69]: await mtcs.rem.mtaos.cmd_resetCorrection.start()
```

```
Out[69]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc1272b3a60>
```

```
In [70]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[70]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc162c6f880>
```

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

```
In [73]: await mtcs.rem.mtaos.cmd_addAberration.set_start(wf=wavefront_errors, timeout=60.)
```

```
Out[73]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc162e48be0>
```

```
In [74]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[74]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc162fde550>
```

```
In [75]: await mtcs.rem.mtaos.cmd_resetCorrection.start()
```

```
Out[75]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc1b09413a0>
```

```
In [76]: await mtcs.rem.mtaos.cmd_issueCorrection.start(timeout=60.)
```

```
Out[76]: <ddsutil.MTAOS_ackcmd_fd03e870 at 0x7fc1b29023a0>
```

```
In [72]: print(wavefront_errors[3])
```

```
2.0
```

Stop Tracking

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

```
In [77]: t = time.Time(utils.current_tai(), format="unix", scale="tai")
t.format = "isot"
print(t.utc)
```

```
2022-06-21T18:55:16.066
```

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 notebook.

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()
```