

# Closed Loop ComCam Image Ingestion and Application of Correction

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-T2229 (<https://jira.lsstcorp.org/secure/Tests.jspa#testCase/LVV-T2229>)

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 executed by B. Quint

Run the setup.ipnyb notebook to bring all components up and in their enabled position. Check Chronograph.

Bring ComCom online and tranistion it to EnabledState. Check Chronograph.

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

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

/tmp/ipykernel_3740/1665379685.py:2: DeprecationWarning: Call to deprecated function (or staticmethod) get_node. (Please use lsst.rsp.get_node())
nb.utils.get_node()
'yagan0'
```

```
Out[2]:
```

```
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

import yaml

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="%s(%s)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: mtpgt: 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: mtrrotator: 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 0.04 sec
MTHexapod INFO: Read historical data in 0.09 sec
MTHexapod.electrical WARNING: electrical DDS read queue is filling: 21 of 100 elements
MTHexapod.application WARNING: application DDS read queue is filling: 21 of 100 elements
[None, None, None, None, None, None, None, None, None]

MTHexapod.electrical WARNING: electrical DDS read queue is filling: 10 of 100 elements
MTHexapod.actuators WARNING: actuators DDS read queue is filling: 21 of 100 elements
MTHexapod.application WARNING: application DDS read queue is filling: 10 of 100 elements
MTHexapod.actuators WARNING: actuators DDS read queue is filling: 11 of 100 elements
```

```
In [9]: comcam = ComCam(domain=domain, log=log)
```

```
[setup.ComCam DEBUG: cccamera: Adding all resources.
[setup.ComCam DEBUG: ccheaderservice: Adding all resources.
[setup.ComCam DEBUG: ccarchiver: Adding all resources.
```

```
In [10]: comcam.set_rem_loglevel(40)
```

```
In [11]: await comcam.start_task
```

```
Out[11]: [None, None, None]
```

```
In [ ]: await comcam.enable()
```

Find a target around az = 120° and el = 60° and rotator angle at PhysicalSky and 1.8°.

At this position, the rotator stays within a couple of degrees of its initial position. This is because the CCW is not running (MTmount in simulation mode).

target -> az = 120°, el = 60°

Slew to target:

```
In [40]: await mtcs.slew_icrs(ra="20:28:18.74", dec="-87:28:19.9", rot_type=RotType.Physical, rot=0)
```

```
[setup.MTCS DEBUG: Setting rotator to physical fixed position 0.0 deg. Rotator will not track.
[setup.MTCS WARNING: Camera cable wrap following disabled in MTMount.
[setup.MTCS DEBUG: Stop tracking.
[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: 0.00
[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: 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: mtpgt: <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: mtrrotator: <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 rotator in position event.
[setup.MTCS DEBUG: Wait for MTMount elevation in position event.
[setup.MTCS DEBUG: MTMount elevation in position: False.
[setup.MTCS DEBUG: Wait for MTMount azimuth in position event.
[setup.MTCS DEBUG: MTMount azimuth in position: False.
[setup.MTCS INFO: Got False
[setup.MTCS DEBUG: Rotator not in position
[setup.MTCS DEBUG: Mount target: private_revCode: bdc080ba, private_sndStamp: 1643391068.163802, private_rcvStamp: 1643391068.1641915, private_seqNum: 13227, private_identity: MTMount, private_origin: 52777, elevation: 32.82060355925074, elevationVelocity: -3.535573275819742e-05, azimuth: 180.56153570327666, azimuthVelocity: 0.000221664669963457, tailTime: 1643391068.22252, trackId: 8, tracksys: SIDERIAL, radesys: ICRS, priority: 0
[setup.MTCS INFO: Got True
[setup.MTCS INFO: Rotator in position.
[setup.MTCS DEBUG: [Tel]: Az = +000.021[-179.5]; El = +089.989[-57.2] [Rot]: +000.000[-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 DEBUG: [Tel]: Az = +040.717[+139.8]; El = +069.656[-36.8] [Rot]: +000.000[-0.0] [Dome] Az = +000.000; El = +000.000
[setup.MTCS DEBUG: [Tel]: Az = +085.976[+94.6]; El = +047.052[-14.2] [Rot]: +000.000[-0.0] [Dome] Az = +000.000; El = +000.000
[setup.MTCS INFO: MTMount elevation in position: True.
[setup.MTCS DEBUG: [Tel]: Az = +131.273[+49.3]; El = +032.820[-0.0] [Rot]: +000.000[+0.0] [Dome] Az = +000.000; El = +000.000
[setup.MTCS DEBUG: [Tel]: Az = +176.411[+4.2]; El = +032.820[-0.0] [Rot]: +000.000[-0.0] [Dome] Az = +000.000; El = +000.000
[setup.MTCS INFO: MTMount azimuth in position: True.

(<ICRS Coordinate: (ra, dec) in deg
(307.07808333, -87.47219444)>,
<Angle 0. deg>)
```

Once the different components are ready (M1M3, M2, rotator and CCW, hexapods) and tracking, take an image using the take\_image command in playback mode. This second image should be the one that uses the correction calculated with the first slew.

```
In [41]: #exp_focus = await comcam.take_object(15)
#print(f"Target exposure: {exp_focus}")
```

Using the Camera Hexapod, piston ComCam +1mm

```
In [42]: await mtcs.move_camera_hexapod(x=0, y=0, z=1000, u=0, v=0)
```

```
[setup.MTCS INFO: Camera Hexapod compensation mode enabled. Move will offset with respect to LUT.
[setup.MTCS DEBUG: Wait for Camera Hexapod in position event.
[setup.MTCS DEBUG: Camera Hexapod in position: True.
[setup.MTCS DEBUG: Camera Hexapod already in position. Handling potential race condition.
[setup.MTCS INFO: Camera Hexapod in position: False.
[setup.MTCS INFO: Camera Hexapod in position: True.
```

While tracking, take an image with ComCam and check that the header is containing the right telemetry

```
In [43]: exp_intra = await comcam.take_object(15)
print(f"Target 1 exposure: {exp_intra}")

[setup.ComCam DEBUG: Generating group_id
[setup.ComCam DEBUG: imagetype: OBJECT, TCS synchronization not configured.
[setup.ComCam DEBUG: OBJECT 0001 - 0001
Target 1 exposure: [2022012800011]
```

Using the Camera Hexapod, piston ComCam to -1mm

```
In [44]: await mtcs.move_camera_hexapod(x=0, y=0, z=-1000, u=0, v=0)
```

```
[setup.MTCS INFO: Camera Hexapod compensation mode enabled. Move will offset with respect to LUT.
[setup.MTCS DEBUG: Wait for Camera Hexapod in position event.
[setup.MTCS DEBUG: Camera Hexapod in position: True.
[setup.MTCS DEBUG: Camera Hexapod already in position. Handling potential race condition.
[setup.MTCS INFO: Camera Hexapod in position: False.
[setup.MTCS INFO: Camera Hexapod in position: True.
```

While tracking, take an image with ComCam and check that the header is containing the right telemetry.

```
In [45]: exp_extra = await comcam.take_object(15)
print(f"Target 1 exposure: {exp_extra}")

[setup.ComCam DEBUG: Generating group_id
[setup.ComCam DEBUG: imagetype: OBJECT, TCS synchronization not configured.
[setup.ComCam DEBUG: OBJECT 0001 - 0001
Target 1 exposure: [2022012800012]
```

Put the hexapod back to 0mm.

```
In [46]: await mtcs.move_camera_hexapod(x=0, y=0, z=0, u=0, v=0)
```

```
[setup.MTCS INFO: Camera Hexapod compensation mode enabled. Move will offset with respect to LUT.
[setup.MTCS DEBUG: Wait for Camera Hexapod in position event.
[setup.MTCS DEBUG: Camera Hexapod in position: True.
[setup.MTCS DEBUG: Camera Hexapod already in position. Handling potential race condition.
[setup.MTCS INFO: Camera Hexapod in position: False.
[setup.MTCS INFO: Camera Hexapod in position: True.
```

If using MTMount Simulator and CCW Following Mode Disabled, stop tracking to prevent the Rotator to hit the limit switches.

```
In [47]: await mtcs.stop_tracking()
```

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

Use the MTAOS to calculate the required offsets to be sent to M1M3, M2 and the hexapods

Process wavefront data

```
In [34]: run_wep_config = yaml.safe_dump(
dict(
    tasks=dict(
        isr=dict(
            config=dict(
                doOverscan=False,
                doAverageGains=False,
            ),
            generateDonutCatalogWcsTask=dict(
                config={
                    "filterName": "phot_g_mean",
                    "connections.refCatalogs": "gaia_dr2_20200414",
                    "donutSelector.sourceLimit": 10,
                }
            )
        )
    )
)
```

```
In [51]: await mtcs.rem.mtaos.cmd_runWEP.set_start(visitId=exp_intra[0] - 2021111900000,
extraId=exp_extra[0] - 2021111900000,
config=run_wep_config)

-----
AckError Traceback (most recent call last)
/tmp/ipykernel_3740/2927336332.py in <module>
----> 1 await mtcs.rem.mtaos.cmd_runWEP.set_start(visitId=exp_intra[0] - 2021111900000,
      2 extraId=exp_extra[0] - 2021111900000,
      3 config=run_wep_config)

/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-0.7.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.py in set_start(self, timeout, wait_done, **kwargs)
--> 421         self.set(**kwargs)
--> 423         return await self.start(timeout=timeout, wait_done=wait_done)
424
425     async def start(
    /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-0.7.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.py in start(self, data, timeout, wait_done)
    481         )
    482         self.salinfo._running_cmds[seq_num] = cmd_info
--> 483         return await cmd_info.next_ackcmd(timeout=timeout)

/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-0.7.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.py in 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=1414931401, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result='Failed: Error running pipeline task: '
```

```
In [49]: print(exp_intra, exp_extra)

[2022012800011] [2022012800012]
```

Apply the resulting offsets to the M1M3, M2 and the hexapods.

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

Issue the corrections

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

Query the butler to verify that the images are there and check the metadata. This step must be verified using a separate notebook.

## Wrap Up and Shut Down

This cell is not currently included as part of the test execution, but included here as needed to shutdown the systems

```
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.OFFLINE, 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 [ ]: await comcam.standby()
```

```
In [ ]:
```