LVV-T2213 - Look-up Table Application from MTMount Elevation Changes

This notebook was originally written by Bo Xin in the lsst-ts/ts_notebooks repository.
It is a modified version with updated commands and simplified steps.

Make sure you run this notebook on TTS before running at the summit.

Requirements

This notebooks require some extra repositories to be installed locally so it can grab some constants and some look-up tables.
Here is a list of which repos are required to run this notebook:

- lsst-ts/ts_cRIOpy
- lsst-ts/ts_m1m3support
- lsst-ts/ts_config_mttcs
- lsst-sitcom/M2_FEA

Since every user has a different setup, the paths might be slightly different. It is recommended to have all the repositories cloned under `$HOME/notebooks`. You might end up with many repositories and adding an extra folder with the name of the organization they belong might help to find them on GitHub later. For example, this repository would be located in `$HOME/notebooks/lsst-sitcom/notebooks_vandv`. The paths below consider this directory structure but, of course, you are free to organize your folders as you please.

In order to have the required repositories available, open a terminal and run the following commands:

```
git clone https://github.com/lsst-ts/ts_cRIOpy
$HOME/notebooks/lsst-ts/ts_cRIOpy
git clone https://github.com/lsst-ts/ts_m1m3support.git
$HOME/notebooks/lsst-ts/ts_m1m3support
git clone https://github.com/lsst-ts/ts_config_mttcs
$HOME/notebooks/lsst-ts/ts_config_mttcs
git clone https://github.com/lsst-sitcom/M2_FEA
$HOME/notebooks/lsst-sitcom/M2_FEA
```

And add these lines to your `$HOME/notebooks/.user_setup` file:

```
export LSST_DDS_DOMAIN_ID=0
setup -j notebooks_vandv -r $HOME/notebooks/lsst-
```

```
sitcom/notebooks_vandv
setup -j ts_cRIOpy -r $HOME/notebooks/lsst-ts/ts_cRIOpy
```

Finally, you will need to put M1M3 and M2 to use the mount for the look-up table calculations. For M2, you can check the M2 Summit Manual page in Confluence.

In [1]:

```
%load_ext autoreload
%autoreload 2
```

In [2]:

```
from lsst.sitcom import vandv

exec_info = vandv.ExecutionInfo()
print(exec_info)
```

```
lsst.ts.utils.tai INFO: Update leap second table
lsst.ts.utils.tai INFO: current_tai uses the system TAI clock
Executed by b1quint on 2022-06-17T21:49:28.989.
  Running in yagan07 at summit
```

Setup Notebook for Test

- Import all libraries
- Get the remotes ready

In [14]:

```
test_execution = "LVV-E1252" # Updated execution
```

In [4]:

```
import asyncio
import os
import yaml

import astropy.units as u
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from astropy import time
from astropy.coordinates import AltAz, ICRS, EarthLocation, Angle, FK5
from datetime import datetime, timedelta

from lsst_efd_client import EfdClient
from lsst.ts import utils, salobj
from lsst.ts.cRIOpy import M1M3FATable
from lsst.ts.observatory.control.maintel.mtcs import MTCS, MTCSUsages
from lsst.ts.observatory.control import RotType

import lsst.sitcom.vandv as vandv
```

In [5]:

```
print(os.environ["OSPL_URI"])
print(os.environ["LSST_DDS_PARTITION_PREFIX"])
print(os.environ["LSST_DDS_DOMAIN_ID"])
```

```
file:///home/b1quint/WORK/ts_ddsconfig/config/ospl-shmem.xml
summit
0
```

In [6]:
```
logging.basicConfig(format="%(name)s:%(message)s", level=logging.DEBUG)
```

In [7]:
```
log = logging.getLogger("setup")
log.level = logging.DEBUG
```

In [8]:
```
domain = salobj.Domain()
```

In [9]:
```
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 0.01 sec
MTHexapod INFO: Read historical data in 0.02 sec
MTHexapod.electrical ERROR: tel_electrical DDS read queue is full (100 elem
ents); data may be lost
```

In [10]:
```
await mtcs.start_task
```

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

Out[10]:
```
[None, None, None, None, None, None, None, None, None, None]
```

```
MTRotator.rotation ERROR: tel_rotation DDS read queue is full (100 element
s); data may be lost
MTHexapod.application ERROR: tel_application DDS read queue is full (100 el
ements); data may be lost
MTMount.azimuth ERROR: tel_azimuth DDS read queue is full (100 elements); d
ata may be lost
MTRotator.motors ERROR: tel_motors DDS read queue is full (100 elements); d
ata may be lost
MTPtg.mountPosition ERROR: tel_mountPosition DDS read queue is full (100 el
ements); data may be lost
MTRotator.electrical ERROR: tel_electrical DDS read queue is full (100 elem
ents); data may be lost
MTHexapod.actuators ERROR: tel_actuators DDS read queue is full (100 elemen
ts); data may be lost
MTRotator.ccwFollowingError ERROR: tel_ccwFollowingError DDS read queue is
 full (100 elements); data may be lost
MTM1M3.hardpointActuatorData ERROR: tel_hardpointActuatorData DDS read queu
e 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.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue
 is full (100 elements); data may be lost
MTM1M3.powerSupplyData ERROR: tel_powerSupplyData DDS read queue is full (1
00 elements); data may be lost
MTM1M3.imsData ERROR: tel_imsData DDS read queue is full (100 elements); da
ta may be lost
MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue
 is full (100 elements); data may be lost
MTM1M3.accelerometerData ERROR: tel_accelerometerData DDS read queue is ful
l (100 elements); data may be lost
MTM1M3.powerSupplyData ERROR: tel_powerSupplyData DDS read queue is full (1
00 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); da
ta may be lost
MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue
 is full (100 elements); data may be lost
MTM1M3.powerSupplyData ERROR: tel_powerSupplyData DDS read queue is full (1
00 elements); data may be lost
MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue
 is full (100 elements); data may be lost
MTM1M3.accelerometerData ERROR: tel_accelerometerData DDS read queue is ful
l (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); da
ta may be lost
MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData 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); da
ta may be lost
MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue
 is full (100 elements); data may be lost
MTM1M3.accelerometerData ERROR: tel_accelerometerData DDS read queue is ful
l (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); da
ta may be lost
MTM1M3.logevent_appliedForces ERROR: evt_appliedForces DDS read queue is fu
ll (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); da
ta may be lost
MTM1M3.logevent_appliedElevationForces ERROR: evt_appliedElevationForces DD
S 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); da
ta may be lost
MTM1M3.accelerometerData ERROR: tel_accelerometerData DDS read queue is ful
l (100 elements); data may be lost
MTM1M3.imsData ERROR: tel_imsData DDS read queue is full (100 elements); da
ta may be lost
MTM1M3.logevent_appliedCylinderForces ERROR: evt_appliedCylinderForces 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.logevent_appliedAzimuthForces ERROR: evt_appliedAzimuthForces DDS re
ad 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.logevent_appliedActiveOpticForces ERROR: evt_appliedActiveOpticForce
s 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); da
ta may be lost
```

Switching Components from Hardware to Simulator

In [11]:

```
# Put components in OFFLINE and restart their simulator using ArgoCD
await mtcs.set_state(
    salobj.State.OFFLINE,
    components=[
        "mtmount",
        "mtrotator",
        "mthexapod_1",
        # "mthexapod_2",  # M2Hexapod Is always a simulator on Level 3
        # "mtm2",  # Cycle 174
    ]
)
```

```
setup.MTCS DEBUG: [mtmount]::[<State.ENABLED: 2>, <State.DISABLED: 1>, <Sta
te.STANDBY: 5>, <State.OFFLINE: 4>]
setup.MTCS DEBUG: [mtrotator]::[<State.ENABLED: 2>, <State.DISABLED: 1>, <S
tate.STANDBY: 5>, <State.OFFLINE: 4>]
setup.MTCS DEBUG: [mthexapod_1]::[<State.ENABLED: 2>, <State.DISABLED: 1>,
 <State.STANDBY: 5>, <State.OFFLINE: 4>]
```

```
setup.MTCS ERROR: Unable to transition mtm2 to <State.OFFLINE: 4> NoneType:
None
.
Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 193, in next_ackcmd
    ackcmd = await self._wait_task
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 218, in _basic_next_ackcmd
    ackcmd = await asyncio.wait_for(
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/asyncio/tasks.py", line 501, in wait_for
    raise exceptions.TimeoutError()
asyncio.exceptions.TimeoutError

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
7, in set_summary_state
    await cmd.start(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 485, in start
    return await cmd_info.next_ackcmd(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 209, in next_ackcmd
    raise base.AckTimeoutError(
lsst.ts.salobj.base.AckTimeoutError: msg='Timed out waiting for command ack
nowledgement', ackcmd=(ackcmd private_seqNum=1803175370, ack=<SalRetCode.CM
D_NOACK: -301>, error=0, result='No command acknowledgement seen')

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
9, in set_summary_state
    raise RuntimeError(
RuntimeError: Error on cmd=cmd_standby, initial_state=2: msg='Timed out wai
ting for command acknowledgement', ackcmd=(ackcmd private_seqNum=180317537
0, ack=<SalRetCode.CMD_NOACK: -301>, error=0, result='No command acknowledg
ement seen')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [11], in <cell line: 2>()
      1 # Put components in OFFLINE and restart their simulator using ArgoCD
----> 2 await mtcs.set_state(
      3     salobj.State.OFFLINE,
      4     components=[
      5         "mtmount",
      6         "mtrotator",
      7         "mthexapod_1",
      8         # "mthexapod_2",  # M2Hexapod Is always a simulator on Level 3
      9         "mtm2",
     10     ]
     11 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:732, in RemoteGroup.set_state(self, state, overrides,
 components)
    729         self.log.debug(f"[{comp}]::{ret_val[i]!r}")
    731 if error_flag:
--> 732     raise RuntimeError(
    733         f"Failed to transition {failed_components} to "
    734         f"{salobj.State(state)!r}."
    735     )
    736 else:
    737     self.log.info(f"All components in {salobj.State(state)!r}.")

RuntimeError: Failed to transition ['mtm2'] to <State.OFFLINE: 4>.
```

In [15]:

```
# Cycle 174 - M1M3 and M2 should be running in hardware mode
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS ERROR: Unable to transition mtm2 to <State.ENABLED: 2> NoneType:
None
.
Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
7, in set_summary_state
    await cmd.start(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 485, in start
    return await cmd_info.next_ackcmd(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 195, in next_ackcmd
    raise base.AckError(msg="Command failed", ackcmd=ackcmd)
lsst.ts.salobj.base.AckError: msg='Command failed', ackcmd=(ackcmd private_
seqNum=531314070, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result='Faile
d: start command failed.')

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
9, in set_summary_state
    raise RuntimeError(
RuntimeError: Error on cmd=cmd_enable, initial_state=5: msg='Command faile
d', ackcmd=(ackcmd private_seqNum=531314070, ack=<SalRetCode.CMD_FAILED: -3
02>, error=1, result='Failed: start command failed.')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [15], in <cell line: 2>()
      1 # Cycle 174 - M1M3 and M2 should be running in hardware mode
----> 2 await mtcs.set_state(
      3     salobj.State.ENABLED,
      4     components=[
      5         "mtm2",
      6     ]
      7 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:732, in RemoteGroup.set_state(self, state, overrides,
 components)
    729         self.log.debug(f"[{comp}]::{ret_val[i]!r}")
    731 if error_flag:
--> 732     raise RuntimeError(
    733         f"Failed to transition {failed_components} to "
    734         f"{salobj.State(state)!r}."
    735     )
    736 else:
    737     self.log.info(f"All components in {salobj.State(state)!r}.")

RuntimeError: Failed to transition ['mtm2'] to <State.ENABLED: 2>.
```

Then, when all the CSCs above are in OFFLINE, use ArgoCD to bring their correspondent simulators online.

Before I deal with MTM2, I will put `mtmount`, `mtrotator`, and `mtcamhexapod` into the ENABLED state.

In [16]:
```
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mtmount",
    ]
)
```

```
setup.MTCS DEBUG: [mtmount]::[<State.STANDBY: 5>, <State.DISABLED: 1>, <Sta
te.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [17]:
```
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mtrotator",
    ]
)
```

```
setup.MTCS DEBUG: [mtrotator]::[<State.STANDBY: 5>, <State.DISABLED: 1>, <S
tate.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [18]:
```
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mthexapod_1",
    ]
)
```

```
setup.MTCS DEBUG: [mthexapod_1]::[<State.STANDBY: 5>, <State.DISABLED: 1>,
 <State.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

And I will move the mount to the Zenith.

In [19]:
```
await mtcs.point_azel(az=0, el=90)
await mtcs.stop_tracking()
```

```
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: 0.10
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 WARNING: mtm2 not in <State.ENABLED: 2>: <State.DISABLED: 1>
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: 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: False.
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.
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [19], in <cell line: 1>()
----> 1 await mtcs.point_azel(az=0, el=90)
      2 await mtcs.stop_tracking()

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/base_tcs.py:335, in BaseTCS.point_azel(self, az, el, rot_tel, target_n
ame, wait_dome, slew_timeout)
    332 check = self.set_azel_slew_checks(wait_dome=wait_dome)
    334 try:
--> 335     await self._slew_to(
    336         getattr(self.rem, self.ptg_name).cmd_azElTarget,
    337         slew_timeout=slew_timeout,
    338         check=check,
    339     )
    340 except salobj.AckError as ack_err:
    341     self.log.error(
    342         f"Command to slew to azEl target rejected: {ack_err.ackcmd.res
ult}"
    343     )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:289, in MTCS._slew_to(self, slew_cmd, slew_timeout, of
fset_cmd, stop_before_slew, wait_settle, check)
    284         getattr(self.rem, comp).evt_summaryState.flush()
    285         self.scheduled_coro.append(
    286             asyncio.create_task(self.check_component_state(comp))
    287         )
--> 289 await self.process_as_completed(self.scheduled_coro)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:1075, in RemoteGroup.process_as_completed(self, tasks)
   1073 except Exception as e:
   1074     await self.cancel_not_done(tasks)
-> 1075     raise e
   1076 else:
   1077     await self.cancel_not_done(tasks)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:1072, in RemoteGroup.process_as_completed(self, tasks)
   1070 for res in asyncio.as_completed(tasks):
   1071     try:
-> 1072         ret_val = await res
   1073     except Exception as e:
   1074         await self.cancel_not_done(tasks)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/asyncio/tasks.py:619, in as_completed.<locals>._wait_for_one
()
    616 if f is None:
    617     # Dummy value from _on_timeout().
    618     raise exceptions.TimeoutError
--> 619 return f.result()

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:495, in RemoteGroup.check_component_state(self, compon
ent, desired_state)
    493 if state != desired_state:
    494     self.log.warning(f"{component} not in {desired_state!r}: {state!r}
")
--> 495     raise RuntimeError(
    496         f"{component} state is {state!r}, expected {desired_state!r}"
    497     )
    498 else:
    499     self.log.debug(f"{component}: {state!r}")

RuntimeError: mtm2 state is <State.DISABLED: 1>, expected <State.ENABLED: 2>
```

In [20]:

```
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mtmount",
    ]
)
```

```
setup.MTCS DEBUG: [mtmount]::[<State.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [22]:

```
await mtcs.rem.mtmount.cmd_moveToTarget.set_start(azimuth=0, elevation=90)
```

```
---------------------------------------------------------------------------
AckError                                  Traceback (most recent call last)
Input In [22], in <cell line: 1>()
----> 1 await mtcs.rem.mtmount.cmd_moveToTarget.set_start(azimuth=0, elevation
=90)

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:418, i
n RemoteCommand.set_start(self, timeout, wait_done, **kwargs)
    377 """Create a new ``self.data``, set zero or more fields,
    378 and start the command.
    379
   (...)
    415     If ``data`` is not None and not an instance of `DataType`.
    416 """
    417 self.set(**kwargs)
--> 418 return await self.start(timeout=timeout, wait_done=wait_done)

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, i
n 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, i
n 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=918809354, ack=<
SalRetCode.CMD_FAILED: -302>, error=1, result='Failed: Tracking is enabled. St
op tracking before moving.')
```

```
In [23]: await mtcs.stop_tracking()
```

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

```
In [24]: await mtcs.rem.mtmount.cmd_moveToTarget.set_start(azimuth=0, elevation=90)
```

```
Out[24]: <ddsutil.MTMount_ackcmd_d68fb318 at 0x7f440d396a60>
```

```
In [25]: await mtcs.set_state(
             salobj.State.STANDBY,
             components=[
                 "mtm2",
             ]
         )
```

```
setup.MTCS DEBUG: [mtm2]::[<State.DISABLED: 1>, <State.STANDBY: 5>]
setup.MTCS INFO: All components in <State.STANDBY: 5>.
```

```
In [26]: await mtcs.set_state(
             salobj.State.DISABLED,
             components=[
                 "mtm2",
             ]
         )
```

```
setup.MTCS ERROR: Unable to transition mtm2 to <State.DISABLED: 1> NoneTyp
e: None
.
Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
7, in set_summary_state
    await cmd.start(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 485, in start
    return await cmd_info.next_ackcmd(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 195, in next_ackcmd
    raise base.AckError(msg="Command failed", ackcmd=ackcmd)
lsst.ts.salobj.base.AckError: msg='Command failed', ackcmd=(ackcmd private_
seqNum=1890096691, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result='Fail
ed: Timeout in connection. Host: hexrot-vm01.cp.lsst.org, ports: 50000 and
50001')

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
9, in set_summary_state
    raise RuntimeError(
RuntimeError: Error on cmd=cmd_start, initial_state=5: msg='Command faile
d', ackcmd=(ackcmd private_seqNum=1890096691, ack=<SalRetCode.CMD_FAILED: -
302>, error=1, result='Failed: Timeout in connection. Host: hexrot-vm01.c
p.lsst.org, ports: 50000 and 50001')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [26], in <cell line: 1>()
----> 1 await mtcs.set_state(
      2     salobj.State.DISABLED,
      3     components=[
      4         "mtm2",
      5     ]
      6 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/control/remote_group.py:732, in RemoteGroup.set_state(self, state, overrides, components)
    729         self.log.debug(f"[{comp}]::{ret_val[i]!r}")
    731 if error_flag:
--> 732     raise RuntimeError(
    733         f"Failed to transition {failed_components} to "
    734         f"{salobj.State(state)!r}."
    735     )
    736 else:
    737     self.log.info(f"All components in {salobj.State(state)!r}.")

RuntimeError: Failed to transition ['mtm2'] to <State.DISABLED: 1>.
```

In [27]:

```
await mtcs.set_state(
    salobj.State.OFFLINE,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS DEBUG: [mtm2]::[<State.DISABLED: 1>, <State.STANDBY: 5>, <State.OFFLINE: 4>]
setup.MTCS INFO: All components in <State.OFFLINE: 4>.
```

In [28]:

```
await mtcs.set_state(
    salobj.State.DISABLED,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS DEBUG: [mtm2]::[<State.STANDBY: 5>, <State.DISABLED: 1>]
setup.MTCS INFO: All components in <State.DISABLED: 1>.
```

In [29]:

```
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS ERROR: Unable to transition mtm2 to <State.ENABLED: 2> NoneType:
None
.
Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
7, in set_summary_state
    await cmd.start(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 485, in start
    return await cmd_info.next_ackcmd(timeout=timeout)
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote_command.
py", line 195, in next_ackcmd
    raise base.AckError(msg="Command failed", ackcmd=ackcmd)
lsst.ts.salobj.base.AckError: msg='Command failed', ackcmd=(ackcmd private_
seqNum=531314071, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result='Faile
d: start command failed.')

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scip
ipe-3.0.0/lib/python3.8/site-packages/lsst/ts/salobj/csc_utils.py", line 15
9, in set_summary_state
    raise RuntimeError(
RuntimeError: Error on cmd=cmd_enable, initial_state=1: msg='Command faile
d', ackcmd=(ackcmd private_seqNum=531314071, ack=<SalRetCode.CMD_FAILED: -3
02>, error=1, result='Failed: start command failed.')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [29], in <cell line: 1>()
----> 1 await mtcs.set_state(
      2     salobj.State.ENABLED,
      3     components=[
      4         "mtm2",
      5     ]
      6 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:732, in RemoteGroup.set_state(self, state, overrides,
 components)
    729         self.log.debug(f"[{comp}]::{ret_val[i]!r}")
    731 if error_flag:
--> 732     raise RuntimeError(
    733         f"Failed to transition {failed_components} to "
    734         f"{salobj.State(state)!r}."
    735     )
    736 else:
    737     self.log.info(f"All components in {salobj.State(state)!r}.")

RuntimeError: Failed to transition ['mtm2'] to <State.ENABLED: 2>.
```

In [30]:

```
await mtcs.set_state(
    salobj.State.STANDBY,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS ERROR: Unable to transition mtm2 to <State.STANDBY: 5> NoneType: None
.
Traceback (most recent call last):
  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/csc_utils.py", line 157, in set_summary_state
    await cmd.start(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", line 485, in start
    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", line 195, in next_ackcmd
    raise base.AckError(msg="Command failed", ackcmd=ackcmd)
lsst.ts.salobj.base.AckError: msg='Command failed', ackcmd=(ackcmd private_seqNum=1803175373, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result='')

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  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/csc_utils.py", line 159, in set_summary_state
    raise RuntimeError(
RuntimeError: Error on cmd=cmd_standby, initial_state=3: msg='Command failed', ackcmd=(ackcmd private_seqNum=1803175373, ack=<SalRetCode.CMD_FAILED: -302>, error=1, result='')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [30], in <cell line: 1>()
----> 1 await mtcs.set_state(
      2     salobj.State.STANDBY,
      3     components=[
      4         "mtm2",
      5     ]
      6 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/control/remote_group.py:732, in RemoteGroup.set_state(self, state, overrides, components)
    729         self.log.debug(f"[{comp}]::{ret_val[i]!r}")
    731 if error_flag:
--> 732     raise RuntimeError(
    733         f"Failed to transition {failed_components} to "
    734         f"{salobj.State(state)!r}."
    735     )
    736 else:
    737     self.log.info(f"All components in {salobj.State(state)!r}.")

RuntimeError: Failed to transition ['mtm2'] to <State.STANDBY: 5>.
```

The errors above are caused by a difference between the inclinometer angle and the mount elevation angle.

I closed the EUI and rolled back the configuration where M2 would look for the elevation in the mtmount. Right now, I cannot transition M2 CSC to STANDBY.
So I will kill the M2 CSC in ArgoCD and start again.

In [32]:

```
await mtcs.set_state(
    salobj.State.DISABLED,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS DEBUG: [mtm2]::[<State.STANDBY: 5>, <State.DISABLED: 1>]
setup.MTCS INFO: All components in <State.DISABLED: 1>.
```

In [33]:

```
await mtcs.set_state(
    salobj.State.ENABLED,
    components=[
        "mtm2",
    ]
)
```

```
setup.MTCS DEBUG: [mtm2]::[<State.DISABLED: 1>, <State.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [35]:

```
m2_conf = mtcs.rem.mtm2.evt_configurationApplied.get()
```

In [36]:

```
m2_conf.configurations
```

Out[36]:

```
'_init.yaml,_summit.yaml'
```

In [38]:

```
print( mtcs.rem.mtm2.evt_inclinationTelemetrySource.get() )
```

```
private_revCode: 60057e9f, private_sndStamp: 1655510918.7971323, private_rcvSt
amp: 1655510918.7974973, private_seqNum: 1, private_identity: MTM2, private_or
igin: 23490, source: 1, priority: 0
```

In [40]:

```
print( mtcs.rem.mtm2.tel_zenithAngle.get() )
```

```
private_revCode: 02204b0f, private_sndStamp: 1655511451.7878368, private_rcvSt
amp: 1655511451.7879968, private_seqNum: 170, private_identity: MTM2, private_
origin: 23490, measured: 0.41, inclinometerRaw: 89.47, inclinometerProcessed:
89.59
```

In [41]:

```
print( mtcs.rem.mtmount.tel_elevation.get() )
```

```
private_revCode: b99a7959, private_sndStamp: 1655511639.5354877, private_rcvSt
amp: 1655511639.5356975, private_seqNum: 60247, private_identity: MTMount, pri
vate_origin: 52812, actualPosition: 90.0, demandPosition: 90.0, actualVelocit
y: 0.0, demandVelocity: 0.0, actualAcceleration: 0.0, actualTorque: 0.0, times
tamp: 1655511639.535343
```

We decided not to make M2 to use the LUT for the tests tonight considering that we do not have enought support.
So we will run this and the following tests without vefirying M2.

Helper Functions

In [93]:

```
async def get_data_from_efd(location, start, end):
    """
    Retrieves data relevant for analysis.

    Parameters
    ----------
    location : str
        In which Test Stand are you running this test?
    start : str or datetime
        Start of the time-window.
    end : str or datetime
        End of the time-window.
    """
    if location == "summit":
        client = EfdClient("summit_efd")
    elif location == "tucson":
        client = EfdClient("tucson_teststand_efd")
    else:
        raise ValueError(
            "Location does not match any valid options {summit|tucson}"
        )

    _df_mount_el = await client.select_time_series(
        "lsst.sal.MTMount.elevation",
        fields="actualPosition",
        start=start.utc,
        end=end.utc,
    )

    _df_m1m3 = await client.select_time_series(
        "lsst.sal.MTM1M3.forceActuatorData",
        fields="*",
        start=start.utc,
        end=end.utc,
    )

    # M2 B1
    _df_m2 = await client.select_time_series(
        "lsst.sal.MTM2.axialForce",
        fields="*",
        start=start.utc,
        end=end.utc,
    )

    # CamHex Y position
    _df_camhex = await client.select_time_series(
        "lsst.sal.MTHexapod.application",
        fields=[
            "position1"
        ],
        index=1,
        start=start.utc,
        end=end.utc,
    )

    # M2Hex Y position
    _df_m2hex = await client.select_time_series(
        "lsst.sal.MTHexapod.application",
        fields=[
            "position1"
        ],
        index=2,
        start=start.utc,
        end=end.utc,
    )

    # Rename columns
    _df_mount_el = _df_mount_el.rename(
        columns={"actualPosition": "elevation"})
    _df_m1m3 = _df_m1m3.rename(
        columns={c: f"mtm1m3.forceActuatorData.{c}" for c in _df_m1m3.columns})
    _df_m2 = _df_m2.rename(
        columns={c: f"mtm2.axialForce.{c}" for c in _df_m2.columns})
    _df_camhex = _df_camhex.rename(
        columns={c: f"mthexapod_1.application.{c}" for c in _df_camhex.columns}
    _df_m2hex = _df_m2hex.rename(
        columns={c: f"mthexapod_2.application.{c}" for c in _df_m2hex.columns})

    # Join dataframes
    _df = _df_mount_el

    _df = pd.merge(
        _df,
        _df_m1m3,
        left_index=True,
        right_index=True,
        how="outer"
    )

    _df = pd.merge(
        _df,
        _df_m2,
        left_index=True,
        right_index=True,
        how="outer"
    )

    _df = pd.merge(
        _df,
        _df_camhex,
        left_index=True,
        right_index=True,
        how="outer"
    )

    _df = pd.merge(
        _df,
        _df_m2hex,
        left_index=True,
        right_index=True,
        how="outer"
    )

    return _df
```

In [65]:

```
async def plotM1M3Forces(mtm1m3):

    fel = await mtm1m3.evt_appliedElevationForces.aget(timeout=10.)
    faz = await mtm1m3.evt_appliedAzimuthForces.aget(timeout=10.)
    fth = await mtm1m3.evt_appliedThermalForces.aget(timeout=10.)
    fba = await mtm1m3.evt_appliedBalanceForces.aget(timeout=10.)
    fac = await mtm1m3.evt_appliedAccelerationForces.aget(timeout=10.)
    fve = await mtm1m3.evt_appliedVelocityForces.aget(timeout=10.)
    fst = await mtm1m3.evt_appliedStaticForces.aget(timeout=10.)
    fab = await mtm1m3.evt_appliedAberrationForces.aget(timeout=10.)
    fof = await mtm1m3.evt_appliedOffsetForces.aget(timeout=10.)
    fao = await mtm1m3.evt_appliedActiveOpticForces.aget(timeout=10.)
    fapp = await mtm1m3.evt_appliedForces.aget(timeout=10.)

    ftel = await mtm1m3.tel_forceActuatorData.next(flush=True, timeout=10.)

    # Get the position of the actuators
    fat = np.array(M1M3FATable.FATABLE)
    xact = np.float64(fat[:, M1M3FATable.FATABLE_XPOSITION])
    yact = np.float64(fat[:, M1M3FATable.FATABLE_YPOSITION])

    # Create the plot
    fig, ax = plt.subplots(3,1, figsize=(15,8))
    print(fel.xForces)
    ax[0].plot(fel.xForces, '-o', label='elevation');
    ax[0].plot(fba.xForces, label='FB')
    ax[0].plot(fst.xForces, label='static')
    ax[0].plot(ftel.xForce, '-v', label='measured')
    ax[0].legend()
    ax[0].set_title('XForces')

    ax[1].plot(fel.yForces, '-o', label='elevation');
    ax[1].plot(fba.yForces, label='FB')
    ax[1].plot(fst.yForces, label='static')
    ax[1].plot(ftel.yForce, '-v', label='measured')
    ax[1].legend()
    ax[1].set_title('YForces')
    ax[2].plot(fel.zForces, '-o', label='elevation');
    ax[2].plot(fba.zForces, label='FB')
    ax[2].plot(fst.zForces, label='static')
    ax[2].plot(fao.zForces, label='AOS')
    ax[2].plot(ftel.zForce, '-v', label='measured')
    ax[2].set_title('ZForces')
    ax[2].legend()

    fig2, ax=plt.subplots( 1,3, figsize = [15,4])
    aa = np.array(fao.zForces)
    img = ax[0].scatter(xact, yact, c=aa)
    ax[0].axis('equal')
    ax[0].set_title('AOS forces')
    fig.colorbar(img, ax=ax[0])

    aa = np.array(fel.zForces)
    img = ax[1].scatter(xact, yact, c=aa)
    ax[1].axis('equal')
    ax[1].set_title('elevation forces')
    fig.colorbar(img, ax=ax[1])

    aa = np.array(fst.zForces)
    img = ax[2].scatter(xact, yact, c=aa)
    ax[2].axis('equal')
    ax[2].set_title('static forces')
    fig.colorbar(img, ax=ax[2])
```

Start all the components, and put them in an enabled state.

Depending on which test cycle this is being executed in, each component is either a hardware component or a simulator:

- [ ] M1M3
- [ ] M2
- [ ] M2 Hexapod
- [ ] Camera Hexapod
- [ ] MTMount

The best/simplest way of doing this is running the notebook that executes the LVV-T2344 test case.
This notebook puts all the components in a ENABLED state.

In [42]:

```
# Verify that all the components have heartbeats
await mtcs.assert_liveliness()
```

In [43]:

```
# Verify that all the components are enabled
await mtcs.assert_all_enabled()
```

Move mtmount to Zenith

Command the mount to elevation = 90, azimuth = 0, so that we can start m1m3 with LUT in mount telemetry mode).

In [44]:

```
await mtcs.point_azel(az=0, el=90)
await mtcs.stop_tracking()
```

```
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: -0.10
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: 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: False.
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: 1655511997.9772472, private_rcvStamp: 1655511997.9775076, private_seqNu
m: 3283, private_identity: MTMount, private_origin: 52812, elevation: 90.0,
elevationVelocity: 0.0, azimuth: 0.0, azimuthVelocity: 0.0, taiTime: 165551
1998.0364313, trackId: 3, tracksys: LOCAL, radesys: , priority: 0
setup.MTCS DEBUG: [Tel]: Az = +000.000[  +0.0]; El = +022.000[ +68.0] [Ro
t]: -000.098[  +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: MTRotator in position: True.
setup.MTCS DEBUG: MTRotator in position True. Waiting settle time 3.0s
setup.MTCS DEBUG: No new in position event in the last 3.0s. Assuming MTMou
nt elevation in position.
setup.MTCS DEBUG: MTMount elevation in position True. Waiting settle time
 3.0s
setup.MTCS DEBUG: No new in position event in the last 3.0s. Assuming MTMou
nt azimuth in position.
setup.MTCS DEBUG: MTMount azimuth in position True. Waiting settle time 3.0
s
setup.MTCS DEBUG: Stop tracking.
```

Get M1M3 Ready

- Raise the mirror,
- Turn on Balance Forces,
- Clear forces

Need to have M1M3 LUT use mount telemetry.

In [46]:
```
await mtcs.lower_m1m3() # M1M3 was previously raised
```

```
setup.MTCS DEBUG: M1M3 current detailed state {<DetailedState.ACTIVEENGINEE
RING: 11>, <DetailedState.ACTIVE: 7>}, executing command...
setup.MTCS DEBUG: process as completed...
setup.MTCS DEBUG: M1M3 detailed state 8
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
setup.MTCS DEBUG: M1M3 detailed state 5
```

In [47]:
```
# M1M3 LUT use mount telemetry

# Put M1M3 in offline
await mtcs.set_state(salobj.State.OFFLINE, components=["mtm1m3"])

# Access the m1m3-crio machine via SSH
# ssh admin@m1m3-crio-ss.cp.lsst.org
# the password is available in 1password, MainTel Vault

# Change the UseInclinometer parameter in the file below from True to False
# vi /var/lib/ts-M1M3support/Sets/Default/1/ForceActuatorSettings.yaml

# Start M1M3 back again
```

```
setup.MTCS DEBUG: [mtm1m3]::[<State.ENABLED: 2>, <State.DISABLED: 1>, <Stat
e.STANDBY: 5>, <State.OFFLINE: 4>]
setup.MTCS INFO: All components in <State.OFFLINE: 4>.
```

When M1M3 LUT is using the mount to get the elevation, avoid changes greater than 1 deg per command and avoid going lower than 82.5 deg

In [48]:
```
# If M1M3 was enabled before, disabled it first and enabled again to start fres
await mtcs.set_state(salobj.State.ENABLED, components=["mtm1m3"], overrides={"n
```

```
setup.MTCS DEBUG: [mtm1m3]::[<State.STANDBY: 5>, <State.DISABLED: 1>, <Stat
e.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [49]:
```
# If M1M3 is not raised yet, use this command to raise it.
await mtcs.raise_m1m3()
```

```
setup.MTCS DEBUG: M1M3 current detailed state {<DetailedState.PARKEDENGINEE
RING: 9>, <DetailedState.PARKED: 5>}, executing command...
setup.MTCS DEBUG: process as completed...
setup.MTCS DEBUG: M1M3 detailed state 6
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
setup.MTCS WARNING: mtm1m3 not in <State.ENABLED: 2>: <State.FAULT: 3>
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [49], in <cell line: 2>()
      1 # If M1M3 is not raised yet, use this command to raise it.
----> 2 await mtcs.raise_m1m3()

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:650, in MTCS.raise_m1m3(self)
    648 async def raise_m1m3(self):
    649     """Raise M1M3."""
--> 650     await self._execute_m1m3_detailed_state_change(
    651         execute_command=self._handle_raise_m1m3,
    652         initial_detailed_states={
    653             MTM1M3.DetailedState.PARKED,
    654             MTM1M3.DetailedState.PARKEDENGINEERING,
    655         },
    656         final_detailed_states={
    657             MTM1M3.DetailedState.ACTIVE,
    658             MTM1M3.DetailedState.ACTIVEENGINEERING,
    659         },
    660     )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:712, in MTCS._execute_m1m3_detailed_state_change(self,
execute_command, initial_detailed_states, final_detailed_states)
    708 if m1m3_detailed_state.detailedState in initial_detailed_states:
    709     self.log.debug(
    710         f"M1M3 current detailed state {initial_detailed_states!r}, exe
cuting command..."
    711     )
--> 712     await execute_command()
    713 elif m1m3_detailed_state.detailedState in final_detailed_states:
    714     self.log.info(
    715         f"M1M3 current detailed state {final_detailed_states!r}. Nothi
ng to do."
    716     )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:736, in MTCS._handle_raise_m1m3(self)
    733 else:
    734     await self.rem.mtm1m3.cmd_raiseM1M3.set_start(timeout=self.long_ti
meout)
--> 736 await self._handle_m1m3_detailed_state(
    737     expected_m1m3_detailed_state=MTM1M3.DetailedState.ACTIVE,
    738     unexpected_m1m3_detailed_states={
    739         MTM1M3.DetailedState.LOWERING,
    740     },
    741 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:795, in MTCS._handle_m1m3_detailed_state(self, expecte
d_m1m3_detailed_state, unexpected_m1m3_detailed_states)
    772 """Handle m1m3 detailed state.
    773 
    774 Parameters
   (...)
    780     these states, raise an exception.
    781 """
    783 m1m3_raise_check_tasks = [
    784     asyncio.create_task(
    785         self._wait_for_mtm1m3_detailed_state(
   (...)
    793     ),
    794 ]
--> 795 await self.process_as_completed(m1m3_raise_check_tasks)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:1075, in RemoteGroup.process_as_completed(self, tasks)
   1073 except Exception as e:
   1074     await self.cancel_not_done(tasks)
-> 1075     raise e
   1076 else:
   1077     await self.cancel_not_done(tasks)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:1072, in RemoteGroup.process_as_completed(self, tasks)
   1070 for res in asyncio.as_completed(tasks):
   1071     try:
-> 1072         ret_val = await res
   1073     except Exception as e:
   1074         await self.cancel_not_done(tasks)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/asyncio/tasks.py:619, in as_completed.<locals>._wait_for_one
()
    616 if f is None:
    617     # Dummy value from _on_timeout().
    618     raise exceptions.TimeoutError
--> 619 return f.result()

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:495, in RemoteGroup.check_component_state(self, compon
ent, desired_state)
    493 if state != desired_state:
    494     self.log.warning(f"{component} not in {desired_state!r}: {state!r}
")
--> 495     raise RuntimeError(
    496         f"{component} state is {state!r}, expected {desired_state!r}"
    497     )
    498 else:
    499     self.log.debug(f"{component}: {state!r}")

RuntimeError: mtm1m3 state is <State.FAULT: 3>, expected <State.ENABLED: 2>
```

In [50]:

```
# If M1M3 was enabled before, disabled it first and enabled again to start fres
await mtcs.set_state(salobj.State.STANDBY, components=["mtm1m3"], overrides={"m
```

```
setup.MTCS DEBUG: [mtm1m3]::[<State.FAULT: 3>, <State.STANDBY: 5>]
setup.MTCS INFO: All components in <State.STANDBY: 5>.
```

In [51]:

```
# If M1M3 was enabled before, disabled it first and enabled again to start fres
await mtcs.set_state(salobj.State.ENABLED, components=["mtm1m3"], overrides={"m
```

```
setup.MTCS DEBUG: [mtm1m3]::[<State.STANDBY: 5>, <State.DISABLED: 1>, <Stat
e.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [52]:

```
# If M1M3 is not raised yet, use this command to raise it.
await mtcs.raise_m1m3()
```

```
setup.MTCS DEBUG: M1M3 current detailed state {<DetailedState.PARKEDENGINEE
RING: 9>, <DetailedState.PARKED: 5>}, executing command...
setup.MTCS DEBUG: process as completed...
setup.MTCS DEBUG: M1M3 detailed state 6
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
MTM1M3.logevent_logMessage ERROR: evt_logMessage DDS read queue is full (10
0 elements); data may be lost
MTM1M3.logevent_logMessage ERROR: evt_logMessage DDS read queue is full (10
0 elements); data may be lost
setup.MTCS WARNING: mtm1m3 not in <State.ENABLED: 2>: <State.FAULT: 3>
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Input In [52], in <cell line: 2>()
      1 # If M1M3 is not raised yet, use this command to raise it.
----> 2 await mtcs.raise_m1m3()

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:650, in MTCS.raise_m1m3(self)
    648 async def raise_m1m3(self):
    649     """Raise M1M3."""
--> 650     await self._execute_m1m3_detailed_state_change(
    651         execute_command=self._handle_raise_m1m3,
    652         initial_detailed_states={
    653             MTM1M3.DetailedState.PARKED,
    654             MTM1M3.DetailedState.PARKEDENGINEERING,
    655         },
    656         final_detailed_states={
    657             MTM1M3.DetailedState.ACTIVE,
    658             MTM1M3.DetailedState.ACTIVEENGINEERING,
    659         },
    660     )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:712, in MTCS._execute_m1m3_detailed_state_change(self,
execute_command, initial_detailed_states, final_detailed_states)
    708 if m1m3_detailed_state.detailedState in initial_detailed_states:
    709     self.log.debug(
    710         f"M1M3 current detailed state {initial_detailed_states!r}, exe
cuting command..."
    711     )
--> 712     await execute_command()
    713 elif m1m3_detailed_state.detailedState in final_detailed_states:
    714     self.log.info(
    715         f"M1M3 current detailed state {final_detailed_states!r}. Nothi
ng to do."
    716     )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:736, in MTCS._handle_raise_m1m3(self)
    733 else:
    734     await self.rem.mtm1m3.cmd_raiseM1M3.set_start(timeout=self.long_ti
meout)
--> 736 await self._handle_m1m3_detailed_state(
    737     expected_m1m3_detailed_state=MTM1M3.DetailedState.ACTIVE,
    738     unexpected_m1m3_detailed_states={
    739         MTM1M3.DetailedState.LOWERING,
    740     },
    741 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/maintel/mtcs.py:795, in MTCS._handle_m1m3_detailed_state(self, expecte
d_m1m3_detailed_state, unexpected_m1m3_detailed_states)
    772 """Handle m1m3 detailed state.
    773 
    774 Parameters
   (...)
    780     these states, raise an exception.
    781 """
    783 m1m3_raise_check_tasks = [
    784     asyncio.create_task(
    785         self._wait_for_mtm1m3_detailed_state(
   (...)
    793     ),
    794 ]
--> 795 await self.process_as_completed(m1m3_raise_check_tasks)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:1075, in RemoteGroup.process_as_completed(self, tasks)
   1073 except Exception as e:
   1074     await self.cancel_not_done(tasks)
-> 1075     raise e
   1076 else:
   1077     await self.cancel_not_done(tasks)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:1072, in RemoteGroup.process_as_completed(self, tasks)
   1070 for res in asyncio.as_completed(tasks):
   1071     try:
-> 1072         ret_val = await res
   1073     except Exception as e:
   1074         await self.cancel_not_done(tasks)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/asyncio/tasks.py:619, in as_completed.<locals>._wait_for_one
()
    616 if f is None:
    617     # Dummy value from _on_timeout().
    618     raise exceptions.TimeoutError
--> 619 return f.result()

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/remote_group.py:495, in RemoteGroup.check_component_state(self, compon
ent, desired_state)
    493 if state != desired_state:
    494     self.log.warning(f"{component} not in {desired_state!r}: {state!r}
")
--> 495     raise RuntimeError(
    496         f"{component} state is {state!r}, expected {desired_state!r}"
    497     )
    498 else:
    499     self.log.debug(f"{component}: {state!r}")

RuntimeError: mtm1m3 state is <State.FAULT: 3>, expected <State.ENABLED: 2>
```

```
setup.MTCS DEBUG: M1M3 detailed state 13
setup.MTCS DEBUG: M1M3 detailed state 2
```

It seems that M1M3 is not happy to use the mount elevation to calculate the LUT.

So we cannot run this test now.

I will recover and put the M1M3 LUT to use the inclinometer elevation.

In [53]:

```
# If M1M3 was enabled before, disabled it first and enabled again to start fres
await mtcs.set_state(salobj.State.OFFLINE, components=["mtm1m3"], overrides={"n
```

```
setup.MTCS DEBUG: M1M3 detailed state 4
setup.MTCS DEBUG: M1M3 detailed state 3
setup.MTCS DEBUG: [mtm1m3]::[<State.FAULT: 3>, <State.STANDBY: 5>, <State.O
FFLINE: 4>]
setup.MTCS INFO: All components in <State.OFFLINE: 4>.
```

In [54]:
```
# If M1M3 was enabled before, disabled it first and enabled again to start fres
await mtcs.set_state(salobj.State.ENABLED, components=["mtm1m3"], overrides={"m
```

```
setup.MTCS DEBUG: M1M3 detailed state 1
setup.MTCS DEBUG: M1M3 detailed state 5
setup.MTCS DEBUG: [mtm1m3]::[<State.STANDBY: 5>, <State.DISABLED: 1>, <Stat
e.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [55]:
```
# If M1M3 is not raised yet, use this command to raise it.
await mtcs.raise_m1m3()
```

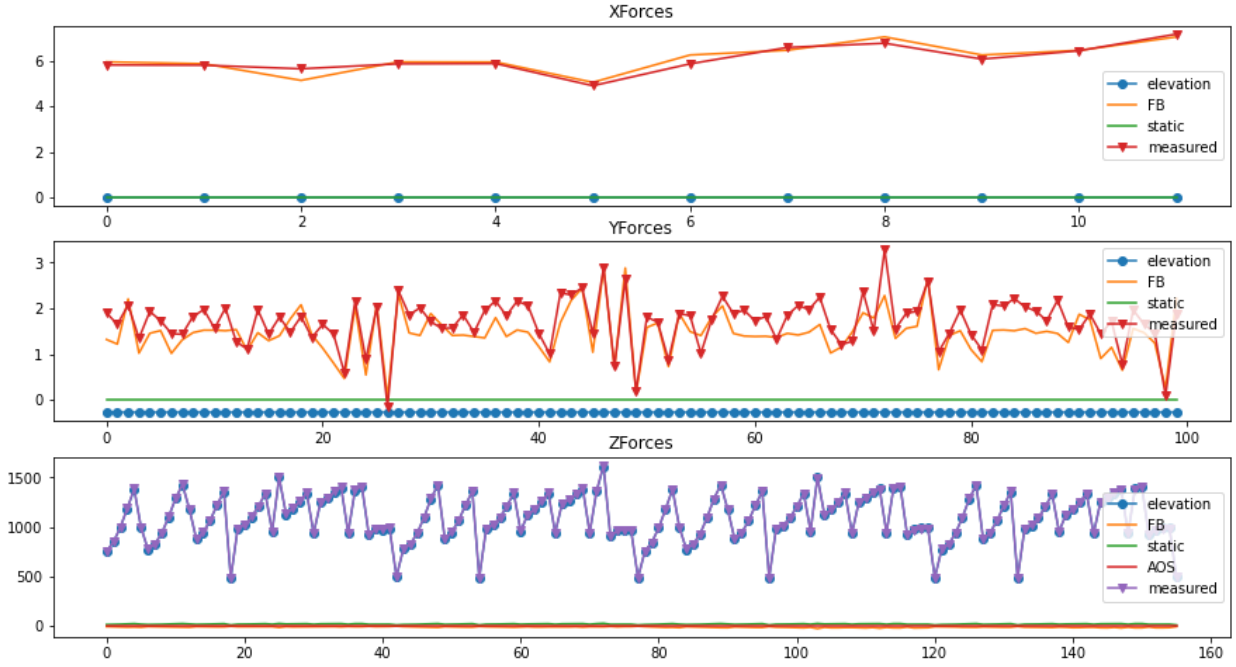
```
setup.MTCS DEBUG: M1M3 current detailed state {<DetailedState.PARKEDENGINEE
RING: 9>, <DetailedState.PARKED: 5>}, executing command...
setup.MTCS DEBUG: M1M3 detailed state 6
setup.MTCS DEBUG: process as completed...
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2>
setup.MTCS DEBUG: M1M3 detailed state 7
setup.MTCS DEBUG: M1M3 detailed state 7
```

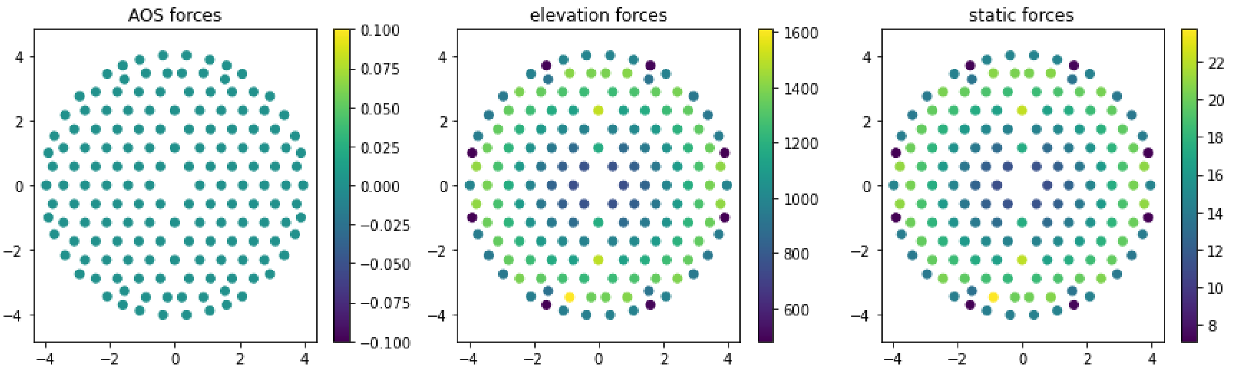
In [57]:
```
# Enables M1M3 Force Balance system using the hardpoints
await mtcs.enable_m1m3_balance_system()
```

```
setup.MTCS DEBUG: Enabling hardpoint corrections.
```

In [66]:
```
await plotM1M3Forces(mtcs.rem.mtm1m3)
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

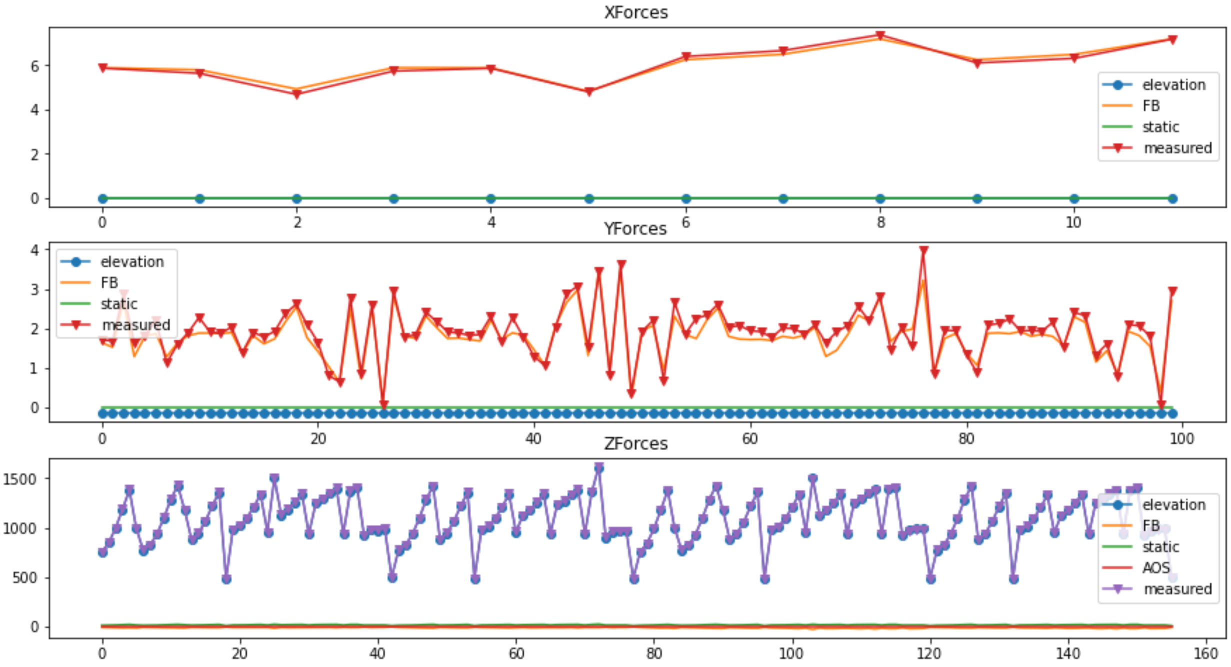


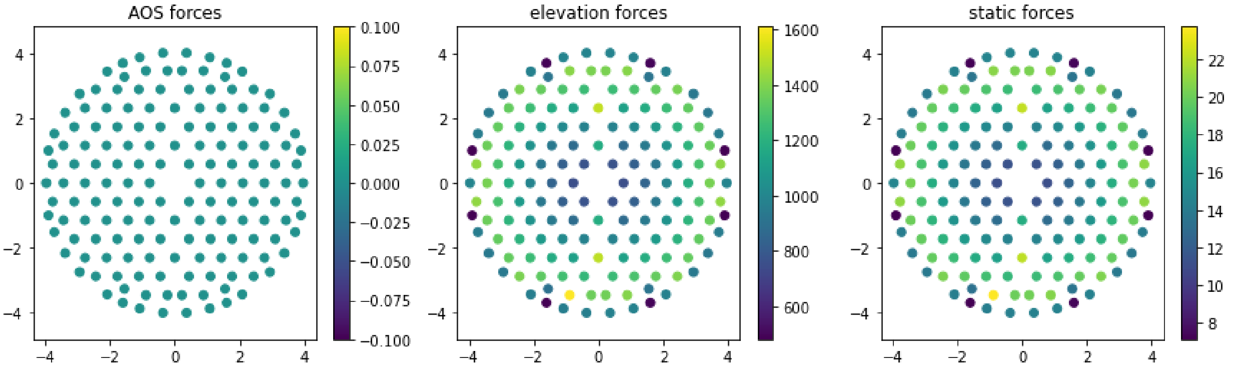


In [67]:
```
# Resets the Aberration Forces and the Active Optics Forces
await mtcs.reset_m1m3_forces()
```

In [68]:
```
await plotM1M3Forces(mtcs.rem.mtm1m3)
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```





Get M2 Ready

- Turn on Force Balance system
- Clear forces

In [69]:
```
# Disable and Enable M2 so we can assure to start fresh
await mtcs.set_state(salobj.State.STANDBY, components=["mtm2"])
await mtcs.set_state(salobj.State.ENABLED, components=["mtm2"])
```

```
setup.MTCS DEBUG: [mtm2]::[<State.ENABLED: 2>, <State.DISABLED: 1>, <State.
STANDBY: 5>]
setup.MTCS INFO: All components in <State.STANDBY: 5>.
setup.MTCS DEBUG: [mtm2]::[<State.STANDBY: 5>, <State.DISABLED: 1>, <State.
ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [70]:
```
# Enabled M2 Force Balance system
await mtcs.enable_m2_balance_system()
```

```
setup.MTCS INFO: M2 force balance system already enabled. Nothing to do.
```

In [73]:
```
# Resets the Active Optics Forces
await mtcs.reset_m2_forces()
```

In [74]:
```
# Need to have M2 LUT use mount telemetry
# todo: how to do that?
```

Get CamHex Ready

- Check config
- Make sure LUT is on, and has valid inputs
- Make sure hex is at LUT position

In [75]:
```
# Disable and Enable CamHex so we can assure to start fresh
await mtcs.set_state(salobj.State.DISABLED, components=["mthexapod_1"])
await mtcs.set_state(salobj.State.ENABLED, components=["mthexapod_1"])
```

```
setup.MTCS DEBUG: [mthexapod_1]::[<State.ENABLED: 2>, <State.DISABLED: 1>]
setup.MTCS INFO: All components in <State.DISABLED: 1>.
setup.MTCS DEBUG: [mthexapod_1]::[<State.DISABLED: 1>, <State.ENABLED: 2>]
setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

In [76]:
```
# Check the configuration
await vandv.hexapod.get_hexapod_configuration(mtcs.rem.mthexapod_1)
```

```
MTHexapodID: 1, private_revCode: 047bbc0a, private_sndStamp: 1655508583.969399
2, private_rcvStamp: 1655508583.9697952, private_seqNum: 1, private_identity:
MTHexapod:1, private_origin: 52068, configurations: _init.yaml,_summit.yaml, v
ersion: v0.8.1-0-g0400d07, url: file:///home/saluser/ts_config_mttcs/MTHexapo
d/v3, schemaVersion: v3, otherInfo: , priority: 0

Pivot at (0.0, 0.0, 500000.0) microns
 maxXY = 11400.0 microns, maxZ = 13100.0 microns
 maxUV = 0.36 deg, maxW = 0.1 deg
```

In [77]:
```
# Enable compensation mode for CamHex
await mtcs.enable_compensation_mode("mthexapod_1")
```

```
setup.MTCS DEBUG: Setting mthexapod_1 compensation mode from False to True.
```

In [78]:
```
# Reset the Camera Hexapod position
```

```
await mtcs.reset_camera_hexapod_position()
```

```
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: False.
setup.MTCS INFO: Camera Hexapod in position: True.
setup.MTCS DEBUG: Camera Hexapod in position True. Waiting settle time 5.0s
```

In [79]:

```
# After resetting the Camera Hexapod position, we want to make sure that
# the compensation and non-compensation values are the same.
await vandv.hexapod.print_hexapod_uncompensation_values(mtcs.rem.mthexapod_1)
await vandv.hexapod.print_hexapod_compensation_values(mtcs.rem.mthexapod_1)
```

```
Uncompensated position
      0.00        0.00        0.00       0.000000    0.000000    0.000000    2022-0
6-18 01:16:41.946214912
Compensated position
     -0.93 um    -652.98 um     295.56 um     -0.017752 deg   0.000000 deg
0.000000 deg    2022-06-18 01:16:41.946400512
```

In [80]:

```
# Need to have CamHex LUT use mount telemetry
await vandv.hexapod.check_hexapod_lut(mtcs.rem.mthexapod_1)
```

```
Does the hexapod has enough inputs to do LUT compensation? (If the below times
out, we do not.)
compsensation mode enabled? True
mount elevation =  90.0
mount azimth =  0.0
rotator angle =  0.0
temperature from sensors on the hex =  0.0
x = -0.93
y = -652.98
z = 295.56
u = -0.02
v = 0.00
w = 0.00
```

Get M2Hex Ready

- Check config
- Make sure LUT is on, and has valid inputs
- Make sure M2Hex is at LUT position

In [81]:

```
# Check the configuration
await vandv.hexapod.get_hexapod_configuration(mtcs.rem.mthexapod_2)
```

```
MTHexapodID: 2, private_revCode: 047bbc0a, private_sndStamp: 1655479264.763605
6, private_rcvStamp: 1655502631.014701, private_seqNum: 1, private_identity: M
THexapod:2, private_origin: 64430, configurations: _init.yaml,_summit.yaml, ve
rsion: v0.8.1-0-g0400d07, url: file:///home/saluser/ts_config_mttcs/MTHexapod/
v3, schemaVersion: v3, otherInfo: , priority: 0

Pivot at (0.0, 0.0, 500000.0) microns
 maxXY = 10500.0 microns, maxZ = 8900.0 microns
 maxUV = 0.175 deg, maxW = 0.05 deg
```

In [82]:
```
# Enable compensation mode for M2Hex
await mtcs.enable_compensation_mode("mthexapod_2")
```

```
setup.MTCS DEBUG: Setting mthexapod_2 compensation mode from False to True.
```

In [83]:
```
# Reset the M2 Hexapod position
await mtcs.reset_m2_hexapod_position()
```

```
setup.MTCS INFO: M2 Hexapod compensation mode enabled. Move will offset wit
h respect to LUT.
setup.MTCS DEBUG: Wait for M2 Hexapod in position event.
setup.MTCS DEBUG: M2 Hexapod in position: True.
setup.MTCS DEBUG: M2 Hexapod already in position. Handling potential race c
ondition.
setup.MTCS INFO: M2 Hexapod in position: False.
setup.MTCS INFO: M2 Hexapod in position: True.
setup.MTCS DEBUG: M2 Hexapod in position True. Waiting settle time 5.0s
```

In [84]:
```
# After resetting the Camera Hexapod position, we want to make sure that
# the compesation and non-compensation values are the same.
await vandv.hexapod.print_hexapod_uncompensation_values(mtcs.rem.mthexapod_2)
await vandv.hexapod.print_hexapod_compensation_values(mtcs.rem.mthexapod_2)
```

```
Uncompensated position
      0.00       0.00       0.00      0.000000    0.000000    0.000000    2022-0
6-18 01:17:33.277017856
Compensated position
     -1.84 um    -572.60 um     304.70 um     -0.006649 deg   0.000000 deg
0.000000 deg    2022-06-18 01:17:33.277177344
```

In [85]:
```
# Need to have CamHex LUT use mount telemetry
await vandv.hexapod.check_hexapod_lut(mtcs.rem.mthexapod_1)
```

```
Does the hexapod has enough inputs to do LUT compensation? (If the below times
out, we do not.)
compsensation mode enabled? True
mount elevation =  90.0
mount azimth =  0.0
rotator angle =  0.0
temperature from sensors on the hex =  0.0
x = -0.93
y = -652.98
z = 295.56
u = -0.02
v = 0.00
w = 0.00
```

Gather Data - Without AO

- command the mount to elevation =86 deg, azimuth = 0
- wait 39s
- command the mount to elevation = 82 deg, azimuth = 0.

In [86]:
```
# Set this to True when you actually want to run this test
t_start = time.Time(time.Time.now(), format="unix", scale="utc")
t_start.format = "isot"
```

```
print(f"Gathering data - without AO - Start time: {t_start.utc}")

await asyncio.sleep(39.)

# Slew to 86 deg
await mtcs.rem.mtmount.cmd_moveToTarget.set_start(azimuth=0, elevation=86)
await asyncio.sleep(39.)

# Slew to 82.5 deg
await mtcs.rem.mtmount.cmd_moveToTarget.set_start(azimuth=0, elevation=82.5)
await asyncio.sleep(39.)

t_end = time.Time(time.Time.now(), format="unix", scale="utc")
t_end.format = "isot"
print(f"Gathering data - without AO - End time: {t_end.utc}")

await mtcs.stop_tracking()
```

```
Gathering data - without AO - Start time: 2022-06-18T01:22:19.040
Gathering data - without AO - End time: 2022-06-18T01:24:20.290
setup.MTCS DEBUG: Stop tracking.
```

Plot Data

In [ ]:

```
# t_start = "2022-06-06T16:33:47.387"
# t_start = time.Time(t_start, format="isot", scale="tai")

# t_end = "2022-06-06T16:36:25.569"
# t_end = time.Time(t_end, format="isot", scale="tai")
```

Plot Optics vs Time

Plot the following as a function of time during the above process:

- mount elevation
- m1m3 actuator 101 z force
- m2 actuator B1 force
- camera hex y position
- m2 hex y position

In [98]:

```
t_start.utc
```

Out[98]:

```
<Time object: scale='utc' format='isot' value=2022-06-18T01:22:19.040>
```

In [94]:

```
df = await get_data_from_efd(
    exec_info.loc,
    t_start,
    t_end)
```

```
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 12
 of 100 elements
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
2 of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 1
2 of 100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 11 of
100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 12 of
100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 11 of
100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 11 of
100 elements
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 15
 of 100 elements
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 15
 of 100 elements
Empty DataFrame
Columns: []
Index: []
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: 1
5 of 100 elements
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 fu
ll (100 elements); data may be lost
MTM1M3.logevent_appliedElevationForces ERROR: evt_appliedElevationForces DD
S 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 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
```

In [88]:

```
df["elevation"].dropna()
```

Out[88]:

```
2022-06-18 01:22:19.053000+00:00    90.0
2022-06-18 01:22:19.089000+00:00    22.0
2022-06-18 01:22:19.104000+00:00    90.0
2022-06-18 01:22:19.155000+00:00    90.0
2022-06-18 01:22:19.206000+00:00    90.0
                                    ... 
2022-06-18 01:24:20.135000+00:00    82.5
2022-06-18 01:24:20.185000+00:00    82.5
2022-06-18 01:24:20.228000+00:00    22.0
2022-06-18 01:24:20.236000+00:00    82.5
2022-06-18 01:24:20.287000+00:00    82.5
Name: elevation, Length: 2975, dtype: float64
```

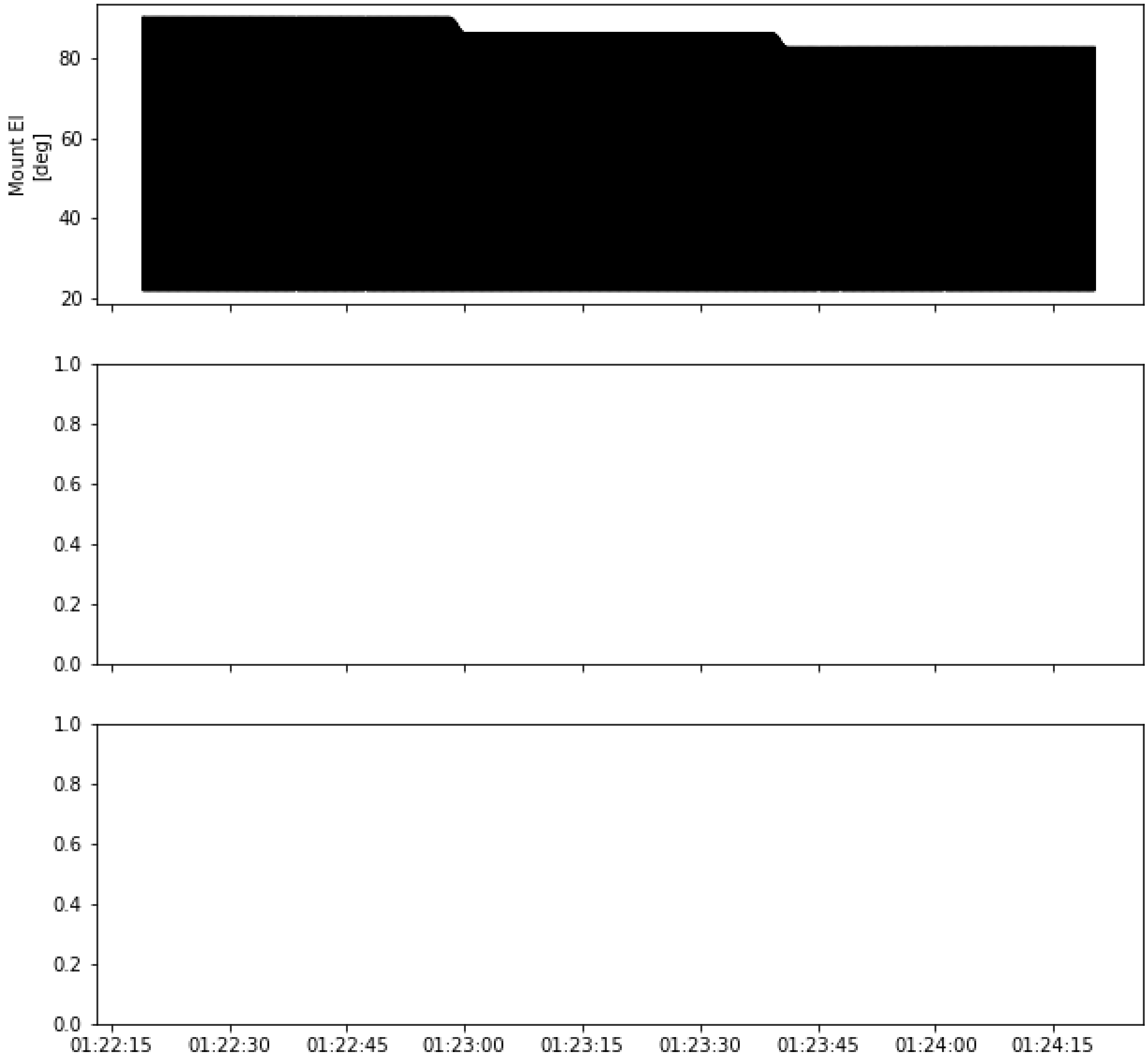
In [ ]:

In [92]:
```
print([c for c in df.columns if "mtm2" in c])
```

```
[]
```

In [89]:
```
fig, axs = plt.subplots(figsize=(10, 10), nrows=3, sharex=True)

axs[0].plot(df["elevation"].dropna(), "k", label="Mount Elevation")
axs[0].set_ylabel("Mount El\n[deg]")

axs[1].plot(df["mtm2.axialForce.applied0"].dropna(), "C1^-", label="applied")
axs[1].plot(df["mtm2.axialForce.lutGravity0"].dropna(), "C2v-", label="Gravity
axs[1].set_ylabel("M2 Forces\n[--]")

axs[2].plot(df["mthexapod_1.application.position1"].dropna(), "C3x-", label="Ca
axs[2].plot(df["mthexapod_2.application.position1"].dropna(), "C4+-", label="M2
axs[2].set_ylabel("Hexapod Position\n[um]")

for ax in axs:
    ax.grid(":", alpha=0.5)
    ax.legend()

fig.suptitle(f"{test_execution} - M1M3/M2/Hexs/Elevation vs Time")
fig.tight_layout(h_pad=0.3)
fig.patch.set_facecolor('white')

fig.savefig(f"plots/{test_execution}_m1m3_m2_hexs_el_vs_time.png")
plt.show()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/pandas/core/indexes/base.py:3621, in Index.get
_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/pandas/_libs/index.pyx:136, in pandas._libs.in
dex.IndexEngine.get_loc()

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/pandas/_libs/index.pyx:163, in pandas._libs.in
dex.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.P
yObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.P
yObjectHashTable.get_item()

KeyError: 'mtm2.axialForce.applied0'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [89], in <cell line: 6>()
      3 axs[0].plot(df["elevation"].dropna(), "k", label="Mount Elevation")
      4 axs[0].set_ylabel("Mount El\n[deg]")
----> 6 axs[1].plot(df["mtm2.axialForce.applied0"].dropna(), "C1^-", label="ap
plied")
      7 axs[1].plot(df["mtm2.axialForce.lutGravity0"].dropna(), "C2v-", label=
"Gravity LUT")
      8 axs[1].set_ylabel("M2 Forces\n[--]")

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/pandas/core/frame.py:3505, in DataFrame.__geti
tem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/pandas/core/indexes/base.py:3623, in Index.get
_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'mtm2.axialForce.applied0'
```



```
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 18 of 100 elements
MTHexapod.electrical WARNING: tel_electrical DDS read queue is filling: 18 of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 18 of 100 elements
MTHexapod.application WARNING: tel_application DDS read queue is filling: 18 of 100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 17 of 100 elements
MTHexapod.actuators WARNING: tel_actuators DDS read queue is filling: 17 of 100 elements
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 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
```

In [ ]:

```
vandv.m1m3.plot_m1m3_and_elevation(df)
```

M1M3 Elevation Forces vs LUT

Check the M1M3 elevation forces match what we expect from the implemented LUT.

In [ ]:

```
elevation = await mtcs.rem.mtmount.tel_elevation.aget(timeout=10.)
print(elevation, "\n")
```

The look-up table for X has basically zeroes.
At least for now.
So it does not make much sense evaluating it now.

The lut for Y has non-zeroes and zeroes values.
in order to compare, we can drop the zeroes components.

For z, we are good. We can perform a direct comparison.

In [ ]:

```
lut_el_xforces = vandv.m1m3.lut_elevation_xforces(elevation.actualPosition)
lut_el_yforces = vandv.m1m3.lut_elevation_yforces(elevation.actualPosition)
lut_el_zforces = vandv.m1m3.lut_elevation_zforces(elevation.actualPosition)
```

In [ ]:

```
fel = await mtcs.rem.mtm1m3.evt_appliedElevationForces.aget(timeout=10.)
```

In [ ]:

```
fig, ax = plt.subplots(figsize=(15, 3))

ax.plot(fel.xForces, "C0^-", label="Applied")
ax.set_ylabel("Elevation xForces [??]")
ax.grid(":", alpha=0.2)
ax.legend()

fig.suptitle(f"{test_execution} - M1M3 Elevation Forces")
fig.tight_layout(h_pad=0.3)
fig.patch.set_facecolor('white')

fig.savefig(f"plots/{test_execution}_m1m3_fel_xForces.png")
plt.show()
```

In [ ]:

```
fig, axs = plt.subplots(figsize=(15, 6), nrows=2, sharex=True)

axs[0].plot(fel.yForces, "C0^-", label="Applied")
axs[0].plot(lut_el_yforces[lut_el_yforces != 0], "C1v-", label="LUT")
```

```
axs[0].set_ylabel("Elevation yForces [??]")
axs[0].grid(":", alpha=0.2)
axs[0].legend()

axs[1].plot(fel.yForces - lut_el_yforces[lut_el_yforces != 0], label="Applied -
axs[1].set_ylabel("Elefation yForces \n difference [??]")
axs[1].grid(":", alpha=0.2)

fig.suptitle(f"{test_execution} - M1M3 Elevation yForces")
fig.tight_layout(h_pad=0.3)
fig.patch.set_facecolor('white')

fig.savefig(f"plots/{test_execution}_m1m3_fel_yForces.png")
plt.show()
```

In [ ]:

```
fig, axs = plt.subplots(figsize=(15, 6), nrows=2, sharex=True)

axs[0].plot(fel.zForces, "C0^-", label="Applied")
axs[0].plot(lut_el_zforces, "C1v-", label="LUT")
axs[0].set_ylabel("Elevation zForces [??]")
axs[0].grid(":", alpha=0.2)
axs[0].legend()

axs[1].plot(fel.zForces - lut_el_zforces, label="Applied - LUT forces")
axs[1].set_ylabel("Elefation zForces \n difference [??]")
axs[1].grid(":", alpha=0.2)

fig.suptitle(f"{test_execution} - M1M3 Elevation zForces")
fig.tight_layout(h_pad=0.3)
fig.patch.set_facecolor('white')

fig.savefig(f"plots/{test_execution}_m1m3_fel_zForces.png")
plt.show()
```

The following plots are extracted from Bo's Notebooks.
The originals are in lsst-ts/ts_notebooks/bxin/ptg2m1m3.

M2 Elevation Forces vs LUT

Check the M2 elevation forces match what we expect from the implemented LUT.

In [ ]:

```
vandv.m2.plot_m2_actuators()
```

In [ ]:

```
axialForces = await mtcs.rem.mtm2.tel_axialForce.aget(timeout=2)
tangentForces = await mtcs.rem.mtm2.tel_tangentForce.aget(timeout=2)
```

In [ ]:

```
vandv.m2.plotM2Forces(axialForces, tangentForces)
```

CamHex Vs LUT

Check the camera hexapod LUT compensations match what we expect from the implemented LUT

In [ ]:

```
a = mtcs.rem.mthexapod_1.evt_compensationOffset.get()
elevCoeff, tCoeff = vandv.hexapod.coeffs_from_lut(index=1)
elev = a.elevation

await vandv.hexapod.print_hexapod_position(mtcs.rem.mthexapod_1)
await vandv.hexapod.print_predicted_compensation(elevCoeff, elev)
await vandv.hexapod.print_hexapod_uncompensation_values(mtcs.rem.mthexapod_1)
await vandv.hexapod.print_hexapod_compensation_values(mtcs.rem.mthexapod_1)
```

M2Hex vs LUT

In [ ]:

```
a = mtcs.rem.mthexapod_1.evt_compensationOffset.get()
elevCoeff, tCoeff = vandv.hexapod.coeffs_from_lut(index=2)
elev = a.elevation

await vandv.hexapod.print_hexapod_uncompensation_values(mtcs.rem.mthexapod_2)
await vandv.hexapod.print_predicted_compensation(elevCoeff, elev)
await vandv.hexapod.print_hexapod_position(mtcs.rem.mthexapod_2)
await vandv.hexapod.print_hexapod_compensation_values(mtcs.rem.mthexapod_2)
```

Close up

In [ ]:

```
# Put the telescope back to the original position
# Specially if running at TTS
target = mtcs.radec_from_azel(az=0, el=80)

await mtcs.slew_icrs(ra=target.ra, dec=target.dec, rot_type=RotType.Physical, r
await mtcs.stop_tracking()
```

In [ ]:

```
await mtcs.set_state(state=salobj.State.STANDBY, components=["mtaos"])
```

In [ ]:

```
await mtcs.lower_m1m3()
```

In [ ]:

```
await mtcs.set_state(state=salobj.State.STANDBY, components=["mtm1m3"])
```

In [ ]:

```
await mtcs.set_state(state=salobj.State.STANDBY, components=["mtm2"])
```

In [ ]:

```
await mtcs.set_state(state=salobj.State.STANDBY, components=["mthexapod_1"])
```

In [ ]:

```
await mtcs.set_state(state=salobj.State.STANDBY, components=["mthexapod_2"])
```

In [ ]:

```
if exec_info.loc == "summit":
    await mtcs.standby()

else:
    # Bring the system back to the original state
    await mtcs.set_state(
        state=salobj.State.ENABLED,
        components=[
            "mtm1m3",
            "mtm2",
            "mthexapod_1",
            "mthexapod_2",
            "mtaos",
        ],
        overrides={
            "mtm1m3": "Default"
        }
    )
```

Wrap Up

In [ ]:
```
await mtcs.lower_m1m3()
```

In [ ]:
```
await mtcs.set_state(state=salobj.State.STANDBY, components=["mtm1m3", "mtm2",
```

In [ ]:
```
await mtcs.set_state(state=salobj.State.ENABLED, components=["mtm2", "mthexapod
```