LVV-T2232 - M1M3 Integration with SAL

The objective of this test case is to verify the latest M1M3 commands, events, and telemetry defined by the latest version of the XML.
This test case will exercise the functionality of the M1M3 on the 3rd level of the Summit and meets the following criteria:

- Only requires the most current version of SAL
- Only requires the M1M3 surrogate to be loaded on the cell
- Requires the use of the DDS and the EFD

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 2022-06-15T14:44:29.479.
  Running in yagan07 at summit
```

LVV-T1996 (1.0) M1M3 DDS Startup Procedure

In [3]:
```
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 [4]:
```
logging.basicConfig(format="%(name)s:%(message)s", level=logging.DEBUG)
```

In [5]:
```
log = logging.getLogger("setup")
log.level = logging.DEBUG
```

In [6]:
```
domain = salobj.Domain()
```

In [7]:
```
mtcs = MTCS(domain=domain, log=log)
mtcs.set_rem_loglevel(40)
```

```
setup.MTCS DEBUG: mtmount: Adding all resources.
setup.MTCS DEBUG: mtptg: Adding all resources.
setup.MTCS DEBUG: mtaos: Adding all resources.
setup.MTCS DEBUG: mtm1m3: Adding all resources.
setup.MTCS DEBUG: mtm2: Adding all resources.
setup.MTCS DEBUG: mthexapod_1: Adding all resources.
setup.MTCS DEBUG: mthexapod_2: Adding all resources.
setup.MTCS DEBUG: mtrotator: Adding all resources.
setup.MTCS DEBUG: mtdome: Adding all resources.
setup.MTCS DEBUG: mtdometrajectory: Adding all resources.
MTHexapod INFO: Read historical data in 0.05 sec
MTHexapod INFO: Read historical data in 0.09 sec
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.hardpointActuatorData ERROR: tel_hardpointActuatorData DDS read queu
e 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.accelerometerData ERROR: tel_accelerometerData DDS read queue is ful
l (100 elements); data may be lost
MTM1M3.hardpointMonitorData ERROR: tel_hardpointMonitorData DDS read queue
 is full (100 elements); data may be lost
```

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

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

```
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.accelerometerData ERROR: tel_accelerometerData DDS read queue is ful
l (100 elements); data may be lost
```

In [9]:
```
index = 22321285  # Test Case + Test Execution

start_time = datetime.now()
script = salobj.Controller("Script", index=index)
```

```
Script INFO: Read historical data in 0.00 sec
MTM1M3.inclinometerData ERROR: tel_inclinometerData DDS read queue is full
 (100 elements); data may be lost
```

In [10]:
```
await mtcs.set_state(state=salobj.State.DISABLED, components=["mtm1m3"], overri
```

```
setup.MTCS DEBUG: [mtm1m3]::[<State.DISABLED: 1>]
setup.MTCS INFO: All components in <State.DISABLED: 1>.
```

In [11]:
```
await mtcs.set_state(state=salobj.State.ENABLED, components=["mtm1m3"])
```

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

In [12]:
```
script.log.info("LVV-T12232 - LVV-E1285 - Start")
```

```
Script INFO: LVV-T12232 - LVV-E1285 - Start
```

Telemetry Verification

Verify the MTM1M3_forceActuatorData telemetry data is being published to the EFD with the following parameters:

- primaryCylinderForce
- secondaryCylinderForce
- xForce
- yForce
- zForce
- fx
- fy
- fz
- mx
- my
- mz
- timestamp
- forceMagnitude

Check Chronograph - M1M3 Status.

In [ ]:
```
if exec_info.loc == "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}"
    )
```

In [ ]:
```
start = time.Time("2022-06-14T20:20", scale="utc", format="isot")
end = time.Time("2022-06-14T20:30", scale="utc", format="isot")
```

In [ ]:
```
df = await client.select_time_series(
    "lsst.sal.MTM1M3.forceActuatorData",
    fields="*",
    start=start.utc,
    end=end.utc,
)
```

In [ ]:
```
df
```

In [ ]:
```
df.iloc[0]
```

Verify the MTM1M3_forceActuatorPressure telemetry data is being published to the EFD with the following parameters:

- timestamps
- primaryCylinderPullPressures
- primaryCylinderPushPressures
- secondaryCylinderPullPressures
- secondaryCylinderPushPressures

In [ ]:
```
fap_df = await client.select_time_series(
    "lsst.sal.MTM1M3.forceActuatorPressure",
    fields="*",
    start=start.utc,
    end=end.utc,
)
```

In [ ]:
```
fap_df
```

Verify the MTM1M3_inclinometerDatatelemetry data is being published to the EFD with the following parameters:

- timestamp
- inclinometerAngle

In [ ]:
```
df_id = await client.select_time_series(
    "lsst.sal.MTM1M3.outerLoopData",
    fields="*",
    start=start.utc,
    end=end.utc,
)
```

In [ ]:
```
df_id
```

In [ ]:
```
script.log.info("LVV-T12232 - LVV-E1285 - END")
```

---

61 - Engineering Mode Test

While the M1M3 is enabled and in the PARKED state, send an `MTM1M3_command_enterEngineering` command.

`PARKED` means that it is not raised.

In [13]:
```
await mtcs.rem.mtm1m3.cmd_enterEngineering.set_start()
```

Out[13]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa17be32cd0>
```

In [17]:
```
from lsst.ts.idl.enums.MTM1M3 import DetailedState

m1m3_dstate = mtcs.rem.mtm1m3.evt_detailedState.get()

print(DetailedState.PARKEDENGINEERING == m1m3_dstate.detailedState)
```

```
True
```

With the system in the ParkedEngineering state and the M1M3 cell lights off, send an MTM1M3_command_turnLightsOn command.

In [22]:
```
await mtcs.rem.mtm1m3.cmd_turnLightsOn.set_start()
```

Out[22]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa17321f280>
```

In [23]:
```
print( mtcs.rem.mtm1m3.evt_cellLightStatus.get() )
```

```
private_revCode: c34d42d2, private_sndStamp: 1655305398.8532453, private_rcvSt
amp: 1655305398.853845, private_seqNum: 2, private_identity: MTM1M3, private_o
rigin: 11199, timestamp: 1655305398.852754, cellLightsCommandedOn: True, cellL
ightsOutputOn: True, cellLightsOn: False, priority: 0
```

In [24]:
```
print( mtcs.rem.mtm1m3.evt_cellLightWarning.get() )
```

```
private_revCode: 8bbed6ea, private_sndStamp: 1655305398.9532137, private_rcvSt
amp: 1655305398.9538088, private_seqNum: 2, private_identity: MTM1M3, private_
origin: 11199, timestamp: 1655305398.9527543, anyWarning: True, cellLightsOutp
utMismatch: False, cellLightsSensorMismatch: True, priority: 0
```

With the system in the ParkedEngineering state and the M1M3 cell lights off, send an MTM1M3_command_turnLightsOff command.

In [19]:
```
await mtcs.rem.mtm1m3.cmd_turnLightsOff.set_start()
```

Out[19]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1d4c554f0>
```

In [20]:
```
print( mtcs.rem.mtm1m3.evt_cellLightStatus.get() )
```

```
private_revCode: c34d42d2, private_sndStamp: 1655305190.4684613, private_rcvSt
amp: 1655305190.469076, private_seqNum: 1, private_identity: MTM1M3, private_o
rigin: 11199, timestamp: 1655305190.4682348, cellLightsCommandedOn: False, cel
lLightsOutputOn: False, cellLightsOn: False, priority: 0
```

In [21]:
```
print( mtcs.rem.mtm1m3.evt_cellLightWarning.get() )
```

```
private_revCode: 8bbed6ea, private_sndStamp: 1655305190.4684963, private_rcvSt
amp: 1655305190.4691052, private_seqNum: 1, private_identity: MTM1M3, private_
origin: 11199, timestamp: 1655305190.4682348, anyWarning: False, cellLightsOut
putMismatch: False, cellLightsSensorMismatch: False, priority: 0
```

With the system in the ParkedEngineering state and the telescope is not moving, send an MTM1M3_command_setAirSlewFlag command to open the booster valves.

In [26]:
```
print( mtcs.rem.mtm1m3.evt_forceActuatorState.get() )
```

```
private_revCode: 4f59e56a, private_sndStamp: 1655300866.5326025, private_rcvSt
amp: 1655304276.9380443, private_seqNum: 1, private_identity: MTM1M3, private_
origin: 11199, timestamp: 0.0, ilcState: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], slewFlag: False, staticForcesApplie
d: False, elevationForcesApplied: False, azimuthForcesApplied: False, thermalF
orcesApplied: False, offsetForcesApplied: False, accelerationForcesApplied: Fa
lse, velocityForcesApplied: False, activeOpticForcesApplied: False, aberration
ForcesApplied: False, balanceForcesApplied: False, supportPercentage: 0.0, pri
ority: 0
```

In [27]:
```
await mtcs.rem.mtm1m3.cmd_setAirSlewFlag.set_start()
```

Out[27]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1730fb0a0>
```

In [28]:
```
print( mtcs.rem.mtm1m3.evt_forceActuatorState.get() )
```

```
private_revCode: 4f59e56a, private_sndStamp: 1655300866.5326025, private_rcvSt
amp: 1655304276.9380443, private_seqNum: 1, private_identity: MTM1M3, private_
origin: 11199, timestamp: 0.0, ilcState: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], slewFlag: False, staticForcesApplie
d: False, elevationForcesApplied: False, azimuthForcesApplied: False, thermalF
orcesApplied: False, offsetForcesApplied: False, accelerationForcesApplied: Fa
lse, velocityForcesApplied: False, activeOpticForcesApplied: False, aberration
ForcesApplied: False, balanceForcesApplied: False, supportPercentage: 0.0, pri
ority: 0
```

In the ParkedEngineering state, send an MTM1M3_command_testHardpoint command.

In [30]:
```
await mtcs.rem.mtm1m3.cmd_testHardpoint.set_start(hardpointActuator=1)
```

```
---------------------------------------------------------------------------
AckError                                  Traceback (most recent call last)
Input In [30], in <cell line: 1>()
----> 1 await mtcs.rem.mtm1m3.cmd_testHardpoint.set_start(hardpointActuator=1)

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=1754694943, ack=
<SalRetCode.CMD_FAILED: -302>, error=0, result='Failed: The command TestHardpo
int is not valid in the ParkedEngineeringState.')
```

In the enabled state, send an MTM1M3_command_moveHardpointActuators command.

In [31]:
```
await mtcs.rem.mtm1m3.cmd_moveHardpointActuators.set_start(steps=[1000]*6)
```

Out[31]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa16257b280>
```

While the M1M3 is still in motion, send an MTM1M3_command_stopHardpointMotion command.

In [33]:
```
# The range of the actuators is up to 64k
await mtcs.rem.mtm1m3.cmd_moveHardpointActuators.set_start(steps=[10000]*6)
await asyncio.sleep(3)
await mtcs.rem.mtm1m3.cmd_stopHardpointMotion.set_start()
```

Out[33]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1d48c9820>
```

In [34]:
```
# Turned the air on again since I turned it off before by accident
await mtcs.rem.mtm1m3.cmd_turnAirOn.set_start()
```

Out[34]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa16af9ee80>
```

In the enabled state, send an MTM1M3_command_moveHardpointActuators command.

In [35]:
```
await mtcs.rem.mtm1m3.cmd_moveHardpointActuators.set_start(steps=[-10000]*6)
```

Out[35]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa162338d90>
```

Enabled Force Actuator Test

Verify the MTM1M3_logevent_enabledForceActuators event is published

In [44]:
```
await mtcs.rem.mtm1m3.cmd_disableForceActuator.set_start(actuatorId=225)
```

Out[44]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1d4c0fb20>
```

In [46]:
```
print( mtcs.rem.mtm1m3.evt_enabledForceActuators.get().forceActuatorEnabled[60]
```

```
False
```

In [47]:
```
await mtcs.rem.mtm1m3.cmd_enableForceActuator.set_start(actuatorId=225)
```

Out[47]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa17355e5e0>
```

In [48]:
```
print( mtcs.rem.mtm1m3.evt_enabledForceActuators.get().forceActuatorEnabled[60]
```

```
True
```

In the enabled state of the Engineering mode, disable at least two force actuators by sending an MTM1M3_command_disableForceActuator command one at a time.

Note: Any Force actuators can be disabled as long as they're not near neighbors or next to near neighbors. The actuators will also need to be disabled one at a time. For example, use force actuators 208 (index 44) and 417 (index 130).

In [54]:
```
await mtcs.rem.mtm1m3.cmd_enableAllForceActuators.set_start()
```

Out[54]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa17be32130>
```

In [55]:
```
await mtcs.rem.mtm1m3.cmd_disableForceActuator.set_start(actuatorId=208)
print( mtcs.rem.mtm1m3.evt_enabledForceActuators.get().forceActuatorEnabled[44]
```

```
True
```

In [56]:
```
print( mtcs.rem.mtm1m3.evt_enabledForceActuators.get().forceActuatorEnabled[44]
```

```
False
```

In [57]:
```
await mtcs.rem.mtm1m3.cmd_disableForceActuator.set_start(actuatorId=417)
await asyncio.sleep(0.1)  # This is only to deal with the async behavior
print( mtcs.rem.mtm1m3.evt_enabledForceActuators.get().forceActuatorEnabled[13(
```

```
False
```

In the enabled state of the Engineering mode, send an MTM1M3_command_enableAllForceActuators command.

In [58]:
```
await mtcs.rem.mtm1m3.cmd_enableAllForceActuators.set_start()
await asyncio.sleep(0.1)  # This is only to deal with the async behavior
print("Actuator 208 enabled? ", mtcs.rem.mtm1m3.evt_enabledForceActuators.get()
print("Actuator 417 enabled? ", mtcs.rem.mtm1m3.evt_enabledForceActuators.get()
print("All actuators enabled?", all(mtcs.rem.mtm1m3.evt_enabledForceActuators.g
```

```
Actuator 208 enabled?  True
Actuator 417 enabled?  True
All actuators enabled? True
```

Raise M1M3

With the M1M3 in the enabled state, send an MTM1M3_command_raiseM1M3 command.

In [59]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
```

Out[59]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa17345c2b0>
```

Before the M1M3 is fully raised, send an MTM1M3_command_abortRaiseM1M3 command.

In [60]:
```
await mtcs.rem.mtm1m3.cmd_lowerM1M3.set_start()
```

Out[60]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa162354130>
```

In [61]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
await asyncio.sleep(10)
await mtcs.rem.mtm1m3.cmd_abortRaiseM1M3.set_start()
```

Out[61]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa16254d2b0>
```

Send an MTM1M3_command_raiseM1M3 command and allow the M1M3 to be fully raised.

In [62]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
```

Out[62]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa162354850>
```

With the M1M3 fully raised and the system in the ActiveEngineering state, send an MTM1M3_command_lowerM1M3 command.

In [63]:
```
await mtcs.rem.mtm1m3.cmd_lowerM1M3.set_start()
```

Out[63]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1624dd790>
```

With the M1M3 lowered and the system still in the ParkedEngineering state, send an MTM1M3_command_raiseM1M3 command.

In [64]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
```

Out[64]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1625843d0>
```

While the M1M3 is still being raised, send an MTM1M3_command_disableHardpointChase command.

In [65]:
```
await mtcs.rem.mtm1m3.cmd_lowerM1M3.set_start()
```

Out[65]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa1734cd610>
```

In [66]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
```

Out[66]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa162590610>
```

In [67]:
```
await mtcs.rem.mtm1m3.cmd_disableHardpointChase.set_start()
```

Out[67]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa151560bb0>
```

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

Out[68]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa173562430>
```

With the M1M3 lowered and the system still in the ParkedEngineering state, send an MTM1M3_command_raiseM1M3 command.

- The command is accepted and the M1M3 starts to raise.
- The MTM1M3_logevent_detailedState event transitions from PARKED to RAISING

In [73]:
```
await mtcs.rem.mtm1m3.cmd_lowerM1M3.set_start()
```

```
---------------------------------------------------------------------------
AckError                                  Traceback (most recent call last)
Input In [73], in <cell line: 1>()
----> 1 await mtcs.rem.mtm1m3.cmd_lowerM1M3.set_start()

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=1138293853, ack=
<SalRetCode.CMD_FAILED: -302>, error=0, result='Failed: The command LowerM1M3
 is not valid in the LoweringEngineeringState.')
```

In [74]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
await asyncio.sleep(5)
print( mtcs.rem.mtm1m3.evt_detailedState.get() )
```

```
private_revCode: 0c019ad7, private_sndStamp: 1655318749.4652853, private_rcvSt
amp: 1655318749.4663076, private_seqNum: 27, private_identity: MTM1M3, private
_origin: 11199, timestamp: 1655318749.4652297, detailedState: 10, priority: 0
```

In [76]:
```
print( mtcs.rem.mtm1m3.evt_detailedState.get() == DetailedState.RAISINGENGINEEI
```

```
False
```

While the M1M3 is still being raised, send an MTM1M3_command_disableHardpointCorrections command. Wait 10seconds after sending the MTM1M3_command_disableHardpointCorrections command and send the MTM1M3_command_enableHardpointCorrections command while the M1M3 is still being raised.

In [70]:
```
await mtcs.rem.mtm1m3.cmd_lowerM1M3.set_start()
```

Out[70]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa162590550>
```

In [71]:
```
await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
await asyncio.sleep(10)
await mtcs.rem.mtm1m3.cmd_disableHardpointCorrections.set_start()
await asyncio.sleep(10)
await mtcs.rem.mtm1m3.cmd_enableHardpointCorrections()
```

```
---------------------------------------------------------------------------
AckError                                  Traceback (most recent call last)
Input In [71], in <cell line: 3>()
      1 await mtcs.rem.mtm1m3.cmd_raiseM1M3.set_start()
      2 await asyncio.sleep(10)
----> 3 await mtcs.rem.mtm1m3.cmd_disableHardpointCorrections.set_start()
      4 await asyncio.sleep(10)
      5 await mtcs.rem.mtm1m3.cmd_enableHardpointCorrections()

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=596616353, ack=<
SalRetCode.CMD_FAILED: -302>, error=0, result='Failed: The command DisableHard
pointCorrections is not valid in the RaisingEngineeringState.')
```

In the ACTIVEENGINEERING state, send an MTM1M3_command_updatePID command with the following parameters:

- pid: 2 (any number 1-6)

- timestep: 0.03s (default is 0.02)
- p: 0.02
- i: 3.0
- d: 0
- n: 0

In [77]:
```
await mtcs.rem.mtm1m3.cmd_updatePID.set_start(
    pid=2, timestep=0.03, p=0.02, i=3.0, d=0, n=0)
```

Out[77]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa16254d5e0>
```

In [80]:
```
print( mtcs.rem.mtm1m3.evt_forceActuatorState.get().balanceForcesApplied )
```

```
False
```

In [79]:
```
print( mtcs.rem.mtm1m3.evt_pidInfo.get() )
```

```
private_revCode: 358059b0, private_sndStamp: 1655319205.6906455, private_rcvSt
amp: 1655319205.6913736, private_seqNum: 36, private_identity: MTM1M3, private
_origin: 11199, timestamp: 1655319205.6906328, timestep: [0.02, 0.03, 0.02, 0.
02, 0.02, 0.02], p: [0.204309678403032, 0.02, 1.35400433655061, 1.714452756693
78, 1.78691443348212, 0.355712659308793], i: [3.63116506150057, 3.0, 3.8764697
2825983, 4.99165132468295, 6.25142910457108, 2.11474788021138], d: [0.0, 0.0,
0.0, 0.0, 0.0, 0.0], n: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0], calculatedA: [0.204309
678403032, 0.02, 1.35400433655061, 1.71445275669378, 1.78691443348212, 0.35571
2659308793], calculatedB: [-0.3359960555760526, 0.049999999999999996, -2.63047
92785360234, -3.329072486893901, -3.4488002848728185, -0.6691303610133583], ca
lculatedC: [0.1316863771730206, -0.06999999999999999, 1.2764749419854133, 1.61
4619730200121, 1.6618858513906984, 0.3134177017045654], calculatedD: [2.0, 2.
0, 2.0, 2.0, 2.0, 2.0], calculatedE: [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0], pri
ority: 0
```

In the ACTIVEENGINEERING state, send an MTM1M3_command_resetPID command for 2

In [82]:
```
await mtcs.rem.mtm1m3.cmd_resetPID.set_start(pid=2)
```

Out[82]:
```
<ddsutil.MTM1M3_ackcmd_91759c3a at 0x7fa173620610>
```

In [83]:
```
print( mtcs.rem.mtm1m3.evt_forceActuatorState.get().balanceForcesApplied )
```

```
False
```

In [84]:
```
print( mtcs.rem.mtm1m3.evt_pidInfo.get() )
```

private_revCode: 358059b0, private_sndStamp: 1655319577.7431917, private_rcvSt
amp: 1655319577.7435439, private_seqNum: 37, private_identity: MTM1M3, private
_origin: 11199, timestamp: 1655319577.7431788, timestep: [0.02, 0.02, 0.02, 0.
02, 0.02, 0.02], p: [0.204309678403032, 0.0190936349895327, 1.35400433655061,
1.71445275669378, 1.78691443348212, 0.355712659308793], i: [3.63116506150057,
1.90936349895328, 3.87646972825983, 4.99165132468295, 6.25142910457108, 2.1147
4788021138], d: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0], n: [0.0, 0.0, 0.0, 0.0, 0.0,
0.0], calculatedA: [0.204309678403032, 0.0190936349895327, 1.35400433655061,
1.71445275669378, 1.78691443348212, 0.355712659308793], calculatedB: [-0.33599
60555760526, 2.0122792321330962e-16, -2.6304792785360234, -3.329072486893901,
-3.4488002848728185, -0.6691303610133583], calculatedC: [0.1316863771730206, -
0.019093634989532902, 1.2764749419854133, 1.614619730200121, 1.661885851390698
4, 0.3134177017045654], calculatedD: [2.0, 2.0, 2.0, 2.0, 2.0, 2.0], calculate
dE: [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0], priority: 0

In the ACTIVEENGINEERING state, send an MTM1M3_command_runMirrorForceProfile command of (10,10,10,10,10,10)

In [87]:
```
await mtcs.rem.mtm1m3.cmd_runMirrorForceProfile.set_start(
    xForce=[0]*1000,
    yForce=10, zForce=10, xMoment=10, yMoment=10, zMoment=10)
```

```
---------------------------------------------------------------------------
TypeError                                 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/topics/write_topic.py:341, in W
riteTopic._basic_write(self)
    340 try:
--> 341     self._writer.write(self.data)
    342 except struct.error as e:

File dds.pyx:2711, in dds.DataWriter.write()

File dds.pyx:2712, in dds.DataWriter.write()

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/ddsutil.py:899, in _dds_type_support.<locals>.
_serialize(self, o)
    897     raise TypeError("Incorrect data type")
--> 899 result = o._serialize()
    900 # print("Result: ", str(result))

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/ddsutil.py:472, in _create_class.<locals>._ser
ialize(self)
    471 fmt = self._get_packing_fmt()
--> 472 args = self._get_packing_args()
    473 # print("Packing format: ", fmt)
    474 # print("Packing args: ", args)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/ddsutil.py:498, in _create_class.<locals>._get
_packing_args(self)
    497     val = getattr(self, m)
--> 498     _compute_packing_args(mem_type, args, val)
    499 return args

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/ddsutil.py:681, in _compute_packing_args(mem_t
ype, args, val)
    680 elif mem_type.tag ==  ARRAY_TAG:
--> 681     _compute_array_packing_args(mem_type, args, val)
    683 else:

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-3.
0.0/lib/python3.8/site-packages/ddsutil.py:746, in _compute_array_packing_args
(mem_type, args, val)
    745 for i in range(array_size):
--> 746     _compute_packing_args(array_type, args, val[i])

TypeError: 'int' object is not subscriptable

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

ValueError                                Traceback (most recent call last)
Input In [87], in <cell line: 1>()
----> 1 await mtcs.rem.mtm1m3.cmd_runMirrorForceProfile.set_start(
      2     xForce=10, yForce=10, zForce=10, xMoment=10, yMoment=10, zMoment=1
0)

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:473, i
n RemoteCommand.start(self, data, timeout, wait_done)
    471 try:
    472     self._in_start = True
--> 473     data_written = await super().write()
    474 finally:
    475     self._in_start = False

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/write_topic.py:324, in W
riteTopic.write(self)
    314 async def write(self) -> type_hints.BaseMsgType:
    315     """Write the current data and return a copy of the data written.
    316
    317     Returns
   (...)
    322         (as found in RemoteCommand).
    323     """
--> 324     self._basic_write()
    325     return copy.copy(self.data)

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/write_topic.py:347, in W
riteTopic._basic_write(self)
    343     raise ValueError(
    344         f"{self.name} write({self.data}) failed: one or more fields in
valid"
    345     ) from e
    346 except TypeError as e:
--> 347     raise ValueError(
    348         f"{self.name} write({self.data}) failed: "
    349         f"perhaps one or more array fields has been set to a scalar"
    350     ) from e
    351 except IndexError as e:
    352     raise ValueError(
    353         f"{self.name} write({self.data}) failed: "
    354         f"probably at least one array field is too short"
    355     ) from e

ValueError: runMirrorForceProfile write(private_revCode: e07b1a31, private_snd
Stamp: 1655320002.1880858, private_rcvStamp: 0.0, private_seqNum: 1400921734,
 private_identity: b1quint@nb-b1quint, private_origin: 28665, xForce: 10, yFor
ce: 10, zForce: 10, xMoment: 10, yMoment: 10, zMoment: 10) failed: perhaps one
or more array fields has been set to a scalar
```

Send an MTM1M3_command_abortProfile command.

In [ ]:
```
await mtcs.rem.mtm1m3.cmd_abortProfile.set_start()
```

Verify the MTM1M3_logevent_forceActuatorBumpTestStatus event is published to the EFD.

In [90]:
```
print( mtcs.rem.mtm1m3.evt_forceActuatorBumpTestStatus.get() )
t = time.Time(mtcs.rem.mtm1m3.evt_forceActuatorBumpTestStatus.get().private_snd
t.format = "isot"
print(t)
```

```
private_revCode: cc34db51, private_sndStamp: 1655318749.4649513, private_rcvSt
amp: 1655318749.4661322, private_seqNum: 7, private_identity: MTM1M3, private_
origin: 11199, timestamp: 0.0, actuatorId: -1, primaryTestTimestamps: [0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], pr
imaryTest: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1], secondaryTestTimestamps: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], secondaryTest: [1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1], priority: 0
2022-06-15T18:46:26.465
```

In [91]:
```
script.log.info("LVV-T12232 - LVV-E1285 - END")
```

```
Script INFO: LVV-T12232 - LVV-E1285 - END
```

In [ ]:
```
```