MTAOS handling of rejected commands This notebook is used for the level 3 integration tests from test plan LVV-P81 (https://jira.lsstcorp.org/secure/Tests.jspa#/testPlan/LVV-P81) as part of test cylce LVV-C176 (https://jira.lsstcorp.org/secure/Tests.jspa#/testCycle/LVV-C176). The following tests are currently run as part of this notebook: LVV-T2193 (https://jira.lsstcorp.org/secure/Tests.jspa#/testCase/LVV-T2193) Execution steps are separated by horizontal lines. Upon completion, save the notebook and its output as a pdf file to be attached to the test execution in JIRA. Last updated by E. Dennihy 20211020 Load all the needed libraries. Get the remotes ready Code in the notebook including section: "Check the summary state of each CSC". In [1]: %load\_ext autoreload %autoreload 2 In [2]: import rubin jupyter utils.lab.notebook as nb nb.utils.get\_node() /tmp/ipykernel\_5878/1665379685.py:2: DeprecationWarning: Call to deprecated function (or staticmethod) get\_node. (Please use lsst.rsp.get\_node()) nb.utils.get node() 'yagan03' Out[2]: In [3]: import os import sys import asyncio import logging import pandas as pd import numpy as np from matplotlib import pyplot as plt from lsst.ts import salobj from lsst.ts.observatory.control.maintel import MTCS, ComCam from lsst.ts.observatory.control import RotType lsst.ts.utils.tai INFO: Update leap second table lsst.ts.utils.tai INFO: current\_tai uses the system TAI clock In [4]: logging.basicConfig(format="%(name)s:%(message)s", level=logging.DEBUG) In [5]: log = logging.getLogger("setup") log.level = logging.DEBUG In [6]: domain = salobj.Domain() In [7]: mtcs = MTCS(domain=domain, log=log) mtcs.set rem loglevel(40) setup.MTCS DEBUG: mtmount: Adding all resources. setup.MTCS DEBUG: mtptg: Adding all resources. setup.MTCS DEBUG: mtaos: Adding all resources. setup.MTCS DEBUG: mtm1m3: Adding all resources. setup.MTCS DEBUG: mtm2: Adding all resources. setup.MTCS DEBUG: mthexapod\_1: Adding all resources. setup.MTCS DEBUG: mthexapod\_2: Adding all resources. setup.MTCS DEBUG: mtrotator: Adding all resources. setup.MTCS DEBUG: mtdome: Adding all resources. setup.MTCS DEBUG: mtdometrajectory: Adding all resources. MTHexapod INFO: Read historical data in 0.03 sec MTHexapod INFO: Read historical data in 0.04 sec In [8]: await mtcs.start task [None, None, None, None, None, None, None, None, None] Out[8]: Ready M1M3: Raise mirror, turn on FB, clear forces Need to have M1M3 LUT use its inclinometer. Ready M2: Turn on FB, clear forces Need to have M2 LUT use its inclinometer Get camera hexapod ready: check config; make sure LUT is on, and has valid inputs; make sure hex is at LUT position Get M2 hexapod ready: check config; make sure LUT is on, and has valid inputs; make sure hex is at LUT position Slew to the next target. Choose a target such that the rotator stays within a couple of degrees of its initial position. This is because the CCW is not running (MTmount in simulation mode). In [9]: await mtcs.rem.mtrotator.cmd\_move.set\_start(position = 0) <lsst.ts.salobj. ddsutil.MTRotator ackcmd f04f56ad at 0x7f05344036a0> Out[9]: In [14]: target = await mtcs.find\_target(el=60, az=120, mag\_limit=8) print(target) HD 224705 In [15]: await mtcs.slew object(target, rot type=RotType.PhysicalSky, rot=1.9) setup.MTCS INFO: Slewing to HD 224705: 00 00 02.3924 -40 35 28.331 setup.MTCS DEBUG: Setting rotator physical position to 1.9 deg. Rotator will track sky. setup.MTCS WARNING: Camera cable wrap following disabled in MTMount. setup.MTCS DEBUG: Stop tracking. setup.MTCS DEBUG: Wait 5.0s for rotator to settle down. setup.MTCS DEBUG: Workaround for rotator trajectory problem. Moving rotator to its current position: 0.00 setup.MTCS DEBUG: Wait for MTRotator in position event. setup.MTCS DEBUG: MTRotator in position: True. setup.MTCS DEBUG: MTRotator already in position. Handling potential race condition. setup.MTCS DEBUG: No new in position event in the last 5.0s. Assuming MTRotator in position. setup.MTCS DEBUG: Sending slew command. setup.MTCS DEBUG: Scheduling check coroutines setup.MTCS DEBUG: process as completed... setup.MTCS DEBUG: Monitor position started. setup.MTCS DEBUG: Waiting for Target event from mtmount. setup.MTCS DEBUG: mtmount: <State.ENABLED: 2> setup.MTCS DEBUG: mtptg: <State.ENABLED: 2> setup.MTCS DEBUG: mtaos: <State.ENABLED: 2> setup.MTCS DEBUG: mtm1m3: <State.ENABLED: 2> setup.MTCS DEBUG: mtm2: <State.ENABLED: 2> setup.MTCS DEBUG: mthexapod\_1: <State.ENABLED: 2> setup.MTCS DEBUG: mthexapod\_2: <State.ENABLED: 2> setup.MTCS DEBUG: mtrotator: <State.ENABLED: 2> setup.MTCS DEBUG: mtdome: <State.ENABLED: 2> setup.MTCS DEBUG: mtdometrajectory: <State.ENABLED: 2> setup.MTCS DEBUG: Wait for mtmount in position events. setup.MTCS DEBUG: Wait for dome in position event. setup.MTCS DEBUG: Wait for rotator in position event. setup.MTCS DEBUG: Wait for MTMount elevation in position event. setup.MTCS DEBUG: MTMount elevation in position: False. setup.MTCS DEBUG: Wait for MTMount azimuth in position event. setup.MTCS DEBUG: MTMount azimuth in position: False. setup.MTCS INFO: Got False setup.MTCS DEBUG: Rotator not in position setup.MTCS DEBUG: Mount target: private\_revCode: bdcb00ba, private\_sndStamp: 1643392371.647594, private\_rcvStamp: 1643392371.647884, private\_seqNum: 16200, priva te\_identity: MTMount, private\_origin: 52777, elevation: 59.91819521858605, elevationVelocity: 0.003140485509615884, azimuth: 119.5017551194959, azimuthVelocity: 0.0009631680331918805, taiTime: 1643392371.7061503, trackId: 4, tracksys: SIDEREAL, radesys: ICRS, priority: 0 setup.MTCS INFO: MTMount azimuth in position: True. setup.MTCS INFO: MTMount elevation in position: True. setup.MTCS DEBUG: [Tel]: Az = +119.275[ +0.2]; El = +060.273[ -0.4] [Rot]: +000.003[ +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: Got True setup.MTCS INFO: Rotator in position. clear all corrections using cmd\_resetCorrection In [34]: await mtcs.rem.mtaos.cmd resetCorrection.start() <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f04d1d73dc0> In [35]: await mtcs.rem.mtaos.cmd issueCorrection.start(timeout=60.) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f04d1da9b50> Add 1um of z7 to the system via OFC, issue the corrections. Compare the corrections sent vs forces and position changes applied. This is currently done in a separate notebook or on Chronograf. In [18]: wavefront\_errors = np.zeros(19) In [19]: wavefront errors[3]=1.0 In [20]: await mtcs.rem.mtaos.cmd\_addAberration.set\_start(wf=wavefront\_errors, timeout=10) <lsst.ts.salobj. ddsutil.MTAOS ackcmd fd03e870 at 0x7f04d1c49760> Out[20]: In [21]: await mtcs.rem.mtaos.cmd\_issueCorrection.start(timeout=60.) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f05345e86d0> Make plots using telemetry from each component to verify the changes in the DOFs. This step does not currently involve running any commands in this notebook. This step must be verified using a separate noteboook. Put M2 hexapod in DISABLED state (so that we can test command rejection). In [22]: await mtcs.set\_state(salobj.State.DISABLED, components=["mthexapod 2"]) setup.MTCS DEBUG: [mthexapod\_2]::[<State.ENABLED: 2>, <State.DISABLED: 1>] setup.MTCS INFO: All components in <State.DISABLED: 1>. Add 1um of z7 to the system via OFC. Expect m2 hexapod corrections are rejected, and all other corrections applied, then undone. In [23]: await mtcs.rem.mtaos.cmd addAberration.set start(wf=wavefront errors, timeout=10) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f04d1f176d0> In [24]: await mtcs.rem.mtaos.cmd issueCorrection.start(timeout=60.) Traceback (most recent call last) /tmp/ipykernel 5878/285352443.py in <module> ---> 1 await mtcs.rem.mtaos.cmd\_issueCorrection.start(timeout=60.) /opt/lsst/software/stack/conda/miniconda3-py38 4.9.2/envs/lsst-scipipe-0.7.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote command.py in start(self, data , timeout, wait\_done) 481 self.salinfo.\_running\_cmds[seq\_num] = cmd info 482 --> 483 return await cmd\_info.next\_ackcmd(timeout=timeout) /opt/lsst/software/stack/conda/miniconda3-py38 4.9.2/envs/lsst-scipipe-0.7.0/lib/python3.8/site-packages/lsst/ts/salobj/topics/remote command.py in next ackcmd(self , timeout) 199 ackcmd = await self. wait task if ackcmd.ack in self.failed ack codes: 200 raise base.AckError(msg="Command failed", ackcmd=ackcmd) --> 201 202 return ackcmd 203 except asyncio.TimeoutError: AckError: msg='Command failed', ackcmd=(ackcmd private seqNum=640532057, ack=<SalRetCode.CMD FAILED: -302>, error=1, result="Failed: Failed to apply correction to: ['m2hex']. ") Re-enable M2 hexapod Make it ready for AOS In [25]: await mtcs.set\_state(salobj.State.ENABLED, components=["mthexapod\_2"]) setup.MTCS DEBUG: [mthexapod\_2]::[<State.DISABLED: 1>, <State.ENABLED: 2>] setup.MTCS INFO: All components in <State.ENABLED: 2>. Re-issue the correction. In [26]: await mtcs.rem.mtaos.cmd\_addAberration.set\_start(wf=wavefront\_errors, timeout=10) <lsst.ts.salobj. ddsutil.MTAOS ackcmd fd03e870 at 0x7f04a6f1d3d0> await mtcs.rem.mtaos.cmd\_issueCorrection.start(timeout=60.) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f04d1a5b430> Reject the latest corrections. In [28]: await mtcs.rem.mtaos.cmd\_rejectCorrection.start() <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f049ecec4c0> In [29]: await mtcs.rem.mtaos.cmd\_issueCorrection.start(timeout=60.) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f04d2276d30> Add 2um of z7 via OFC In [30]: wavefront\_errors[3] = 2.0 In [31]: wavefront\_errors 0., 0.]) In [32]: await mtcs.rem.mtaos.cmd\_addAberration.set\_start(wf=wavefront\_errors, timeout=10) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f049ecece50> In [33]: await mtcs.rem.mtaos.cmd\_issueCorrection.start(timeout=60.) <lsst.ts.salobj.\_ddsutil.MTAOS\_ackcmd\_fd03e870 at 0x7f04d22d8520> Stop Tracking In [36]: await mtcs.stop\_tracking() setup.MTCS DEBUG: Stop tracking. Wrap up. Put each component to the following states: mtaos --> standby m1m3 --> standby m2 --> standby camera hex --> standby m2 hex --> standby In []: await mtcs.set state(salobj.State.STANDBY, components=["mtaos"]) In []: await mtcs.lower m1m3() In []: await mtcs.set state(salobj.State.STANDBY, components=["mtm1m3"]) In [ ]: await mtcs.set\_state(salobj.State.STANDBY, components=["mtm2"]) In [ ]: await mtcs.set\_state(salobj.State.STANDBY, components=["mthexapod\_1"]) In []: await mtcs.set\_state(salobj.State.STANDBY, components=["mthexapod\_2"]) In [ ]:

await mtcs.standby()