LVV-T2190-plots

March 11, 2022

1 LVV-T2190 Plots

This notebook is designed to query the EFD and make diagnostics plots for the execution of Test Case LVV-T2190.

When executing the tests, duplicate the notebook and rename it using the test execution name.

1.1 Test executed in YYYY-MM-DD

[1]:
```
import os
import sys
import logging

import numpy as np
import pandas as pd

from astropy.time import Time
from astropy import units as u
from datetime import timedelta, datetime

import lsst_efd_client

import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm

from pandas.plotting import register_matplotlib_converters
```

[2]:
```
%config Application.log_level="DEBUG"
```

[3]:
```
%matplotlib inline
```

1.2 Time window for the test execution.

Update the cells below to reflect the time when the test was executed.

This is the time window used to query the EFD.

[ ]:
```
test_execution = ""
time_start_tai = 1636480426.529292-60.
```

```
time_end_tai = 1636480581.6822271
```

[ ]:
```
test_execution = ""
time_start_tai = 1639495414.3939805-300
time_end_tai = 1639495414.3939805+300
```

[ ]:
```
test_execution = "LVV-E1723"
time_start_tai = 1647009551.261028
time_end_tai = 1647009594.87075
```

[4]:
```
test_execution = "LVV-E1725"
time_start_tai = 1647009929.0
time_end_tai = 1647010049.0
```

[5]:
```
start = Time(time_start_tai, format="unix_tai", scale="tai")
end = Time(time_end_tai, format="unix_tai", scale="tai")
```

1.3 Initialization

We start by setting up a logger for the notebook and configuring the EFD Client.

[6]:
```
log = logging.getLogger("LVV-T2190")
log.setLevel(logging.DEBUG)
```

[7]:
```
lsst_efd_client.EfdClient.list_efd_names()
```

[7]:
```
['summit_efd',
 'ncsa_teststand_efd',
 'ldf_stable_efd',
 'ldf_int_efd',
 'base_efd',
 'tucson_teststand_efd',
 'test_efd']
```

[8]:
```
efd_name = "summit_efd"
```

[9]:
```
client = lsst_efd_client.EfdClient(efd_name)
```

[10]:
```
start.strftime("%m/%d/%Y, %H:%M:%S"), end.strftime("%m/%d/%Y, %H:%M:%S")
```

[10]:
```
('03/11/2022, 14:45:37', '03/11/2022, 14:47:37')
```

[11]:
```
log.debug(f"{start.utc}, {end}")
```

```
<IPython.core.display.HTML object>
```

1.4 Displaying results

1.4.1 Display degrees of freedom

The degrees of freedom are the first step performed by the OFC in converting the wavefront errors into corrections.

It is composed of two parts, the "aggregated" and the "visit" degrees of freedom. The "aggregated" is the combination of all corrections computed so far whereas the "visit" contains only the degrees of freedom from the last correction.

These values are published as vectors of 50 elements each in the "degreeOfFreedom" event. As with the `annularZernikeCoeff` case above we need to query them individually and then build the vectors afterwards.

[12]:
```
degrees_of_freedom = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_degreeOfFreedom',
    [f"aggregatedDoF{i}" for i in range(50)] + [f"visitDoF{i}" for i in␣
 ↪range(50)],
    start.utc,
    end.utc
)
```

[13]:
```
degrees_of_freedom
```

[13]:

	aggregatedDoF0	aggregatedDoF1	\
2022-03-11 14:45:59.046000+00:00	0.000000	0.000000	
2022-03-11 14:46:07.608000+00:00	0.169121	0.054919	
2022-03-11 14:46:11.591000+00:00	0.216516	0.074500	
2022-03-11 14:46:12.783000+00:00	0.169121	0.054919	
2022-03-11 14:46:19.340000+00:00	0.216516	0.074500	
2022-03-11 14:46:21.011000+00:00	0.169121	0.054919	
2022-03-11 14:46:24.817000+00:00	0.385636	0.129420	

	aggregatedDoF2	aggregatedDoF3	\
2022-03-11 14:45:59.046000+00:00	0.000000	0.000000	
2022-03-11 14:46:07.608000+00:00	-71.852360	-11.856128	
2022-03-11 14:46:11.591000+00:00	-128.192546	-20.266218	
2022-03-11 14:46:12.783000+00:00	-71.852360	-11.856128	
2022-03-11 14:46:19.340000+00:00	-128.192546	-20.266218	
2022-03-11 14:46:21.011000+00:00	-71.852360	-11.856128	
2022-03-11 14:46:24.817000+00:00	-200.044907	-32.122346	

	aggregatedDoF4	aggregatedDoF5	\
2022-03-11 14:45:59.046000+00:00	0.000000	0.000000	
2022-03-11 14:46:07.608000+00:00	-0.008164	0.008265	
2022-03-11 14:46:11.591000+00:00	-0.012232	0.090462	
2022-03-11 14:46:12.783000+00:00	-0.008164	0.008265	
2022-03-11 14:46:19.340000+00:00	-0.012232	0.090462	

```
2022-03-11 14:46:21.011000+00:00       -0.008164        0.008265
2022-03-11 14:46:24.817000+00:00       -0.020396        0.098726

                                  aggregatedDoF6  aggregatedDoF7  \
2022-03-11 14:45:59.046000+00:00        0.000000        0.000000
2022-03-11 14:46:07.608000+00:00       -0.025150       37.968733
2022-03-11 14:46:11.591000+00:00       -0.039064       62.612988
2022-03-11 14:46:12.783000+00:00       -0.025150       37.968733
2022-03-11 14:46:19.340000+00:00       -0.039064       62.612988
2022-03-11 14:46:21.011000+00:00       -0.025150       37.968733
2022-03-11 14:46:24.817000+00:00       -0.064214      100.581721

                                  aggregatedDoF8  aggregatedDoF9  …  \
2022-03-11 14:45:59.046000+00:00        0.000000        0.000000  …
2022-03-11 14:46:07.608000+00:00       -0.419880       -0.002405  …
2022-03-11 14:46:11.591000+00:00        0.018287       -0.002876  …
2022-03-11 14:46:12.783000+00:00       -0.419880       -0.002405  …
2022-03-11 14:46:19.340000+00:00        0.018287       -0.002876  …
2022-03-11 14:46:21.011000+00:00       -0.419880       -0.002405  …
2022-03-11 14:46:24.817000+00:00       -0.401594       -0.005281  …

                                  visitDoF40  visitDoF41  visitDoF42  \
2022-03-11 14:45:59.046000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:07.608000+00:00   -0.000021    0.000042    0.000041
2022-03-11 14:46:11.591000+00:00   -0.000024   -0.000058    0.000039
2022-03-11 14:46:12.783000+00:00   -0.000021    0.000042    0.000041
2022-03-11 14:46:19.340000+00:00   -0.000024   -0.000058    0.000039
2022-03-11 14:46:21.011000+00:00   -0.000021    0.000042    0.000041
2022-03-11 14:46:24.817000+00:00   -0.000045   -0.000016    0.000080

                                  visitDoF43  visitDoF44    visitDoF45  \
2022-03-11 14:45:59.046000+00:00    0.000000    0.000000  0.000000e+00
2022-03-11 14:46:07.608000+00:00    0.000018    0.000219  7.525153e-08
2022-03-11 14:46:11.591000+00:00    0.000002    0.000153  2.873308e-08
2022-03-11 14:46:12.783000+00:00    0.000018    0.000219  7.525153e-08
2022-03-11 14:46:19.340000+00:00    0.000002    0.000153  2.873308e-08
2022-03-11 14:46:21.011000+00:00    0.000018    0.000219  7.525153e-08
2022-03-11 14:46:24.817000+00:00    0.000020    0.000372  1.039846e-07

                                    visitDoF46    visitDoF47  visitDoF48  \
2022-03-11 14:45:59.046000+00:00  0.000000e+00  0.000000e+00    0.000000
2022-03-11 14:46:07.608000+00:00  5.763935e-08 -1.898672e-06    0.000464
2022-03-11 14:46:11.591000+00:00  2.691570e-08 -3.683317e-08    0.000204
2022-03-11 14:46:12.783000+00:00  5.763935e-08 -1.898672e-06    0.000464
2022-03-11 14:46:19.340000+00:00  2.691570e-08 -3.683317e-08    0.000204
2022-03-11 14:46:21.011000+00:00  5.763935e-08 -1.898672e-06    0.000464
2022-03-11 14:46:24.817000+00:00  8.455505e-08 -1.935505e-06    0.000668
```

```
                                  visitDoF49
2022-03-11 14:45:59.046000+00:00    0.000000
2022-03-11 14:46:07.608000+00:00    0.001309
2022-03-11 14:46:11.591000+00:00    0.000577
2022-03-11 14:46:12.783000+00:00    0.001309
2022-03-11 14:46:19.340000+00:00    0.000577
2022-03-11 14:46:21.011000+00:00    0.001309
2022-03-11 14:46:24.817000+00:00    0.001886

[7 rows x 100 columns]
```

We need to unpack the data from the EFD query into vectors that are easier to plot.

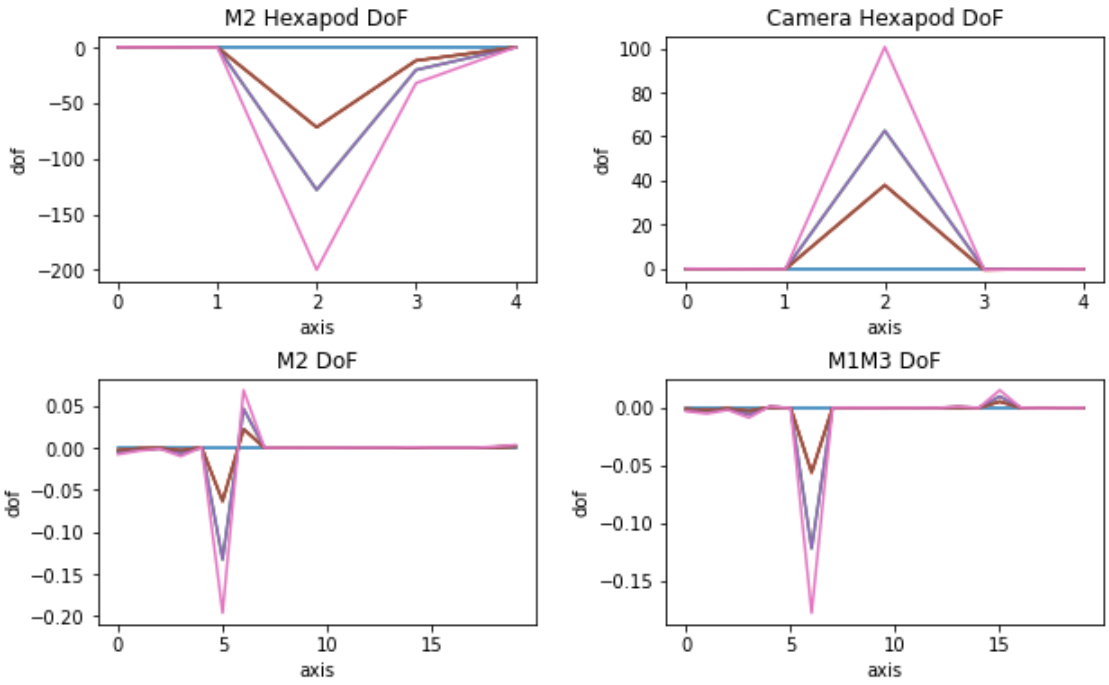
[14]:
```
aggregated_dof = np.array([degrees_of_freedom[f"aggregatedDoF{i}"] for i in␣
↪range(50)]).T
visit_dof = np.array([degrees_of_freedom[f"visitDoF{i}"] for i in range(50)]).T
```

[15]:
```
comp_dof_idx = dict(
            m2HexPos=dict(
                startIdx=0,
                idxLength=5,
                stateOname="M2Hexapod",
            ),
            camHexPos=dict(
                startIdx=5,
                idxLength=5,
                stateOname="cameraHexapod",
            ),
            M1M3Bend=dict(
                startIdx=10, idxLength=20, stateOname="M1M3Bending", rot_mat=1.0
            ),
            M2Bend=dict(startIdx=30, idxLength=20, stateOname="M2Bending",␣
↪rot_mat=1.0),
        )
```

And we finally plot them.

[16]:
```
fig, axes = plt.subplots(2,2, figsize=(10,6))

for i in range(len(aggregated_dof)):
    axes[0][0].plot(
        aggregated_dof[i][
            comp_dof_idx["m2HexPos"]["startIdx"]:
           ␣
↪comp_dof_idx["m2HexPos"]["startIdx"]+comp_dof_idx["m2HexPos"]["idxLength"]
        ]
```

```
    )
    axes[0][0].set_title("M2 Hexapod DoF")
    axes[0][0].set_xlabel("axis")
    axes[0][0].set_ylabel("dof")

    axes[0][1].plot(
        aggregated_dof[i][
            comp_dof_idx["camHexPos"]["startIdx"]:
          ␣
↪comp_dof_idx["camHexPos"]["startIdx"]+comp_dof_idx["camHexPos"]["idxLength"]
        ]
    )

    axes[0][1].set_title("Camera Hexapod DoF")
    axes[0][1].set_xlabel("axis")
    axes[0][1].set_ylabel("dof")

    axes[1][0].plot(
        aggregated_dof[i][
            comp_dof_idx["M2Bend"]["startIdx"]:
          ␣
↪comp_dof_idx["M2Bend"]["startIdx"]+comp_dof_idx["M2Bend"]["idxLength"]
        ]
    )
    axes[1][0].set_title("M2 DoF")
    axes[1][0].set_xlabel("axis")
    axes[1][0].set_ylabel("dof")


    axes[1][1].plot(
        aggregated_dof[i][
            comp_dof_idx["M1M3Bend"]["startIdx"]:
          ␣
↪comp_dof_idx["M1M3Bend"]["startIdx"]+comp_dof_idx["M1M3Bend"]["idxLength"]
        ]
    )
axes[1][1].set_title("M1M3 DoF")
axes[1][1].set_xlabel("axis")
axes[1][1].set_ylabel("dof")

fig.patch.set_facecolor('white')
plt.subplots_adjust(hspace=0.4, wspace=0.3)

fig.savefig("dof.png")
```



1.5 Step 8

1.5.1 Display Camera Hexapod Correction

[17]:
```
cam_hexapod_correction_computed_xyz = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_cameraHexapodCorrection',
    ["x", "y", "z"],
    start.utc,
    end.utc
)

cam_hexapod_correction_computed_uv = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_cameraHexapodCorrection',
    ["u", "v"],
    start.utc,
    end.utc
)
```

[18]:
```
cam_hexapod_correction_applied_xyz = await client.select_time_series(
    'lsst.sal.MTHexapod.logevent_uncompensatedPosition',
    ["x", "y", "z", "MTHexapodID"],
    start.utc,
    end.utc,
    index=1
)
```

```
cam_hexapod_correction_applied_uv = await client.select_time_series(
    'lsst.sal.MTHexapod.logevent_uncompensatedPosition',
    ["u", "v", "MTHexapodID"],
    start.utc,
    end.utc,
    index=1
)
```

[19]:
```
cam_hexapod_correction_command_xyz = await client.select_time_series(
    'lsst.sal.MTHexapod.command_move',
    ["x", "y", "z", "MTHexapodID"],
    start.utc,
    end.utc,
    index=1
)

cam_hexapod_correction_command_uv = await client.select_time_series(
    'lsst.sal.MTHexapod.command_move',
    ["u", "v", "MTHexapodID"],
    start.utc,
    end.utc,
    index=1
)
```

[20]:
```
cam_hexapod_correction_computed_xyz
```

[20]:

	x	y	z
2022-03-11 14:45:59.047000+00:00	0.000000	0.000000	0.000000
2022-03-11 14:46:07.633000+00:00	0.025150	37.968733	-0.008265
2022-03-11 14:46:11.592000+00:00	0.039064	62.612988	-0.090462
2022-03-11 14:46:19.341000+00:00	0.039064	62.612988	-0.090462
2022-03-11 14:46:21.012000+00:00	0.025150	37.968733	-0.008265
2022-03-11 14:46:24.819000+00:00	0.064214	100.581721	-0.098726

[21]:
```
cam_hexapod_correction_computed_uv
```

[21]:

	u	v
2022-03-11 14:45:59.047000+00:00	0.000000	0.000000e+00
2022-03-11 14:46:07.633000+00:00	0.000117	6.679176e-07
2022-03-11 14:46:11.592000+00:00	-0.000005	7.989896e-07
2022-03-11 14:46:19.341000+00:00	-0.000005	7.989896e-07
2022-03-11 14:46:21.012000+00:00	0.000117	6.679176e-07
2022-03-11 14:46:24.819000+00:00	0.000112	1.466907e-06

[22]:
```
cam_hexapod_correction_applied_xyz
```

[22]:

	x	y	z	MTHexapodID
2022-03-11 14:46:02.297000+00:00	0.000000	0.000000	0.000000	1
2022-03-11 14:46:08.350000+00:00	0.025150	37.968733	-0.008265	1
2022-03-11 14:46:12.351000+00:00	0.039064	62.612988	-0.090462	1
2022-03-11 14:46:12.902000+00:00	0.025150	37.968733	-0.008265	1
2022-03-11 14:46:19.904000+00:00	0.039064	62.612988	-0.090462	1
2022-03-11 14:46:22.105000+00:00	0.025150	37.968733	-0.008265	1
2022-03-11 14:46:25.456000+00:00	0.064214	100.581721	-0.098726	1

[23]:
```
cam_hexapod_correction_applied_uv
```

[23]:

	u	v	MTHexapodID
2022-03-11 14:46:02.297000+00:00	0.000000	0.000000e+00	1
2022-03-11 14:46:08.350000+00:00	0.000117	6.679176e-07	1
2022-03-11 14:46:12.351000+00:00	-0.000005	7.989896e-07	1
2022-03-11 14:46:12.902000+00:00	0.000117	6.679176e-07	1
2022-03-11 14:46:19.904000+00:00	-0.000005	7.989896e-07	1
2022-03-11 14:46:22.105000+00:00	0.000117	6.679176e-07	1
2022-03-11 14:46:25.456000+00:00	0.000112	1.466907e-06	1

[24]:
```
cam_hexapod_correction_command_xyz
```

[24]:

	x	y	z	MTHexapodID
2022-03-11 14:46:02.192000+00:00	0.000000	0.000000	0.000000	1
2022-03-11 14:46:08.249000+00:00	0.025150	37.968733	-0.008265	1
2022-03-11 14:46:12.277000+00:00	0.039064	62.612988	-0.090462	1
2022-03-11 14:46:12.784000+00:00	0.025150	37.968733	-0.008265	1
2022-03-11 14:46:21.993000+00:00	0.025150	37.968733	-0.008265	1

[25]:
```
cam_hexapod_correction_command_uv
```

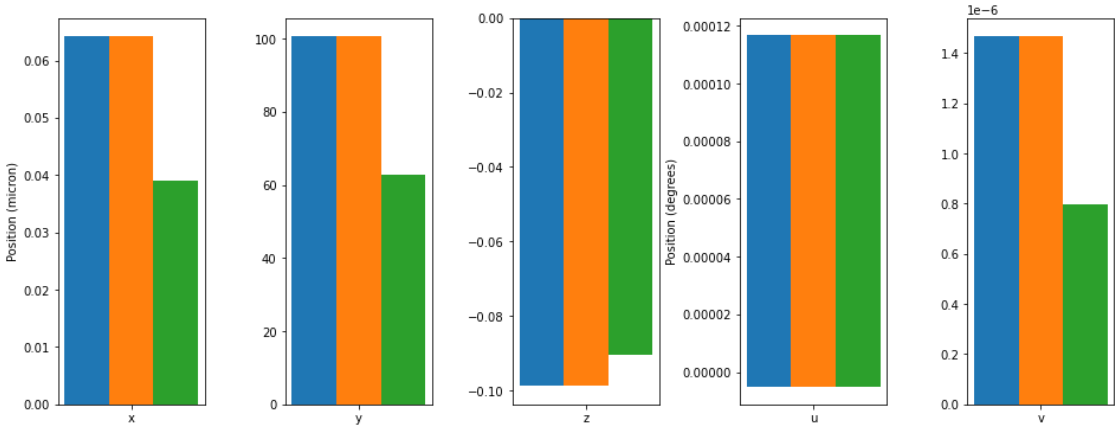
[25]:

	u	v	MTHexapodID
2022-03-11 14:46:02.192000+00:00	0.000000	0.000000e+00	1
2022-03-11 14:46:08.249000+00:00	0.000117	6.679176e-07	1
2022-03-11 14:46:12.277000+00:00	-0.000005	7.989896e-07	1
2022-03-11 14:46:12.784000+00:00	0.000117	6.679176e-07	1
2022-03-11 14:46:21.993000+00:00	0.000117	6.679176e-07	1

[26]:
```
fig = plt.figure(figsize=(16,6))

axis = []
# label = "x"
for panel, label in enumerate("xyz"):

    ax = plt.subplot(1,5,panel+1)

    x = [0.]
```

```
    ax.bar(
        [-0.5],
        cam_hexapod_correction_computed_xyz[label],
        width=0.5
    )
    ax.bar(
        [0.],
        cam_hexapod_correction_applied_xyz[label],
        width=0.5
    )

    ax.bar(
        [0.5],
        cam_hexapod_correction_command_xyz[label],
        width=0.5
    )

    ax.set_xticks([0])
    ax.set_xticklabels([label])
    axis.append(ax)

axis[0].set_ylabel("Position (micron)")

for panel, label in enumerate("uv"):

    ax = plt.subplot(1,5,panel+4)

    x = [0.]

    ax.bar(
        [-0.5],
        cam_hexapod_correction_computed_uv[label],
        width=0.5
    )
    ax.bar(
        [0.],
        cam_hexapod_correction_applied_uv[label],
        width=0.5
    )

    ax.bar(
        [0.5],
        cam_hexapod_correction_command_uv[label],
        width=0.5
    )
```

```
    ax.set_xticks([0])
    ax.set_xticklabels([label])
    axis.append(ax)

axis[3].set_ylabel("Position (degrees)")


plt.subplots_adjust(hspace=0.3, wspace=0.55)
fig.patch.set_facecolor('white')

fig.savefig(f"camera_hexapod_{test_execution}.png")
```



1.5.2 Display M2 Hexapod Correction

[27]:
```
m2_hexapod_correction_computed_xyz = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_m2HexapodCorrection',
    ["x", "y", "z"],
    start.utc,
    end.utc
)

m2_hexapod_correction_computed_uv = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_m2HexapodCorrection',
    ["u", "v"],
    start.utc,
    end.utc
)
```

[28]:
```
m2_hexapod_correction_applied_xyz = await client.select_time_series(
    'lsst.sal.MTHexapod.logevent_uncompensatedPosition',
    ["x", "y", "z", "MTHexapodID"],
```

```
    start.utc,
    end.utc,
    index=2
)

m2_hexapod_correction_applied_uv = await client.select_time_series(
    'lsst.sal.MTHexapod.logevent_uncompensatedPosition',
    ["u", "v", "MTHexapodID"],
    start.utc,
    end.utc,
    index=2
)
```

[29]:
```
m2_hexapod_correction_command_xyz = await client.select_time_series(
    'lsst.sal.MTHexapod.command_move',
    ["x", "y", "z", "MTHexapodID"],
    start.utc,
    end.utc,
    index=2
)

m2_hexapod_correction_command_uv = await client.select_time_series(
    'lsst.sal.MTHexapod.command_move',
    ["u", "v", "MTHexapodID"],
    start.utc,
    end.utc,
    index=2
)
```

[30]:
```
m2_hexapod_correction_command_xyz
```

[30]:

	x	y	z	MTHexapodID
2022-03-11 14:46:19.813000+00:00	-0.07450	-128.192546	-0.216516	2
2022-03-11 14:46:25.352000+00:00	-0.12942	-200.044907	-0.385636	2

[31]:
```
m2_hexapod_correction_computed_xyz
```

[31]:

	x	y	z
2022-03-11 14:45:59.046000+00:00	0.000000	0.000000	0.000000
2022-03-11 14:46:07.632000+00:00	-0.054919	-71.852360	-0.169121
2022-03-11 14:46:11.591000+00:00	-0.074500	-128.192546	-0.216516
2022-03-11 14:46:19.341000+00:00	-0.074500	-128.192546	-0.216516
2022-03-11 14:46:21.012000+00:00	-0.054919	-71.852360	-0.169121
2022-03-11 14:46:24.818000+00:00	-0.129420	-200.044907	-0.385636

[32]:
```
m2_hexapod_correction_applied_xyz
```

[32]:

	x	y	z	MTHexapodID
2022-03-11 14:46:02.194000+00:00	0.000000	0.000000	0.000000	2
2022-03-11 14:46:08.252000+00:00	-0.054919	-71.852360	-0.169121	2
2022-03-11 14:46:19.815000+00:00	-0.074500	-128.192546	-0.216516	2
2022-03-11 14:46:21.994000+00:00	-0.054919	-71.852360	-0.169121	2
2022-03-11 14:46:25.353000+00:00	-0.129420	-200.044907	-0.385636	2

[33]:
```
m2_hexapod_correction_command_uv
```

[33]:

	u	v	MTHexapodID
2022-03-11 14:46:19.813000+00:00	0.005630	0.000003	2
2022-03-11 14:46:25.352000+00:00	0.008923	0.000006	2

[34]:
```
m2_hexapod_correction_computed_uv
```

[34]:

	u	v
2022-03-11 14:45:59.046000+00:00	0.000000	0.000000
2022-03-11 14:46:07.632000+00:00	0.003293	0.000002
2022-03-11 14:46:11.591000+00:00	0.005630	0.000003
2022-03-11 14:46:19.341000+00:00	0.005630	0.000003
2022-03-11 14:46:21.012000+00:00	0.003293	0.000002
2022-03-11 14:46:24.818000+00:00	0.008923	0.000006

[35]:
```
m2_hexapod_correction_applied_uv
```

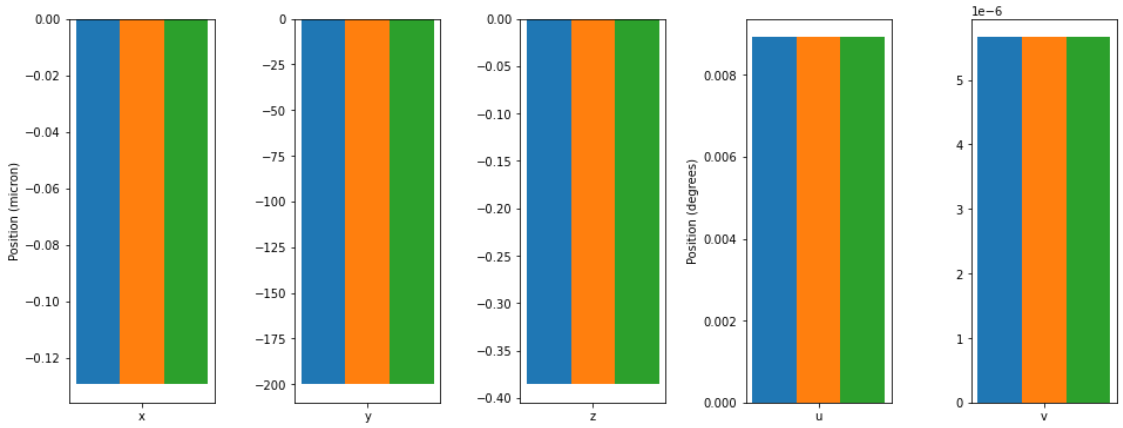
[35]:

	u	v	MTHexapodID
2022-03-11 14:46:02.194000+00:00	0.000000	0.000000	2
2022-03-11 14:46:08.252000+00:00	0.003293	0.000002	2
2022-03-11 14:46:19.815000+00:00	0.005630	0.000003	2
2022-03-11 14:46:21.994000+00:00	0.003293	0.000002	2
2022-03-11 14:46:25.353000+00:00	0.008923	0.000006	2

[36]:
```
fig = plt.figure(figsize=(16,6))

axis = []
# label = "x"
for panel, label in enumerate("xyz"):

    ax = plt.subplot(1,5,panel+1)

    x = [0.]

    ax.bar(
        [-0.5],
        m2_hexapod_correction_computed_xyz[label],
        width=0.5
    )
```

```
    ax.bar(
        [0.],
        m2_hexapod_correction_applied_xyz[label],
        width=0.5
    )

    ax.bar(
        [0.5],
        m2_hexapod_correction_command_xyz[label],
        width=0.5
    )

    ax.set_xticks([0])
    ax.set_xticklabels([label])
    axis.append(ax)

axis[0].set_ylabel("Position (micron)")

for panel, label in enumerate("uv"):

    ax = plt.subplot(1,5,panel+4)

    x = [0.]

    ax.bar(
        [-0.5],
        m2_hexapod_correction_computed_uv[label],
        width=0.5
    )
    ax.bar(
        [0.],
        m2_hexapod_correction_applied_uv[label],
        width=0.5
    )

    ax.bar(
        [0.5],
        m2_hexapod_correction_command_uv[label],
        width=0.5
    )

    ax.set_xticks([0])
    ax.set_xticklabels([label])
    axis.append(ax)

axis[3].set_ylabel("Position (degrees)")
```

```
plt.subplots_adjust(hspace=0.3, wspace=0.55)
fig.patch.set_facecolor('white')

fig.savefig(f"m2_hexapod_{test_execution}.png")
```



1.5.3 Display M2 Correction

[37]:
```
m2_correction = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_m2Correction',
    [f"zForces{i}" for i in range(72)],
    start.utc,
    end.utc
)
```

[38]:
```
m2_correction
```

[38]:

	zForces0	zForces1	zForces2	zForces3	\
2022-03-11 14:45:59.048000+00:00	0.000000	0.000000	0.000000	0.000000	
2022-03-11 14:46:07.634000+00:00	-0.759188	-0.741219	-0.704368	-0.630182	
2022-03-11 14:46:11.594000+00:00	-1.484869	-1.450552	-1.375108	-1.229742	
2022-03-11 14:46:19.343000+00:00	-1.484869	-1.450552	-1.375108	-1.229742	
2022-03-11 14:46:21.013000+00:00	-0.759188	-0.741219	-0.704368	-0.630182	
2022-03-11 14:46:24.820000+00:00	-2.244057	-2.191770	-2.079476	-1.859924	

	zForces4	zForces5	zForces6	zForces7	\
2022-03-11 14:45:59.048000+00:00	0.000000	0.000000	0.000000	0.000000	
2022-03-11 14:46:07.634000+00:00	-0.518267	-0.390221	-0.240212	-0.079020	
2022-03-11 14:46:11.594000+00:00	-1.015420	-0.767578	-0.476014	-0.158691	
2022-03-11 14:46:19.343000+00:00	-1.015420	-0.767578	-0.476014	-0.158691	
2022-03-11 14:46:21.013000+00:00	-0.518267	-0.390221	-0.240212	-0.079020	
2022-03-11 14:46:24.820000+00:00	-1.533687	-1.157800	-0.716226	-0.237711	

```
                                zForces8  zForces9  …  zForces62  \
2022-03-11 14:45:59.048000+00:00  0.000000  0.000000  …   0.000000
2022-03-11 14:46:07.634000+00:00  0.083761  0.241742  …  -0.776518
2022-03-11 14:46:11.594000+00:00  0.164762  0.477563  …  -1.799385
2022-03-11 14:46:19.343000+00:00  0.164762  0.477563  …  -1.799385
2022-03-11 14:46:21.013000+00:00  0.083761  0.241742  …  -0.776518
2022-03-11 14:46:24.820000+00:00  0.248523  0.719305  …  -2.575903

                                zForces63  zForces64  zForces65  zForces66  \
2022-03-11 14:45:59.048000+00:00   0.000000   0.000000   0.000000   0.000000
2022-03-11 14:46:07.634000+00:00  -0.843476  -0.782143  -0.612518  -0.391572
2022-03-11 14:46:11.594000+00:00  -1.939438  -1.806488  -1.430485  -0.913980
2022-03-11 14:46:19.343000+00:00  -1.939438  -1.806488  -1.430485  -0.913980
2022-03-11 14:46:21.013000+00:00  -0.843476  -0.782143  -0.612518  -0.391572
2022-03-11 14:46:24.820000+00:00  -2.782914  -2.588631  -2.043003  -1.305551

                                zForces67  zForces68  zForces69  zForces70  \
2022-03-11 14:45:59.048000+00:00   0.000000   0.000000   0.000000   0.000000
2022-03-11 14:46:07.634000+00:00  -0.138743   0.136610   0.387046   0.604225
2022-03-11 14:46:11.594000+00:00  -0.318616   0.318486   0.910369   1.421071
2022-03-11 14:46:19.343000+00:00  -0.318616   0.318486   0.910369   1.421071
2022-03-11 14:46:21.013000+00:00  -0.138743   0.136610   0.387046   0.604225
2022-03-11 14:46:24.820000+00:00  -0.457359   0.455096   1.297415   2.025296

                                zForces71
2022-03-11 14:45:59.048000+00:00   0.000000
2022-03-11 14:46:07.634000+00:00   0.773663
2022-03-11 14:46:11.594000+00:00   1.795960
2022-03-11 14:46:19.343000+00:00   1.795960
2022-03-11 14:46:21.013000+00:00   0.773663
2022-03-11 14:46:24.820000+00:00   2.569623

[6 rows x 72 columns]
```

[39]:
```
m2_correction_applied = await client.select_time_series(
    'lsst.sal.MTM2.command_applyForces',
    [f"axial{i}" for i in range(72)],
    start.utc,
    end.utc
)
```

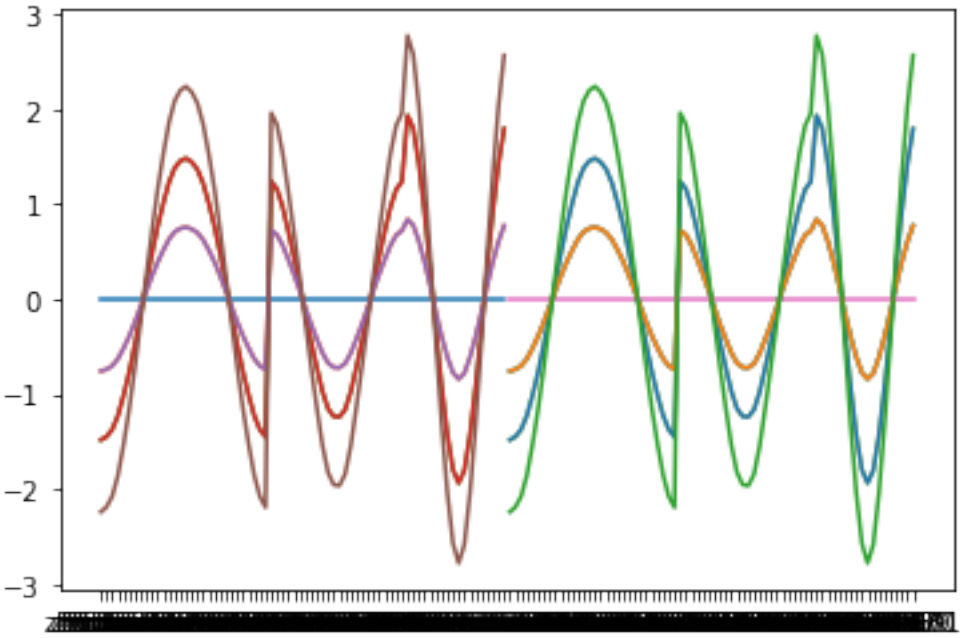
[40]:
```
m2_correction_applied
```

[40]:
```
                                  axial0    axial1    axial2    axial3  \
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  0.000000  0.000000
2022-03-11 14:46:08.250000+00:00 -0.759188 -0.741219 -0.704368 -0.630182
```

```
2022-03-11 14:46:12.277000+00:00 -1.484869 -1.450552 -1.375108 -1.229742
2022-03-11 14:46:12.905000+00:00 -0.759188 -0.741219 -0.704368 -0.630182
2022-03-11 14:46:19.814000+00:00 -1.484869 -1.450552 -1.375108 -1.229742
2022-03-11 14:46:21.993000+00:00 -0.759188 -0.741219 -0.704368 -0.630182
2022-03-11 14:46:25.353000+00:00 -2.244057 -2.191770 -2.079476 -1.859924

                                    axial4    axial5    axial6    axial7  \
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  0.000000  0.000000
2022-03-11 14:46:08.250000+00:00 -0.518267 -0.390221 -0.240212 -0.079020
2022-03-11 14:46:12.277000+00:00 -1.015420 -0.767578 -0.476014 -0.158691
2022-03-11 14:46:12.905000+00:00 -0.518267 -0.390221 -0.240212 -0.079020
2022-03-11 14:46:19.814000+00:00 -1.015420 -0.767578 -0.476014 -0.158691
2022-03-11 14:46:21.993000+00:00 -0.518267 -0.390221 -0.240212 -0.079020
2022-03-11 14:46:25.353000+00:00 -1.533687 -1.157800 -0.716226 -0.237711

                                    axial8    axial9  …   axial62   axial63  \
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  …  0.000000  0.000000
2022-03-11 14:46:08.250000+00:00  0.083761  0.241742  … -0.776518 -0.843476
2022-03-11 14:46:12.277000+00:00  0.164762  0.477563  … -1.799385 -1.939438
2022-03-11 14:46:12.905000+00:00  0.083761  0.241742  … -0.776518 -0.843476
2022-03-11 14:46:19.814000+00:00  0.164762  0.477563  … -1.799385 -1.939438
2022-03-11 14:46:21.993000+00:00  0.083761  0.241742  … -0.776518 -0.843476
2022-03-11 14:46:25.353000+00:00  0.248523  0.719305  … -2.575903 -2.782914

                                   axial64   axial65   axial66   axial67  \
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  0.000000  0.000000
2022-03-11 14:46:08.250000+00:00 -0.782143 -0.612518 -0.391572 -0.138743
2022-03-11 14:46:12.277000+00:00 -1.806488 -1.430485 -0.913980 -0.318616
2022-03-11 14:46:12.905000+00:00 -0.782143 -0.612518 -0.391572 -0.138743
2022-03-11 14:46:19.814000+00:00 -1.806488 -1.430485 -0.913980 -0.318616
2022-03-11 14:46:21.993000+00:00 -0.782143 -0.612518 -0.391572 -0.138743
2022-03-11 14:46:25.353000+00:00 -2.588631 -2.043003 -1.305551 -0.457359

                                   axial68   axial69   axial70   axial71
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  0.000000  0.000000
2022-03-11 14:46:08.250000+00:00  0.136610  0.387046  0.604225  0.773663
2022-03-11 14:46:12.277000+00:00  0.318486  0.910369  1.421071  1.795960
2022-03-11 14:46:12.905000+00:00  0.136610  0.387046  0.604225  0.773663
2022-03-11 14:46:19.814000+00:00  0.318486  0.910369  1.421071  1.795960
2022-03-11 14:46:21.993000+00:00  0.136610  0.387046  0.604225  0.773663
2022-03-11 14:46:25.353000+00:00  0.455096  1.297415  2.025296  2.569623

[7 rows x 72 columns]
```

[41]:
```
plt.plot(m2_correction.T)
plt.plot(m2_correction_applied.T)
```

[41]:
```
[<matplotlib.lines.Line2D at 0x7fc21ed36e50>,
 <matplotlib.lines.Line2D at 0x7fc21ed36e80>,
 <matplotlib.lines.Line2D at 0x7fc21ed36f70>,
 <matplotlib.lines.Line2D at 0x7fc21ecce0d0>,
 <matplotlib.lines.Line2D at 0x7fc21ed26df0>,
 <matplotlib.lines.Line2D at 0x7fc21ecce1f0>,
 <matplotlib.lines.Line2D at 0x7fc21ecce400>]
```



[42]:
```
aa = np.loadtxt('%s/notebooks/M2_FEA/data/M2_1um_72_force.txt'%(os.
  ↪environ["HOME"]))
# to have +x going to right, and +y going up, we need to transpose and reverse␣
  ↪x and y
m2_xact = -aa[:,2]
m2_yact = -aa[:,1]
```

[43]:
```
m2_yact
```

[43]:
```
array([-1.333500e-16, -3.328670e-01, -6.511849e-01, -9.410446e-01,
       -1.189774e+00, -1.386507e+00, -1.522641e+00, -1.592229e+00,
       -1.592229e+00, -1.522641e+00, -1.386507e+00, -1.189774e+00,
       -9.410446e-01, -6.511849e-01, -3.328670e-01,  0.000000e+00,
        3.328670e-01,  6.511849e-01,  9.410446e-01,  1.189774e+00,
        1.386507e+00,  1.522641e+00,  1.592229e+00,  1.592229e+00,
        1.522641e+00,  1.386507e+00,  1.189774e+00,  9.410446e-01,
        6.511849e-01,  3.328670e-01, -1.675856e-01, -4.913528e-01,
```

```
       -7.816342e-01, -1.018647e+00, -1.186244e+00, -1.272997e+00,
       -1.273000e+00, -1.186249e+00, -1.018657e+00, -7.816469e-01,
       -4.913655e-01, -1.676011e-01,  1.675856e-01,  4.913528e-01,
        7.816342e-01,  1.018647e+00,  1.186244e+00,  1.272997e+00,
        1.273000e+00,  1.186249e+00,  1.018657e+00,  7.816469e-01,
        4.913655e-01,  1.676011e-01,  3.893820e-16, -3.427044e-01,
       -6.440729e-01, -8.677580e-01, -9.867773e-01, -9.867773e-01,
       -8.677580e-01, -6.440729e-01, -3.427044e-01,  0.000000e+00,
        3.427044e-01,  6.440729e-01,  8.677580e-01,  9.867773e-01,
        9.867773e-01,  8.677580e-01,  6.440729e-01,  3.427044e-01])
```

[44]:
```
aa = np.array(m2_correction.T)
```

[45]:
```
aa.shape
```

[45]:
```
(72, 6)
```

[46]:
```
m2_correction.T
```

[46]:

	2022-03-11 14:45:59.048000+00:00	2022-03-11 14:46:07.634000+00:00	\
zForces0	0.0	-0.759188	
zForces1	0.0	-0.741219	
zForces2	0.0	-0.704368	
zForces3	0.0	-0.630182	
zForces4	0.0	-0.518267	
…	…	…	
zForces67	0.0	-0.138743	
zForces68	0.0	0.136610	
zForces69	0.0	0.387046	
zForces70	0.0	0.604225	
zForces71	0.0	0.773663	

	2022-03-11 14:46:11.594000+00:00	2022-03-11 14:46:19.343000+00:00	\
zForces0	-1.484869	-1.484869	
zForces1	-1.450552	-1.450552	
zForces2	-1.375108	-1.375108	
zForces3	-1.229742	-1.229742	
zForces4	-1.015420	-1.015420	
…	…	…	
zForces67	-0.318616	-0.318616	
zForces68	0.318486	0.318486	
zForces69	0.910369	0.910369	
zForces70	1.421071	1.421071	
zForces71	1.795960	1.795960	

	2022-03-11 14:46:21.013000+00:00	2022-03-11 14:46:24.820000+00:00
zForces0	-0.759188	-2.244057

```
zForces1                          -0.741219                          -2.191770
zForces2                          -0.704368                          -2.079476
zForces3                          -0.630182                          -1.859924
zForces4                          -0.518267                          -1.533687
…                                       …                                  …
zForces67                         -0.138743                          -0.457359
zForces68                          0.136610                           0.455096
zForces69                          0.387046                           1.297415
zForces70                          0.604225                           2.025296
zForces71                          0.773663                           2.569623

[72 rows x 6 columns]
```

[47]:
```
m2_correction_applied.T
```

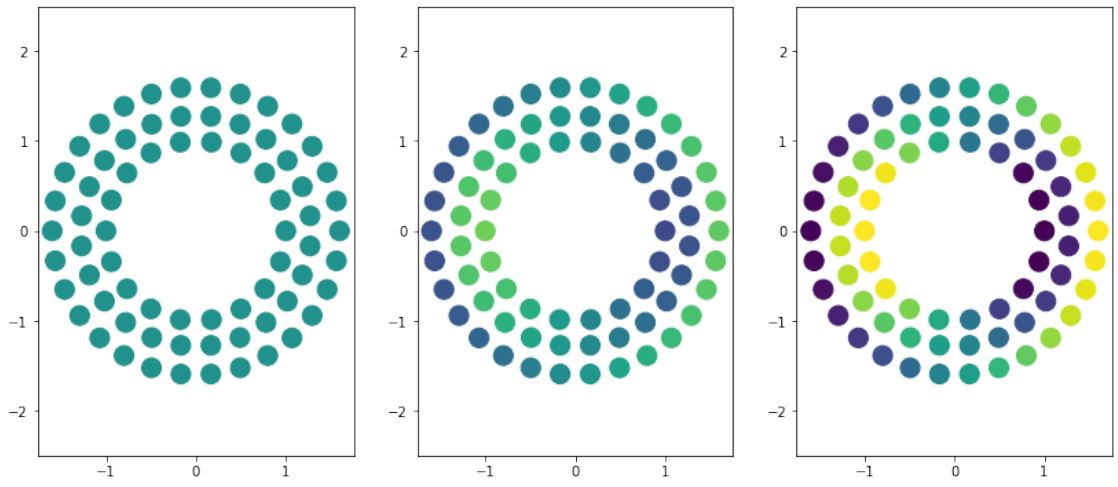
[47]:
```
          2022-03-11 14:46:02.192000+00:00  2022-03-11 14:46:08.250000+00:00  \
axial0                                 0.0                         -0.759188
axial1                                 0.0                         -0.741219
axial2                                 0.0                         -0.704368
axial3                                 0.0                         -0.630182
axial4                                 0.0                         -0.518267
…                                      …                                 …
axial67                                0.0                         -0.138743
axial68                                0.0                          0.136610
axial69                                0.0                          0.387046
axial70                                0.0                          0.604225
axial71                                0.0                          0.773663

          2022-03-11 14:46:12.277000+00:00  2022-03-11 14:46:12.905000+00:00  \
axial0                           -1.484869                         -0.759188
axial1                           -1.450552                         -0.741219
axial2                           -1.375108                         -0.704368
axial3                           -1.229742                         -0.630182
axial4                           -1.015420                         -0.518267
…                                      …                                 …
axial67                          -0.318616                         -0.138743
axial68                           0.318486                          0.136610
axial69                           0.910369                          0.387046
axial70                           1.421071                          0.604225
axial71                           1.795960                          0.773663

          2022-03-11 14:46:19.814000+00:00  2022-03-11 14:46:21.993000+00:00  \
axial0                           -1.484869                         -0.759188
axial1                           -1.450552                         -0.741219
axial2                           -1.375108                         -0.704368
axial3                           -1.229742                         -0.630182
axial4                           -1.015420                         -0.518267
```

```
…                                    …                                    …
axial67                        -0.318616                            -0.138743
axial68                         0.318486                             0.136610
axial69                         0.910369                             0.387046
axial70                         1.421071                             0.604225
axial71                         1.795960                             0.773663

         2022-03-11 14:46:25.353000+00:00
axial0                         -2.244057
axial1                         -2.191770
axial2                         -2.079476
axial3                         -1.859924
axial4                         -1.533687
…                                    …
axial67                        -0.457359
axial68                         0.455096
axial69                         1.297415
axial70                         2.025296
axial71                         2.569623

[72 rows x 7 columns]
```

[48]:
```
fig, axes = plt.subplots(1,3, figsize=(14,6))

for panel, timestamp in enumerate(m2_correction_applied.index):

    img = axes[panel].scatter(
        m2_xact,
        m2_yact,
        c=m2_correction_applied.T[timestamp],
        s=200,
        vmin=-1.5,
        vmax=1.5
    )

    axes[panel].axis('equal')

# axis = fig.add_axes([0,1,0,1])

fig.colorbar(img, ax=axes)
axes[0].set_title("+1um")
axes[1].set_title("zero")
axes[2].set_title("+2um")
fig.patch.set_facecolor('white')
fig.text(
    0.5,
    0.95,
```

```
    "Step 8 - M2 Corrections",
    ha="center",
    weight="bold",
    size="large"
)

fig.savefig("m2.png")
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [48], in <cell line: 3>()
      1 fig, axes = plt.subplots(1,3, figsize=(14,6))
      3 for panel, timestamp in enumerate(m2_correction_applied.index):
----> 5     img = axes[panel].scatter(
      6         m2_xact,
      7         m2_yact,
      8         c=m2_correction_applied.T[timestamp],
      9         s=200,
     10         vmin=-1.5,
     11         vmax=1.5
     12     )
     14     axes[panel].axis('equal')
     16 # axis = fig.add_axes([0,1,0,1])

IndexError: index 3 is out of bounds for axis 0 with size 3
```



1.5.4 Display M1M3 Correction

[49]:
```
FATABLE_XPOSITION = 2
FATABLE_YPOSITION = 3

FATABLE = np.array([
    [0,101,0.776782776,0,-2.158743,'SAA',3,1,'NA',-1,-1,0,-1],
    [1,102,1.442567993,0,-2.158743,'DAA',1,17,'+Y',-1,0,1,0],
    [2,103,2.10837793,0,-2.158743,'DAA',4,17,'+Y',-1,1,2,1],
    [3,104,2.774187988,0,-2.158743,'DAA',2,17,'+Y',-1,2,3,2],
    [4,105,3.439998047,0,-2.158743,'DAA',3,17,'+Y',-1,3,4,3],
    [5,106,3.968012939,0,-2.158743,'SAA',2,1,'NA',-1,-1,5,-1],
    [6,107,0.44386499,-0.57660498,-2.158743,'SAA',1,1,'NA',-1,-1,6,-1],
    [7,108,1.109675049,-0.57660498,-2.158743,'DAA',4,18,'+Y',-1,4,7,4],
    [8,109,1.775484985,-0.57660498,-2.158743,'DAA',2,18,'+Y',-1,5,8,5],
    [9,110,2.441295898,-0.57660498,-2.158743,'DAA',3,18,'+Y',-1,6,9,6],
    [10,111,3.107080078,-0.57660498,-2.158743,'DAA',1,18,'+Y',-1,7,10,7],
    [11,112,3.772891113,-0.57660498,-2.158743,'DAA',4,19,'-X',0,-1,11,8],
    [12,113,0,-1.153209961,-2.158743,'DAA',2,19,'+Y',-1,8,12,9],
    [13,114,0.776782776,-1.153209961,-2.158743,'DAA',3,19,'+Y',-1,9,13,10],
    [14,115,1.442567993,-1.153209961,-2.158743,'DAA',1,19,'+Y',-1,10,14,11],
    [15,116,2.10837793,-1.153209961,-2.158743,'DAA',4,20,'+Y',-1,11,15,12],
    [16,117,2.774187988,-1.153209961,-2.158743,'DAA',2,20,'+Y',-1,12,16,13],
    [17,118,3.439998047,-1.153209961,-2.158743,'DAA',3,20,'+Y',-1,13,17,14],
    [18,119,3.9005,-0.997687012,-2.158743,'SAA',2,2,'NA',-1,-1,18,-1],
    [19,120,0.44386499,-1.729819946,-2.158743,'DAA',1,20,'+Y',-1,14,19,15],
    [20,121,1.109675049,-1.729819946,-2.158743,'DAA',4,21,'+Y',-1,15,20,16],
    [21,122,1.775484985,-1.729819946,-2.158743,'DAA',2,21,'+Y',-1,16,21,17],
    [22,123,2.44127002,-1.729819946,-2.158743,'DAA',3,21,'+Y',-1,17,22,18],
    [23,124,3.107080078,-1.729819946,-2.158743,'DAA',1,21,'+Y',-1,18,23,19],
    [24,125,3.724452881,-1.517949951,-2.158743,'SAA',4,1,'NA',-1,-1,24,-1],
    [25,126,0,-2.306419922,-2.158743,'DAA',2,22,'+Y',-1,19,25,20],
    [26,127,0.776782776,-2.306419922,-2.158743,'DAA',3,22,'+Y',-1,20,26,21],
    [27,128,1.442567993,-2.306419922,-2.158743,'DAA',1,22,'-X',1,-1,27,22],
    [28,129,2.10837793,-2.306419922,-2.158743,'DAA',4,22,'+Y',-1,21,28,23],
    [29,130,2.774187988,-2.306419922,-2.158743,'DAA',2,23,'+Y',-1,22,29,24],
    [30,131,3.387954102,-2.167409912,-2.158743,'SAA',3,2,'NA',-1,-1,30,-1],
    [31,132,0.44386499,-2.883030029,-2.158743,'DAA',1,23,'+Y',-1,23,31,25],
    [32,133,1.109675049,-2.883030029,-2.158743,'DAA',4,23,'+Y',-1,24,32,26],
    [33,134,1.775484985,-2.883030029,-2.158743,'DAA',2,24,'+Y',-1,25,33,27],
    [34,135,2.44127002,-2.883030029,-2.158743,'DAA',3,23,'-X',2,-1,34,28],
    [35,136,2.939364014,-2.745179932,-2.158743,'SAA',4,2,'NA',-1,-1,35,-1],
    [36,137,0.221945206,-3.459629883,-2.158743,'DAA',2,25,'+Y',-1,26,36,29],
    [37,138,0.88772998,-3.459629883,-2.158743,'DAA',3,24,'+Y',-1,27,37,30],
    [38,139,1.553540039,-3.267429932,-2.158743,'SAA',1,2,'NA',-1,-1,38,-1],
    [39,140,2.089733887,-3.436389893,-2.158743,'SAA',4,3,'NA',-1,-1,39,-1],
    [40,141,0.365734589,-4.00525,-2.158743,'SAA',1,3,'NA',-1,-1,40,-1],
```

```
[41,142,1.085088013,-3.87276001,-2.158743,'SAA',2,3,'NA',-1,-1,41,-1],
[42,143,1.60401001,-3.692780029,-2.158743,'SAA',3,3,'NA',-1,-1,42,-1],
[43,207,-0.44386499,-0.57660498,-2.158743,'SAA',1,4,'NA',-1,-1,43,-1],
[44,208,-1.109680054,-0.57660498,-2.158743,'DAA',4,24,'+Y',-1,28,44,31],
[45,209,-1.77548999,-0.57660498,-2.158743,'DAA',2,26,'+Y',-1,29,45,32],
[46,210,-2.441300049,-0.57660498,-2.158743,'DAA',3,25,'+Y',-1,30,46,33],
[47,211,-3.107080078,-0.57660498,-2.158743,'DAA',1,24,'+Y',-1,31,47,34],
[48,212,-3.772889893,-0.57660498,-2.158743,'DAA',4,25,'+X',3,-1,48,35],
[49,214,-0.77678302,-1.153209961,-2.158743,'DAA',3,26,'+Y',-1,32,49,36],
[50,215,-1.442569946,-1.153209961,-2.158743,'DAA',1,25,'+Y',-1,33,50,37],
[51,216,-2.108379883,-1.153209961,-2.158743,'DAA',4,26,'+Y',-1,34,51,38],
[52,217,-2.774189941,-1.153209961,-2.158743,'DAA',2,27,'+Y',-1,35,52,39],
[53,218,-3.44,-1.153209961,-2.158743,'DAA',3,27,'+Y',-1,36,53,40],
[54,219,-3.9005,-0.997687012,-2.158743,'SAA',2,4,'NA',-1,-1,54,-1],
[55,220,-0.44386499,-1.729819946,-2.158743,'DAA',1,26,'+Y',-1,37,55,41],
[56,221,-1.109680054,-1.729819946,-2.158743,'DAA',4,27,'+Y',-1,38,56,42],
[57,222,-1.77548999,-1.729819946,-2.158743,'DAA',2,28,'+Y',-1,39,57,43],
[58,223,-2.44127002,-1.729819946,-2.158743,'DAA',3,28,'+Y',-1,40,58,44],
[59,224,-3.107080078,-1.729819946,-2.158743,'DAA',1,27,'+Y',-1,41,59,45],
[60,225,-3.724449951,-1.517949951,-2.158743,'SAA',4,4,'NA',-1,-1,60,-1],
[61,227,-0.77678302,-2.306419922,-2.158743,'DAA',3,29,'+Y',-1,42,61,46],
[62,228,-1.442569946,-2.306419922,-2.158743,'DAA',1,28,'+X',4,-1,62,47],
[63,229,-2.108379883,-2.306419922,-2.158743,'DAA',4,28,'+Y',-1,43,63,48],
[64,230,-2.774189941,-2.306419922,-2.158743,'DAA',2,29,'+Y',-1,44,64,49],
[65,231,-3.387949951,-2.167409912,-2.158743,'SAA',3,4,'NA',-1,-1,65,-1],
[66,232,-0.44386499,-2.883030029,-2.158743,'DAA',1,29,'+Y',-1,45,66,50],
[67,233,-1.109680054,-2.883030029,-2.158743,'DAA',4,29,'+Y',-1,46,67,51],
[68,234,-1.77548999,-2.883030029,-2.158743,'DAA',2,30,'+Y',-1,47,68,52],
[69,235,-2.44127002,-2.883030029,-2.158743,'DAA',3,30,'+X',5,-1,69,53],
[70,236,-2.939360107,-2.745179932,-2.158743,'SAA',4,5,'NA',-1,-1,70,-1],
[71,237,-0.221945007,-3.459629883,-2.158743,'DAA',2,31,'+Y',-1,48,71,54],
[72,238,-0.88772998,-3.459629883,-2.158743,'DAA',3,31,'+Y',-1,49,72,55],
[73,239,-1.553540039,-3.267429932,-2.158743,'SAA',1,5,'NA',-1,-1,73,-1],
[74,240,-2.08972998,-3.436389893,-2.158743,'SAA',4,6,'NA',-1,-1,74,-1],
[75,241,-0.365734985,-4.00525,-2.158743,'SAA',1,6,'NA',-1,-1,75,-1],
[76,242,-1.085089966,-3.87276001,-2.158743,'SAA',2,5,'NA',-1,-1,76,-1],
[77,243,-1.60401001,-3.692780029,-2.158743,'SAA',3,5,'NA',-1,-1,77,-1],
[78,301,-0.77678302,0,-2.158743,'SAA',3,6,'NA',-1,-1,78,-1],
[79,302,-1.442569946,0,-2.158743,'DAA',1,30,'+Y',-1,50,79,56],
[80,303,-2.108379883,0,-2.158743,'DAA',4,30,'+Y',-1,51,80,57],
[81,304,-2.774189941,0,-2.158743,'DAA',2,32,'+Y',-1,52,81,58],
[82,305,-3.44,0,-2.158743,'DAA',3,32,'+Y',-1,53,82,59],
[83,306,-3.96801001,0,-2.158743,'SAA',2,6,'NA',-1,-1,83,-1],
[84,307,-0.44386499,0.576605408,-2.158743,'SAA',1,7,'NA',-1,-1,84,-1],
[85,308,-1.109680054,0.576605408,-2.158743,'DAA',4,31,'+Y',-1,54,85,60],
[86,309,-1.77548999,0.576605408,-2.158743,'DAA',2,33,'+Y',-1,55,86,61],
[87,310,-2.441300049,0.576605408,-2.158743,'DAA',3,33,'+Y',-1,56,87,62],
```

```
[88,311,-3.107080078,0.576605408,-2.158743,'DAA',1,31,'-Y',-1,57,88,63],
[89,312,-3.772889893,0.576605408,-2.158743,'DAA',4,32,'+X',6,-1,89,64],
[90,313,0,1.15321106,-2.158743,'DAA',2,34,'+Y',-1,58,90,65],
[91,314,-0.77678302,1.15321106,-2.158743,'DAA',3,34,'+Y',-1,59,91,66],
[92,315,-1.442569946,1.15321106,-2.158743,'DAA',1,32,'+Y',-1,60,92,67],
[93,316,-2.108379883,1.15321106,-2.158743,'DAA',4,33,'+Y',-1,61,93,68],
[94,317,-2.774189941,1.15321106,-2.158743,'DAA',2,35,'+Y',-1,62,94,69],
[95,318,-3.44,1.15321106,-2.158743,'DAA',3,35,'+Y',-1,63,95,70],
[96,319,-3.9005,0.997686584,-2.158743,'SAA',2,7,'NA',-1,-1,96,-1],
[97,320,-0.44386499,1.72981604,-2.158743,'DAA',1,33,'+Y',-1,64,97,71],
[98,321,-1.109680054,1.72981604,-2.158743,'DAA',4,34,'+Y',-1,65,98,72],
[99,322,-1.77548999,1.72981604,-2.158743,'DAA',2,36,'+Y',-1,66,99,73],
[100,323,-2.44127002,1.72981604,-2.158743,'DAA',3,36,'+Y',-1,67,100,74],
[101,324,-3.107080078,1.72981604,-2.158743,'DAA',1,34,'+Y',-1,68,101,75],
[102,325,-3.724449951,1.517954956,-2.158743,'SAA',4,7,'NA',-1,-1,102,-1],
[103,326,0,2.306422119,-2.158743,'DAA',2,37,'+Y',-1,69,103,76],
[104,327,-0.77678302,2.306422119,-2.158743,'DAA',3,37,'+Y',-1,70,104,77],
[105,328,-1.442569946,2.306422119,-2.158743,'DAA',1,35,'+X',7,-1,105,78],
[106,329,-2.108379883,2.306422119,-2.158743,'DAA',4,35,'+Y',-1,71,106,79],
[107,330,-2.774189941,2.306422119,-2.158743,'DAA',2,38,'+Y',-1,72,107,80],
[108,331,-3.387949951,2.167406982,-2.158743,'SAA',3,7,'NA',-1,-1,108,-1],
[109,332,-0.44386499,2.8830271,-2.158743,'DAA',1,36,'+Y',-1,73,109,81],
[110,333,-1.109680054,2.8830271,-2.158743,'DAA',4,36,'+Y',-1,74,110,82],
[111,334,-1.77548999,2.8830271,-2.158743,'DAA',2,39,'-Y',-1,75,111,83],
[112,335,-2.44127002,2.8830271,-2.158743,'DAA',3,38,'+X',8,-1,112,84],
[113,336,-2.939360107,2.745180908,-2.158743,'SAA',4,8,'NA',-1,-1,113,-1],
[114,337,-0.221945007,3.45963208,-2.158743,'DAA',2,40,'+Y',-1,76,114,85],
[115,338,-0.88772998,3.45963208,-2.158743,'DAA',3,39,'+Y',-1,77,115,86],
[116,339,-1.553540039,3.267430908,-2.158743,'SAA',1,8,'NA',-1,-1,116,-1],
[117,340,-2.08972998,3.436391113,-2.158743,'SAA',4,9,'NA',-1,-1,117,-1],
[118,341,-0.365734985,4.00525,-2.158743,'SAA',1,9,'NA',-1,-1,118,-1],
[119,342,-1.085089966,3.872762939,-2.158743,'SAA',2,8,'NA',-1,-1,119,-1],
[120,343,-1.60401001,3.692779053,-2.158743,'SAA',3,8,'NA',-1,-1,120,-1],
[121,407,0.44386499,0.576605408,-2.158743,'SAA',1,10,'NA',-1,-1,121,-1],
[122,408,1.109675049,0.576605408,-2.158743,'DAA',4,37,'+Y',-1,78,122,87],
[123,409,1.775484985,0.576605408,-2.158743,'DAA',2,41,'+Y',-1,79,123,88],
[124,410,2.441295898,0.576605408,-2.158743,'DAA',3,40,'+Y',-1,80,124,89],
[125,411,3.107080078,0.576605408,-2.158743,'DAA',1,37,'-Y',-1,81,125,90],
[126,412,3.772891113,0.576605408,-2.158743,'DAA',4,38,'-X',9,-1,126,91],
[127,414,0.776782776,1.15321106,-2.158743,'DAA',3,41,'+Y',-1,82,127,92],
[128,415,1.442567993,1.15321106,-2.158743,'DAA',1,38,'+Y',-1,83,128,93],
[129,416,2.10837793,1.15321106,-2.158743,'DAA',4,39,'+Y',-1,84,129,94],
[130,417,2.774187988,1.15321106,-2.158743,'DAA',2,42,'+Y',-1,85,130,95],
[131,418,3.439998047,1.15321106,-2.158743,'DAA',3,42,'+Y',-1,86,131,96],
[132,419,3.9005,0.997686584,-2.158743,'SAA',2,9,'NA',-1,-1,132,-1],
[133,420,0.44386499,1.72981604,-2.158743,'DAA',1,39,'+Y',-1,87,133,97],
[134,421,1.109675049,1.72981604,-2.158743,'DAA',4,40,'+Y',-1,88,134,98],
```

```
    [135,422,1.775484985,1.72981604,-2.158743,'DAA',2,43,'+Y',-1,89,135,99],
    [136,423,2.44127002,1.72981604,-2.158743,'DAA',3,43,'+Y',-1,90,136,100],
    [137,424,3.107080078,1.72981604,-2.158743,'DAA',1,40,'+Y',-1,91,137,101],
    [138,425,3.724452881,1.517954956,-2.158743,'SAA',4,10,'NA',-1,-1,138,-1],
    [139,427,0.776782776,2.306422119,-2.158743,'DAA',3,44,'+Y',-1,92,139,102],
    [140,428,1.442567993,2.306422119,-2.158743,'DAA',1,41,'-X',10,-1,140,103],
    [141,429,2.10837793,2.306422119,-2.158743,'DAA',4,41,'+Y',-1,93,141,104],
    [142,430,2.774187988,2.306422119,-2.158743,'DAA',2,44,'+Y',-1,94,142,105],
    [143,431,3.387954102,2.167406982,-2.158743,'SAA',3,9,'NA',-1,-1,143,-1],
    [144,432,0.44386499,2.8830271,-2.158743,'DAA',1,42,'+Y',-1,95,144,106],
    [145,433,1.109675049,2.8830271,-2.158743,'DAA',4,42,'+Y',-1,96,145,107],
    [146,434,1.775484985,2.8830271,-2.158743,'DAA',2,45,'-Y',-1,97,146,108],
    [147,435,2.44127002,2.8830271,-2.158743,'DAA',3,45,'-X',11,-1,147,109],
    [148,436,2.939364014,2.745180908,-2.158743,'SAA',4,11,'NA',-1,-1,148,-1],
    [149,437,0.221945206,3.45963208,-2.158743,'DAA',2,46,'+Y',-1,98,149,110],
    [150,438,0.88772998,3.45963208,-2.158743,'DAA',3,46,'+Y',-1,99,150,111],
    [151,439,1.553540039,3.267430908,-2.158743,'SAA',1,11,'NA',-1,-1,151,-1],
    [152,440,2.089733887,3.436391113,-2.158743,'SAA',4,12,'NA',-1,-1,152,-1],
    [153,441,0.365734589,4.00525,-2.158743,'SAA',1,12,'NA',-1,-1,153,-1],
    [154,442,1.085088013,3.872762939,-2.158743,'SAA',2,10,'NA',-1,-1,154,-1],
    [155,443,1.60401001,3.692779053,-2.158743,'SAA',3,10,'NA',-1,-1,155,-1],
])
```

[50]:
```
m1m3_xact = np.float64(FATABLE[:, FATABLE_XPOSITION])
m1m3_yact = np.float64(FATABLE[:, FATABLE_YPOSITION])
```

[51]:
```
m1m3_yact
```

[51]:
```
array([ 0.        ,  0.        ,  0.        ,  0.        ,  0.        ,
        0.        , -0.57660498, -0.57660498, -0.57660498, -0.57660498,
       -0.57660498, -0.57660498, -1.15320996, -1.15320996, -1.15320996,
       -1.15320996, -1.15320996, -1.15320996, -0.99768701, -1.72981995,
       -1.72981995, -1.72981995, -1.72981995, -1.72981995, -1.51794995,
       -2.30641992, -2.30641992, -2.30641992, -2.30641992, -2.30641992,
       -2.16740991, -2.88303003, -2.88303003, -2.88303003, -2.88303003,
       -2.74517993, -3.45962988, -3.45962988, -3.26742993, -3.43638989,
       -4.00525   , -3.87276001, -3.69278003, -0.57660498, -0.57660498,
       -0.57660498, -0.57660498, -0.57660498, -0.57660498, -1.15320996,
       -1.15320996, -1.15320996, -1.15320996, -1.15320996, -0.99768701,
       -1.72981995, -1.72981995, -1.72981995, -1.72981995, -1.72981995,
       -1.51794995, -2.30641992, -2.30641992, -2.30641992, -2.30641992,
       -2.16740991, -2.88303003, -2.88303003, -2.88303003, -2.88303003,
       -2.74517993, -3.45962988, -3.45962988, -3.26742993, -3.43638989,
       -4.00525   , -3.87276001, -3.69278003,  0.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  0.        ,  0.57660541,
        0.57660541,  0.57660541,  0.57660541,  0.57660541,  0.57660541,
        1.15321106,  1.15321106,  1.15321106,  1.15321106,  1.15321106,
```

```
       1.15321106,  0.99768658,  1.72981604,  1.72981604,  1.72981604,
       1.72981604,  1.72981604,  1.51795496,  2.30642212,  2.30642212,
       2.30642212,  2.30642212,  2.30642212,  2.16740698,  2.8830271 ,
       2.8830271 ,  2.8830271 ,  2.8830271 ,  2.74518091,  3.45963208,
       3.45963208,  3.26743091,  3.43639111,  4.00525   ,  3.87276294,
       3.69277905,  0.57660541,  0.57660541,  0.57660541,  0.57660541,
       0.57660541,  0.57660541,  1.15321106,  1.15321106,  1.15321106,
       1.15321106,  1.15321106,  0.99768658,  1.72981604,  1.72981604,
       1.72981604,  1.72981604,  1.72981604,  1.51795496,  2.30642212,
       2.30642212,  2.30642212,  2.30642212,  2.16740698,  2.8830271 ,
       2.8830271 ,  2.8830271 ,  2.8830271 ,  2.74518091,  3.45963208,
       3.45963208,  3.26743091,  3.43639111,  4.00525   ,  3.87276294,
       3.69277905])
```

[52]:
```
m1m3_correction = await client.select_time_series(
    'lsst.sal.MTAOS.logevent_m1m3Correction',
    [f"zForces{i}" for i in range(156)],
    start.utc,
    end.utc
)
```

[53]:
```
m1m3_correction_applied = await client.select_time_series(
    'lsst.sal.MTM1M3.command_applyActiveOpticForces',
    [f"zForces{i}" for i in range(156)],
    start.utc,
    end.utc
)
```

[54]:
```
m1m3_correction
```

[54]:

	zForces0	zForces1	zForces2	zForces3	\
2022-03-11 14:45:59.047000+00:00	0.000000	0.000000	0.000000	0.000000	
2022-03-11 14:46:07.633000+00:00	0.018060	-0.022865	-0.027896	-0.002299	
2022-03-11 14:46:11.593000+00:00	0.026952	-0.037211	-0.041791	-0.000233	
2022-03-11 14:46:19.342000+00:00	0.026952	-0.037211	-0.041791	-0.000233	
2022-03-11 14:46:21.013000+00:00	0.018060	-0.022865	-0.027896	-0.002299	
2022-03-11 14:46:24.819000+00:00	0.045012	-0.060077	-0.069687	-0.002531	

	zForces4	zForces5	zForces6	zForces7	\
2022-03-11 14:45:59.047000+00:00	0.000000	0.000000	0.000000	0.000000	
2022-03-11 14:46:07.633000+00:00	0.027176	0.049886	5.305404	5.580944	
2022-03-11 14:46:11.593000+00:00	0.043319	0.076803	14.361468	13.055202	
2022-03-11 14:46:19.342000+00:00	0.043319	0.076803	14.361468	13.055202	
2022-03-11 14:46:21.013000+00:00	0.027176	0.049886	5.305404	5.580944	
2022-03-11 14:46:24.819000+00:00	0.070496	0.126689	19.666872	18.636147	

	zForces8	zForces9	…	zForces146	\

```
2022-03-11 14:45:59.047000+00:00   0.000000  0.000000  …    0.000000
2022-03-11 14:46:07.633000+00:00   4.462502  2.587142  …   -0.216042
2022-03-11 14:46:11.593000+00:00   9.637859  5.383689  …   -0.175045
2022-03-11 14:46:19.342000+00:00   9.637859  5.383689  …   -0.175045
2022-03-11 14:46:21.013000+00:00   4.462502  2.587142  …   -0.216042
2022-03-11 14:46:24.819000+00:00  14.100361  7.970831  …   -0.391087

                                  zForces147  zForces148  zForces149  \
2022-03-11 14:45:59.047000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:07.633000+00:00    4.494667    6.571590    1.070340
2022-03-11 14:46:11.593000+00:00    9.541871   13.812837    2.476829
2022-03-11 14:46:19.342000+00:00    9.541871   13.812837    2.476829
2022-03-11 14:46:21.013000+00:00    4.494667    6.571590    1.070340
2022-03-11 14:46:24.819000+00:00   14.036538   20.384426    3.547169

                                  zForces150  zForces151  zForces152  \
2022-03-11 14:45:59.047000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:07.633000+00:00    2.656595    3.126895    8.191571
2022-03-11 14:46:11.593000+00:00    5.739522    6.723288   17.177977
2022-03-11 14:46:19.342000+00:00    5.739522    6.723288   17.177977
2022-03-11 14:46:21.013000+00:00    2.656595    3.126895    8.191571
2022-03-11 14:46:24.819000+00:00    8.396116    9.850183   25.369549

                                  zForces153  zForces154  zForces155
2022-03-11 14:45:59.047000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:07.633000+00:00    9.781115    9.392729    8.886160
2022-03-11 14:46:11.593000+00:00   20.375088   19.600555   18.585400
2022-03-11 14:46:19.342000+00:00   20.375088   19.600555   18.585400
2022-03-11 14:46:21.013000+00:00    9.781115    9.392729    8.886160
2022-03-11 14:46:24.819000+00:00   30.156202   28.993284   27.471558

[6 rows x 156 columns]
```

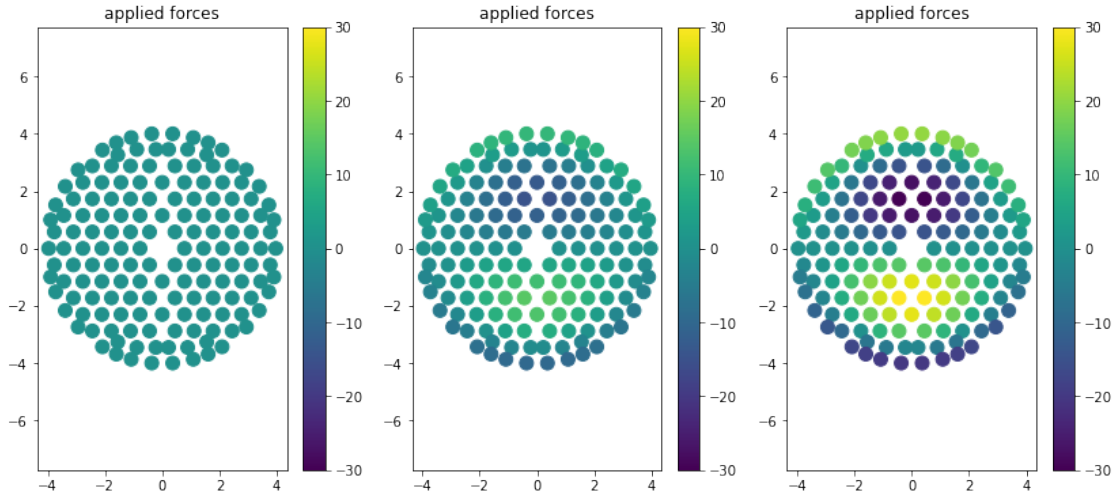
[55]:
```
m1m3_correction_applied
```

[55]:
```
                                  zForces0  zForces1  zForces2  zForces3  \
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  0.000000  0.000000
2022-03-11 14:46:08.249000+00:00  0.018060 -0.022865 -0.027896 -0.002299
2022-03-11 14:46:12.277000+00:00  0.026952 -0.037211 -0.041791 -0.000233
2022-03-11 14:46:12.904000+00:00  0.018060 -0.022865 -0.027896 -0.002299
2022-03-11 14:46:19.814000+00:00  0.026952 -0.037211 -0.041791 -0.000233
2022-03-11 14:46:21.993000+00:00  0.018060 -0.022865 -0.027896 -0.002299
2022-03-11 14:46:25.352000+00:00  0.045012 -0.060077 -0.069687 -0.002531

                                  zForces4  zForces5  zForces6  zForces7  \
2022-03-11 14:46:02.192000+00:00  0.000000  0.000000  0.000000  0.000000
2022-03-11 14:46:08.249000+00:00  0.027176  0.049886  5.305404  5.580944
```

```
2022-03-11 14:46:12.277000+00:00  0.043319  0.076803  14.361468  13.055202
2022-03-11 14:46:12.904000+00:00  0.027176  0.049886   5.305404   5.580944
2022-03-11 14:46:19.814000+00:00  0.043319  0.076803  14.361468  13.055202
2022-03-11 14:46:21.993000+00:00  0.027176  0.049886   5.305404   5.580944
2022-03-11 14:46:25.352000+00:00  0.070496  0.126689  19.666872  18.636147

                                   zForces8  zForces9  …  zForces146  \
2022-03-11 14:46:02.192000+00:00   0.000000  0.000000  …    0.000000
2022-03-11 14:46:08.249000+00:00   4.462502  2.587142  …   -0.216042
2022-03-11 14:46:12.277000+00:00   9.637859  5.383689  …   -0.175045
2022-03-11 14:46:12.904000+00:00   4.462502  2.587142  …   -0.216042
2022-03-11 14:46:19.814000+00:00   9.637859  5.383689  …   -0.175045
2022-03-11 14:46:21.993000+00:00   4.462502  2.587142  …   -0.216042
2022-03-11 14:46:25.352000+00:00  14.100361  7.970831  …   -0.391087

                                  zForces147  zForces148  zForces149  \
2022-03-11 14:46:02.192000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:08.249000+00:00    4.494667    6.571590    1.070340
2022-03-11 14:46:12.277000+00:00    9.541871   13.812837    2.476829
2022-03-11 14:46:12.904000+00:00    4.494667    6.571590    1.070340
2022-03-11 14:46:19.814000+00:00    9.541871   13.812837    2.476829
2022-03-11 14:46:21.993000+00:00    4.494667    6.571590    1.070340
2022-03-11 14:46:25.352000+00:00   14.036538   20.384426    3.547169

                                  zForces150  zForces151  zForces152  \
2022-03-11 14:46:02.192000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:08.249000+00:00    2.656595    3.126895    8.191571
2022-03-11 14:46:12.277000+00:00    5.739522    6.723288   17.177977
2022-03-11 14:46:12.904000+00:00    2.656595    3.126895    8.191571
2022-03-11 14:46:19.814000+00:00    5.739522    6.723288   17.177977
2022-03-11 14:46:21.993000+00:00    2.656595    3.126895    8.191571
2022-03-11 14:46:25.352000+00:00    8.396116    9.850183   25.369549

                                  zForces153  zForces154  zForces155
2022-03-11 14:46:02.192000+00:00    0.000000    0.000000    0.000000
2022-03-11 14:46:08.249000+00:00    9.781115    9.392729    8.886160
2022-03-11 14:46:12.277000+00:00   20.375088   19.600555   18.585400
2022-03-11 14:46:12.904000+00:00    9.781115    9.392729    8.886160
2022-03-11 14:46:19.814000+00:00   20.375088   19.600555   18.585400
2022-03-11 14:46:21.993000+00:00    9.781115    9.392729    8.886160
2022-03-11 14:46:25.352000+00:00   30.156202   28.993284   27.471558

[7 rows x 156 columns]
```

[56]:
```
fig, axes = plt.subplots(1,3, figsize=(14,6))

for ax, time in zip(axes.flatten(), m1m3_correction.T):
```

```
    img = ax.scatter(m1m3_xact, m1m3_yact, c=m1m3_correction.T[time], s=100,␣
↪vmin=-30, vmax=30)
    #plt.jet()
    ax.axis('equal')
    ax.set_title('applied forces')
    fig.colorbar(img, ax=ax)
```



[ ]: