

# ASC\_comapre\_30Nov

November 30, 2023

```
[ ]: from gwpy.timeseries import TimeSeries
      from gwpy.plot import Plot
```

```
[ ]:
```

```
[ ]: ReferenceTime = 1384757118 # 2023-11-23 06:45:00 UTC

      #CheckTime = 1384758918 # 2023-11-23 07:15 UTC - DARM Fuzzy Time
      CheckTime = 1385365518 # 2023/11/30 7:45 UTC - High MicroSeism and Low Freq
      ↪Noise

      dt = 600
```

```
[ ]: loops = {'DHARD': {'dofs': ['P', 'Y']},
              'DSOFT': {'dofs': ['P', 'Y']},
              'CHARD': {'dofs': ['P', 'Y']},
              'CSOFT': {'dofs': ['P', 'Y']},
              'INP1': {'dofs': ['P', 'Y']},
              'INP2': {'dofs': ['P', 'Y']},
              'PRC1': {'dofs': ['P', 'Y']},
              'PRC2': {'dofs': ['P', 'Y']},
              'MICH': {'dofs': ['P', 'Y']},
              'SRC1': {'dofs': ['P', 'Y']},
              'SRC2': {'dofs': ['P', 'Y']},
              }

      for loop in loops.keys():
          chanlist = []
          for dof in loops[loop]['dofs']:
              chanlist.append('H1:ASC-%s_%s_OUT_DQ'%(loop, dof))
          loops[loop]['chanlist'] = chanlist
```

```
[ ]: myplot = {}

      for loop in loops.keys():
          myplot[loop] = Plot(figsize=(50,8))
          ii = 0
          for dof in loops[loop]['dofs']:
              ii += 1
```

```

myplot[loop].add_subplot(1, len(loops[loop]['dofs']), ii)

print('Fetching data for %s'%loops[loop]['chanlist'][ii-1])
refData = TimeSeries.fetch(loops[loop]['chanlist'][ii-1],
↳ ReferenceTime, ReferenceTime+dt).asd(20,10)
checkData = TimeSeries.fetch(loops[loop]['chanlist'][ii-1], CheckTime,
↳ CheckTime+dt).asd(20,10)

ax = myplot[loop].gca()
ax.plot(refData, label='Reference time %s'%ReferenceTime)
ax.plot(checkData, label='Check time %s'%CheckTime)
ax.set_xscale('log')
ax.set_yscale('log')
ax.set_xlabel('Frequency [Hz]')
ax.set_ylabel('ASC Loop Motion [urad/rtHz]')
ax.set_title(loops[loop]['chanlist'][ii-1])
ax.legend()

```

```

Fetching data for H1:ASC-DHARD_P_OUT_DQ
Fetching data for H1:ASC-DHARD_Y_OUT_DQ
Fetching data for H1:ASC-DSOFT_P_OUT_DQ
Fetching data for H1:ASC-DSOFT_Y_OUT_DQ
Fetching data for H1:ASC-CHARD_P_OUT_DQ
Fetching data for H1:ASC-CHARD_Y_OUT_DQ
Fetching data for H1:ASC-CSOFT_P_OUT_DQ
Fetching data for H1:ASC-CSOFT_Y_OUT_DQ
Fetching data for H1:ASC-INP1_P_OUT_DQ
Fetching data for H1:ASC-INP1_Y_OUT_DQ
Fetching data for H1:ASC-INP2_P_OUT_DQ

```

/tmp/ipykernel\_784609/2093495424.py:18: UserWarning: Data has no positive values, and therefore cannot be log-scaled.

```
ax.set_yscale('log')
```

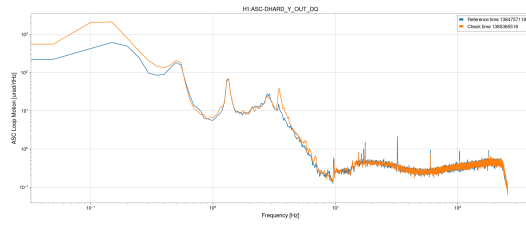
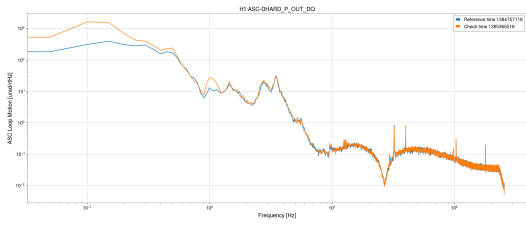
```

Fetching data for H1:ASC-INP2_Y_OUT_DQ
Fetching data for H1:ASC-PRC1_P_OUT_DQ
Fetching data for H1:ASC-PRC1_Y_OUT_DQ
Fetching data for H1:ASC-PRC2_P_OUT_DQ
Fetching data for H1:ASC-PRC2_Y_OUT_DQ
Fetching data for H1:ASC-MICH_P_OUT_DQ
Fetching data for H1:ASC-MICH_Y_OUT_DQ
Fetching data for H1:ASC-SRC1_P_OUT_DQ
Fetching data for H1:ASC-SRC1_Y_OUT_DQ
Fetching data for H1:ASC-SRC2_P_OUT_DQ
Fetching data for H1:ASC-SRC2_Y_OUT_DQ

```

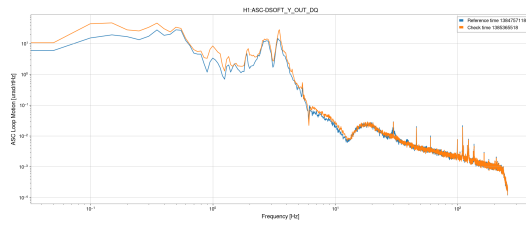
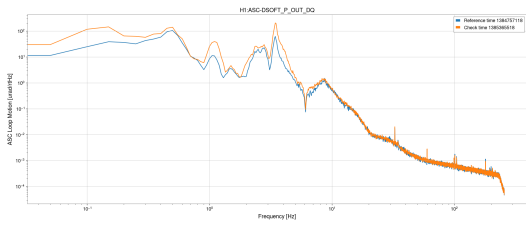
```
[ ]: myplot['DHARD']
```

```
[ ]:
```



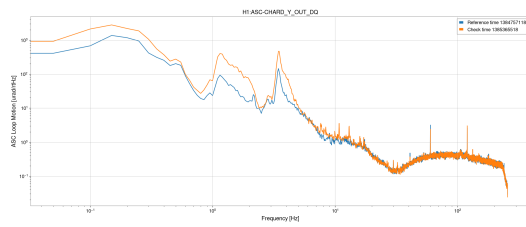
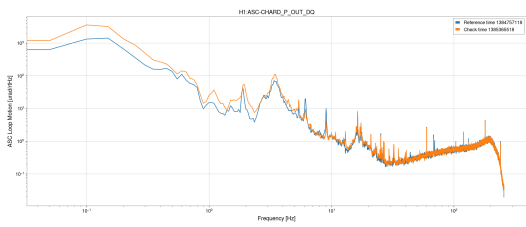
```
[ ]: myplot['DSOFT']
```

```
[ ]:
```



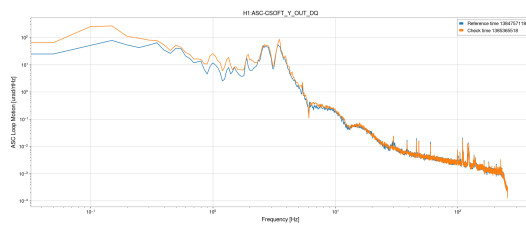
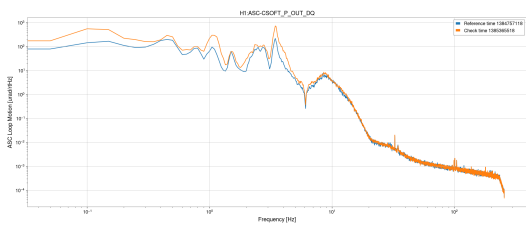
```
[ ]: myplot['CHARD']
```

```
[ ]:
```



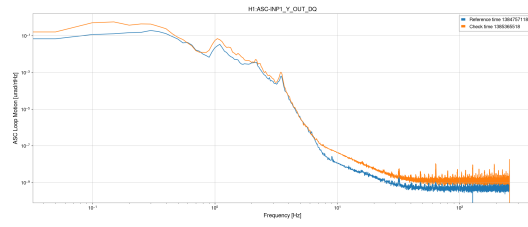
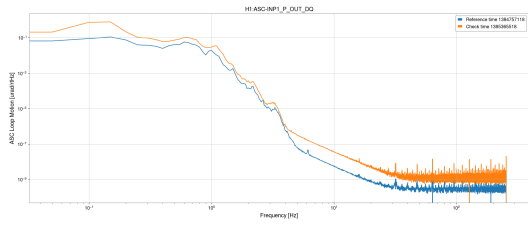
```
[ ]: myplot['CSOFT']
```

```
[ ]:
```



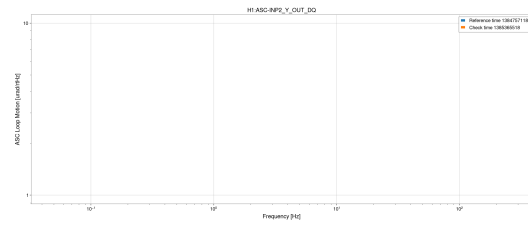
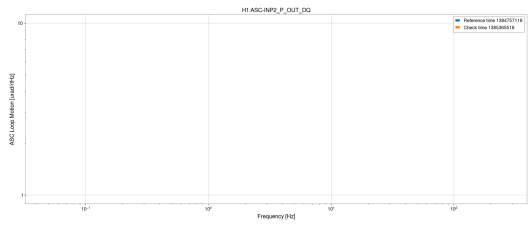
```
[ ]: myplot['INP1']
```

```
[ ]:
```



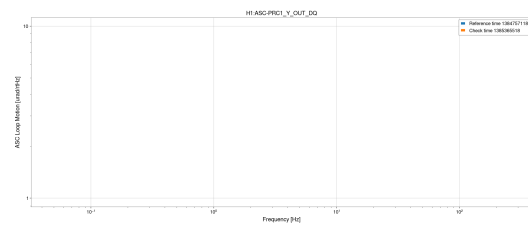
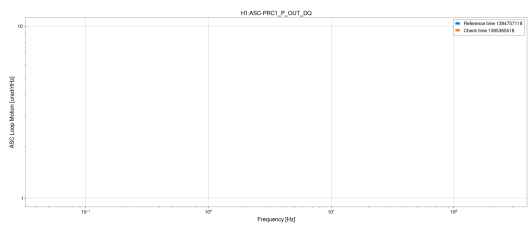
```
[ ]: myplot['INP2']
```

```
[ ]:
```



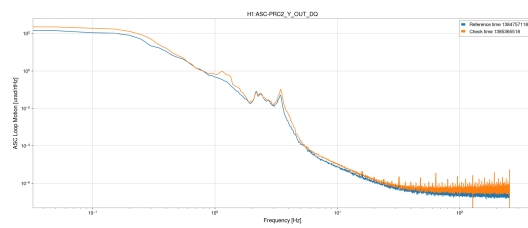
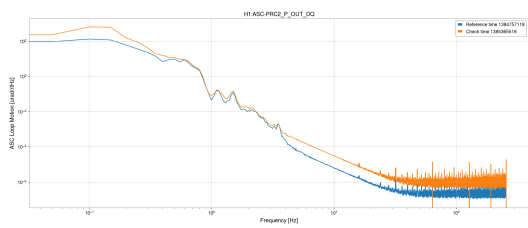
```
[ ]: myplot['PRC1']
```

```
[ ]:
```



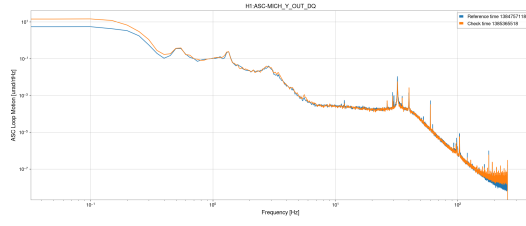
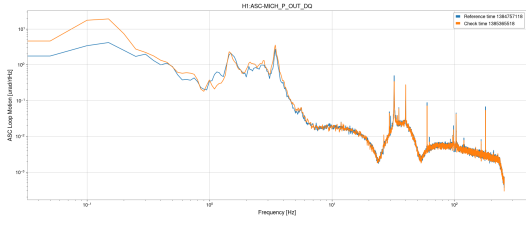
```
[ ]: myplot['PRC2']
```

```
[ ]:
```



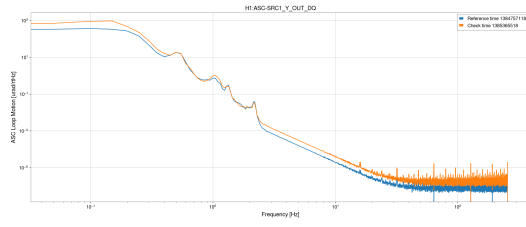
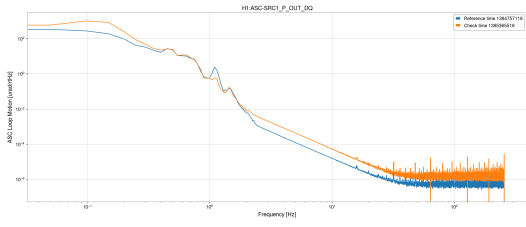
```
[ ]: myplot['MICH']
```

```
[ ]:
```



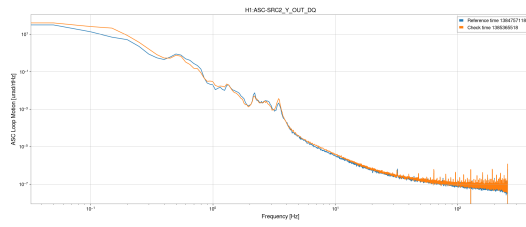
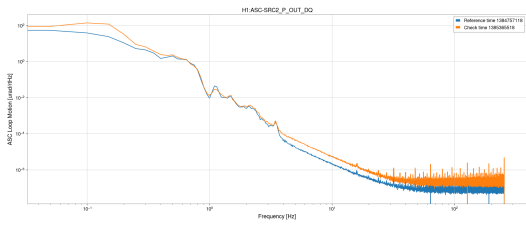
```
[ ]: myplot['SRC1']
```

```
[ ]:
```



```
[ ]: myplot['SRC2']
```

```
[ ]:
```



```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```