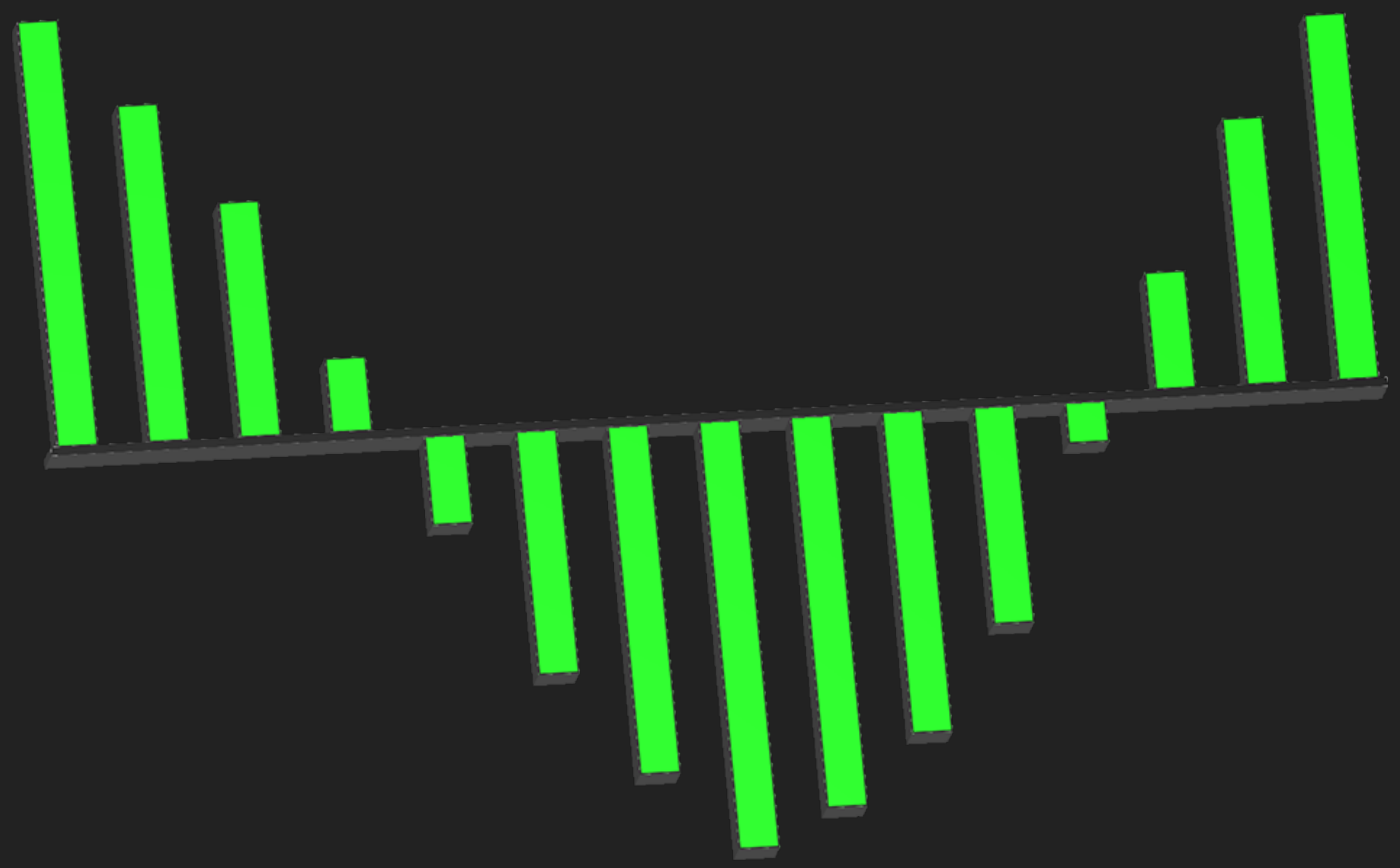
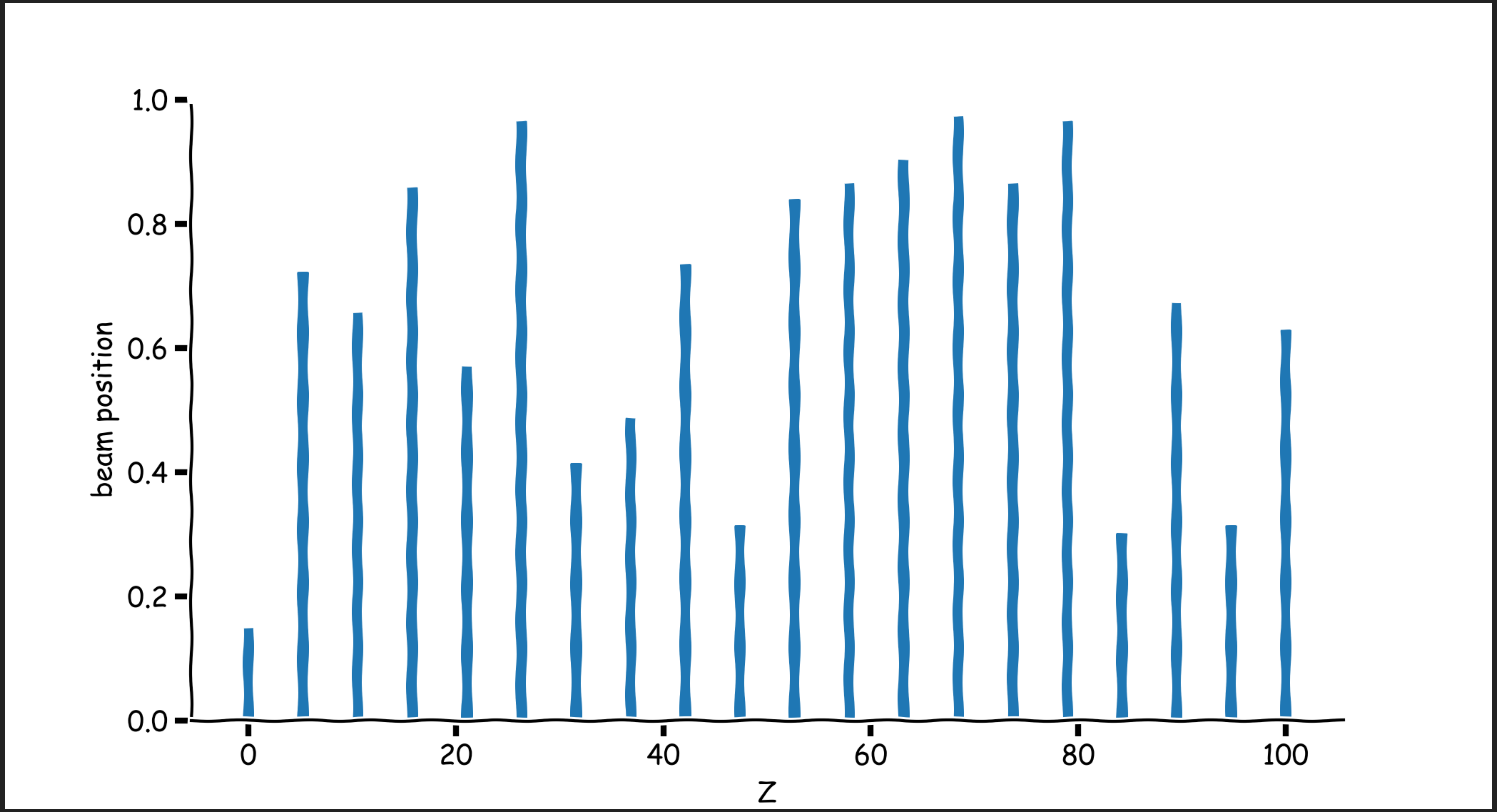


ORBIT DISPLAY

Matt Gibbs

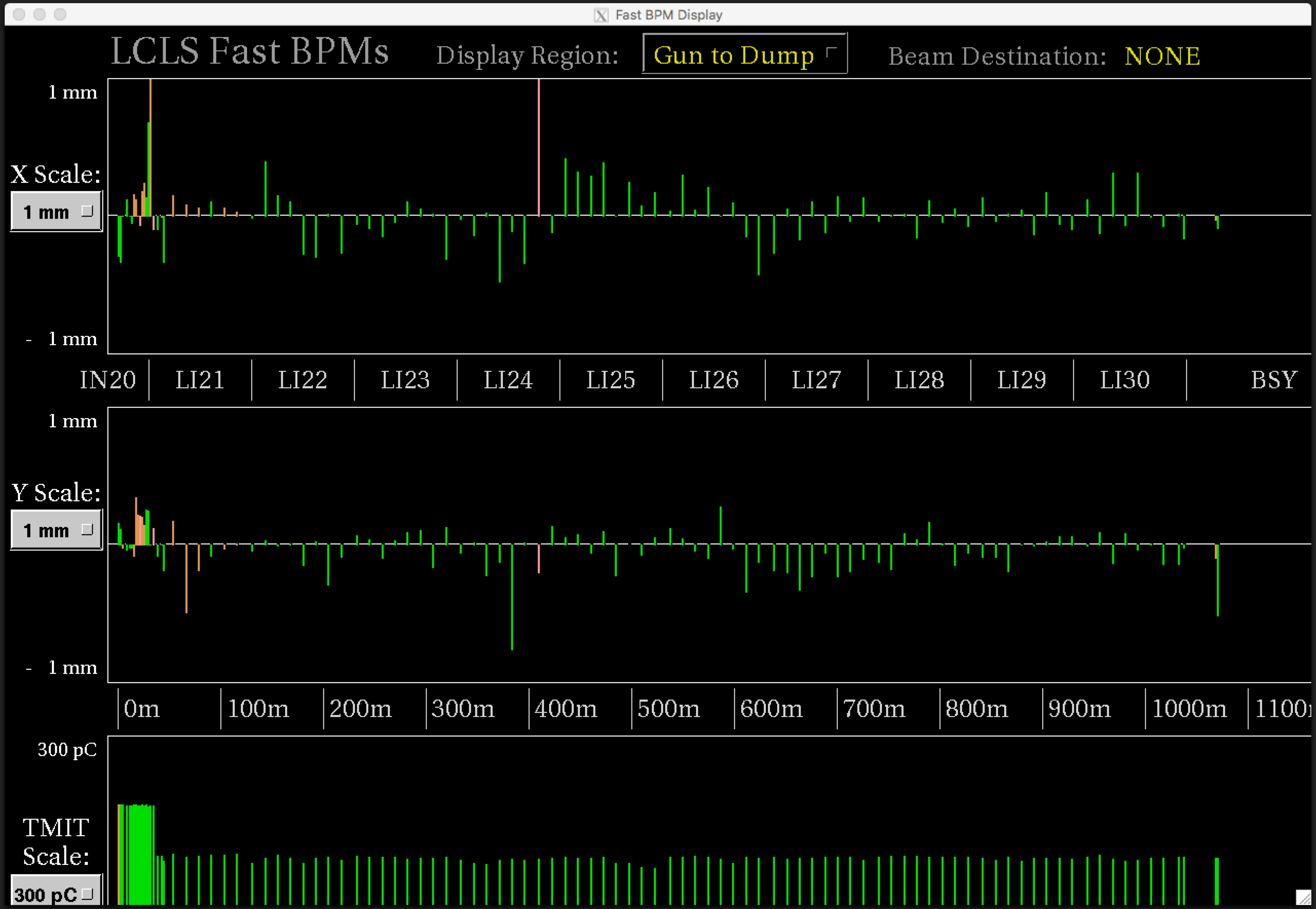
Accelerator Operations Specialist for LCLS

ORBIT DISPLAY



Orbit (noun) - The trajectory of the beam in a particle accelerator. Typically measured at several discrete locations by beam position monitors (BPMs) installed throughout the beam line.

PREVIOUS TOOLS FOR VIEWING THE ORBIT AT SLAC



Fast BPM Display

LCLS Fast BPMs

Display Region: Gun to Dump

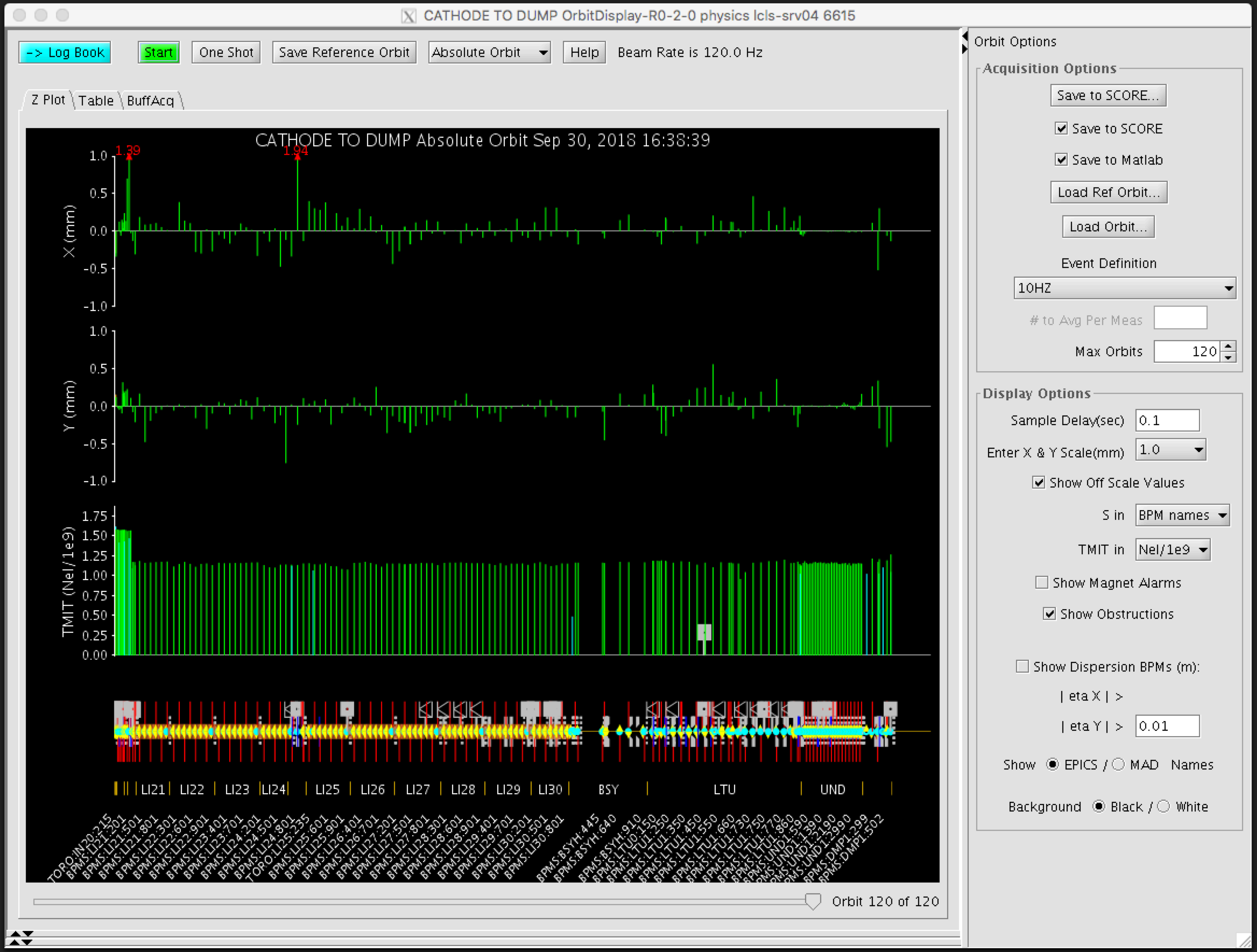
Beam Destination: NONE

X Scale: 1 mm

Y Scale: 1 mm

TMIT Scale: 300 pC

PREVIOUS TOOLS FOR VIEWING THE ORBIT AT SLAC



CATHODE TO DUMP OrbitDisplay-R0-2-0 physics lcls-srv04 6615

Beam Rate is 120.0 Hz

CATHODE TO DUMP Absolute Orbit Sep 30, 2018 16:38:39

Orbit 120 of 120

Orbit Options

Acquisition Options

☑ Save to SCORE

☑ Save to Matlab

Event Definition: 10HZ

to Avg Per Meas:

Max Orbits: 120

Display Options

Sample Delay(sec): 0.1

Enter X & Y Scale(mm): 1.0

☑ Show Off Scale Values

S in: BPM names

TMIT in: Nel/1e9

☐ Show Magnet Alarms

☑ Show Obstructions

☐ Show Dispersion BPMs (m):

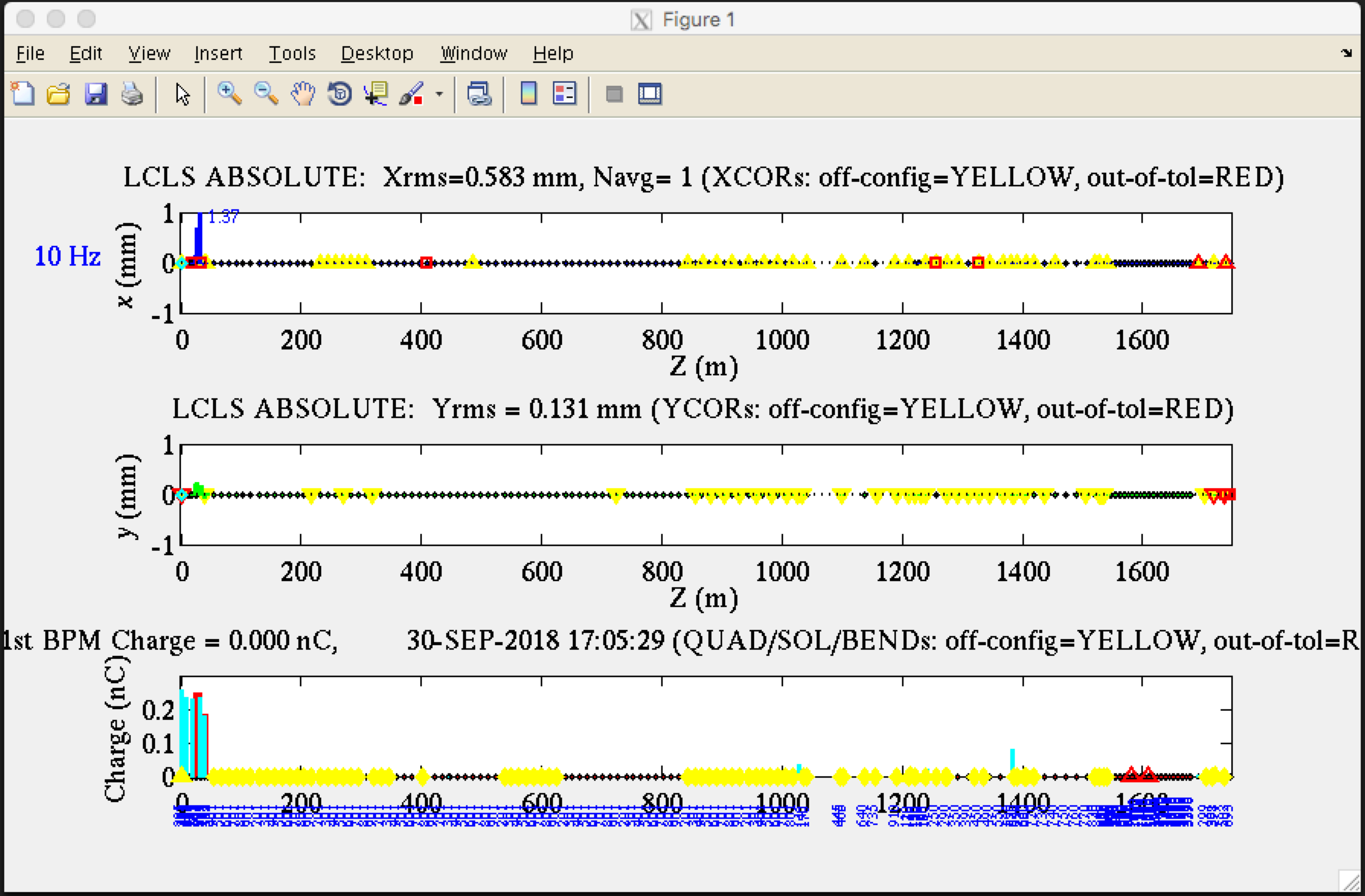
| eta X | >

| eta Y | > 0.01

Show ● EPICS / ○ MAD Names

Background ● Black / ○ White

PREVIOUS TOOLS FOR VIEWING THE ORBIT AT SLAC



LCLS ABSOLUTE: Xrms=0.583 mm, Navg= 1 (XCORs: off-config=YELLOW, out-of-tol=RED)

LCLS ABSOLUTE: Yrms = 0.131 mm (YCORs: off-config=YELLOW, out-of-tol=RED)

1st BPM Charge = 0.000 nC, 30-SEP-2018 17:05:29 (QUAD/SOL/BENDs: off-config=YELLOW, out-of-tol=R

PREVIOUS TOOLS FOR VIEWING THE ORBIT AT SLAC

	EDM “Fast Orbit”	Java “Orbit Display”	MATLAB “BPMs vs. Z”
Update Rate	60 Hz	10 Hz	5 Hz
Reference Orbits	No	Yes, from live data and config saves	Yes, from live data
Saves to Logbook	No	Yes, raster images	Yes, Postscript files
Configurable Viewing Window	Minimal choices	Yes	Multiple pre-defined options
Fit Data to Model	No	Yes	Yes

GOALS FOR A NEW DISPLAY

- 60 Hz update rate
- List of BPMs and their positions automatically retrieved from model
- Reference orbits from live data, saved orbit files, and saved machine configs
- Saves logbook data in publication-usable format
- Fully configurable viewing window
- Can fit orbit data to machine model in real-time

PLATFORM

EDM?

- + Fast
- + Easy to use

- - Minimal interactivity
- - Hard to Maintain

PLATFORM

JAVA?

- Slow to launch and slow to run
- No expertise in Operations Group

PLATFORM

MATLAB?

+ Easy plotting
+ Operators know it
+ Support libraries for LCLS already exist

- Slow launch times
- Subpar GUI tools
- Difficult to achieve 60 Hz update rate

PLATFORM

PYTHON?

- + Easy plotting
- + Operators know it
- + Good GUI tools
- + Fast launch times
- + Fast at run time

- - No support libraries for LCLS

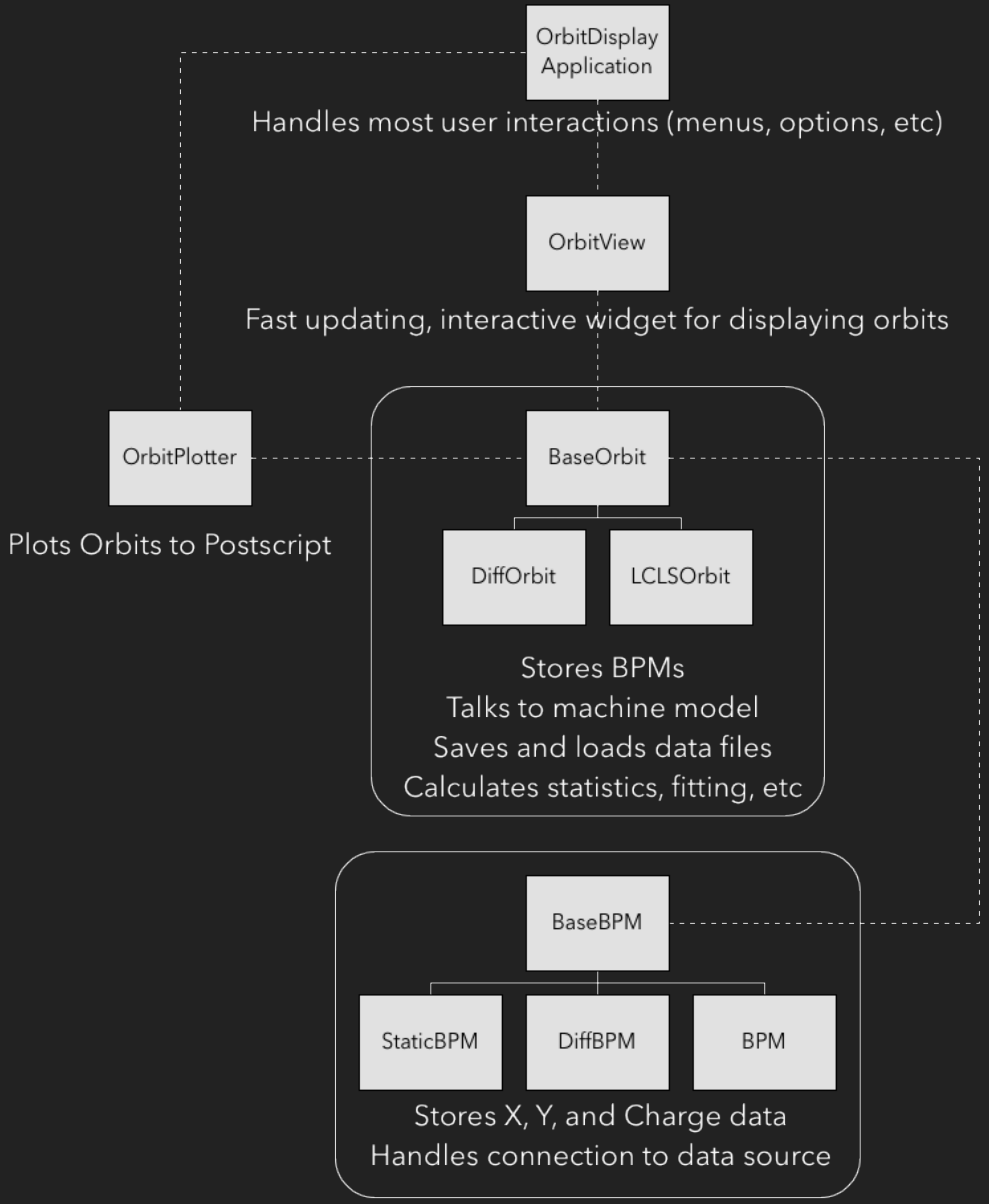
PLATFORM

PYTHON

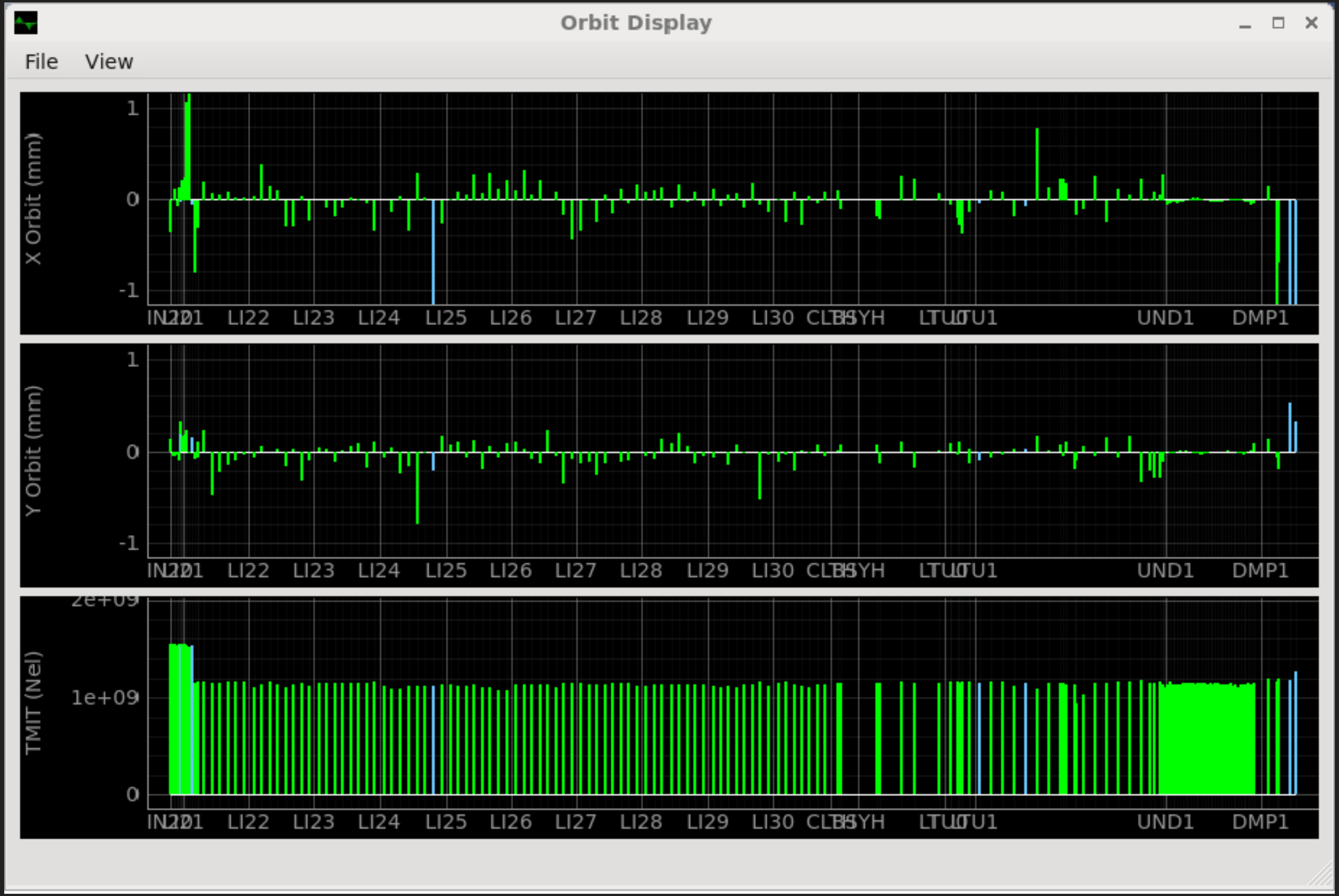
- PyQt
- PyQtGraph
- NumPy
- Matplotlib

ARCHITECTURE

- Modular
- Put as little functionality in GUI classes as possible
- Abstract base classes allow for flexibility in implementation

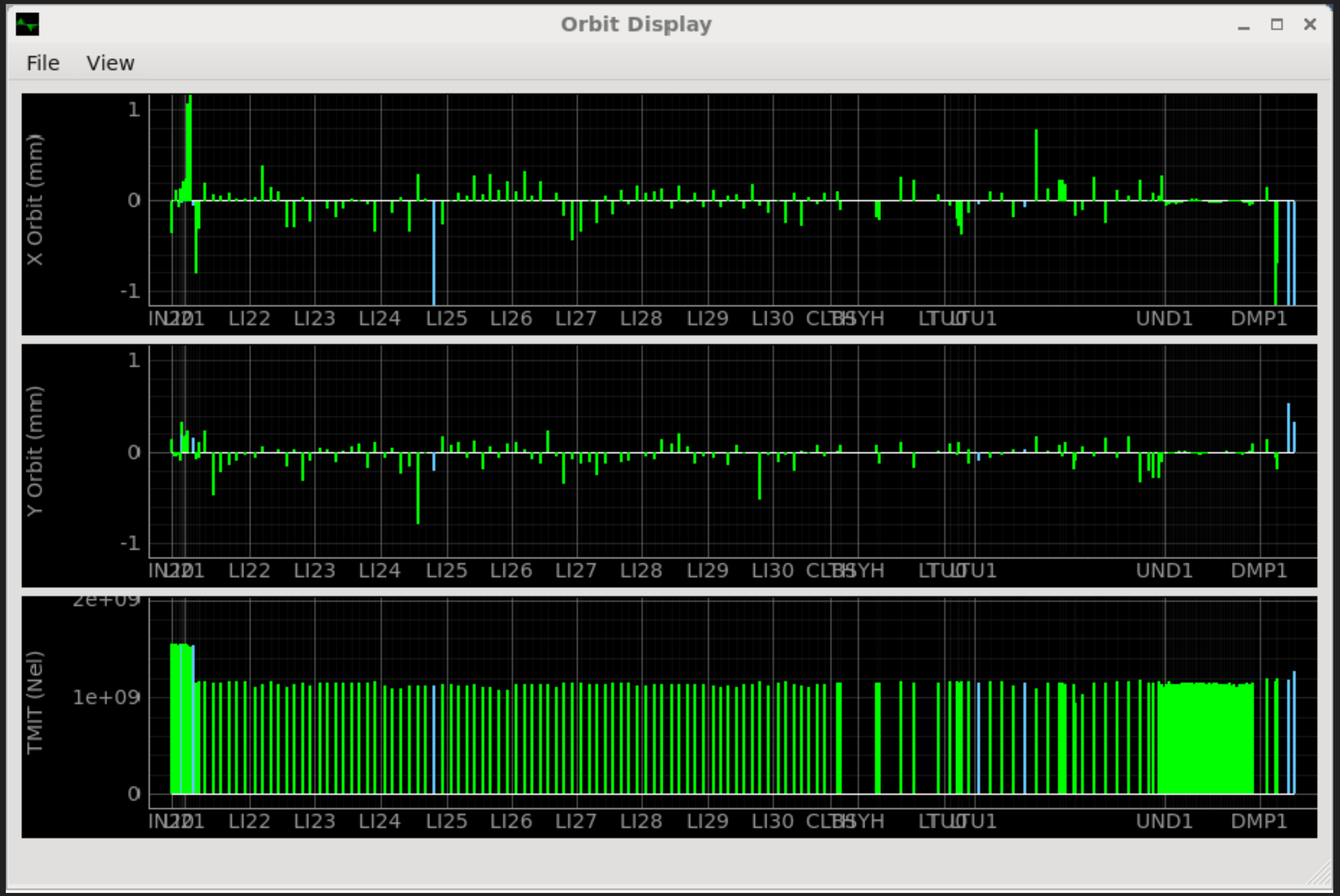

OrbitDisplay Application
Handles most user interactions (menus, options, etc)
OrbitView
Fast updating, interactive widget for displaying orbits
OrbitPlotter
Plots Orbits to Postscript
BaseOrbit
DiffOrbit
LCLSOrbit
Stores BPMs
Talks to machine model
Saves and loads data files
Calculates statistics, fitting, etc
BaseBPM
StaticBPM
DiffBPM
BPM
Stores X, Y, and Charge data
Handles connection to data source

ORBIT DISPLAY

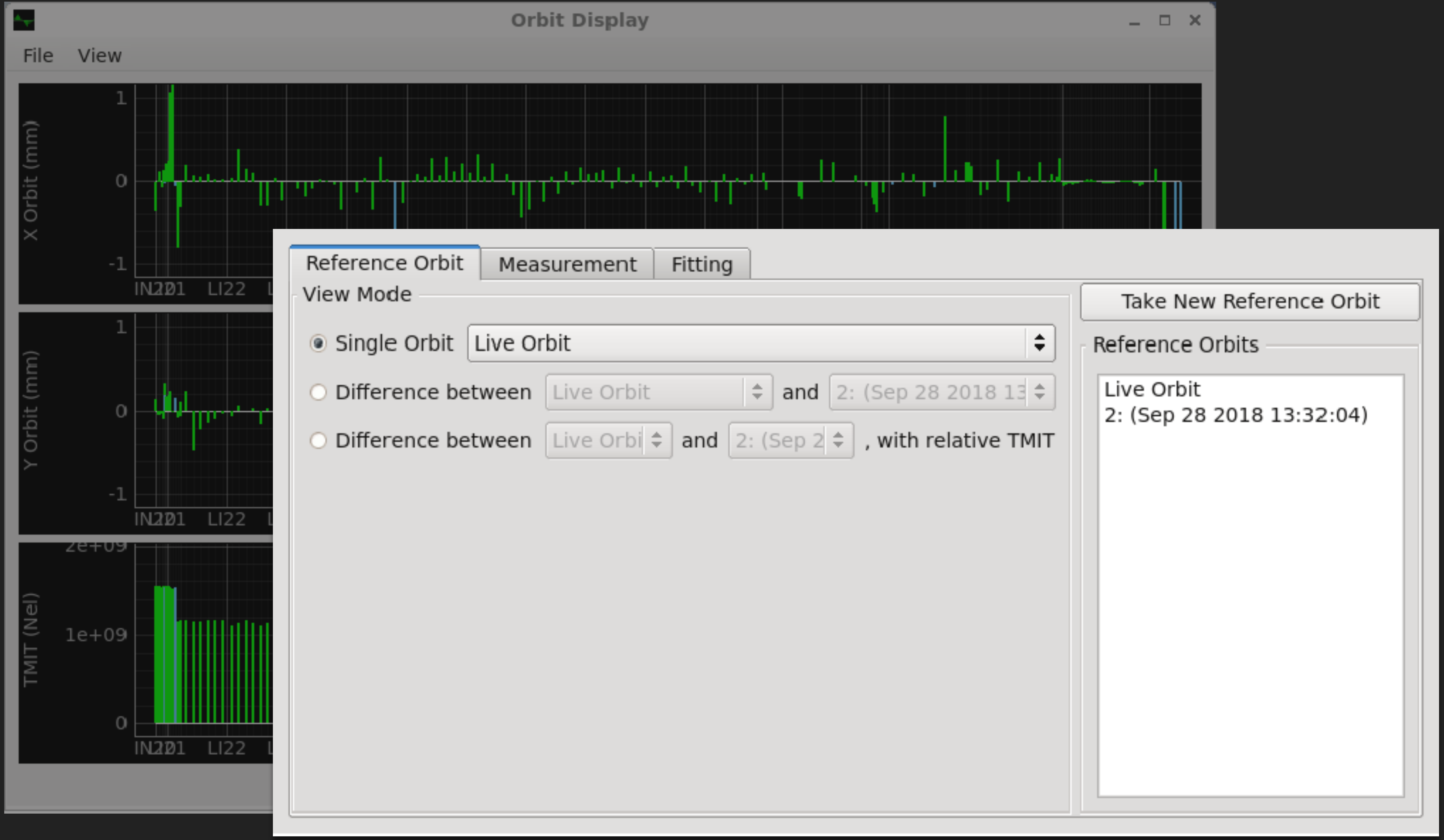


OPERATOR-FRIENDLY FEATURES

- Fast, easy zooming and panning with the mouse - don’t need to lose focus to adjust plot parameters
- Freely resizable to fit the workspace
- Adaptive tick-marks give more info when zoomed in
- Clutter-free main window usable as overhead display too



ORBIT DISPLAY



View Mode

● Single Orbit: Live Orbit

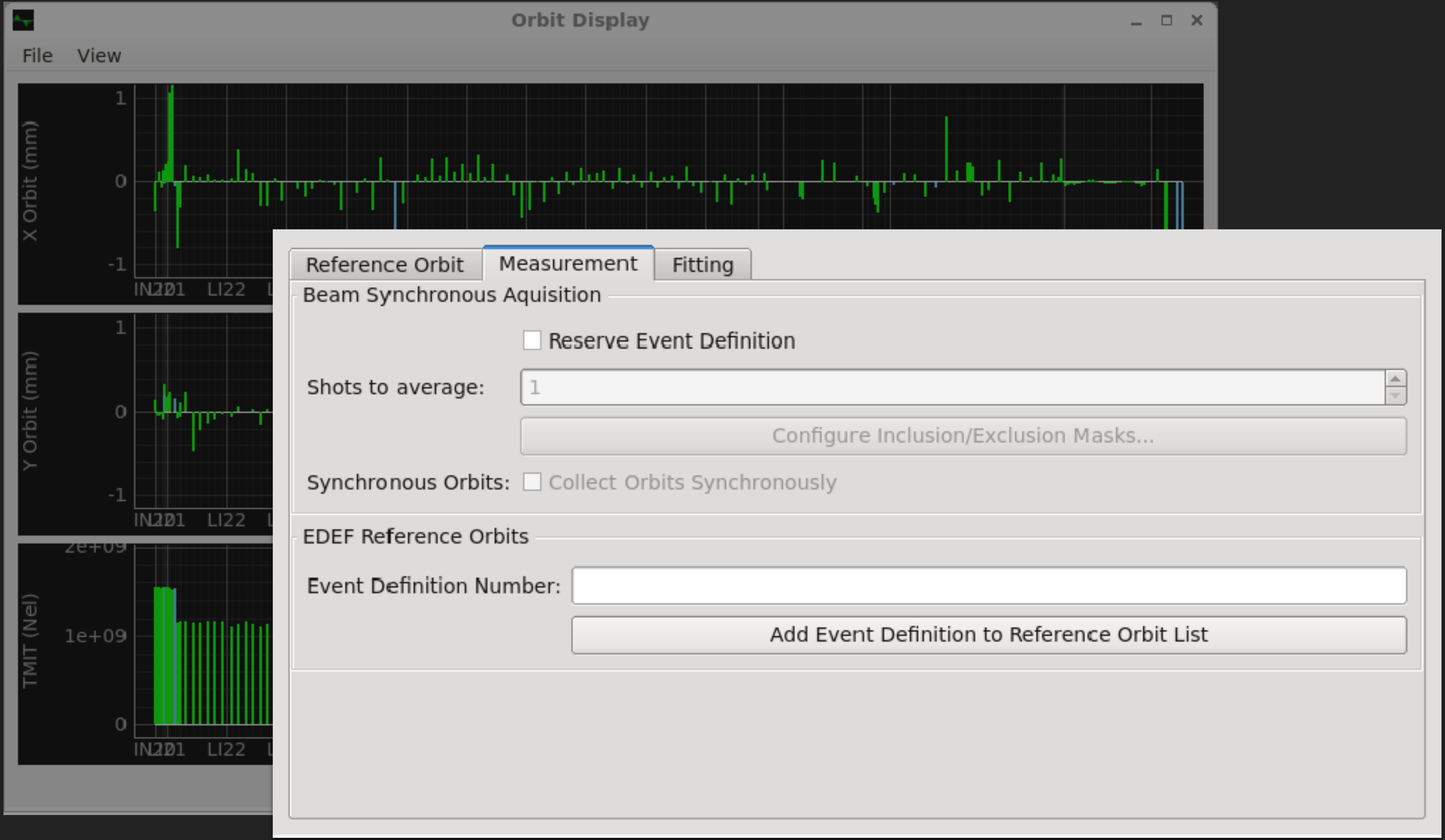
○ Difference between Live Orbit and 2: (Sep 28 2018 13

○ Difference between Live Orbi and 2: (Sep 2, with relative TMIT

Reference Orbits:

Live Orbit
2: (Sep 28 2018 13:32:04)

ORBIT DISPLAY



Beam Synchronous Aquisition

☐ Reserve Event Definition

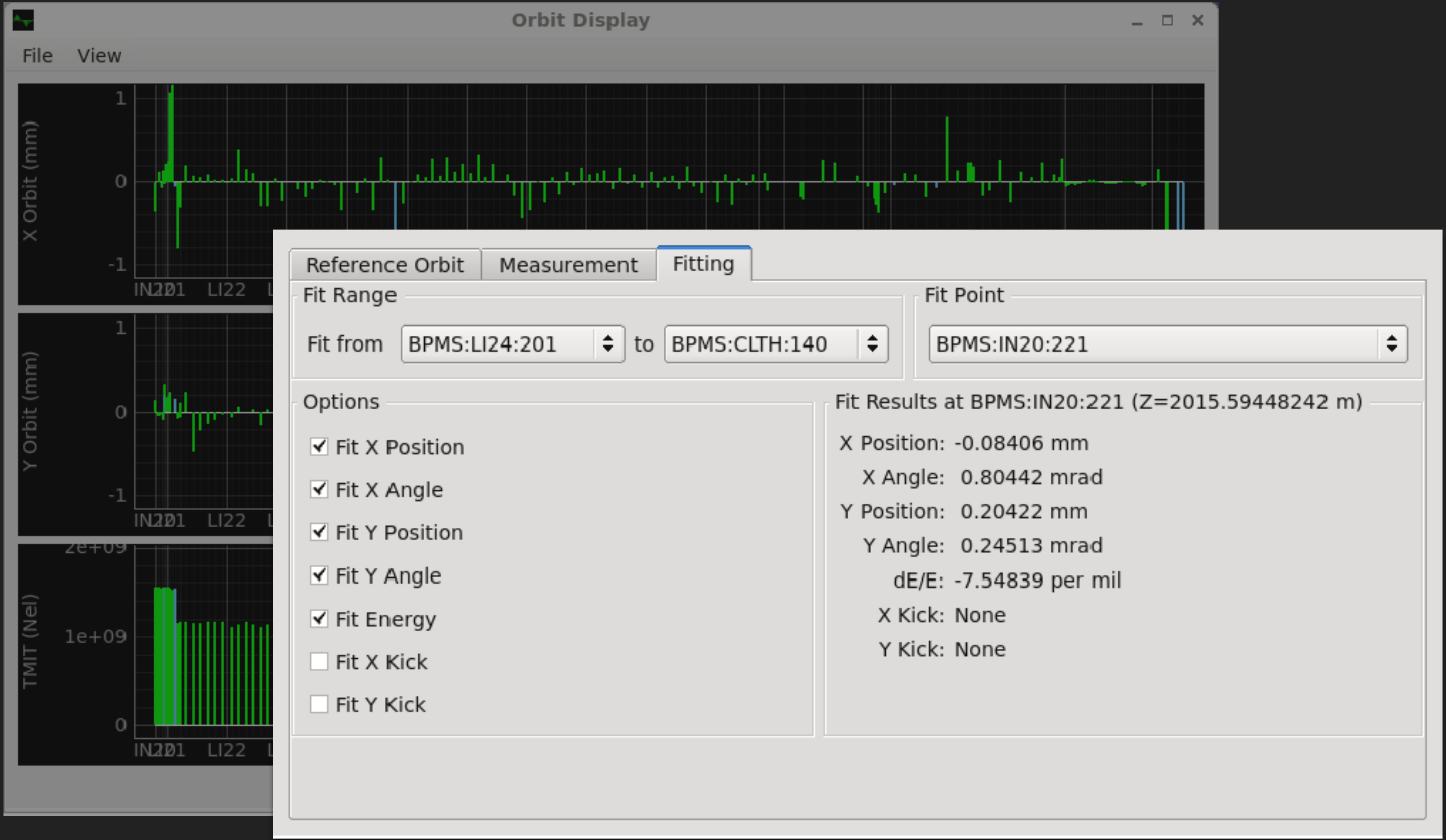
Shots to average: 1

Synchronous Orbits: ☐ Collect Orbits Synchronously

EDEF Reference Orbits

Event Definition Number:

ORBIT DISPLAY



Fit Range

Fit from BPMS:LI24:201 to BPMS:CLTH:140

Fit Point

BPMS:IN20:221

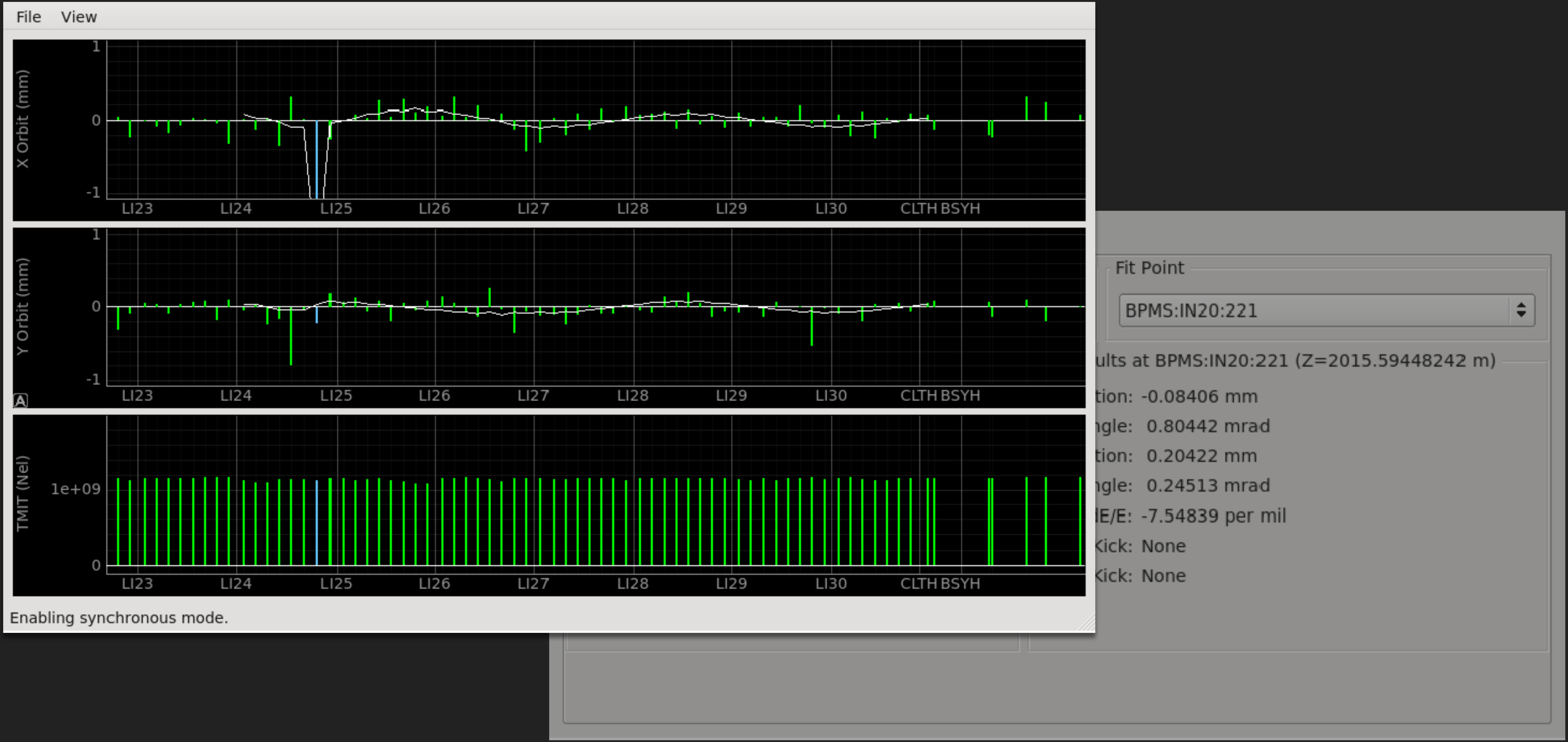
Options

- ☑ Fit X Position
- ☑ Fit X Angle
- ☑ Fit Y Position
- ☑ Fit Y Angle
- ☑ Fit Energy
- ☐ Fit X Kick
- ☐ Fit Y Kick

Fit Results at BPMS:IN20:221 (Z=2015.59448242 m)

```
X Position: -0.08406 mm
   X Angle:  0.80442 mrad
Y Position:  0.20422 mm
   Y Angle:  0.24513 mrad
      dE/E: -7.54839 per mil
    X Kick: None
    Y Kick: None
```

ORBIT DISPLAY



Enabling synchronous mode.

Fit Point

BPMS:IN20:221

ults at BPMS:IN20:221 (Z=2015.59448242 m)

tion: -0.08406 mm

ngle: 0.80442 mrad

tion: 0.20422 mm

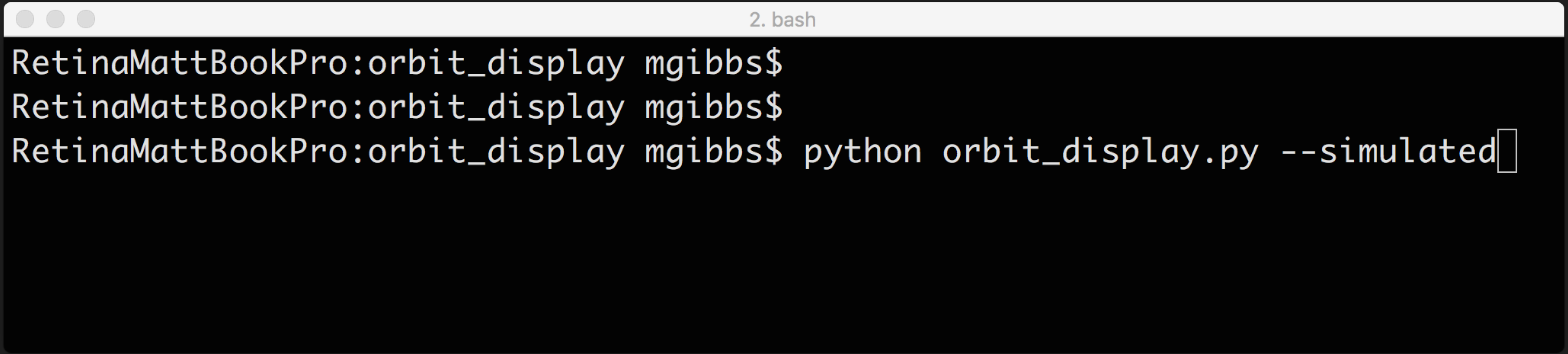
ngle: 0.24513 mrad

lE/E: -7.54839 per mil

Kick: None

Kick: None

SIMULATOR MODE

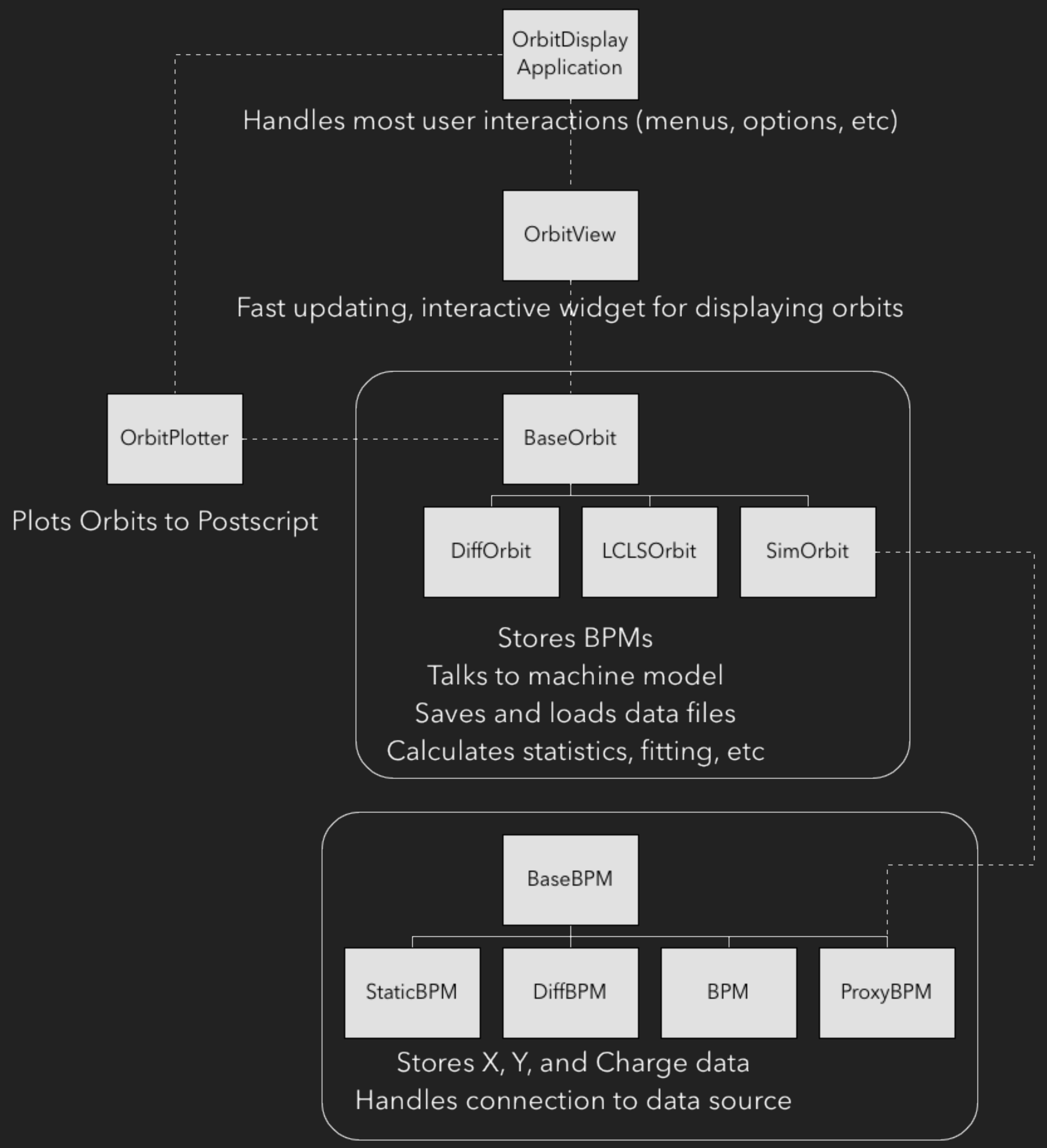


```
RetinaMattBookPro:orbit_display mgibbs$
RetinaMattBookPro:orbit_display mgibbs$
RetinaMattBookPro:orbit_display mgibbs$ python orbit_display.py --simulated
```

- Developing while using live data is not always possible
- No beam = No testing
- Simulator mode uses a 'SimOrbit' class to generate realistic fake data

SIMULATED ORBIT

- SimOrbit is just another BaseOrbit implementation
- No changes to the application or OrbitView necessary, besides adding the command line flag
- “ProxyBPM” lets any NumPy array be the data source


OrbitDisplay Application
Handles most user interactions (menus, options, etc)
OrbitView
Fast updating, interactive widget for displaying orbits
OrbitPlotter
Plots Orbits to Postscript
BaseOrbit
DiffOrbit
LCLSOrbit
SimOrbit
Stores BPMs
Talks to machine model
Saves and loads data files
Calculates statistics, fitting, etc
BaseBPM
StaticBPM
DiffBPM
BPM
ProxyBPM
Stores X, Y, and Charge data
Handles connection to data source

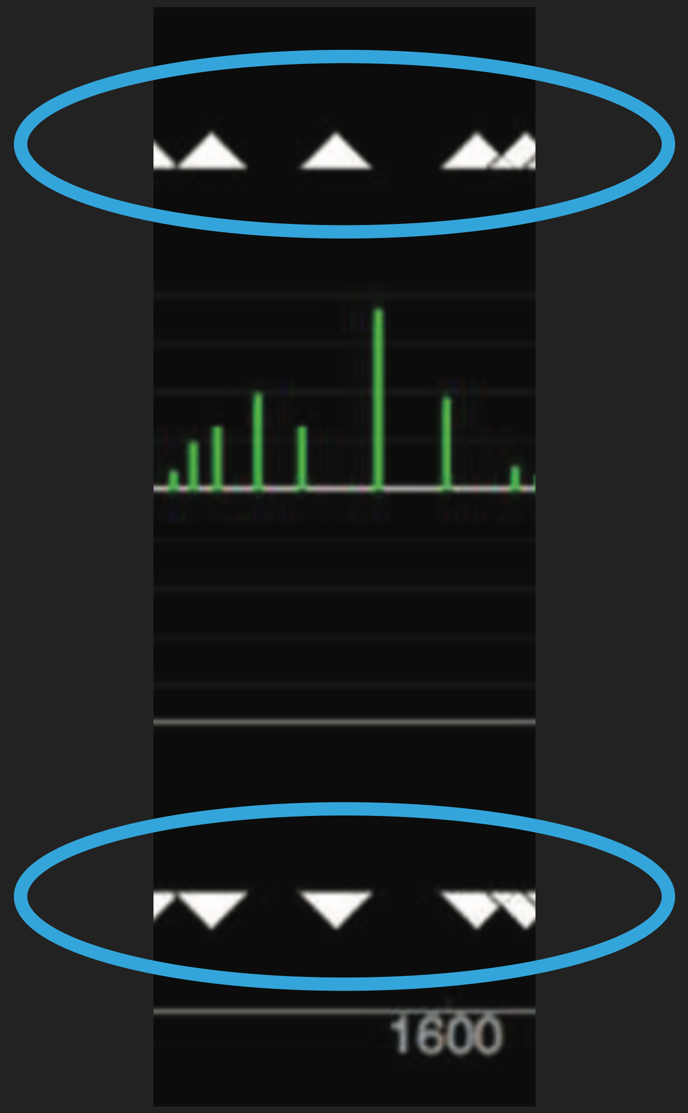
MODIFICATIONS FOR XFEL

- During commissioning, a quick way to steer the beam by hand was needed
- New concepts needed for XFEL version:
  - Multiple beam paths
  - Sub-trains
  - Corrector magnet control overlay
  - DOOCS, not EPICS
  - Two data sources: BPM Front-Ends, and Orbit Middle Layer Service

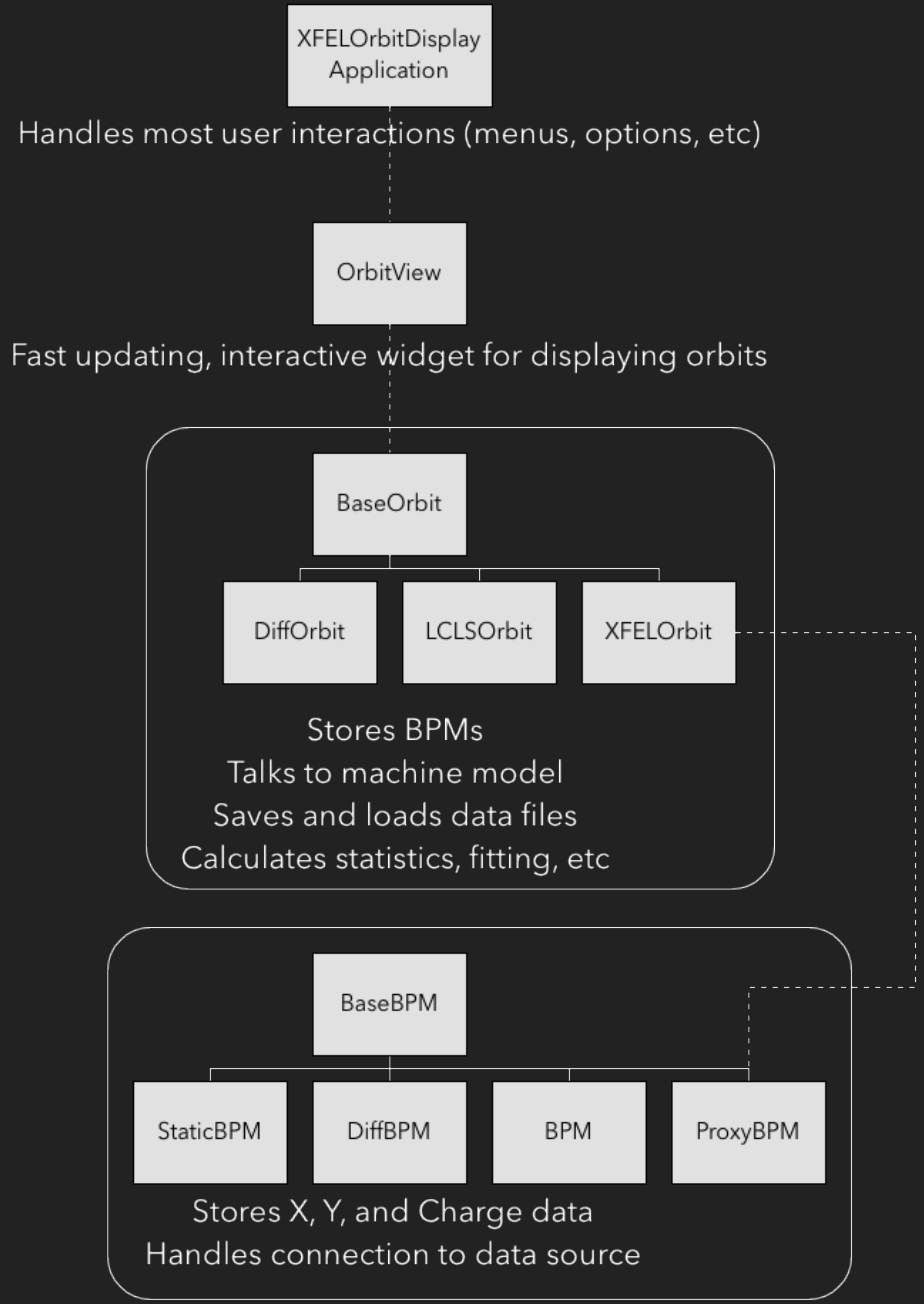
Photo by Dirk Noelle

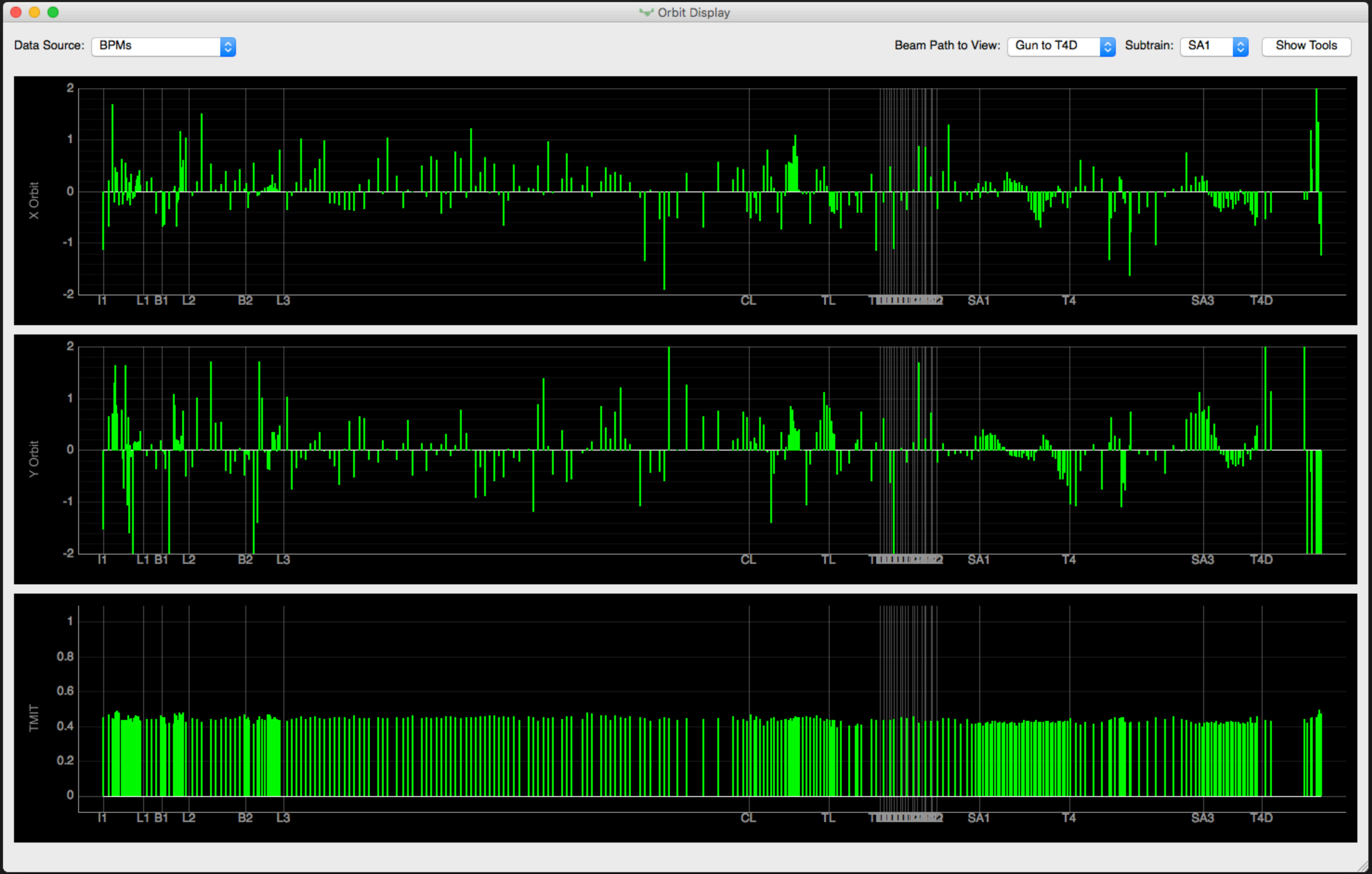
ORBIT DISPLAY

- XFELOrbitDisplay adds widgets for selecting beam destination, sub-train to view, and corrector magnet overlay
- XFELOrbit implements data collection from DOOCS, where it is efficient to get all orbit data as a NumPy array in one call, so ProxyBPM is used again



Steering controls overlaid at z-position of each magnet


XFELOrbitDisplay Application
Handles most user interactions (menus, options, etc)
OrbitView
Fast updating, interactive widget for displaying orbits
BaseOrbit
DiffOrbit
LCLSOrbit
XFELOrbit
Stores BPMs
Talks to machine model
Saves and loads data files
Calculates statistics, fitting, etc
BaseBPM
StaticBPM
DiffBPM
BPM
ProxyBPM
Stores X, Y, and Charge data
Handles connection to data source



Data Source: BPMs

Beam Path to View: Gun to T4D

Subtrain: SA1

CONCLUSION

- Python, PyQt, and PyQtGraph make a good platform for writing very fast interactive accelerator software
- Thinking about the architecture ahead of time pays off - modifications down the road are much easier

THANKS!

QUESTIONS?