# Matlab Middlelayer at Spear3, ALS, Soleil and other Light Sources


Greg Portmann (ALS / Spear3)
Jeff Corbett, Xiaobiao Huang, James SaFranek, Jim Sebek, Andrei Terebilo (Spear3)
Dave Robin and Christoph Steier (ALS)
Laurent Nadolski (Soleil)
Eugene Tan (ASP)
and many others



U.S. DEPARTMENT OF ENERGY | Office of Science

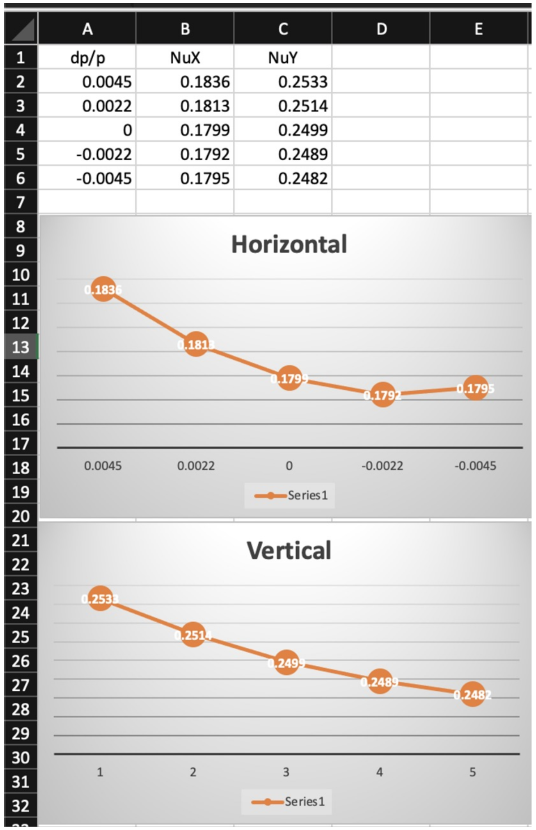
BERKELEY LAB

ALS

October 4, 2023

# I’ve been writing Matlab code for accelerators since about 1993. Why?

- I saw chromaticity measurements being done in the control room in Excel.
- There’s no way we’re going to implement SVD orbit correction, … in Excel. C or C++ was an option but Matlab’s active workspace is ideal for shift work.
  - I believe our first study using the original MML was beam base alignment with Dave Robin.



| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | dp/p | NuX | NuY | | |
| 2 | 0.0045 | 0.1836 | 0.2533 | | |
| 3 | 0.0022 | 0.1813 | 0.2514 | | |
| 4 | 0 | 0.1799 | 0.2499 | | |
| 5 | -0.0022 | 0.1792 | 0.2489 | | |
| 6 | -0.0045 | 0.1795 | 0.2482 | | |

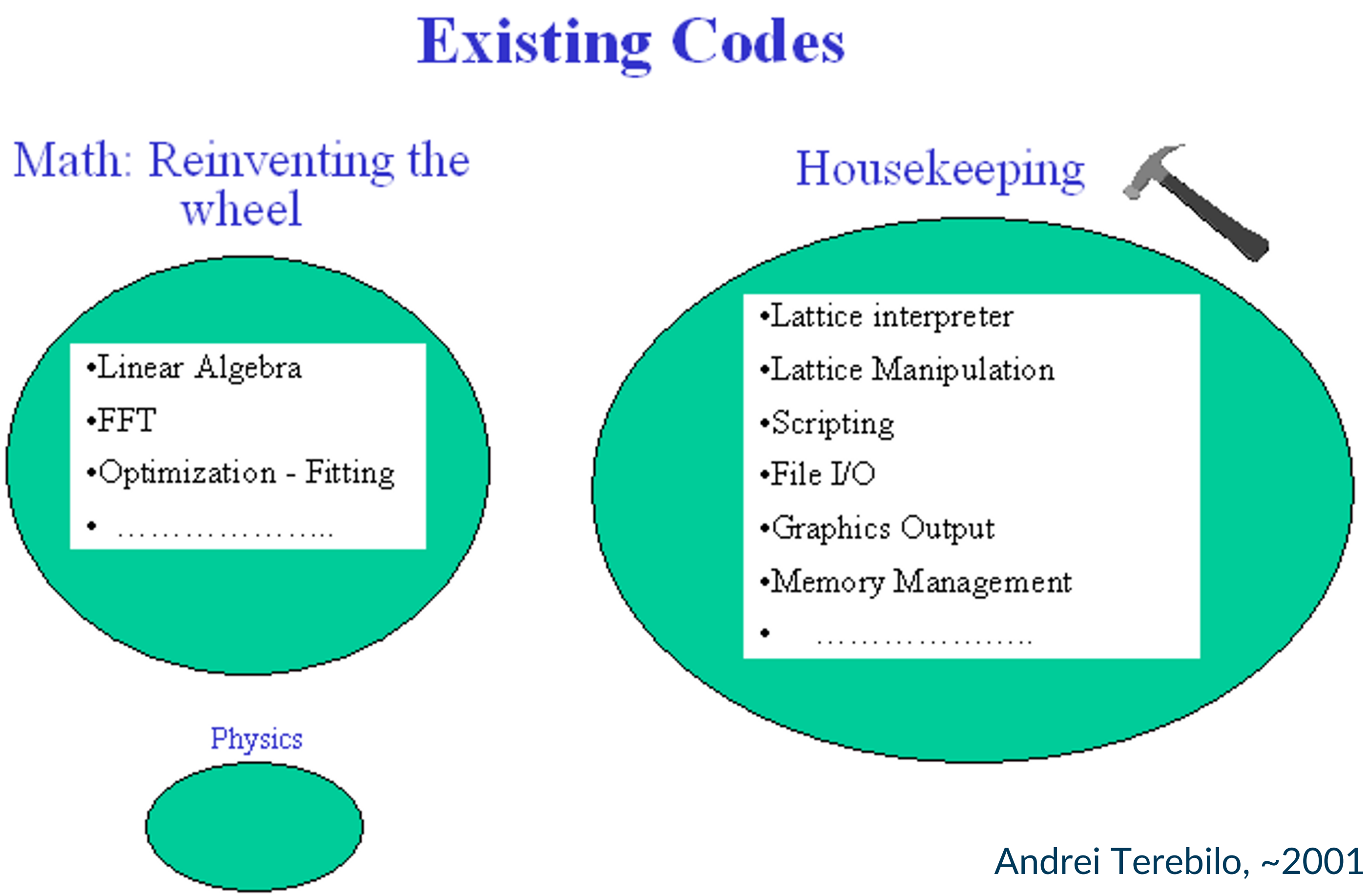
# MML Brief History

- **Started about 1993 at the ALS to automate physics experiments (orbit, tune, chromaticity correction, quadrupole centering, …)**

- **We started using Matlab code during ALS operations out of necessity, and we never stopped**
  - Lack of the people working in high level controls
  - The motivation to change working Matlab code is low and it comes with risk.

- **Spear3 commissioning effort (early 2000s)**
  - Started with the ALS Matlab code and completely rewrote it to be accelerator independent (as much as we could)
  - Andrei Terebilo had written the Matlab tracking code AT
  - LOCO was ported from Fortran to Matlab & AT

# AT – Accelerator Toolbox
# Why did Andrei Chose Matlab?




Andrei Terebilo, ~2001

# The Matlab Toolbox Suite for accelerator physics and commissioning developed in the 2001 – 2004 timeframe for Spear3.

- MiddleLayer + High Level Applications (MML)
  1. Link between applications and the control system (EPICS, Tango, etc.) or simulator
  2. Functions to access accelerator data
  3. Provide a physics function library
- AT – Accelerator Toolbox for simulations
- LOCO – Linear Optics from Closed Orbits (Lattice calibration, etc.)

# Some of the people … a while back.

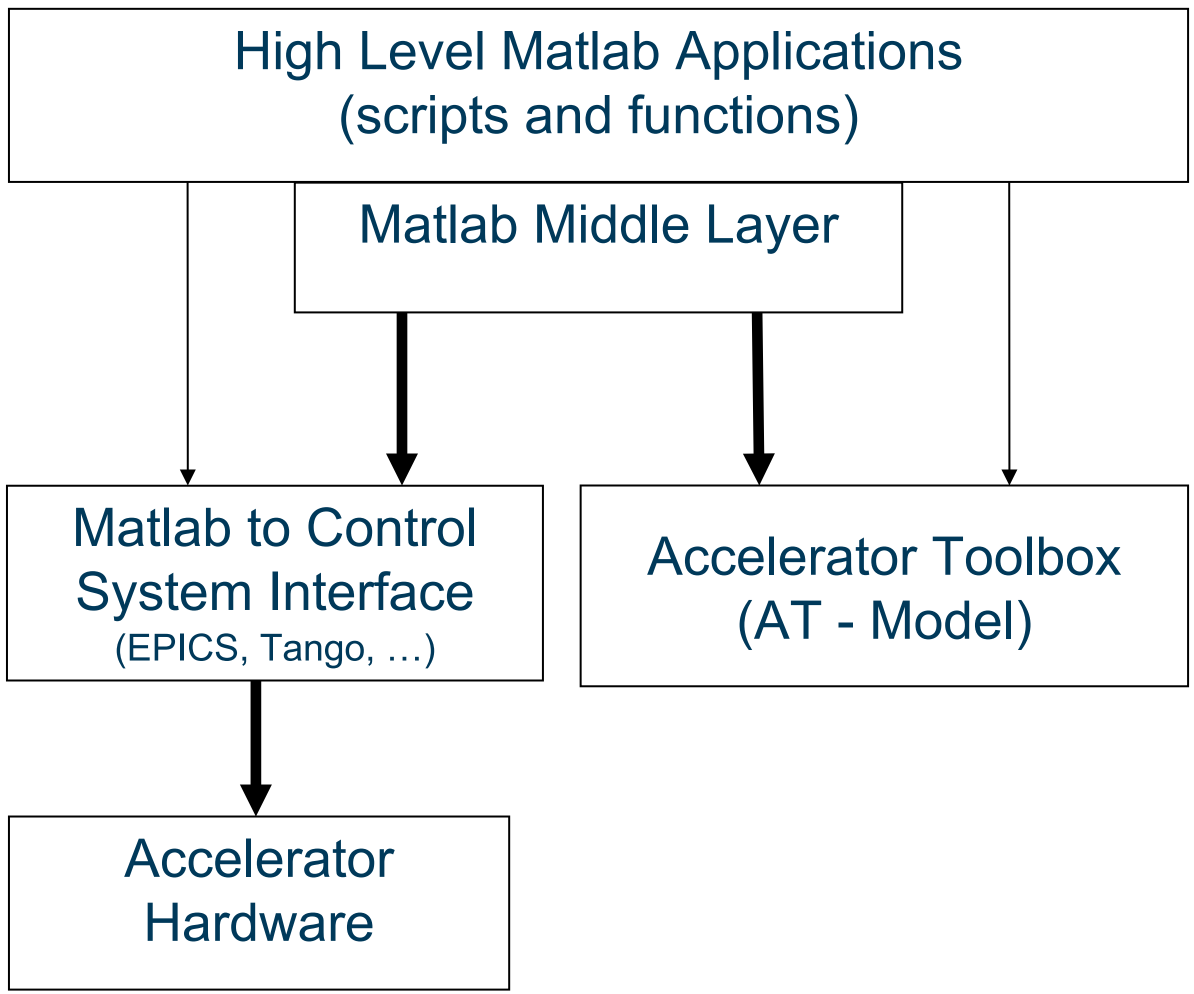
High concentration of MML and AT programmers at this ALS/Spear3 offsite in 2007
Including: Andrei Terebilo (AT)

Missing: the CLS control room

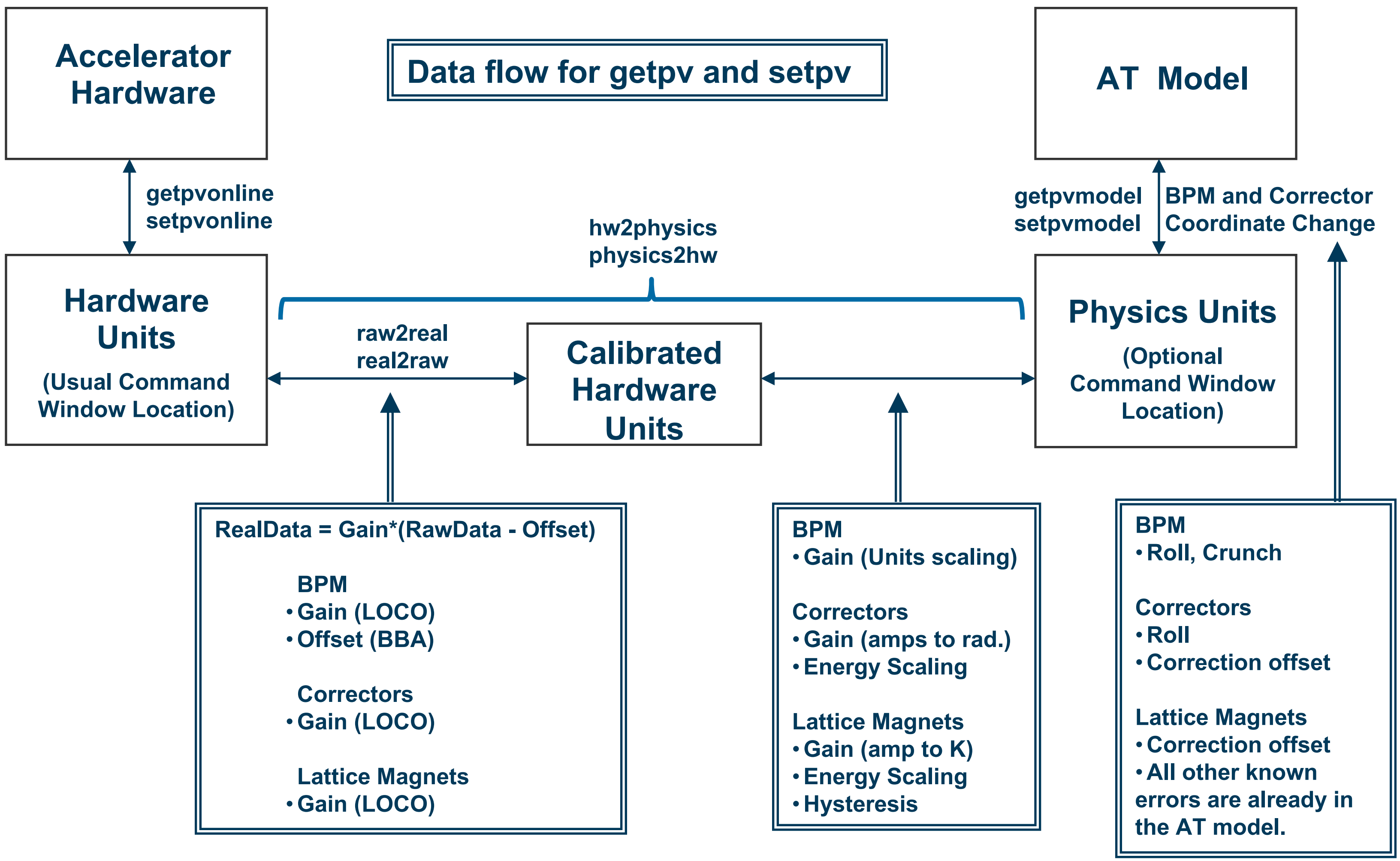
# Design Plan/Goals

- Accelerator Independent
  - Control system independent
    - Two main functions to change control systems
      - getpvonline, setpvonline
      - Minimize the need to know the control system or a channel name.
    - Work on EPICS, Tango, OPC, and other older control systems
  - AT model independent
- Tracking code independent
  - Build on AT. We planned to connect to other modeling code but it hasn't happened so far.
    - getpvmodel, setpvmodel
- Be flexible on where data comes from
  - getdata and getrespmat can branch between Matlab data, model data, or data from a server.
- Nothing fancy, keep it simple, …
  - Clean, readable, commented code
  - No objects (just expose the structures)
    - Although, mml objects are available
  - The target audience was non-professional programmers

# Software Interconnection Diagram



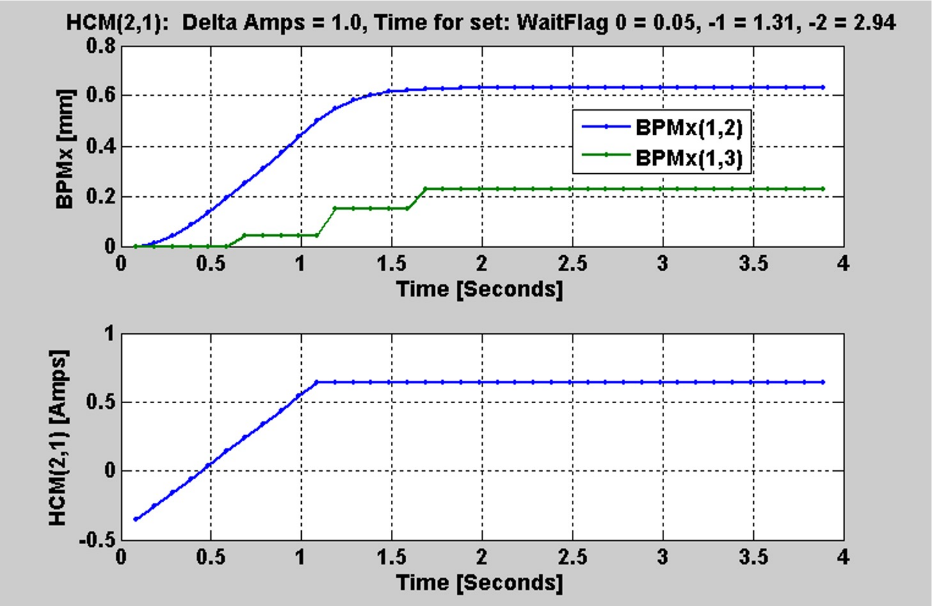
# Middlelayer Data Flow Diagram



# Challenges

- **The most difficult step was to make it accelerator independent.  But it greatly improved the code.**
  - Needed to be very flexible on units and conversions (hw2physics and physics2hw)
  - Forces abstraction instead of hardcoding.
  - Needed the .MemberOf field to know what the families were.
    - X, Y, BPMx, BPMy, BPMz, …
    - HCM, HCOR, HC, XCOR, …
  - …

- **Timing setpoint change and data taking**
  - WaitFlags in setpv.
    Knowing when the setpoint change is complete and when monitors, like orbit and tune data, are ready to read.




- **Data management**

# Data, Data, and more Data

- Beam Position Monitors
  - Channel names, gains, roll, crunch, offsets, golden, standard deviations
- Magnets
  - Channel names, gains, offsets, roll, setpoint-monitor tolerance, amp-to-simulator conversions, hysteresis loops, max/min setpoint
- Response matrices (Orbit, Tune, Chromaticity)
- Lattices (Save and restore)
- Measurement archiving
  - Dispersion, tunes, chromaticity, quadrupole centers
  - …
- …

Management of data within Matlab is easy.
Sharing data with other applications is not so easy.

# MML Setup Data Structures (1)

```
>> ao = getao;
>> ao.HCM
    FamilyName: 'HCM'
    MemberOf: {3×1 cell}
    DeviceList: [98×2 double]
    ElementList: [98×1 double]
    Status: [98×1 double]
    Position: [98×1 double]
    BaseName: {98×1 cell}
    DeviceType: {98×1 cell}
    CommonNames: [98×11 char]
    Monitor: [1×1 struct]
    Setpoint: [1×1 struct]
    Trim: [1×1 struct]
    FF1: [1×1 struct]
```

```
FF2: [1×1 struct]
FFMultiplier: [1×1 struct]
Sum: [1×1 struct]
DAC: [1×1 struct]
RampRate: [1×1 struct]
TimeConstant: [1×1 struct]
OnControl: [1×1 struct]
On: [1×1 struct]
Reset: [1×1 struct]
Ready: [1×1 struct]
AT: [1×1 struct]
Gain: [98×1 double]
Roll: [98×1 double]
```

# MML Setup Data Structures (2)

```
>> ao.HCM.Setpoint
  MemberOf: {9×1 cell}
  Mode: 'Simulator'
  DataType: 'Scalar'
  ChannelNames: [98×19 char]
  HW2PhysicsFcn: @amp2k
  Physics2HWFcn: @k2amp
  Units: 'Hardware'
  HWUnits: 'Ampere'
  PhysicsUnits: 'Radian'
  RunFlagFcn: @getrunflagcm
  Range: [98×2 double]
  Tolerance: [98×1 double]
  DeltaRespMat: [98×1 double]
```

```
>> ao.HCM.Setpoint.MemberOf
    {'PlotFamily'  }
    {'Save/Restore'}
    {'COR'         }
    {'Horizontal'  }
    {'HCM'         }
    {'Magnet'      }
    {'Setpoint'    }
    {'measbpmresp' }
    {'Archive'     }
```

Typically only one or two people at a facility maintain the MML setup data structures.

# MML Data Structures

## Basic Data Structure

```
>> getpv('HCM', 'Monitor', 'Struct')
              Data: [98×1 double]
        FamilyName: 'HCM'
             Field: 'Monitor'
        DeviceList: [98×2 double]
            Status: [98×1 double]
              Mode: 'Online'
             Units: 'Hardware'
       UnitsString: 'Ampere'
    DataDescriptor: 'Get by FamilyName'
         CreatedBy: 'getpv'
          DataTime: [98×1 double]
         TimeStamp: [2023 10 8 16 55 19.8041]
```

## Response Matrix Data Structure

```
>> r = getbpmresp('struct');
>> r(1,1)
               Data: [122×98 double]
            Monitor: [1×1 struct]
           Actuator: [1×1 struct]
      ActuatorDelta: [98×1 double]
           Monitor1: [122×98 double]
           Monitor2: [122×98 double]
              Units: 'Hardware'
        UnitsString: 'mm/Ampere'
                GeV: 1.8909
          TimeStamp: [2023 10 4 16 22 58.9844]
               DCCT: 55.7607
   ModulationMethod: 'bipolar'
           WaitFlag: -2
         ExtraDelay: 0
     DataDescriptor: 'Response Matrix'
          CreatedBy: 'measrespmat'
    OperationalMode: 'Pseudo-Single Bunch (0.18,0.25)'
           FileName:
'/home/als/physbase/mmlt/machine/ALS/StorageRingOpsData/PseudoSingleBunch/GoldenBPMResp_LowEmittance'
```

# I. Basic Calling Syntax (the middelayer part)

**Naming Convention**

Family = Group descriptor (text string)

Field = Subgroup descriptor (text string)

DeviceList = [Sector Element-in-Sector]

**Basic Functions**

```
getpv(Family, Field, DeviceList);
setpv(Family, Field, Value, DeviceList);
steppv(Family, Field, Value, DeviceList);
```

These functions can branch between the model and online.

**Examples:**

```
x = getpv('BPMx', 'Monitor', [3 4;5 2]);
h = getpv('HCM', 'Setpoint', [2 1;12 4]);
setpv('QF', 'Setpoint', 81);
```

# ALS Naming Scheme

## Families

Bend magnets – BEND
Quadrupoles – QF, QD, QFA, QDA
Sextupoles – SF, SD
Skew quadrupoles - SQSF, SQSD
Correctors – HCM, VCM
Beam position monitors – BPMx and BPMy
Insertion devices – ID, EPU
Other - RF, DCCT, TUNE, Energy

## Fields

Setpoint, Monitor, RampRate, RunFlag, DAC, OnOff, Reset, Ready, Voltage, Power, Velocity, HallProbe, etc…

# Name Server: EPICS Channel Finder

At the ALS, the channel finder service tags channels using the same "Accelerator, Family, Field, Device" scheme as in the MML.

Controls Main × | Ops Launcher × | ALS Launcher × | Linac × | MOD1 × | MOD2 × | miniDCCT × | EPBI × | Channel Table × | Channel Table ×

Query: * Acc=SR Family=QF Field=Setpoint Sector=* Device=*

| name ▲ | Acc | Family | Field | Sector | Device | Position | Golden | archive | recordType | iocName |
|---|---|---|---|---|---|---|---|---|---|---|
| SR01C___QF1____AC02 | SR | QF | Setpoint | 1 | 1 | 3.378696 | 96.908063344168553 | Fast | ao | ffbsec01 |
| SR01C___QF2____AC03 | SR | QF | Setpoint | 1 | 2 | 12.677306 | 96.548511743507575 | Fast | ao | ffbsec01 |
| SR02C___QF1____AC02 | SR | QF | Setpoint | 2 | 1 | 19.778696 | 95.675469420609659 | Fast | ao | ffbsec02 |
| SR02C___QF2____AC03 | SR | QF | Setpoint | 2 | 2 | 29.077306 | 96.25629257026965 | Fast | ao | ffbsec02 |
| SR03C___QF1____AC02 | SR | QF | Setpoint | 3 | 1 | 36.178696 | 96.254993755862387 | Fast | ao | ffbsec03 |
| SR03C___QF2____AC03 | SR | QF | Setpoint | 3 | 2 | 45.477306 | 94.968993154771908 | Fast | ao | ffbsec03 |
| SR04C___QF1____AC02 | SR | QF | Setpoint | 4 | 1 | 52.578696 | 97.362668560344588 | Fast | ao | ffbsec04 |
| SR04C___QF2____AC03 | SR | QF | Setpoint | 4 | 2 | 61.879111 | 96.997481310809263 | Fast | ao | ffbsec04 |
| SR05C___QF1____AC02 | SR | QF | Setpoint | 5 | 1 | 68.980501 | 95.703886987555819 | Fast | ao | ffbsec05 |
| SR05C___QF2____AC03 | SR | QF | Setpoint | 5 | 2 | 78.279111 | 95.288593169344963 | Fast | ao | ffbsec05 |
| SR06C___QF1____AC02 | SR | QF | Setpoint | 6 | 1 | 85.380501 | 96.447573180643431 | Fast | ao | ffbsec06 |
| SR06C___QF2____AC03 | SR | QF | Setpoint | 6 | 2 | 94.679111 | 96.5060204101957 | Fast | ao | ffbsec06 |
| SR07C___QF1____AC02 | SR | QF | Setpoint | 7 | 1 | 101.780501 | 95.744347344953354 | Fast | ao | ffbsec07 |
| SR07C___QF2____AC03 | SR | QF | Setpoint | 7 | 2 | 111.079111 | 96.566816850643619 | Fast | ao | ffbsec07 |
| SR08C___QF1____AC02 | SR | QF | Setpoint | 8 | 1 | 118.180501 | 96.1579998863688 | Fast | ao | ffbsec08 |
| SR08C___QF2____AC03 | SR | QF | Setpoint | 8 | 2 | 127.480916 | 97.18719441846909 | Fast | ao | ffbsec08 |
| SR09C___QF1____AC02 | SR | QF | Setpoint | 9 | 1 | 134.582305 | 95.3480583970734 | Fast | ao | ffbsec09 |
| SR09C___QF2____AC03 | SR | QF | Setpoint | 9 | 2 | 143.880916 | 96.182497070246868 | Fast | ao | ffbsec09 |
| SR10C___QF1____AC02 | SR | QF | Setpoint | 10 | 1 | 150.982306 | 96.266783582685619 | Fast | ao | ffbsec10 |
| SR10C___QF2____AC03 | SR | QF | Setpoint | 10 | 2 | 160.280916 | 95.861797805405516 | Fast | ao | ffbsec10 |
| SR11C___QF2____AC03 | SR | QF | Setpoint | 11 | 2 | 176.680916 | 95.543268540128 | Fast | ao | ffbsec11 |
| SR11C:QF1:Setpoint | SR | QF | Setpoint | 11 | 1 | 167.382306 | 95.775254811217167 | Fast | ao | genesys |
| SR12C___QF1____AC02 | SR | QF | Setpoint | 12 | 1 | 183.782306 | 96.484200063890142 | Fast | ao | ffbsec12 |
| SR12C___QF2____AC03 | SR | QF | Setpoint | 12 | 2 | 193.082721 | 106.21448885069287 | Fast | ao | ffbsec12 |

# Scripting Example: Orbit Correction in 6 lines

```
% Get the horizontal response matrix
Rx = getrespmat('BPMx', 'HCM');      % 122x94 matrix at ALS

% Computes the SVD of the response matrix
Ivec = 1:48;
[U, S, V] = svd(Rx, 0);


% Get the vertical orbit
X = getpv('BPMx');

% Find the corrector changes
DeltaAmps = -V(:,Ivec) * S(Ivec,Ivec)^-1 * U(:,Ivec)' *  X;

% Changes the corrector strengths
steppv('HCM', 'Setpoint', DeltaAmps);
```

Add a loop => Slow Orbit feedback

# Controls & Instrumentation

Supported Systems and Devices

- Fast magnets
- Power supplies
- Vacuum systems
- Diagnostic systems
  - Beam Position Monitors (BPMs), fast orbit feedback
  - Scopes
  - Scintillators (fluorescent screens) and CCD cameras
  - Photon Beam Position Monitors (PBPMs),
  - Beam current (DCCT, ICT)
  - Bunch current monitor
  - Beam loss monitors
  - Vacuum chamber/girder/floor motion monitors
  - …
- RF systems
- Machine protection systems (MPS)
- Facilities data (temperatures, …)
- Control room to support operations and physics
- …

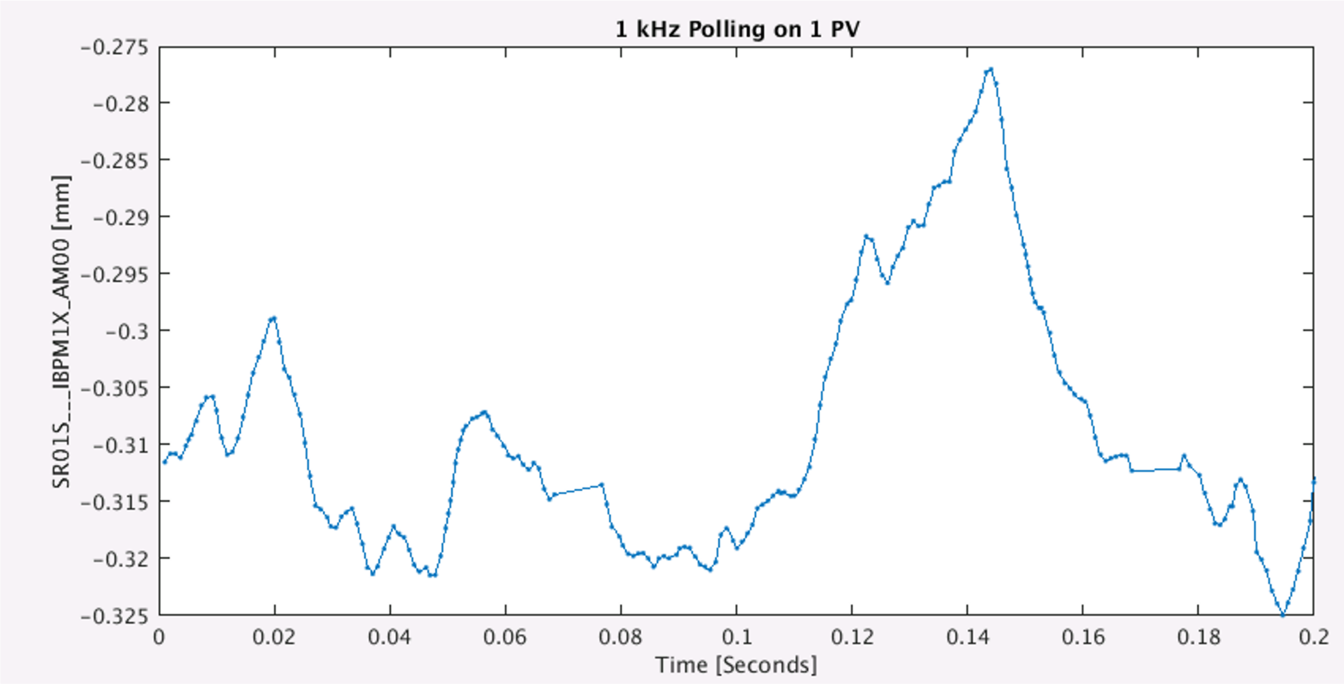
# II. Function Library

There are hundreds of functions for accelerator control

- setorbit – general purpose global orbit correction function
- setorbitbump – general purpose local bump function
- settune – sets the storage ring tune
- setchro – sets the storage ring chromaticity
- measchro – measure the chromaticity
- measdisp – measure the dispersion function
- quadcenter, quadplot – finds the quadrupole center
- physcis2hw – converts between physics and hardware units
- measbpmresp – measure a BPM response matrix
- measlifetime – computes the beam lifetime
- minpv/maxpv – min/max value for family/field
- srcycle – standardizes the storage ring magnets
- scantune – scan in tune space and record the lifetime
- scanaperture – scans the electron beam in the straight sections and monitors lifetime
- finddispquad – finds the setpoint that minimizes the dispersion in the straight sections.
- rmdisp – adjusts the RF frequency to remove the dispersion component of the orbit by fitting the orbit to the dispersion orbit
- Etc. (thousands of scripts and functions have been written)

# Matlab - LabCA Polling Speed

1. One PV -> 1 kHz polling (4 kHz is about the maximum rate)
   x = getpv('BPMx', 'Monitor', [1 1], 0:.001:.2);



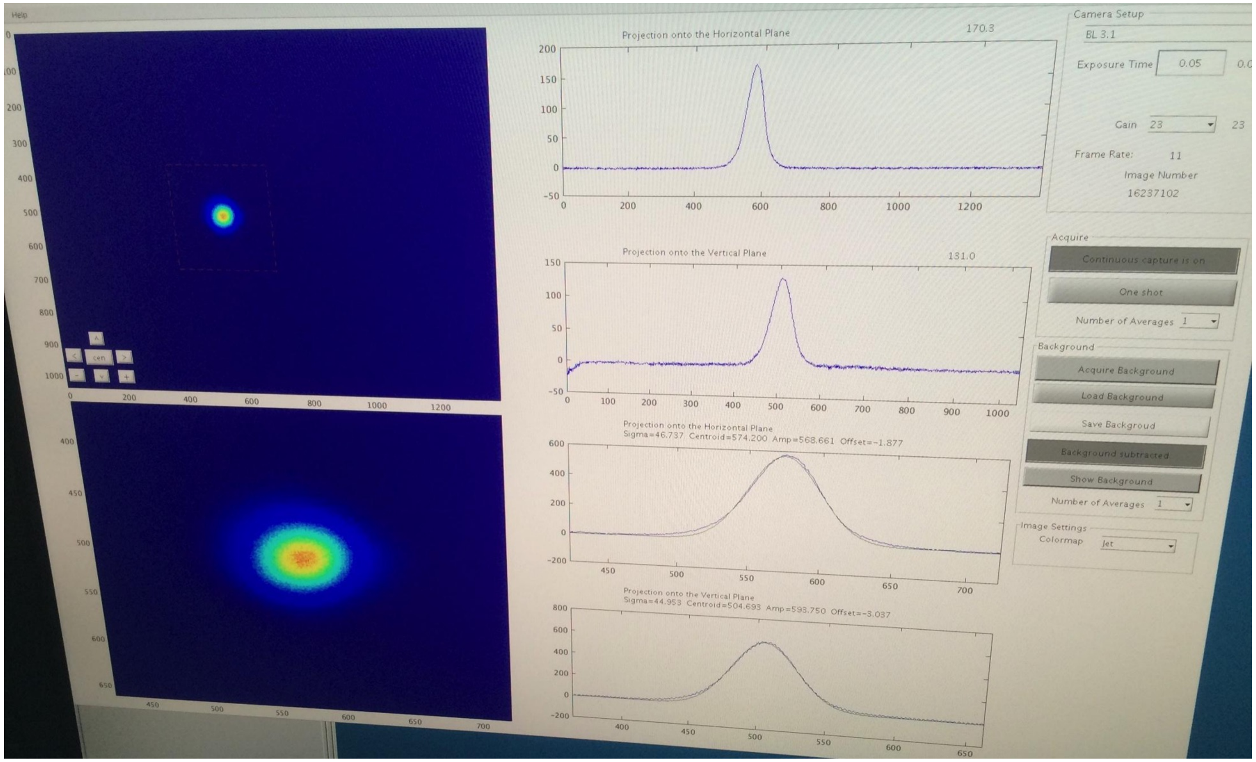

2. 120 PVs -> 1 kHz polling is about the maximum but there is some variation in the sampling time.
   x = getpv('BPMx', 'Monitor', [], 0:.001:.2);

Note: these numbers improve every year.

# Matlab Waveform Speed in EPICS

## BL 3.1 CCD Camera

```
>> tic; Image = getpv('BL31:image1:ArrayData', 'native', 1038*1390); toc
   Elapsed time is 0.08 seconds (12.5 Hz)
  1038*1390*16 / 2^20 = 22.0 Mbits  (~1.44M points in .08 seconds)
```




## BPM

```
>> tic; X = getpv('BR2:BPM3:wfr:TBT:X','waveform'); toc;
   Elapsed time is 0.114679 seconds.   (2.1M points)
```

# MML Applications

# High Level Applications in Matlab

- Save/ restore / configuration control
- Orbit correction and slow orbit feedback
- Insertion device focusing compensation
- Quadrupole centering
- Display (plotfamily, mmlviewer)
- Transport line tuning
- CCD cameras
- Energy Ramping
- General scripting language for machine shifts
- LOCO (Response matrix analysis, machine calibration)
- …

-> Many applications can be run and optimized before the accelerator is build. It's highly recommended to test orbit correction, tune correction, quadrupole centering, and LOCO before finalizing the lattice design.

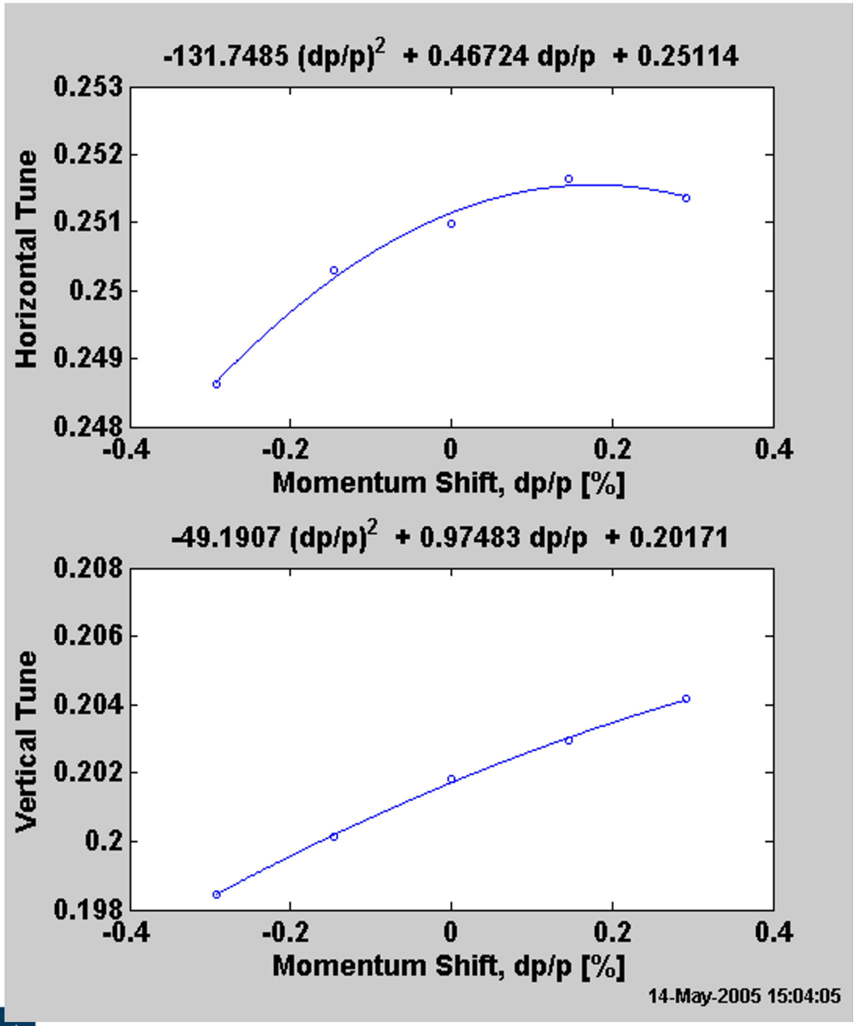
# plotfamily application


See also mmlviewer viewfamily

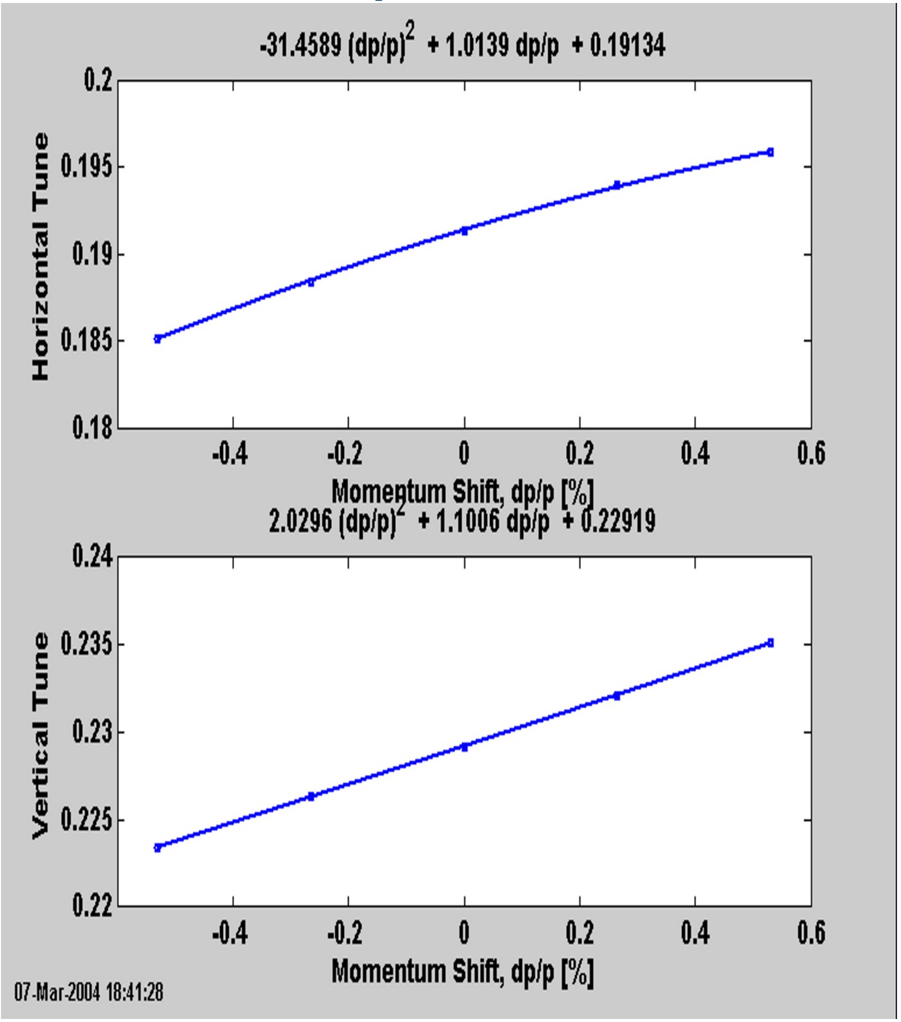
# Chromaticity Measurement

## Accelerator Independent

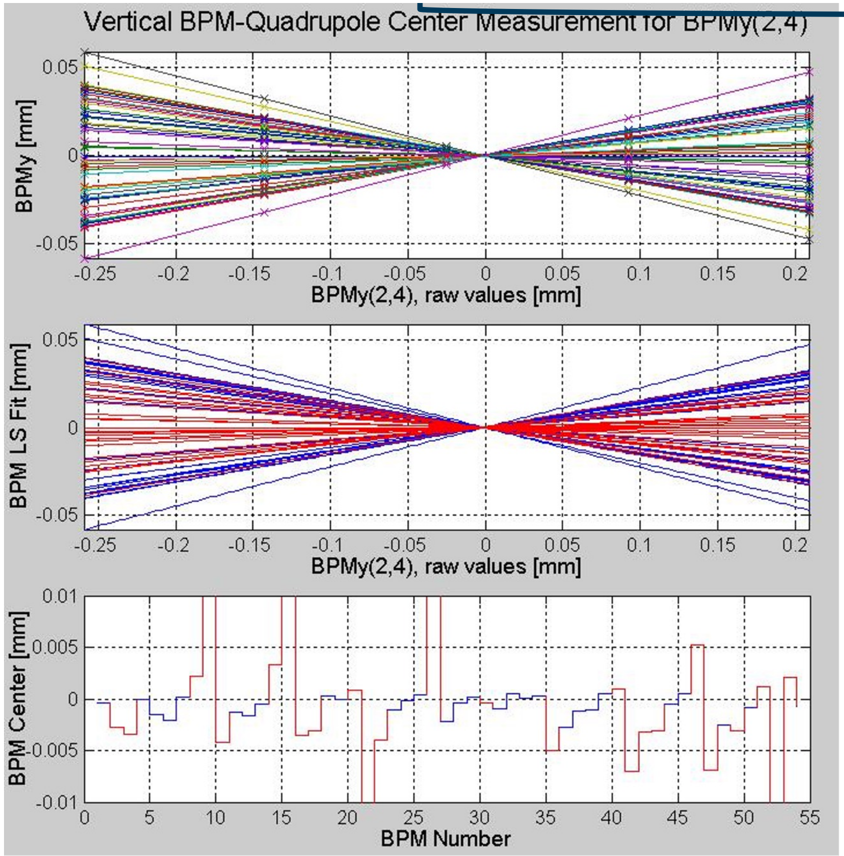
### ALS

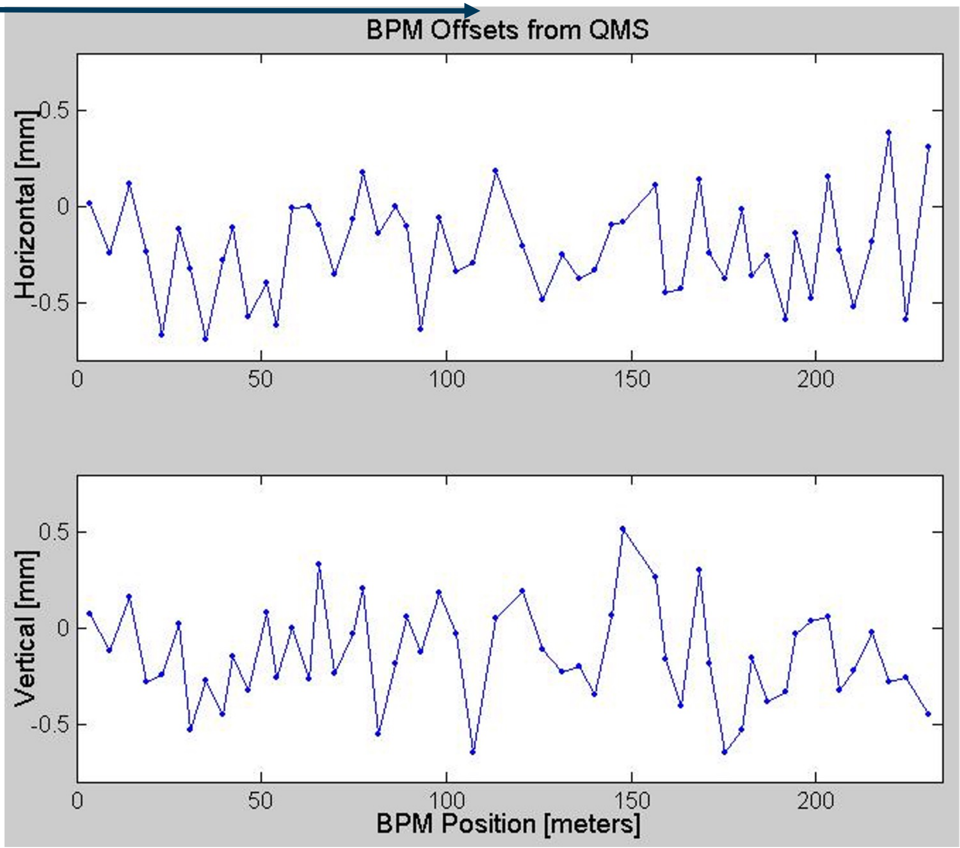



### Spear3

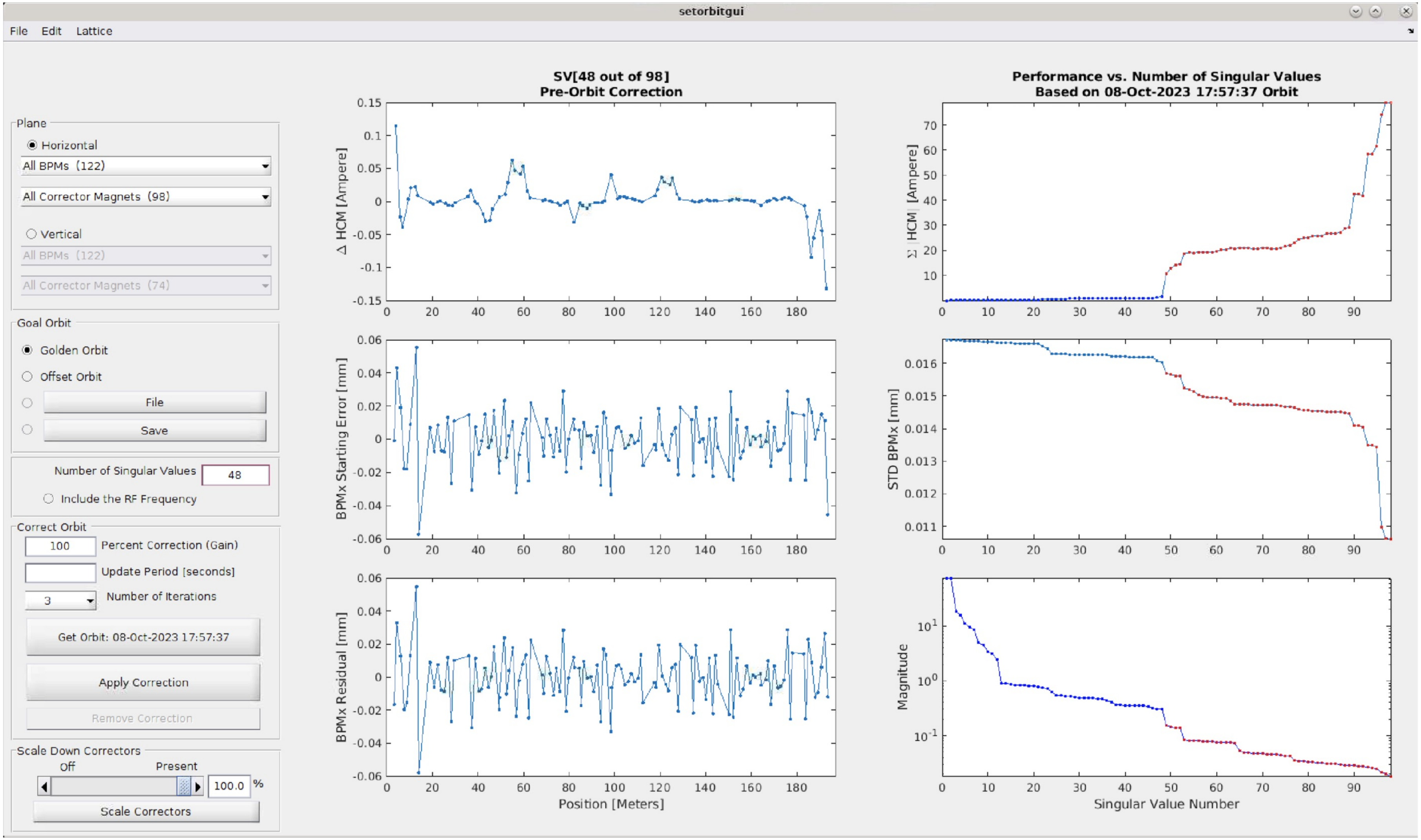


# Beam-based Alignment



# Orbit Correction



# Save/Restore

Saved values come from golden file: /home/als/physbase/mmlt/machine/ALS/StorageRingOpsData/PseudoSingleBunch/GoldenConfig_LowEmittance

## Restore Table

Update ( 3:35:53 PM)

| | Family | Field | Select |
|---|---|---|---|
| 1 | HCM | Setpoint | ☑ |
| 2 | VCM | Setpoint | ☑ |
| 3 | QF | Setpoint | ☑ |
| 4 | QD | Setpoint | ☑ |
| 5 | QFA | Setpoint | ☑ |
| 6 | QDA | Setpoint | ☑ |
| 7 | SF | Setpoint | ☑ |
| 8 | SD | Setpoint | ☑ |
| 9 | SHF | Setpoint | ☑ |
| 10 | SHD | Setpoint | ☑ |
| 11 | SQSF | Setpoint | ☑ |
| 12 | SQSF | RampRate | ☑ |
| 13 | SQSD | Setpoint | ☑ |
| 14 | SQSD | RampRate | ☑ |
| 15 | SQSHF | Setpoint | ☑ |
| 16 | BEND | Setpoint | ☑ |
| 17 | HCMCHICANE | Setpoint | ☑ |
| 18 | HCMCHICANEM | Setpoint | ☑ |
| 19 | TOPSCRAPER | Setpoint | ☑ |

Restores only the selected items

Restore Setpoints

Allow individual changes to the present column

Edit Mode is Off

All | None

## Saved Values for Viewing Only (No Restore)

| | Family | Field | Select |
|---|---|---|---|
| 1 | BPMx | Monitor | ☑ |
| 2 | BPMx | GoldenSetpoint | ☑ |
| 3 | BPMy | Monitor | ☑ |
| 4 | BPMy | GoldenSetpoint | ☑ |
| 5 | HCM | Monitor | ☑ |
| 6 | HCM | Trim | ☑ |
| 7 | HCM | FF1 | ☑ |
| 8 | HCM | FF2 | ☑ |
| 9 | HCM | DAC | ☑ |
| 10 | HCM | RampRate | ☑ |
| 11 | VCM | Monitor | ☑ |
| 12 | VCM | Trim | ☑ |
| 13 | VCM | FF1 | ☑ |
| 14 | VCM | FF2 | ☑ |
| 15 | VCM | DAC | ☑ |
| 16 | VCM | RampRate | ☑ |
| 17 | HCMFOFB | Trim | ☑ |
| 18 | HCMFOFB | FF1 | ☑ |
| 19 | HCMFOFB | FF2 | ☑ |

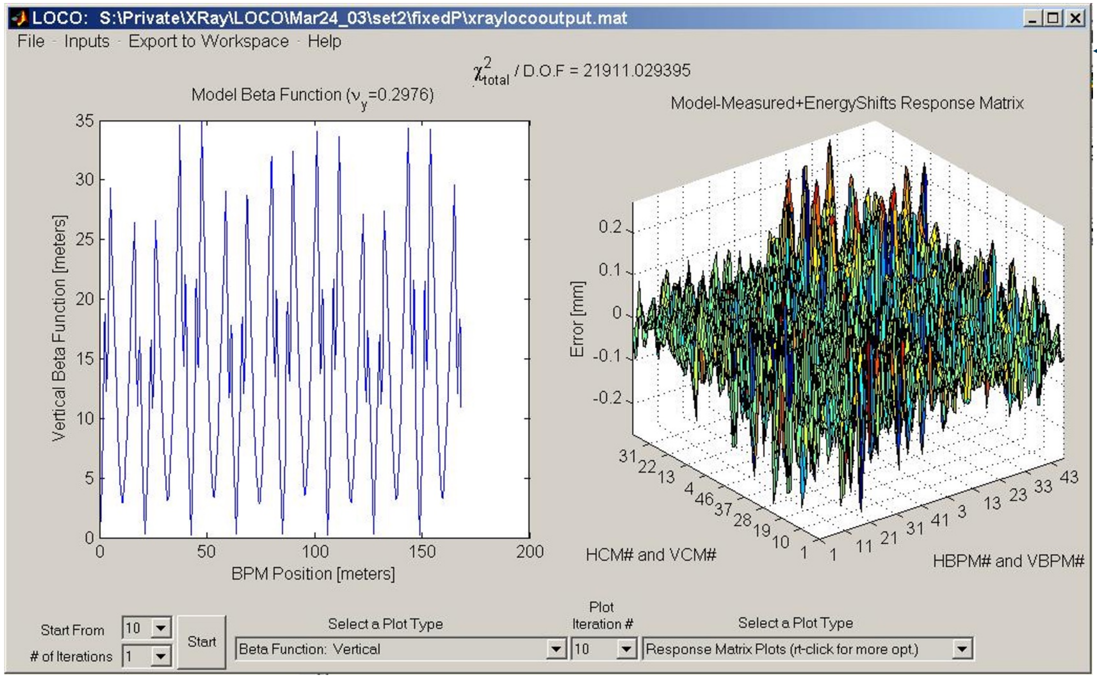
All | None

| | Common Name | Saved | Present | Difference | Channel Name |
|---|---|---|---|---|---|
| 1 | SR01C HCM2 | 2.4834 | 2.1535 | 0.3299 | SR01C___HCM2___AC01 |
| 2 | SR01C HCM3 | -2.8546 | -2.9254 | 0.0708 | SR01C___HCM3___AC02 |
| 3 | SR01C HCM4 | 2.0643 | 2.1394 | -0.0751 | SR01C___HCM4___AC03 |
| 4 | SR01C HCSD1 | -0.1861 | -0.2369 | 0.0508 | SR01C___HCSD1___AC00 |
| 5 | SR01C HCSD2 | 0.4834 | 0.4499 | 0.0336 | SR01C___HCSD2___AC01 |
| 6 | SR01C HCSF1 | -0.4171 | -0.4024 | -0.0146 | SR01C___HCSF1___AC02 |
| 7 | SR01C HCSF2 | -1.6436 | -1.3695 | -0.2740 | SR01C___HCSF2___AC03 |
| 8 | SR02C HCM1 | -0.8611 | -1.2647 | 0.4036 | SR02C___HCM1___AC00 |
| 9 | SR02C HCM2 | 3.2098 | 3.8261 | -0.6163 | SR02C___HCM2___AC01 |
| 10 | SR02C HCM3 | 4.3154 | 3.7702 | 0.5452 | SR02C___HCM3___AC02 |
| 11 | SR02C HCM4 | -1.3294 | -1.2178 | -0.1116 | SR02C___HCM4___AC03 |
| 12 | SR02C HCSD1 | 1.1803 | 1.0369 | 0.1435 | SR02C___HCSD1___AC00 |
| 13 | SR02C HCSD2 | -0.7575 | -0.5868 | -0.1707 | SR02C___HCSD2___AC01 |
| 14 | SR02C HCSF1 | 0.4060 | 0.7393 | -0.3333 | SR02C___HCSF1___AC02 |
| 15 | SR02C HCSF2 | -1.3653 | -1.2345 | -0.1308 | SR02C___HCSF2___AC03 |
| 16 | SR03C HCM1 | -0.9174 | -0.7456 | -0.1718 | SR03C___HCM1___AC00 |
| 17 | SR03C HCM2 | 1.6622 | 1.3624 | 0.2998 | SR03C___HCM2___AC01 |
| 18 | SR03C HCM3 | 5.3772 | 6.2632 | -0.8860 | SR03C___HCM3___AC02 |
| 19 | SR03C HCM4 | -0.7731 | -1.2626 | 0.4894 | SR03C___HCM4___AC03 |
| 20 | SR03C HCSD1 | -1.7643 | -1.4993 | -0.2650 | SR03C___HCSD1___AC00 |
| 21 | SR03C HCSD2 | -0.1011 | -0.1552 | 0.0541 | SR03C___HCSD2___AC01 |
| 22 | SR03C HCSF1 | 0.0987 | 0.1154 | -0.0167 | SR03C___HCSF1___AC02 |
| 23 | SR03C HCSF2 | -0.6371 | -0.4230 | -0.2141 | SR03C___HCSF2___AC03 |
| 24 | SR04C HCM1 | -0.6833 | -0.9424 | 0.2591 | SR04C___HCM1___AC00 |
| 25 | SR04C HCM2 | -1.3519 | -0.6885 | -0.6634 | SR04C___HCM2___AC01 |
| 26 | SR04C HCM3 | -4.3273 | -2.6350 | -1.6923 | SR04C___HCM3___AC02 |
| 27 | SR04C HCM4 | 2.8719 | 1.8860 | 0.9859 | SR04C___HCM4___AC03 |
| 28 | SR04C HCSD1 | 1.6749 | 1.5084 | 0.1665 | SR04C___HCSD1___AC00 |
| 29 | SR04C HCSD2 | -0.5050 | -0.6395 | 0.1345 | SR04C___HCSD2___AC01 |
| 30 | SR04C HCSF1 | -1.1413 | -0.8029 | -0.3383 | SR04C___HCSF1___AC02 |
| 31 | SR04C HCSF2 | 0.0051 | 0.0204 | -0.0153 | SR04C___HCSF2___AC03 |
| 32 | SR04U HCM2 | 3.8831 | 3.8465 | 0.0366 | SR04U___HCM2___AC00 |
| 33 | SR05C HCM1 | 0.4487 | -0.0586 | 0.5073 | SR05C___HCM1___AC00 |
| 34 | SR05C HCM2 | -2.2776 | -1.2428 | -1.0348 | SR05C___HCM2___AC01 |
| 35 | SR05C HCM3 | -3.4285 | -2.6524 | -0.7761 | SR05C___HCM3___AC02 |
| 36 | SR05C HCM4 | 2.3152 | 1.9160 | 0.3992 | SR05C___HCM4___AC03 |
| 37 | SR05C HCSD1 | 1.6053 | 1.5049 | 0.1005 | SR05C___HCSD1___AC00 |
| 38 | SR05C HCSD2 | 1.0667 | 1.1270 | -0.0603 | SR05C___HCSD2___AC01 |
| 39 | SR05C HCSF1 | -1.4808 | -1.2453 | -0.2355 | SR05C___HCSF1___AC02 |
| 40 | SR05C HCSF2 | -0.8600 | -0.7995 | -0.0605 | SR05C___HCSF2___AC03 |
| 41 | SR06C HCM1 | 2.7772 | 2.1691 | 0.6082 | SR06C___HCM1___AC00 |
| 42 | SR06C HCM2 | -3.5768 | -2.4459 | -1.1309 | SR06C___HCM2___AC01 |

| | Common Name | Saved | Present | Difference | Channel Name |
|---|---|---|---|---|---|
| 1 | BPMx(1,2) (Monitor) | -0.2240 | 0.0507 | -0.2747 | SR01C:BPM1:SA:X |
| 2 | BPMx(1,3) (Monitor) | -0.3511 | -0.3541 | 0.0030 | SR01C:BPM2:SA:X |
| 3 | BPMx(1,4) (Monitor) | 0.6160 | -0.0604 | 0.6764 | SR01C:BPM3:SA:X |
| 4 | BPMx(1,5) (Monitor) | 0.1075 | -0.0731 | 0.1806 | SR01C:BPM4:SA:X |
| 5 | BPMx(1,6) (Monitor) | 0.8042 | -0.1226 | 0.9269 | SR01C:BPM5:SA:X |
| 6 | BPMx(1,7) (Monitor) | 0.4016 | -0.0154 | 0.4170 | SR01C:BPM6:SA:X |
| 7 | BPMx(1,9) (Monitor) | -0.1969 | -0.0466 | -0.1503 | SR01C:BPM8:SA:X |
| 8 | BPMx(2,3) (Monitor) | 0.0504 | 0 | 0.0504 | SR02C:BPM2:SA:X |
| 9 | BPMx(2,4) (Monitor) | 0.6727 | -0.0356 | 0.7083 | SR02C:BPM3:SA:X |
| 10 | BPMx(2,5) (Monitor) | 0.9610 | 0 | 0.9610 | SR02C:BPM4:SA:X |
| 11 | BPMx(2,6) (Monitor) | 1.4760 | 0.0429 | 1.4331 | SR02C:BPM5:SA:X |
| 12 | BPMx(2,7) (Monitor) | 0.5538 | -0.0590 | 0.6129 | SR02C:BPM6:SA:X |
| 13 | BPMx(2,8) (Monitor) | -0.7323 | -0.0942 | -0.6381 | SR02C:BPM7:SA:X |
| 14 | BPMx(2,9) (Monitor) | -0.0963 | 0.1865 | -0.2827 | SR02C:BPM8:SA:X |
| 15 | BPMx(1,10) (Monit... | 0.9191 | 0.8278 | 0.0913 | SR02S:IDBPM1:SA:X |
| 16 | BPMx(2,1) (Monitor) | 0.4214 | -0.0093 | 0.4307 | SR02S:IDBPM2:SA:X |
| 17 | BPMx(3,2) (Monitor) | -0.0324 | -0.2462 | 0.2137 | SR03C:BPM1:SA:X |
| 18 | BPMx(3,3) (Monitor) | 0.1959 | -0.0766 | 0.2724 | SR03C:BPM2:SA:X |
| 19 | BPMx(3,4) (Monitor) | 0.6309 | 0.0751 | 0.5558 | SR03C:BPM3:SA:X |
| 20 | BPMx(3,5) (Monitor) | 0.6270 | -4.9772 | 5.6041 | SR03C:BPM4:SA:X |
| 21 | BPMx(3,6) (Monitor) | 0.8826 | -0.2376 | 1.1202 | SR03C:BPM5:SA:X |
| 22 | BPMx(3,7) (Monitor) | 1.1059 | -0.1485 | 1.2544 | SR03C:BPM6:SA:X |
| 23 | BPMx(3,8) (Monitor) | -0.0021 | 0.0370 | -0.0392 | SR03C:BPM7:SA:X |
| 24 | BPMx(3,9) (Monitor) | 0.3373 | -0.2268 | 0.5641 | SR03C:BPM8:SA:X |
| 25 | BPMx(4,2) (Monitor) | 0.4792 | -0.1445 | 0.6237 | SR04C:BPM1:SA:X |
| 26 | BPMx(4,3) (Monitor) | -0.3872 | -0.2835 | -0.1038 | SR04C:BPM2:SA:X |
| 27 | BPMx(4,4) (Monitor) | 0.6497 | 0.0524 | 0.5973 | SR04C:BPM3:SA:X |
| 28 | BPMx(4,7) (Monitor) | 0.6646 | 0.0536 | 0.6109 | SR04C:BPM6:SA:X |
| 29 | BPMx(4,8) (Monitor) | -0.4437 | 0.1143 | -0.5580 | SR04C:BPM7:SA:X |
| 30 | BPMx(4,9) (Monitor) | -0.0949 | -0.0270 | -0.0678 | SR04C:BPM8:SA:X |
| 31 | BPMx(4,5) (Monitor) | 0.5018 | -0.1283 | 0.6302 | SR04C___BPM4XTFAM00 |
| 32 | BPMx(4,6) (Monitor) | 1.2218 | -0.0248 | 1.2467 | SR04C___BPM5XTFAM02 |
| 33 | BPMx(3,10) (Monit... | 1.5293 | 0.0152 | 1.5141 | SR04S:IDBPM1:SA:X |
| 34 | BPMx(4,1) (Monitor) | 1.5479 | 0.0523 | 1.4955 | SR04S:IDBPM2:SA:X |
| 35 | BPMx(3,11) (Monit... | 0.8979 | -0.0351 | 0.9330 | SR04S:IDBPM3:SA:X |
| 36 | BPMx(3,12) (Monit... | 0.8961 | -0.0474 | 0.9435 | SR04S:IDBPM4:SA:X |
| 37 | BPMx(5,2) (Monitor) | -0.2213 | -0.0868 | -0.1345 | SR05C:BPM1:SA:X |
| 38 | BPMx(5,3) (Monitor) | -0.1088 | -0.0101 | -0.0987 | SR05C:BPM2:SA:X |
| 39 | BPMx(5,4) (Monitor) | 0.3370 | 0.1833 | 0.1536 | SR05C:BPM3:SA:X |
| 40 | BPMx(5,5) (Monitor) | 0.5935 | -0.1247 | 0.7182 | SR05C:BPM4:SA:X |
| 41 | BPMx(5,6) (Monitor) | 0.2702 | 0 | 0.2702 | SR05C:BPM5:SA:X |
| 42 | BPMx(5,7) (Monitor) | 0.1819 | -0.0247 | 0.2066 | SR05C:BPM6:SA:X |

# LOCO Optics Analysis

- Calibrate/control optics using orbit response matrix
- Determine quadrupole gradients
- Correct coupling
- Calibrate BPM gains, steering magnets
- Measure local chromaticity and transverse impedance




<u>MATLAB version of LOCO</u>

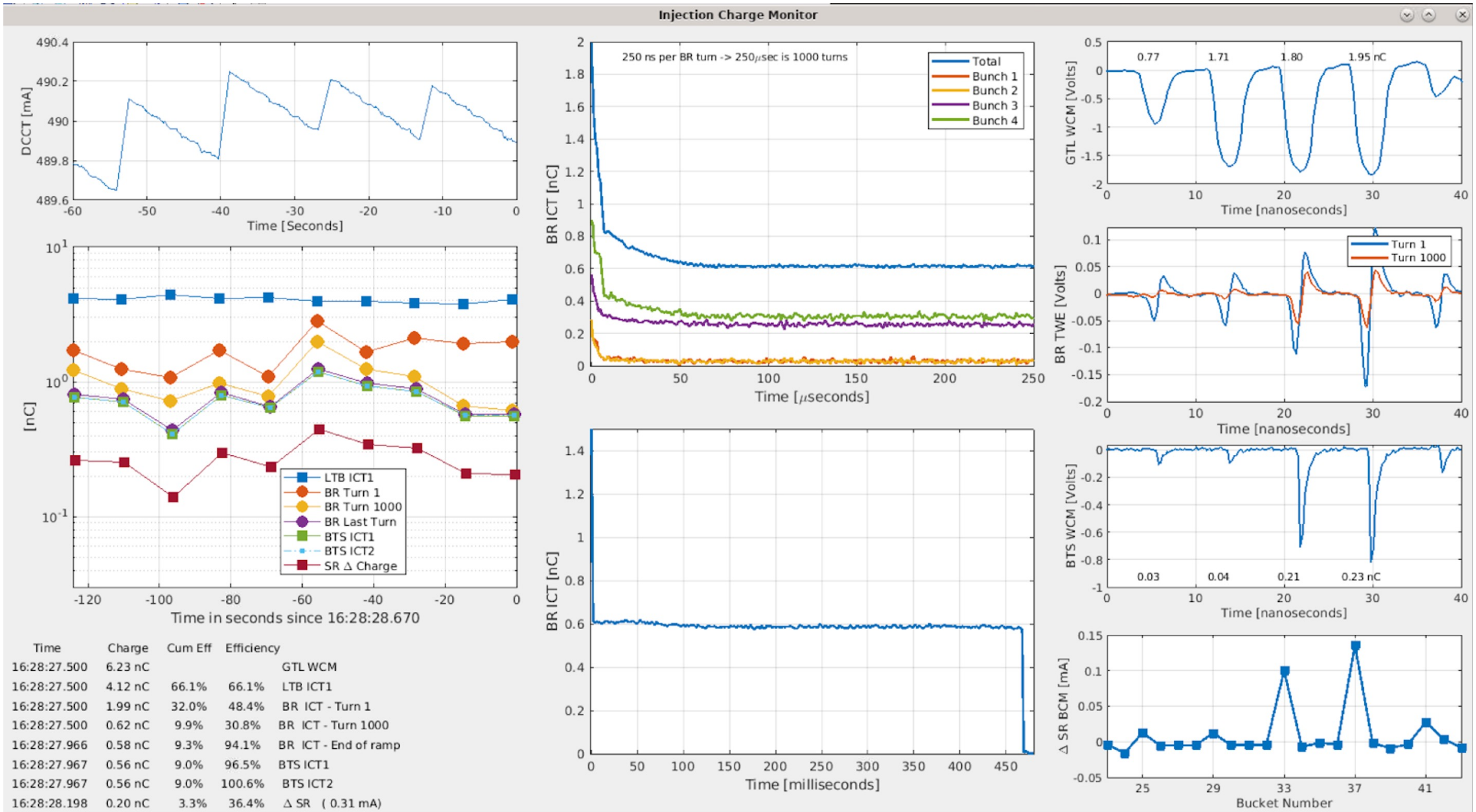
- Rewritten from FORTRAN
- Linked to AT simulator
- Compatible with the MMLT
  - Easy to measure LOCO data with the MML
    - *measlocodata*
  - Relatively easy to import response matrix data
    - *buildlocoinput*
  - Relatively easy to apply result back to the accelerator (*setlocodata*).

# Accelerator Specific Applications

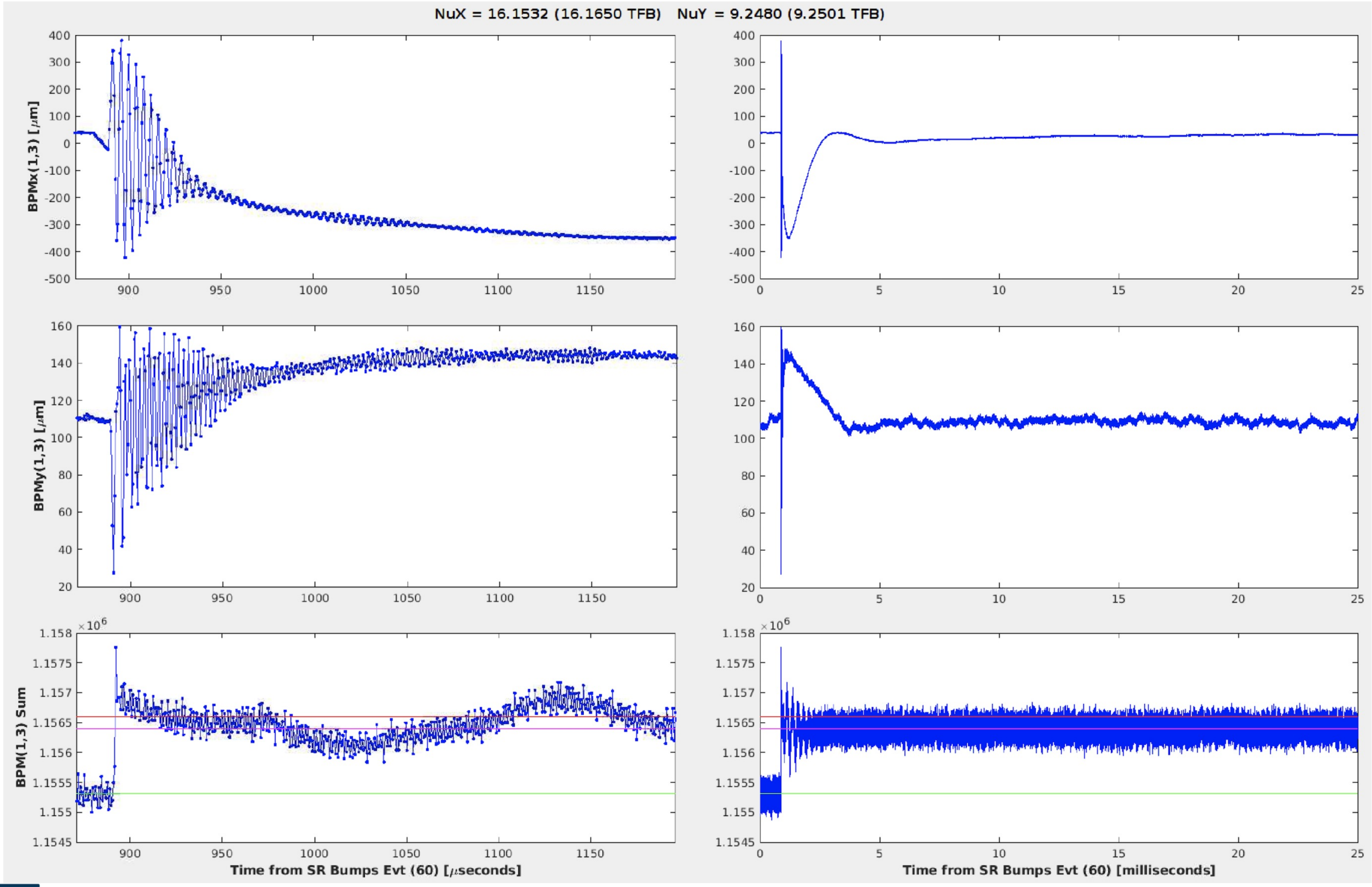
There are many more accelerator specific applications than accelerator independent applications.

However, these applications are typically control system independent so they are often relatively easy to port to a different facility.

# Injection Charge Monitor at ALS

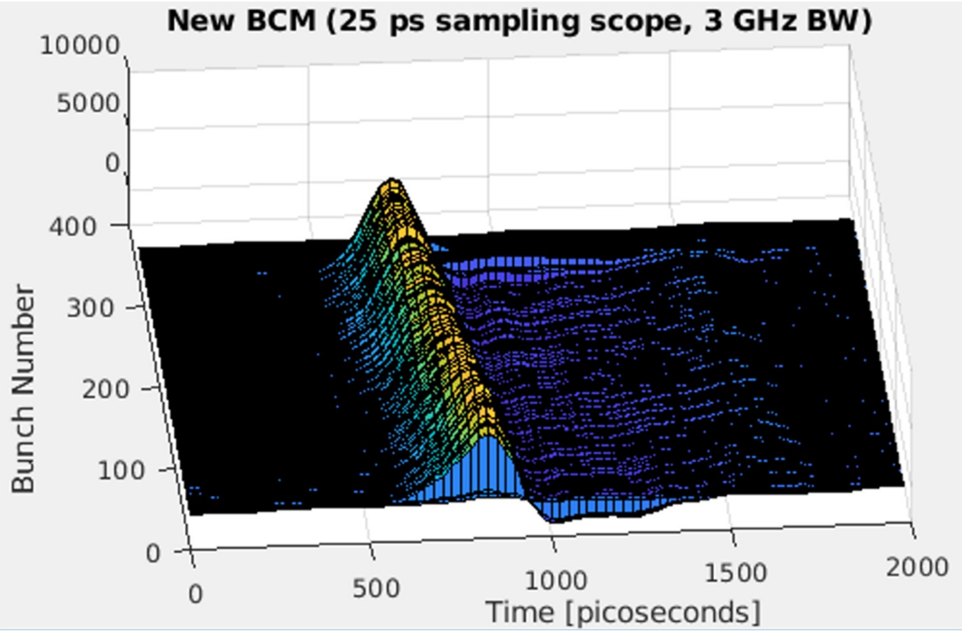


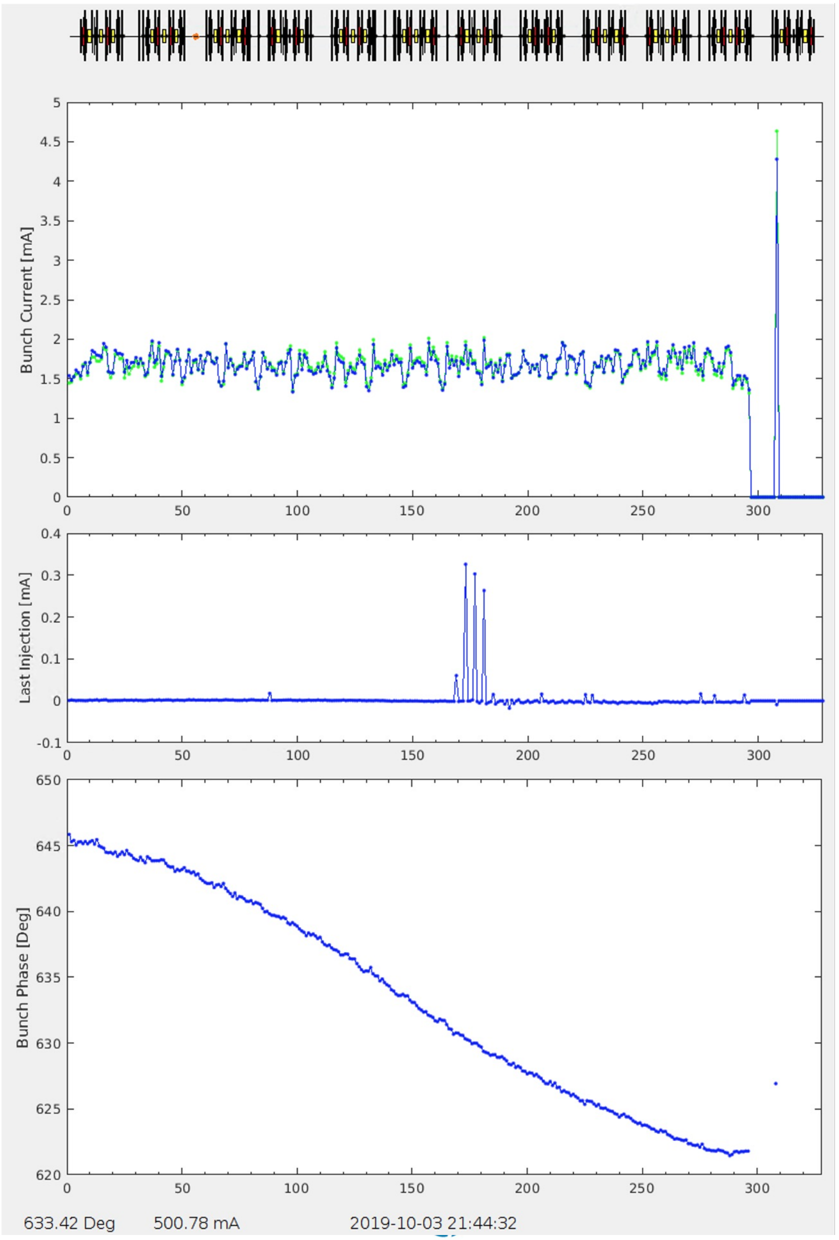
# SR Injection Orbit and Tunes



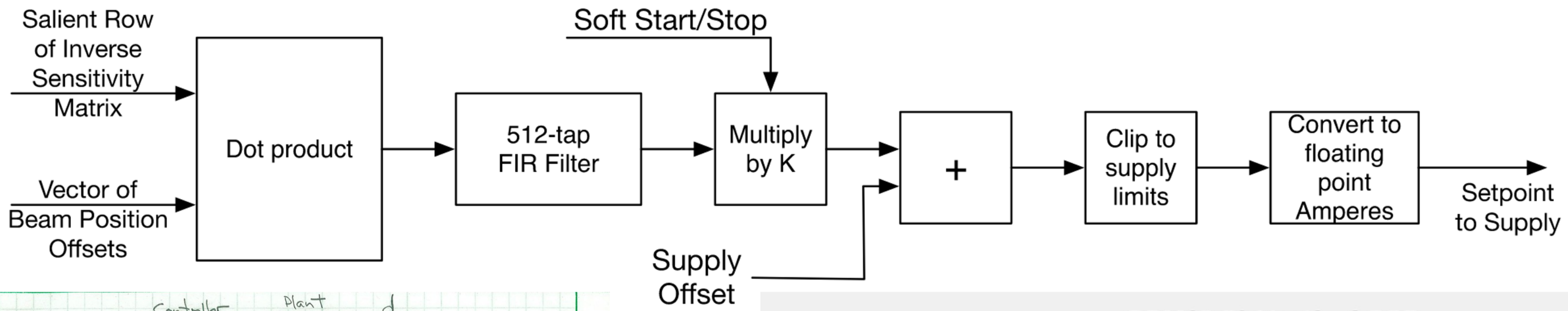
# Bunch Current and Phase

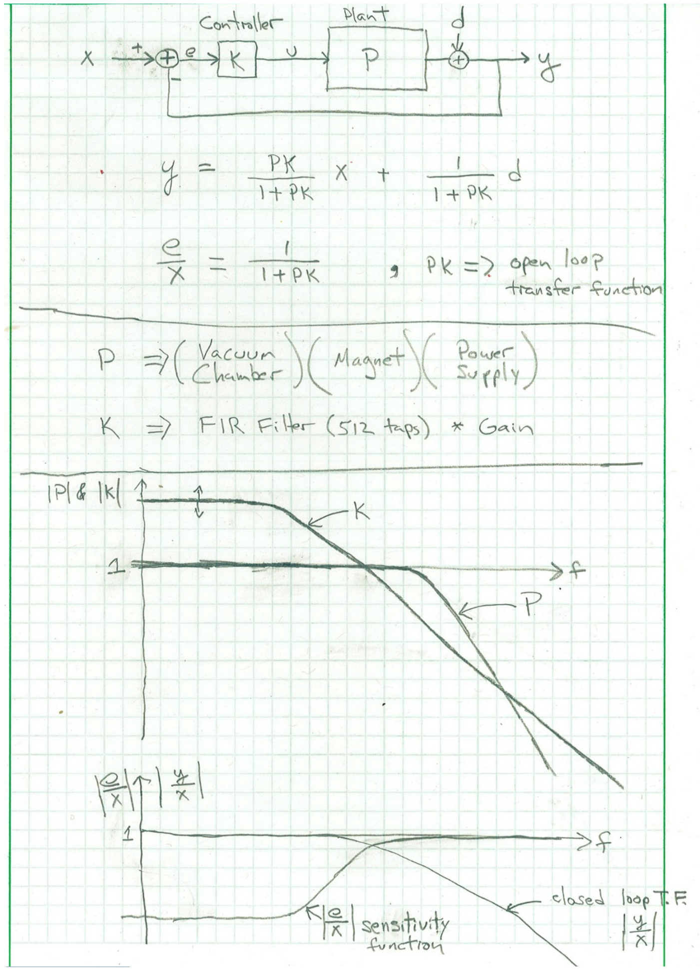
A Matlab application post-process the data from an FPGA based sampling scope and provides bunch current and phase information 24/7 in the control room.

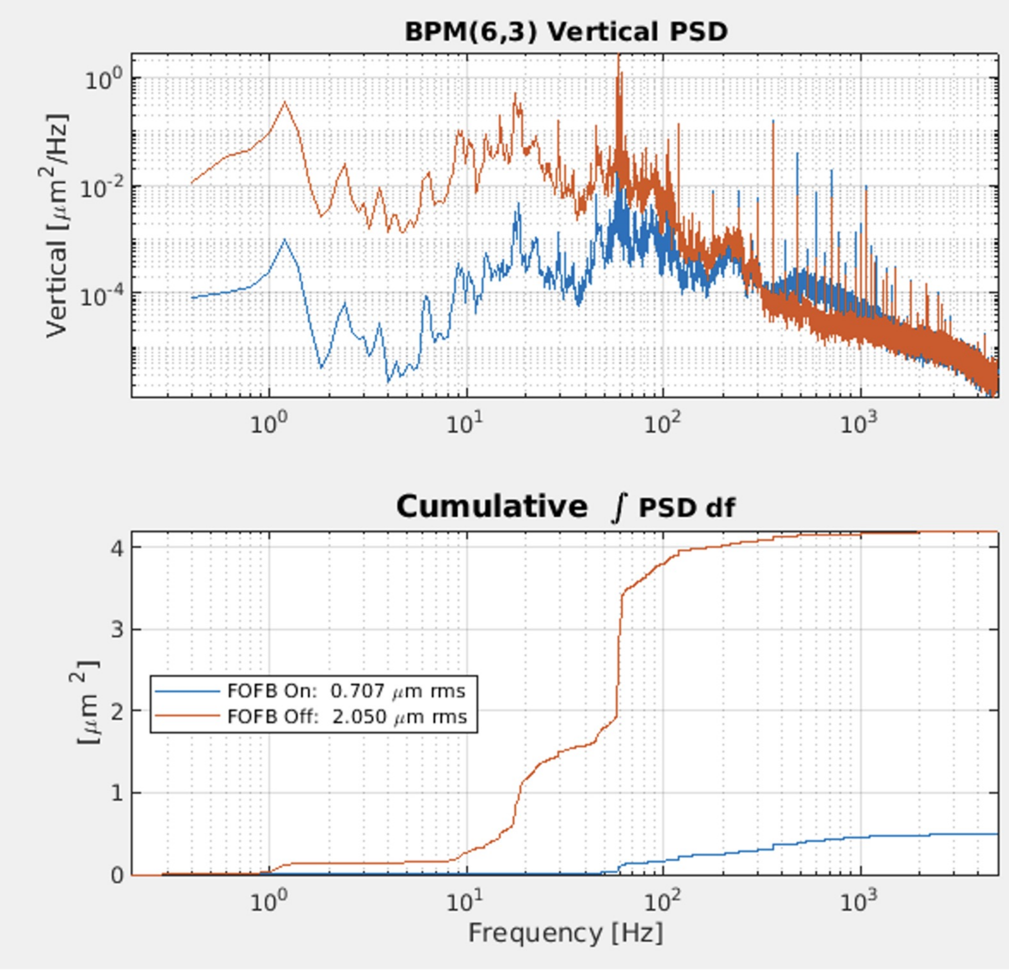


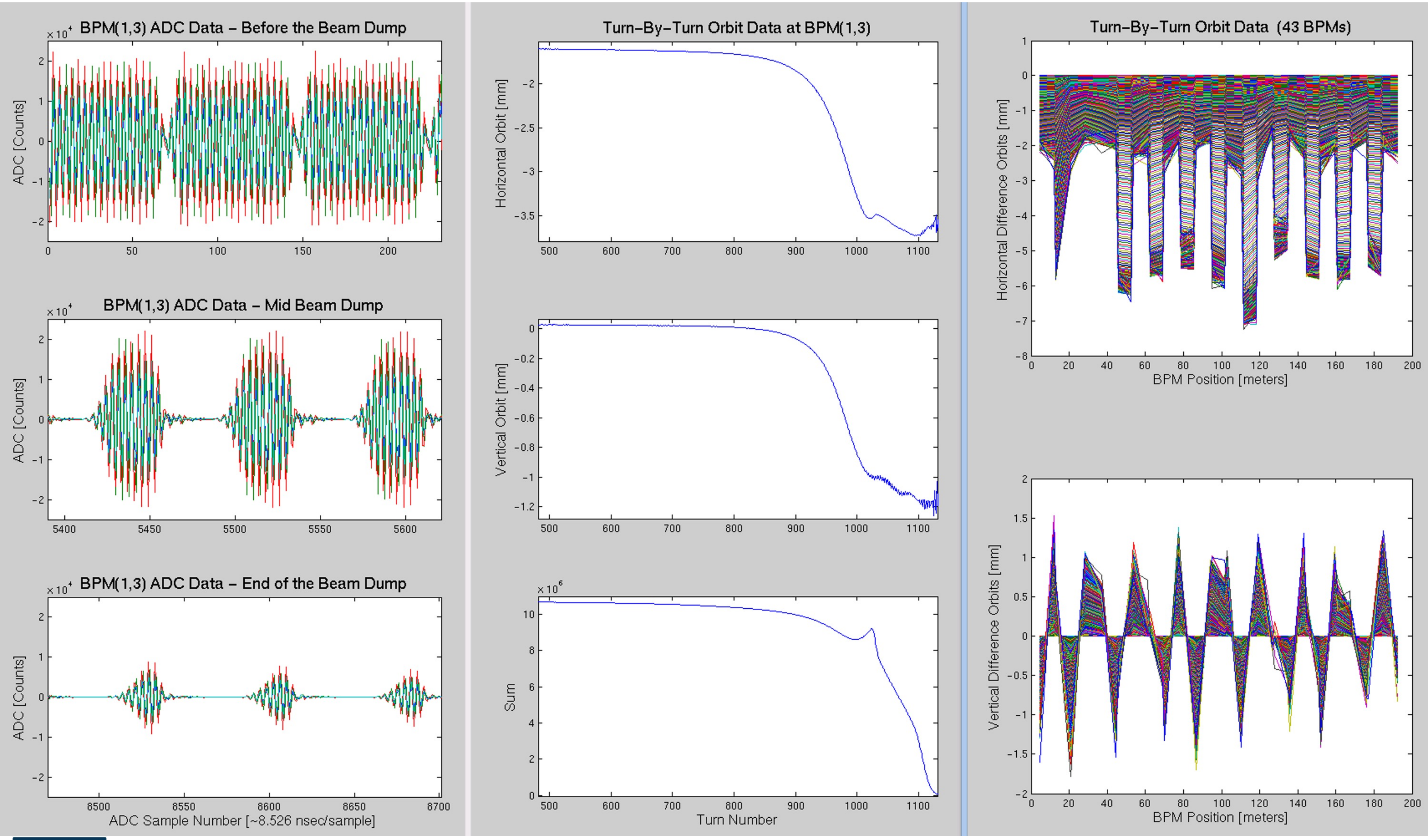
# Fast Orbit Feedback Setup and Monitorting



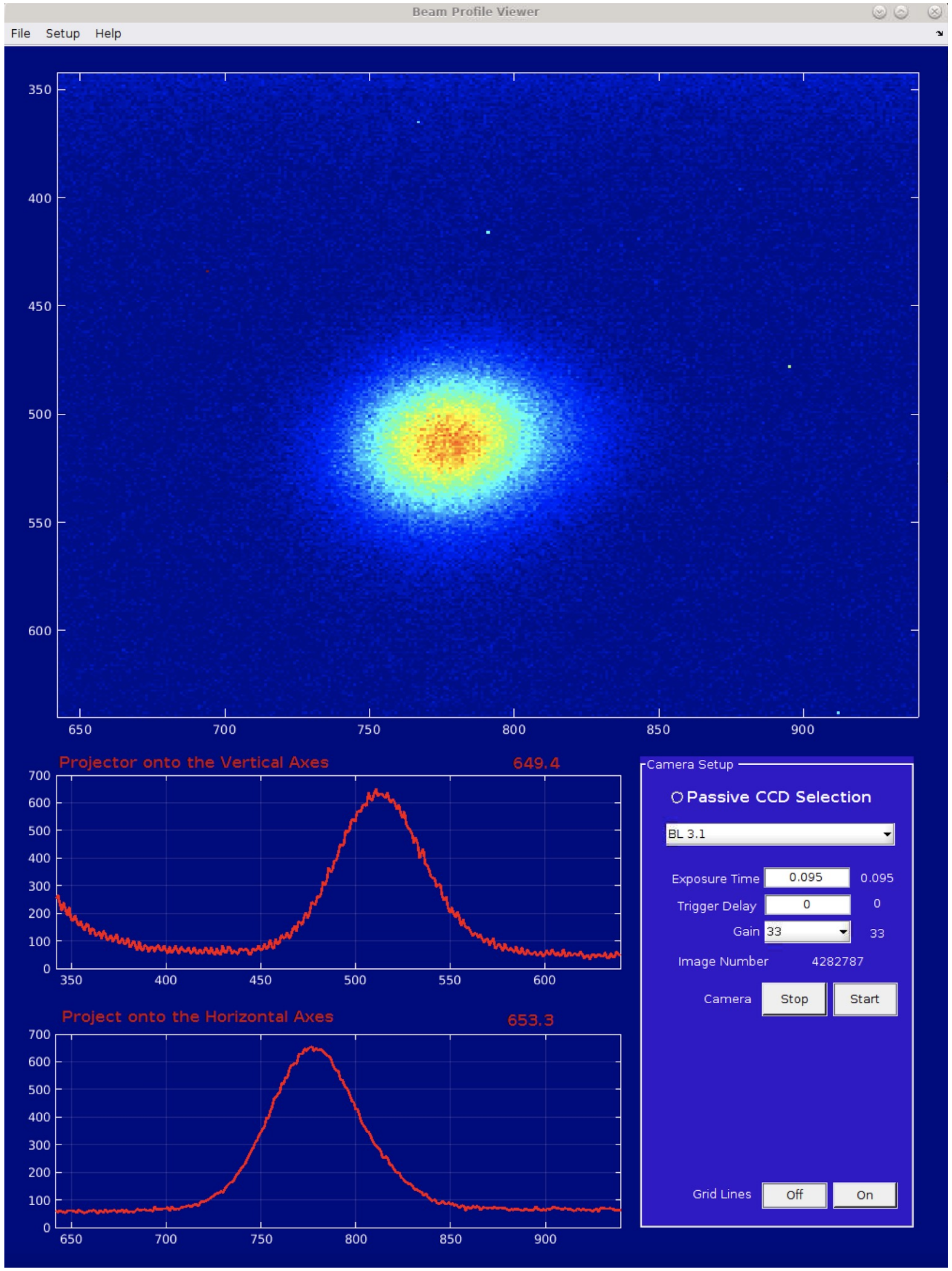



# Beam Dump Orbit Capture

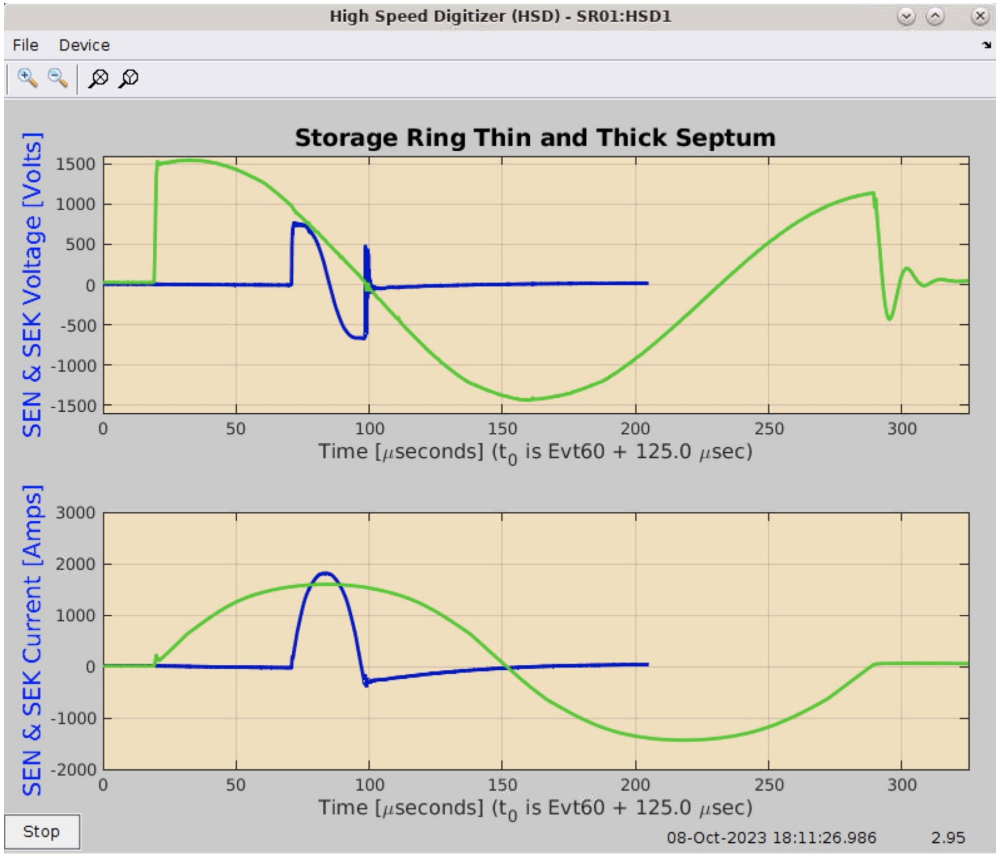


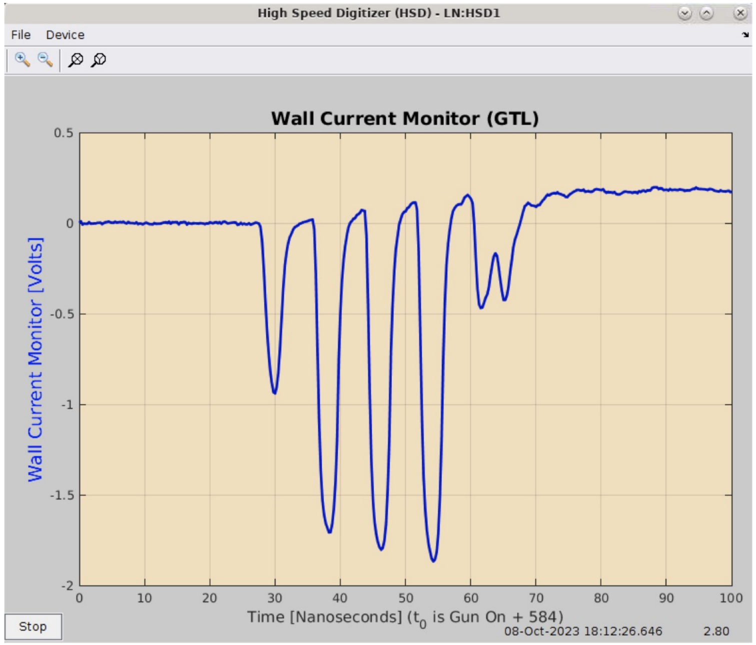
# CCD Cameras

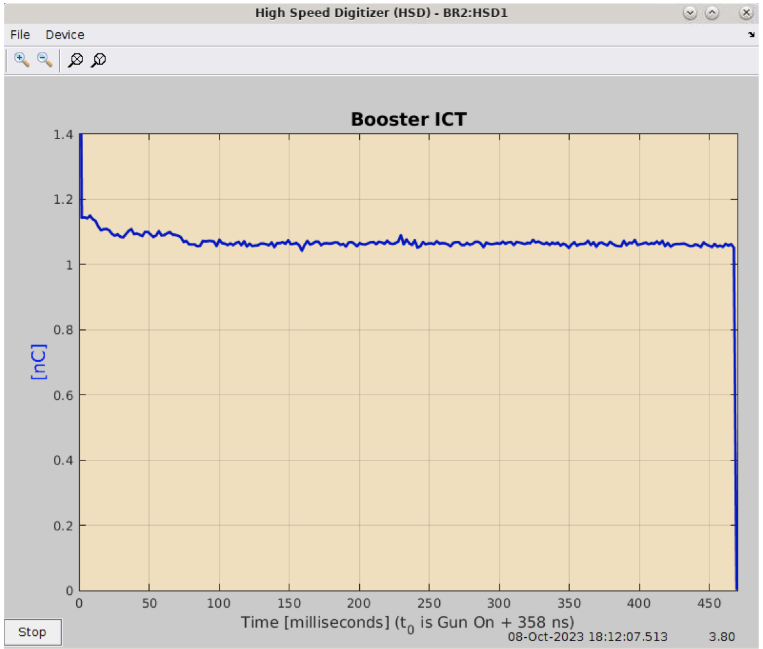


Beam Profile Viewer
File
Setup
Help
Projector onto the Vertical Axes
649.4
Project onto the Horizontal Axes
653.3
Camera Setup
Passive CCD Selection
BL 3.1
Exposure Time
0.095
0.095
Trigger Delay
0
0
Gain
33
33
Image Number
4282787
Camera
Stop
Start
Grid Lines
Off
On

# Scopes


High Speed Digitizer (HSD) - SR01:HSD1
File Device
Storage Ring Thin and Thick Septum
SEN & SEK Voltage [Volts]
Time [μseconds] (t0 is Evt60 + 125.0 μsec)
SEN & SEK Current [Amps]
Time [μseconds] (t0 is Evt60 + 125.0 μsec)
Stop
08-Oct-2023 18:11:26.986
2.95

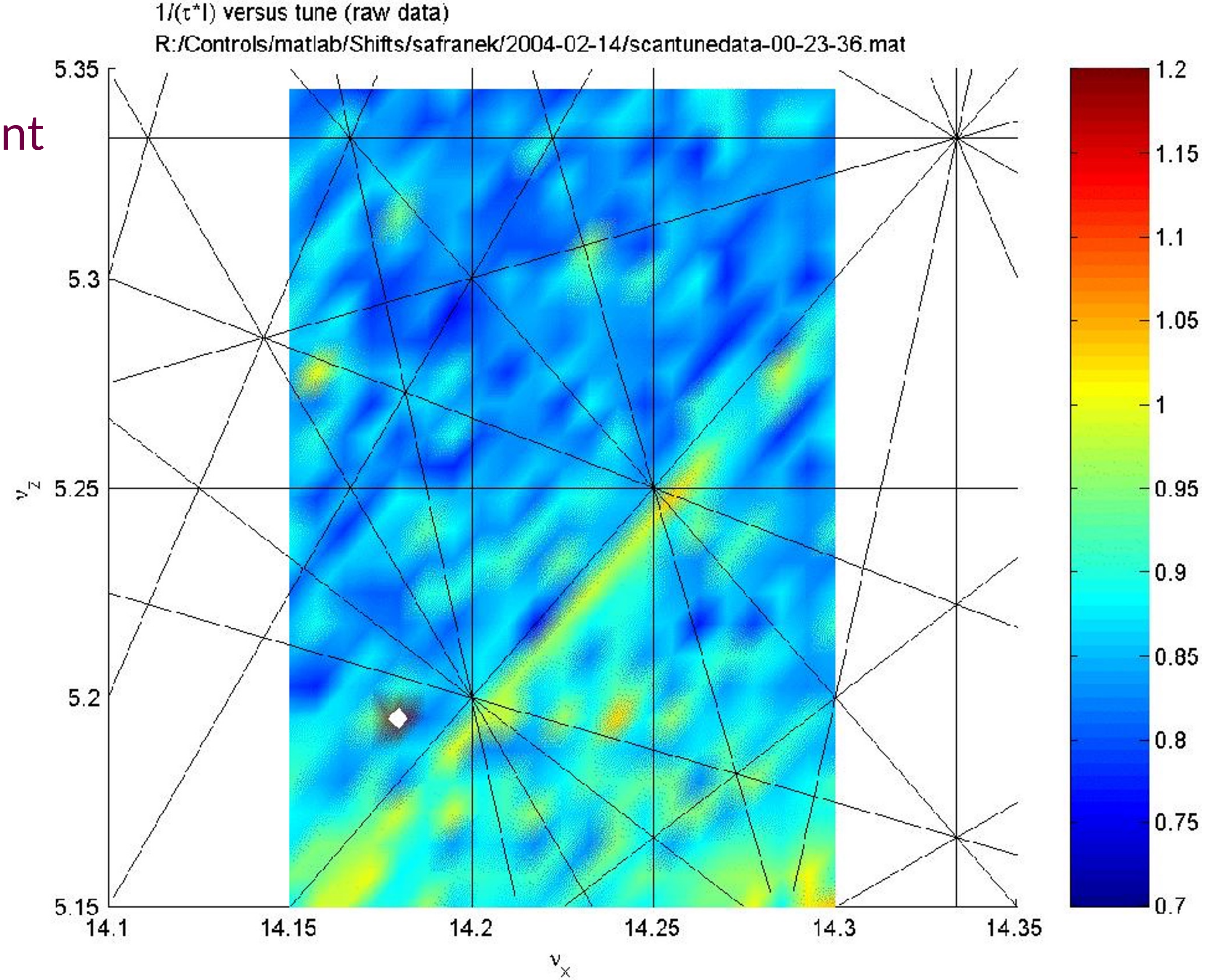

High Speed Digitizer (HSD) - LN:HSD1
File Device
Wall Current Monitor (GTL)
Wall Current Monitor [Volts]
Time [Nanoseconds] (t0 is Gun On + 584)
Stop
08-Oct-2023 18:12:26.646
2.80


High Speed Digitizer (HSD) - BR2:HSD1
File Device
Booster ICT
[nC]
Time [milliseconds] (t0 is Gun On + 358 ns)
Stop
08-Oct-2023 18:12:07.513
3.80

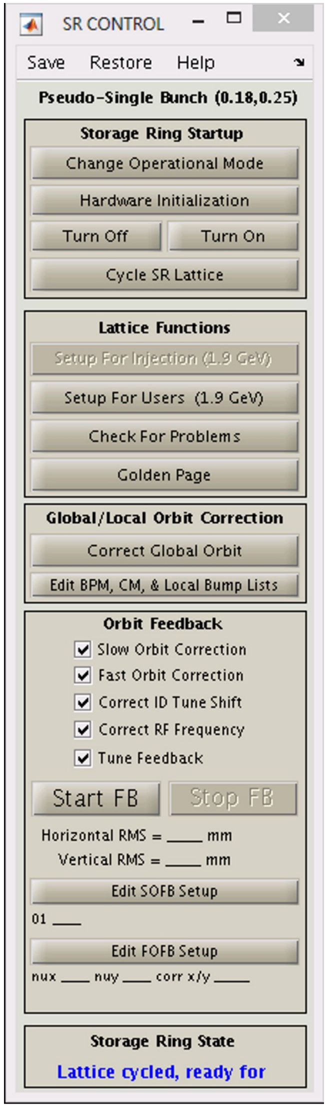
# Spear3 - Lifetime vs. tunes

Spear3 Measurement

- Resonant line:
  - $\nu_x - \nu_y = 9$
- Operating tunes (5.19, 6.23)
- ***Data gathered automatically on owl shift.***



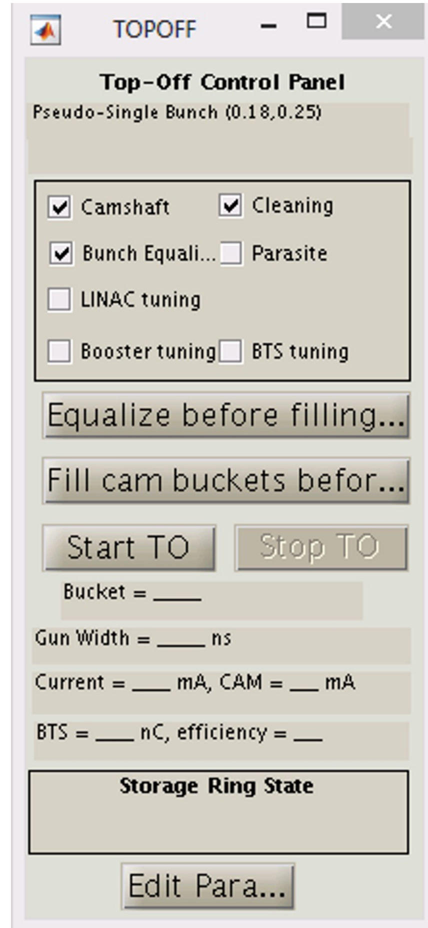
# Storage Ring Control & Topoff




SRControl

- Setups up the lattice, standardizes (cycles) the magnets
- Slow orbit feedback
- ID tune correction
- Tune feedback

Note: 150 to 200 magnets change when an ID moves.



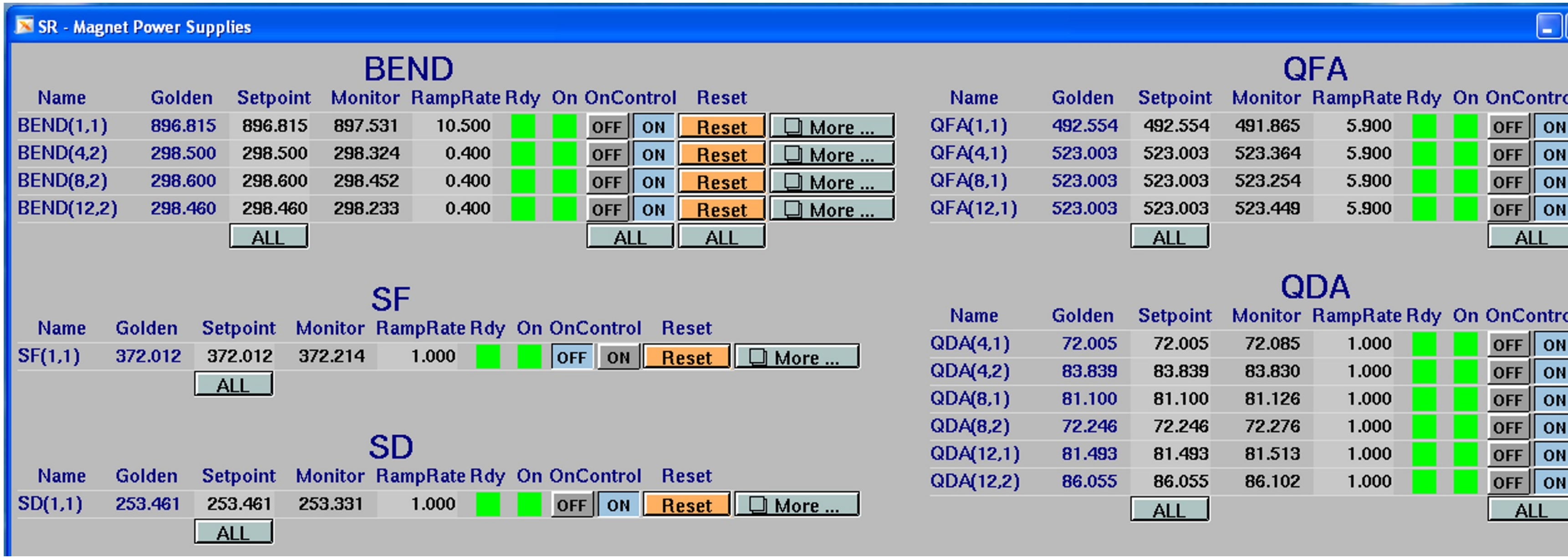

Topoff

- Controls the fill pattern
- Coordinates injector tuning during user operations
- Computes the lifetime and transfer efficiencies.

# MML2EDM

EDM screen generated from with Matlab

- EDM applications can be tedious to build.
- MML has channel names arranged by families.
- Adding/subtracting a device is easy and less error prone.




Recently Pheobus screen generation program was written in Python, but I haven't written the a MML2Pheobus function yet. Since Matlab can call Python functions this should be relatively straightforward to do.

# Who Uses the MML Software

## MML (~15 labs, maybe more, maybe less)

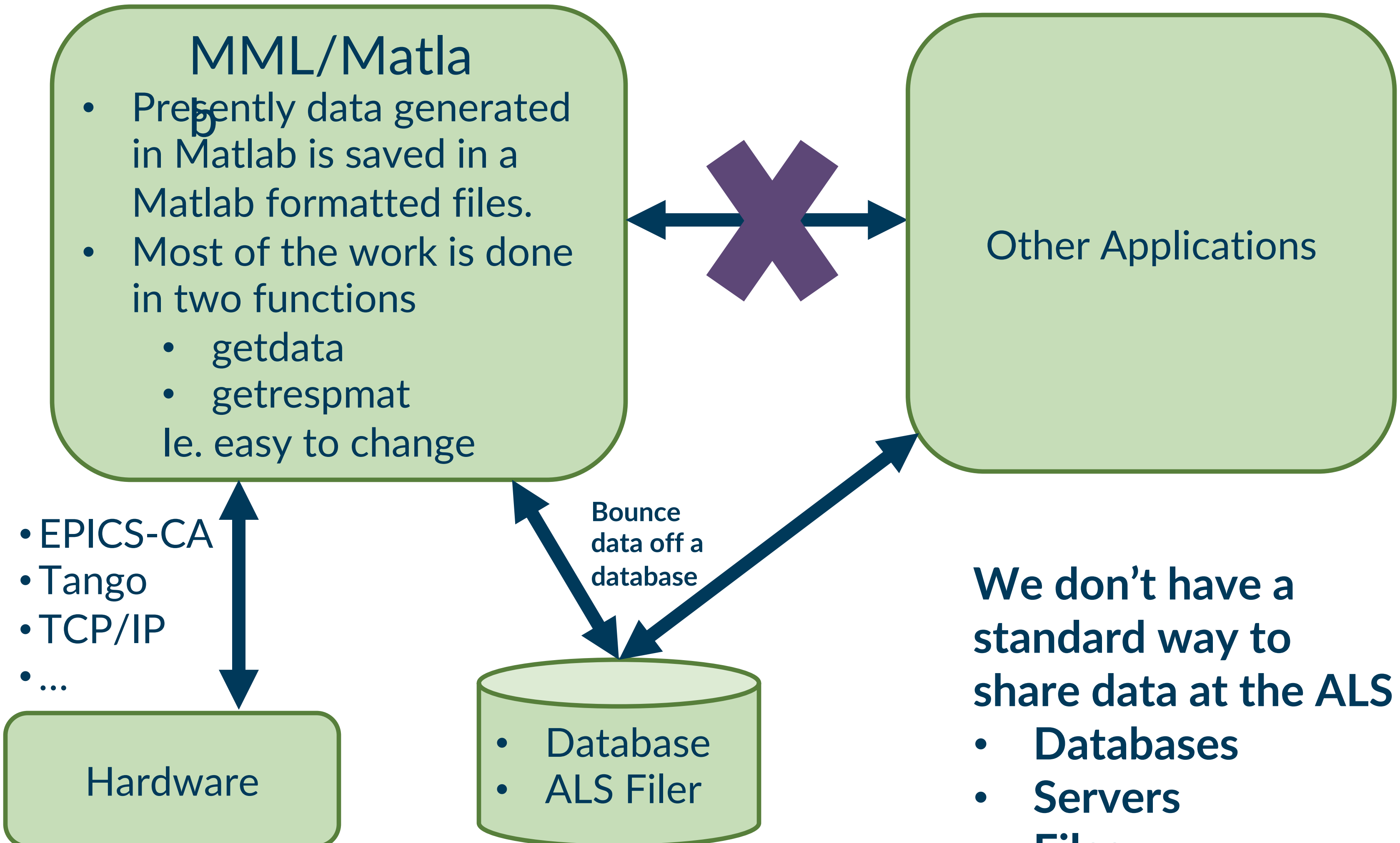
USA: ALS (Berkeley), Spear3 (Stanford), Duke FEL, NSLS-II (Brookhaven)
Canada: CLS
Europe: Soleil (France), Solaris (Poland), DIAMOND (England), ALBA (Spain), ELSA (Germany), MaxIV (Sweden)
Asia: PLS2 (Korea), SSRF (Shanghai), TLS/PLS (Taiwan)
Australia: ASP (Australia)
Dabblers: MLS (Germany), Indus (India), SESAME (Jordan), SNS (USA), SLS (Thailand), Elettra (Italy), LNLS/SIRIUS (Brazil), UMER (USA)

## LOCO (~20, likely more)

MML Users + BESSY–II and MLS (Germany), ELSA (Germany), TLS (Taiwan), LNLS (Brazil), RHIC (USA), ASTRID2 (Denmark)

Disclaimer: I don't know everyone that uses the MMLT & LOCO so it's difficult to know the extent of use.

# The Matlab Centric Problem




**We don't have a standard way to share data at the ALS**

- **Databases**
- **Servers**
- **Files**

# Future?

- There has been very little expansion of the core MML library in the last 10 years.
  - Almost all the expansion in the MML has been in scripting and high level application development at each of the individual accelerator level.

- NSLS-II didn't like that Matlab was such a "thick" client. They wanted their servers to do the bulk of the work and clients like Matlab, Python, Phoebus, … should be thin clients. Clearly there are many good merits for doing this and I think it a good approach but there are also some negative impacts.
  - I prefer a local model in Matlab over a model server. But it's fine to have both.
  - The code isn't as accessible, readable, changeable to the average physicist.
  - Even in simulation mode, you need to be connected the accelerator control system. Running on a laptop on an airplane isn't so easy.
  - Creating an accelerator independent and control system independent server might be a challenge.
  - It will be a challenge to provide all the functionality in the MML on a server and keep up with the weekly demands for new functionality to support physics experimentation and operational changes.
  - …

  Note: the MML can work in a "thin mode." For instance, measrespmat, getrespmat, getdata, etc. can be redirected to a server.

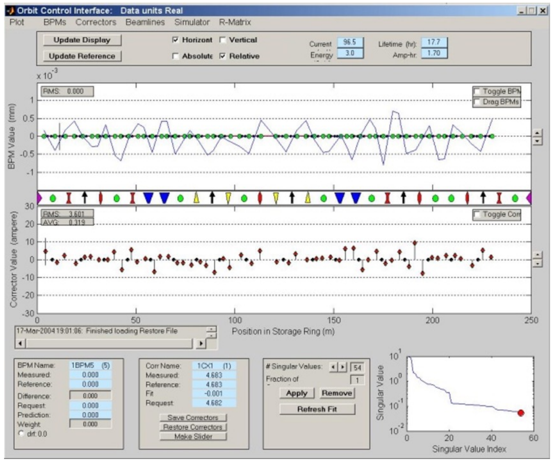
# Conclusion

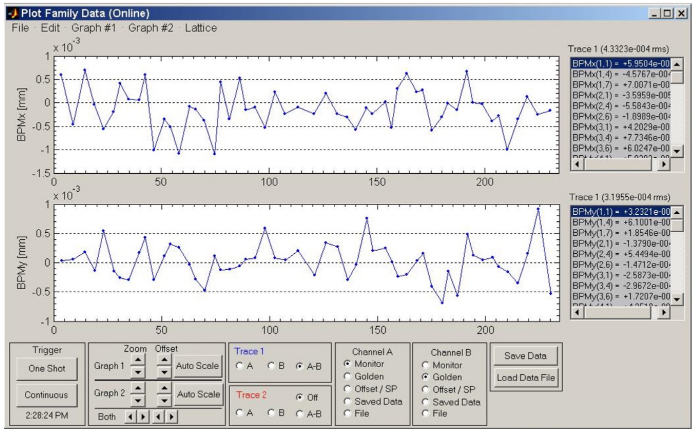
- Relatively easy to use.  Most people start writing useful scripts in a few hours.
- MiddleLayer + LOCO + AT  cover many of the high level software concerns for storage ring physics.  Hence, not every accelerator has to spend resources coding the same algorithms.
- Thousands of dedicated accelerator hours have been spent testing, improving, debugging, and exercising the Middle Layer software.
- It's a good scripting language for machine shifts or it can be the high level setup and control software for a storage ring.
- Integration of the AT model is good for debugging software without using accelerator time.
- Having machine independence software has fostered collaboration and code sharing between the laboratories.

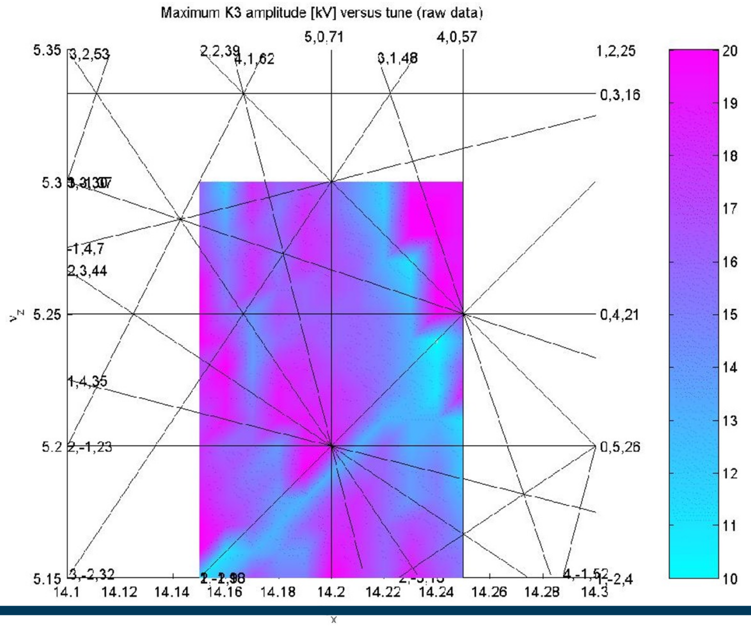
# Extra Slides

# Spear3 Commissioning Tools

- Basic Machine Setup and Control
  - Orbit, Tune, Chromaticity
  - Monitoring
- Fast scripting language for commissioning shifts
- Numerical algorithms and graphics for fast data processing



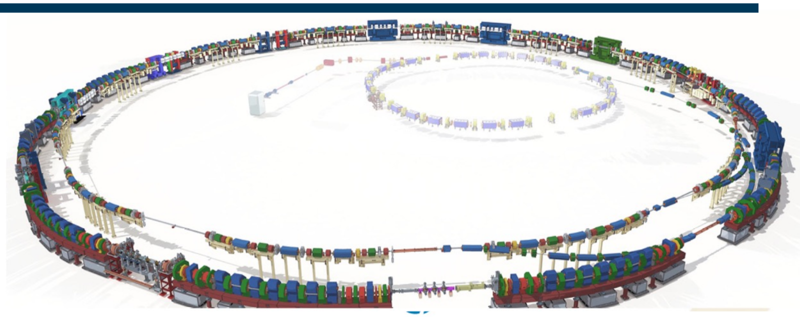




**Dynamic aperture vs ($\nu_x$, $\nu_y$)**

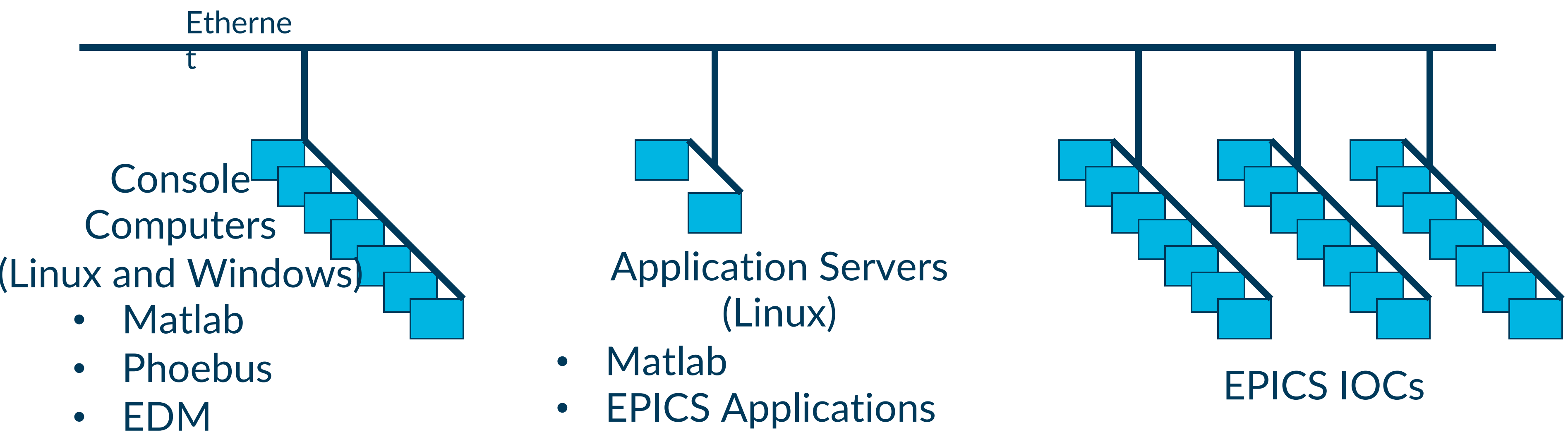
## ALS-U

- AR commissioning ~2 years
- SR commissioning ~5 years

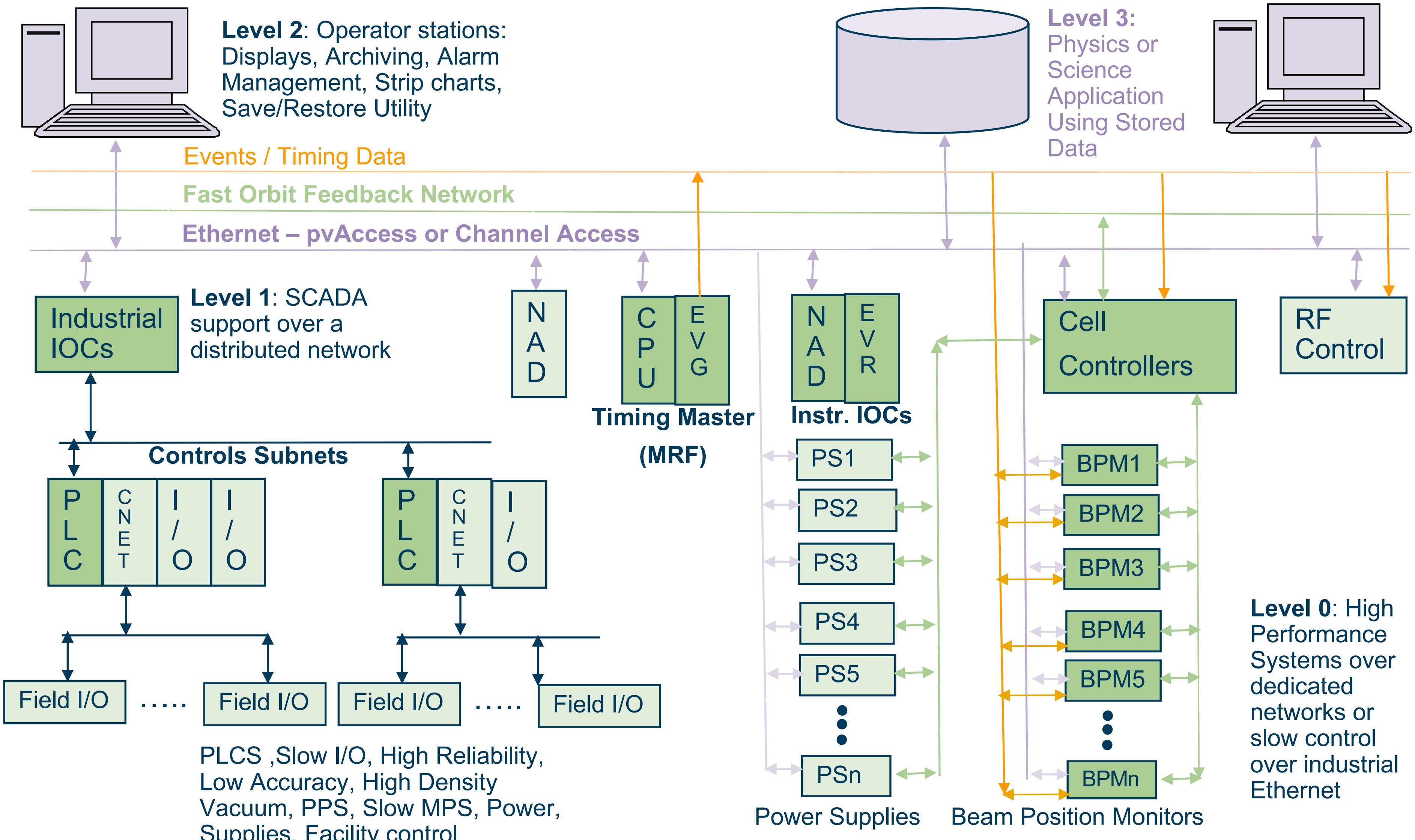


BERKELEY LAB

# ALS Computer Layout




# Architecture

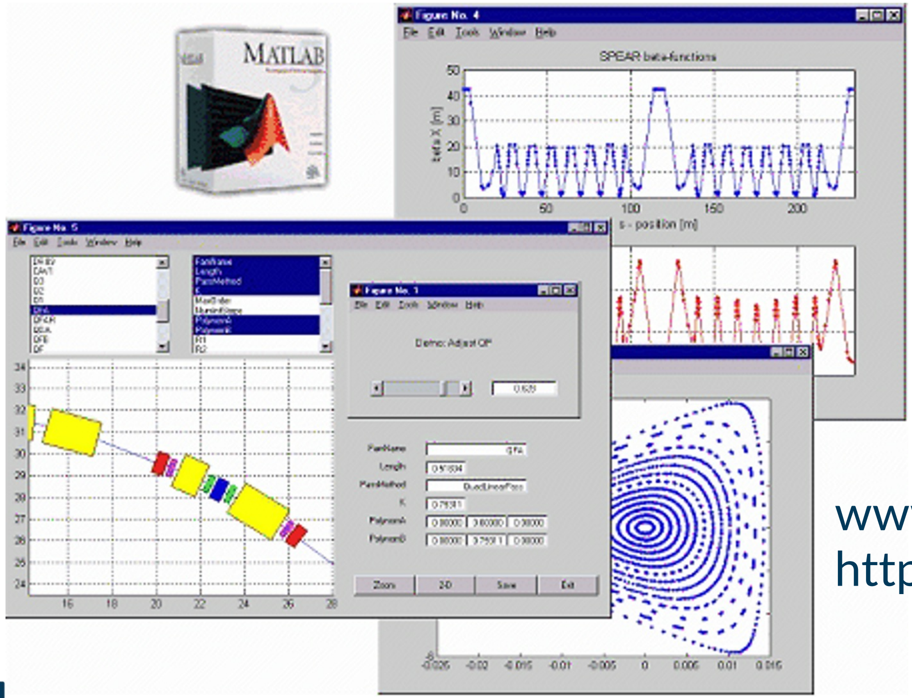


# AT – Accelerator Toolbox

Andrei Terebilo (AT) creator, was being maintained by Xiaobiao Huang (SLAC/SSRL) and Boaz Nash (Tech-X). Moved to git hub.

MATLAB ® Toolbox for Particle Accelerator Modeling

Accelerator Toolbox is a collection of tools to model particle accelerators and beam transport lines in MATLAB environment. It is being developed by Accelerator Physics Group at Stanford Synchrotron Radiation Laboratory for the ongoing design and future operation needs of SPEAR3 Synchrotron Light Source.

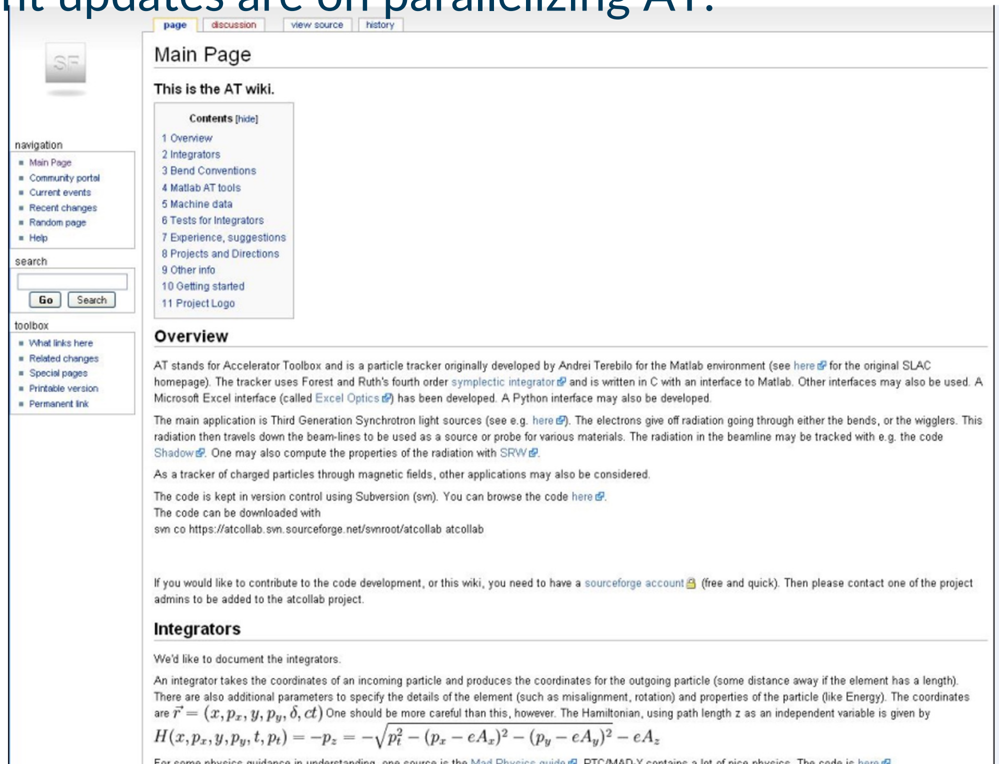



www-ssrl.slac.stanford.edu/at/welcome.html
http://www.slac.stanford.edu/~terebilo/at/

# AT – Accelerator Toolbox

New sourceforge project for AT set up by Boaz Nash (Tech-X)
https://sourceforge.net/apps/mediawiki/atcollab/index.php?title=Main_Page

Most recent updates are on parallelizing AT.


page | discussion | view source | history

navigation
- Main Page
- Community portal
- Current events
- Recent changes
- Random page
- Help

search

Go | Search

toolbox
- What links here
- Related changes
- Special pages
- Printable version
- Permanent link

## Main Page

**This is the AT wiki.**

Contents [hide]



### Overview

AT stands for Accelerator Toolbox and is a particle tracker originally developed by Andrei Terebilo for the Matlab environment (see here for the original SLAC homepage). The tracker uses Forest and Ruth's fourth order symplectic integrator and is written in C with an interface to Matlab. Other interfaces may also be used. A Microsoft Excel interface (called Excel Optics) has been developed. A Python interface may also be developed.

The main application is Third Generation Synchrotron light sources (see e.g. here). The electrons give off radiation going through either the bends, or the wigglers. This radiation then travels down the beam-lines to be used as a source or probe for various materials. The radiation in the beamline may be tracked with e.g. the code Shadow. One may also compute the properties of the radiation with SRW.

As a tracker of charged particles through magnetic fields, other applications may also be considered.

The code is kept in version control using Subversion (svn). You can browse the code here.
The code can be downloaded with
svn co https://atcollab.svn.sourceforge.net/svnroot/atcollab atcollab

If you would like to contribute to the code development, or this wiki, you need to have a sourceforge account (free and quick). Then please contact one of the project admins to be added to the atcollab project.

### Integrators

We'd like to document the integrators.

An integrator takes the coordinates of an incoming particle and produces the coordinates for the outgoing particle (some distance away if the element has a length). There are also additional parameters to specify the details of the element (such as misalignment, rotation) and properties of the particle (like Energy). The coordinates are $\vec{r} = (x, p_x, y, p_y, \delta, ct)$ One should be more careful than this, however. The Hamiltonian, using path length z as an independent variable is given by

$$H(x, p_x, y, p_y, t, p_t) = -p_z = -\sqrt{p_t^2 - (p_x - eA_x)^2 - (p_y - eA_y)^2} - eA_z$$

For some physics guidance in understanding, one source is the Mad Physics guide. PTC/MAD-X contains a lot of nice physics. The code is here.

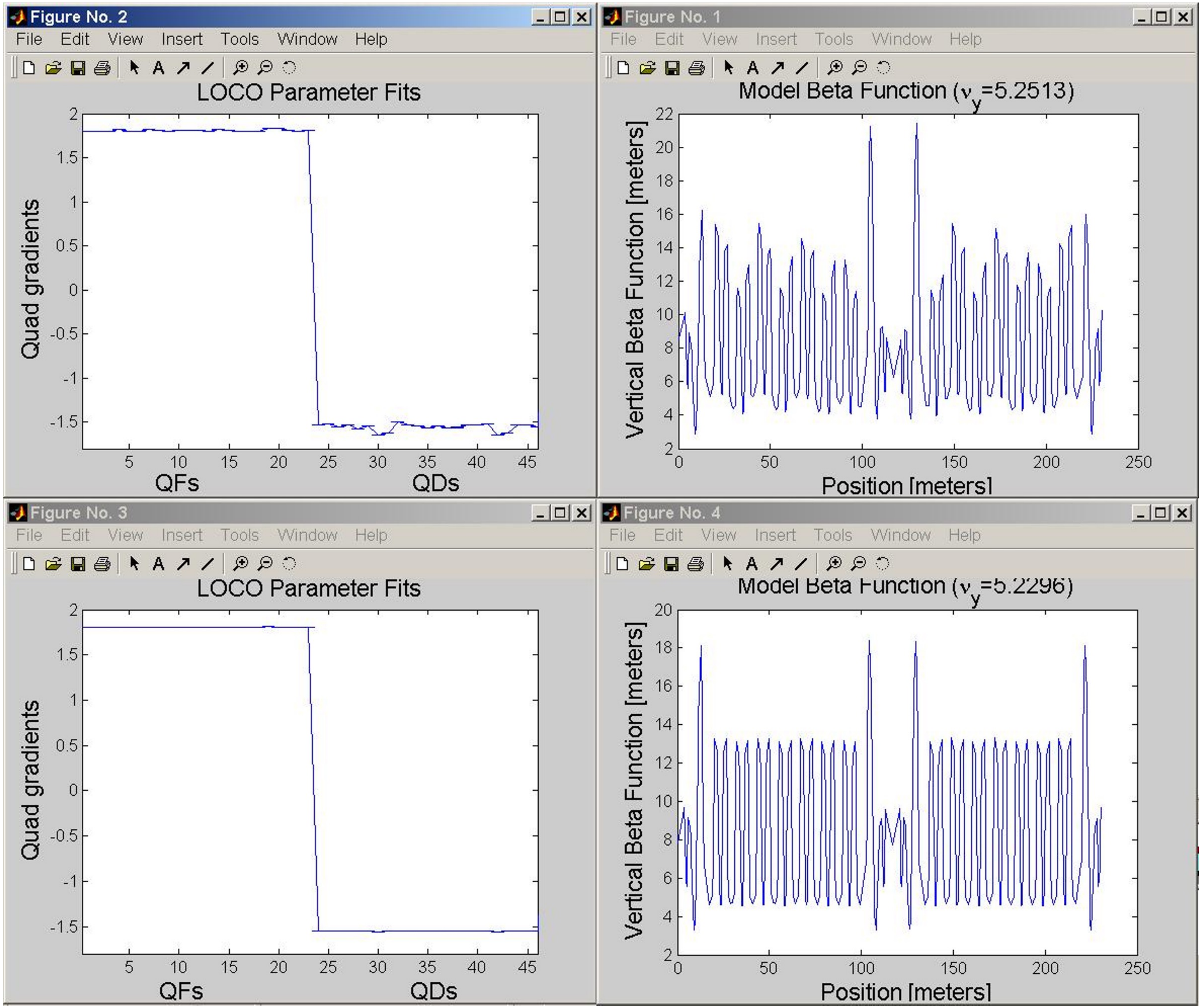
BERKELEY LAB

ADVANCED LIGHT SOURCE

# ID focusing correction

Before

After



# Spear3 Commissioning Team (partial)