The Matlab Middle Layer Suite of tools for Accelerator Simulation, Control, Measurement and data Analysis

Christoph Steier (CSteier@lbl.gov)

Lawrence Berkeley National Laboratory

Based on previous lecture by Greg Portmann (the main author)

Jeff Corbett and Andrei Terebilo, SSRL/SLAC

James Safranek (SSRL), Christoph Steier and David Robin (ALS)

Goal

- ❖ Develop an easy scripting method to experiment with accelerators (accelerator independent)
  - Remove the control system details from the physicist (like channel names and how to connect to the computer control system)
  - Easy access to important data (offsets, gains, rolls, max/min, etc.)
- ❖ Integrate simulation and online control. Make working on an accelerator more like simulation codes.
- ❖ Integrate data taking and data analysis tools
- ❖ Develop a software library of common tasks (orbit correction, tune correction, chromaticity, ID compensation, etc.)
- ❖ Develop a high level control applications to automate the setup and control of a storage ring.

History

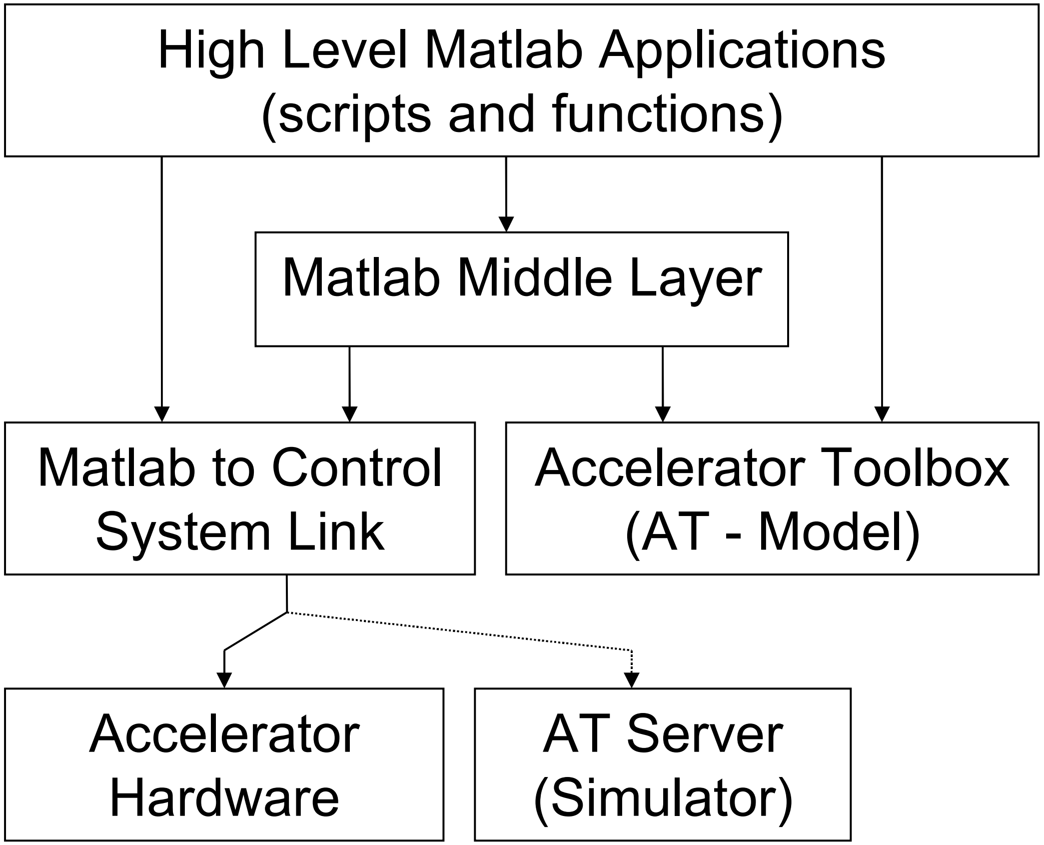
- Development of Matlab Middle Layer started >10 years ago at ALS
  - Greg Portmann, Winfried Decking, David Robin, Christoph Steier
- Other accelerators had similar requirements and went similar (or somewhat different routes)
  - APS – Tcl/TK
  - ELSA – EPOS and later Matlab
  - DESY – Matlab
- Later on AT (Terebilo, et al.) and LOCO (Portmann and Safranek) were ported to Matlab (from pascal and fortran) and combined with middle layer and many controls interfaces other than EPICS were added
- Now very widely used at more than ½ of 3rd generation light sources and some colliders (in some cases only LOCO, not full middle layer)

Why Matlab

- Matrix programming language
  (variables default to a double precision matrix)
- Extensive built-in math libraries
- Active workspace for experimentation and algorithms development
- Easy of import/export of data
- Graphics
- Compact code and good readability
- Adequate GUI capabilities
- Platform Independent

- MiddleLayer + High Level Applications
  1. Link between applications and control system or simulator.
  2. Functions to access accelerator data.
  3. Provide a physics function library.
- MCA, LabCA, SCAIII – Matlab to EPICS links
- AT – Accelerator Toolbox for simulations
- LOCO – Linear Optics from Closed Orbits (Calibration)

Software Interconnection Diagram



Basic Calling Syntax

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);
```

Examples:

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

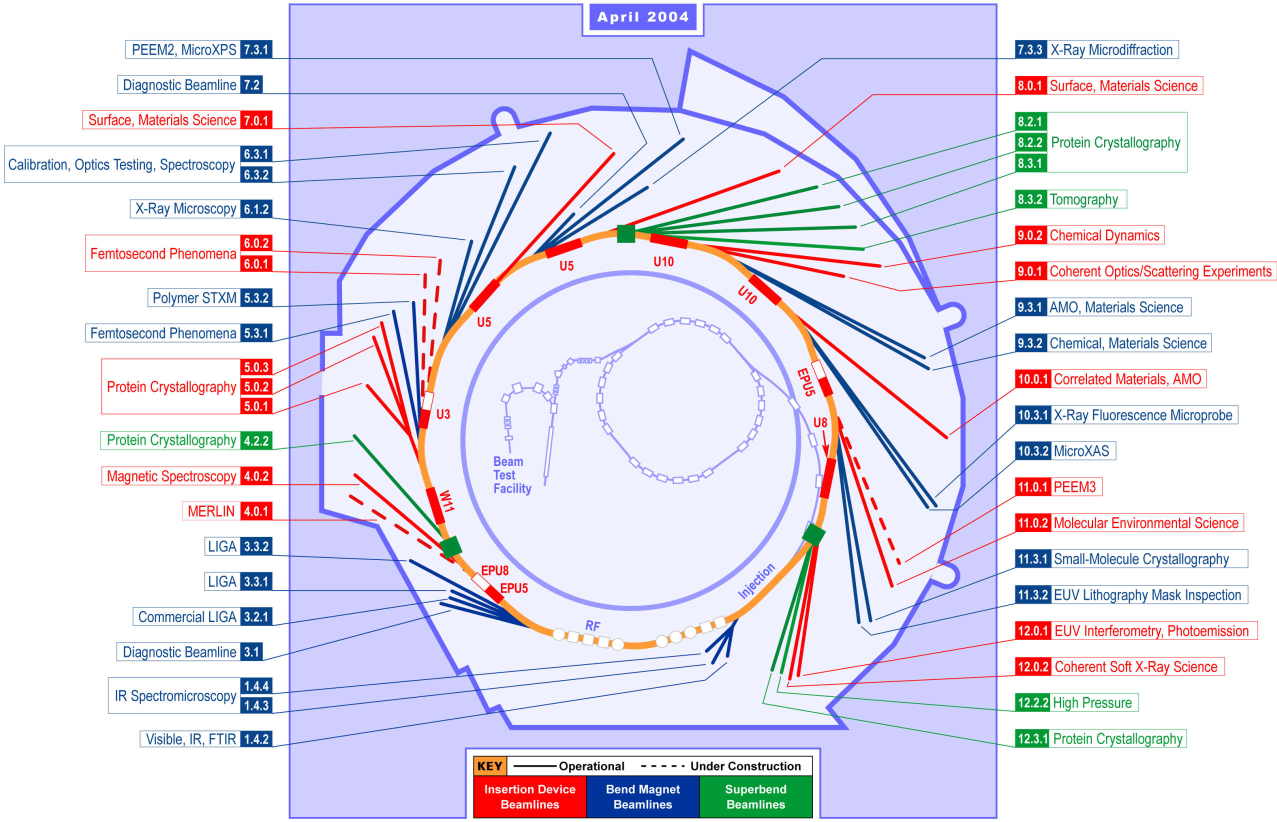
Example we will use very often: ALS

Advanced Light Source (ALS) at Berkeley, 1.9 GeV, 400 mA, optimized for VUV+soft x-rays, covers IR – hard x-rays (40 keV), 200 m circumference

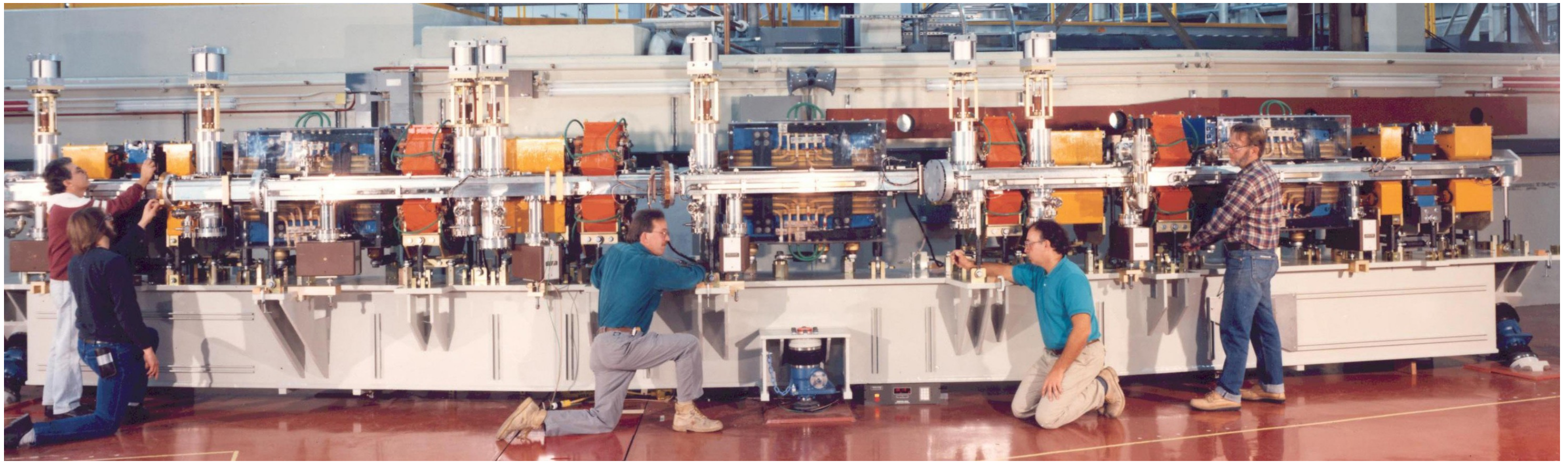
jc/ALSaerial/11-96

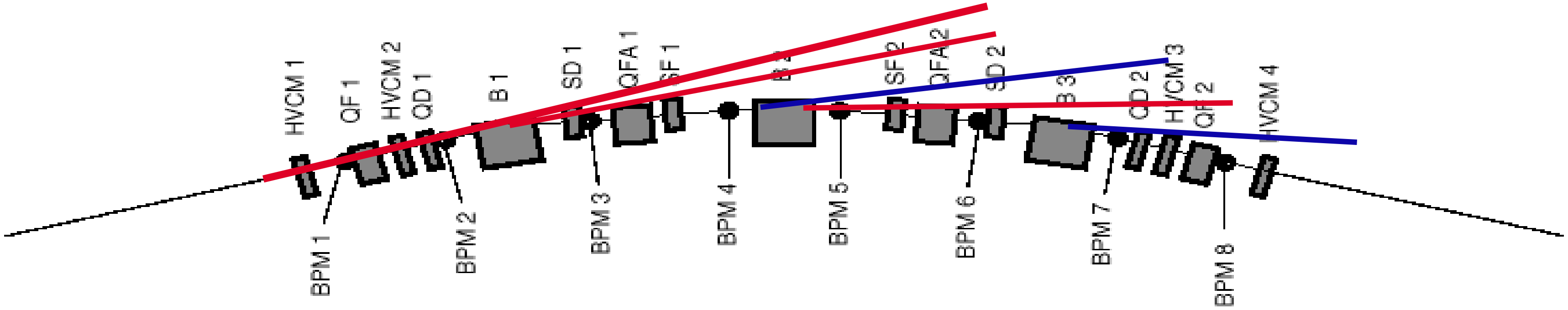
ALS Parameters and Beamlines

Nominal Energy	1.5-1.9 GeV
Circumference	196.8 m
RF frequency	499.642 MHz
Harmonic number	328
Beam current	400 mA multibunch (future 500 mA) 65 mA two-bunch
Nat. emittance	6.3 nm at 1.9 GeV
Emittance Coupling	Typical about 2% (future 0.4%)
Nat. energy spread	0.097%
Refill period	3 daily fills multibunch 12 two-bunch (future top-off about every 30 s)



One ALS Arc





- 12 nearly identical arcs – TBA; aluminum vacuum chamber
- 96+52 beam position monitors in each plane (original+Bergoz)
- 8 horizontal, 6 vertical corrector magnets per arc (94/70 total+chicanes)
- Beam based alignment capability in all quadrupoles

ALS Naming Scheme

Families

Bend magnets – BEND

Quadrupoles – QF, QD, QFA, QDA

Sextupoles – SF, SD

Skew quadrupoles - SQSF, SQSD

Correctors – HCM, VCM, VCBSC

Beam position monitors – BPMx and BPMy

Insertion devices – ID, EPU

Other - RF, DCCT, TUNE, GeV

Fields

Setpoint, Monitor, RampRate, RunFlag, DAC, OnOff,

Reset, Ready, Voltage, Power, Velocity, HallProbe, etc…

Function Library (examples)

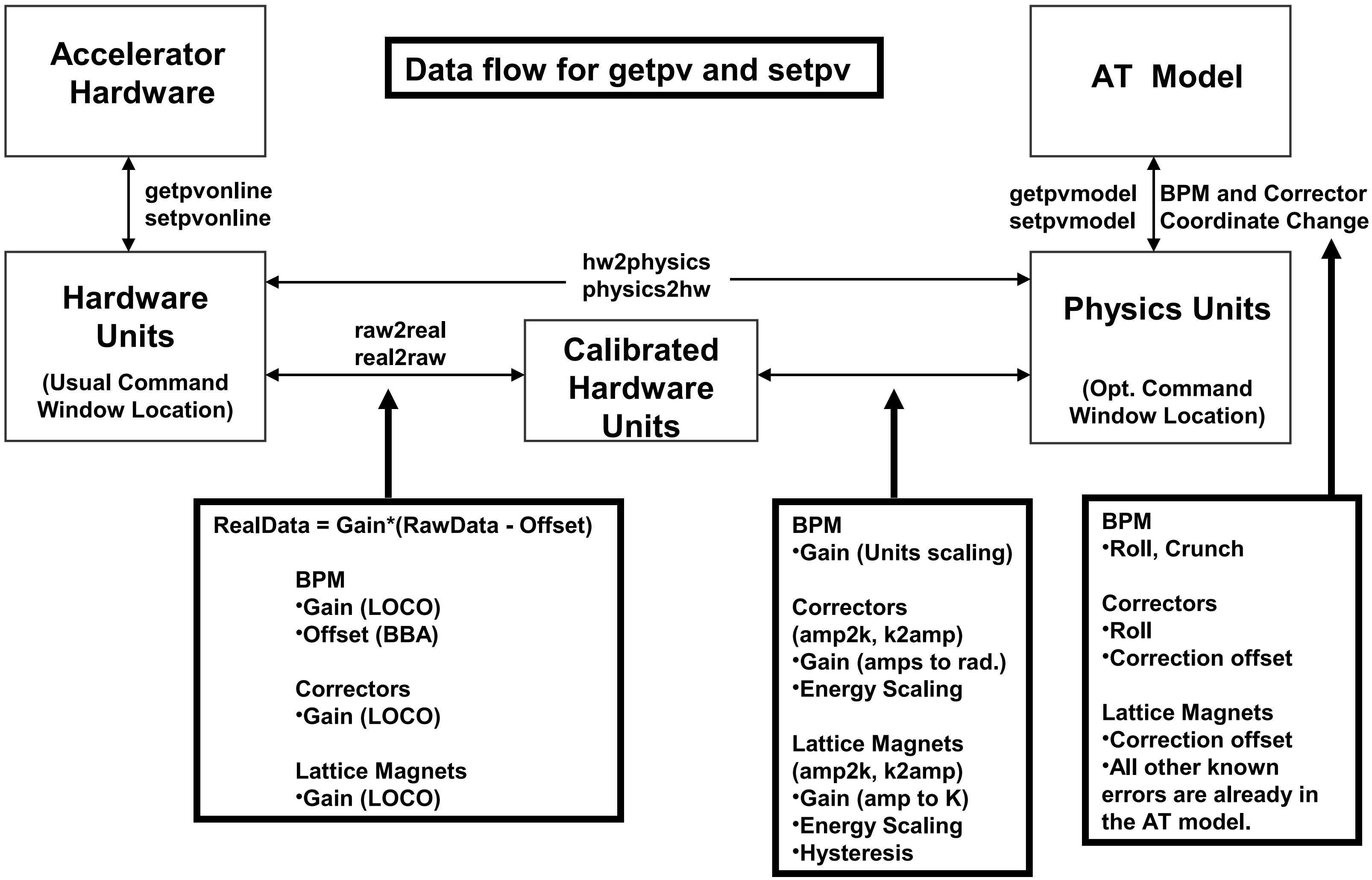
There are hundreds of functions for accelerator control

- 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

Data Management

- 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, etc.

MiddleLayer Data Flow Diagram



How to Use the Model or Online Accelerator

- Make the model the default

  >> switch2sim

- Make the accelerator the default

  >> switch2online

- Mixed mode – use keyword overrides

  'Simulator' – Run the same code as online just use the AT model for input/output.

  'Model' – Some code uses the AT model more directly (like measbpmresp or measchro)

  Note: ‘Model’ and ‘Simulator’ are often the same.

How to Switch Between Hardware and Physics Units

- Make the hardware units the default

  >> switch2hw
- Make the physics units the default

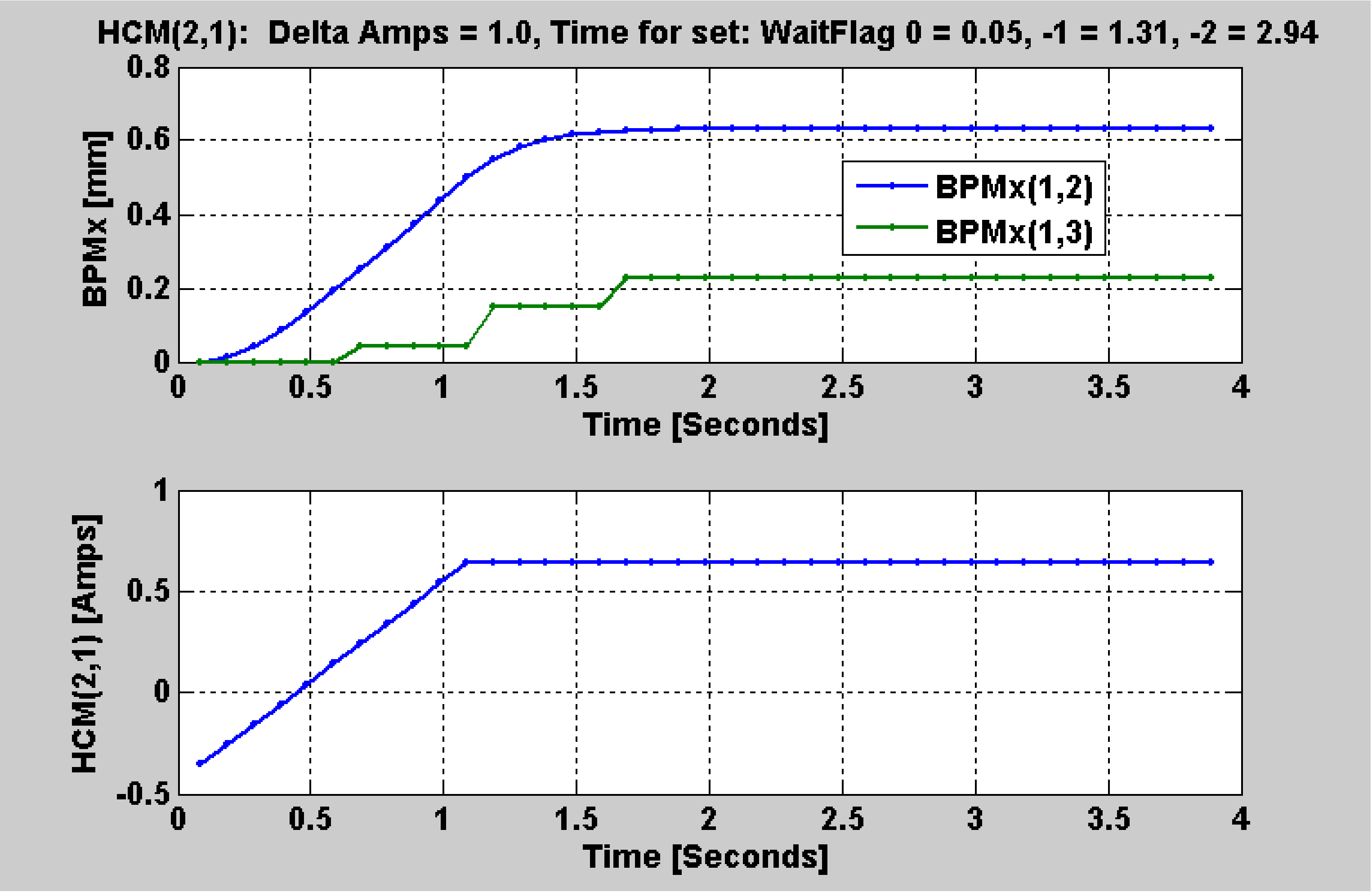
  >> switch2physics
- Mixed mode – use keyword overrides

  'Hardware' – Force hardware units for this function.

  'Physics' – Force physics units for this function.

  Example:

```
>> Amp = getpv('QF', 'Hardware');
>>   K  =  getpv('QF', 'Physics');
```



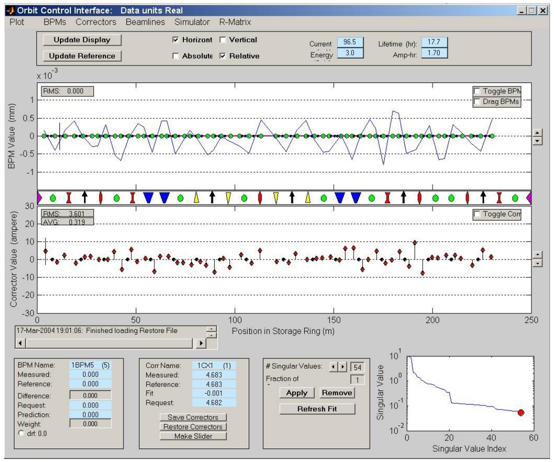
Default WaitFlag = 0

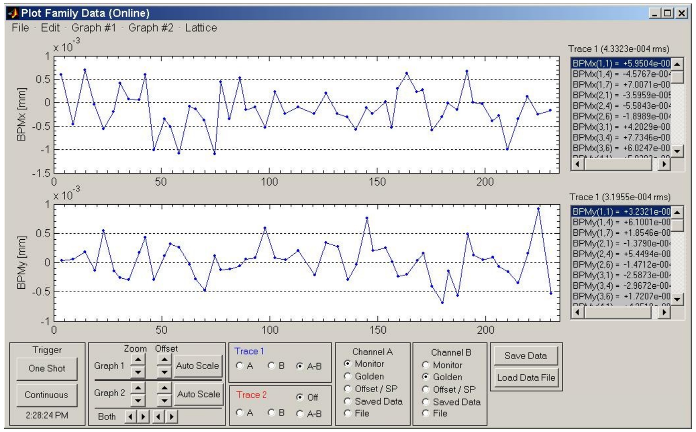
High Level Applications

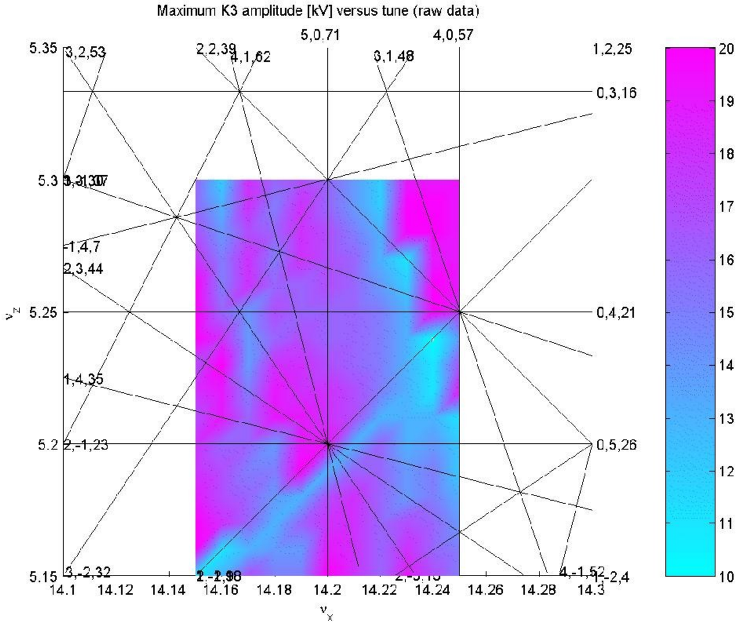
- Magnet lattice save / restore / configuration control
- Energy Ramping
- Slow orbit feedback
- Insertion device compensation
- Quadrupole centering
- Display (plotfamily) / Diagnostics
- LOCO (Response matrix analysis)

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

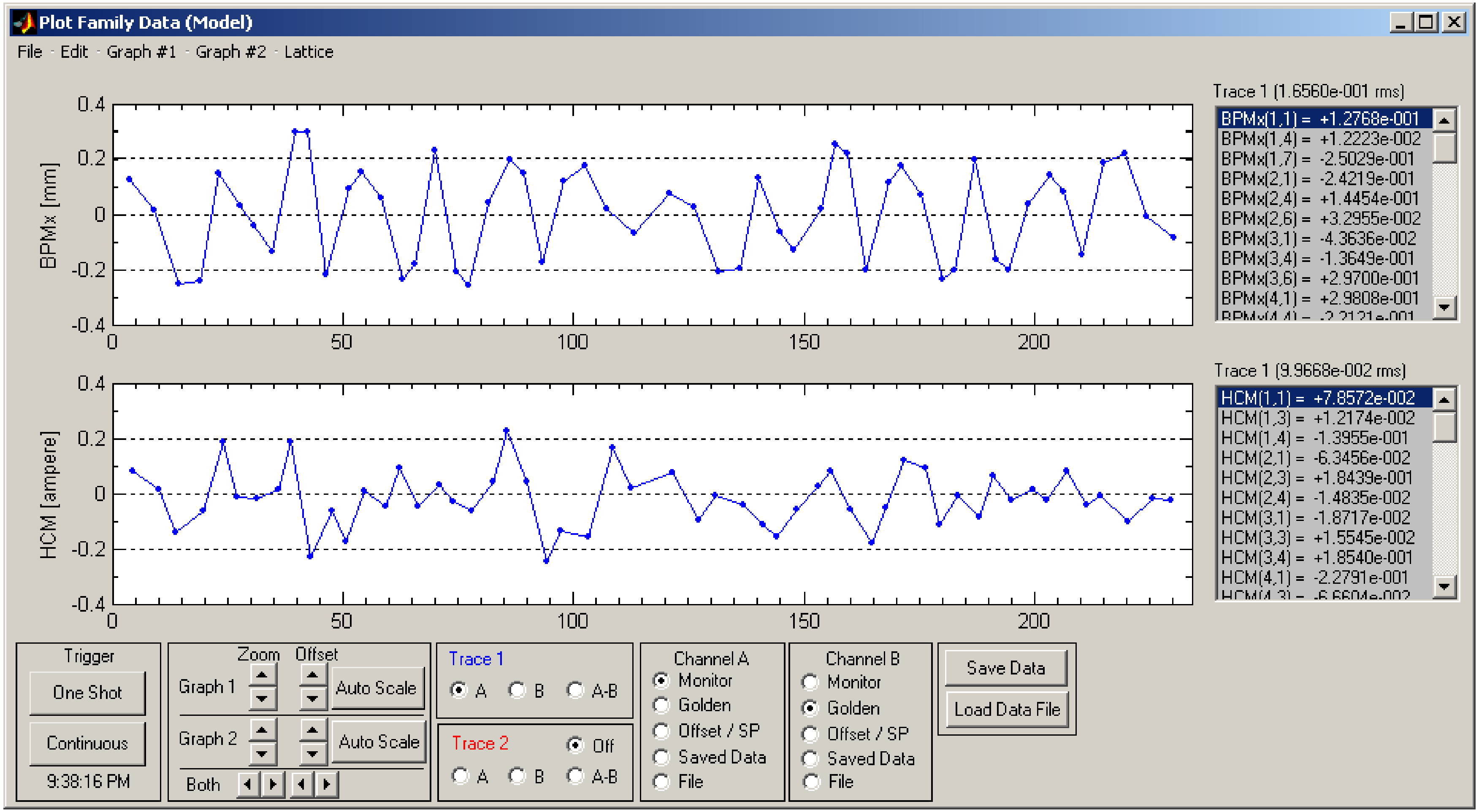






← **Dynamic aperture vs (ν_x, ν_y)**

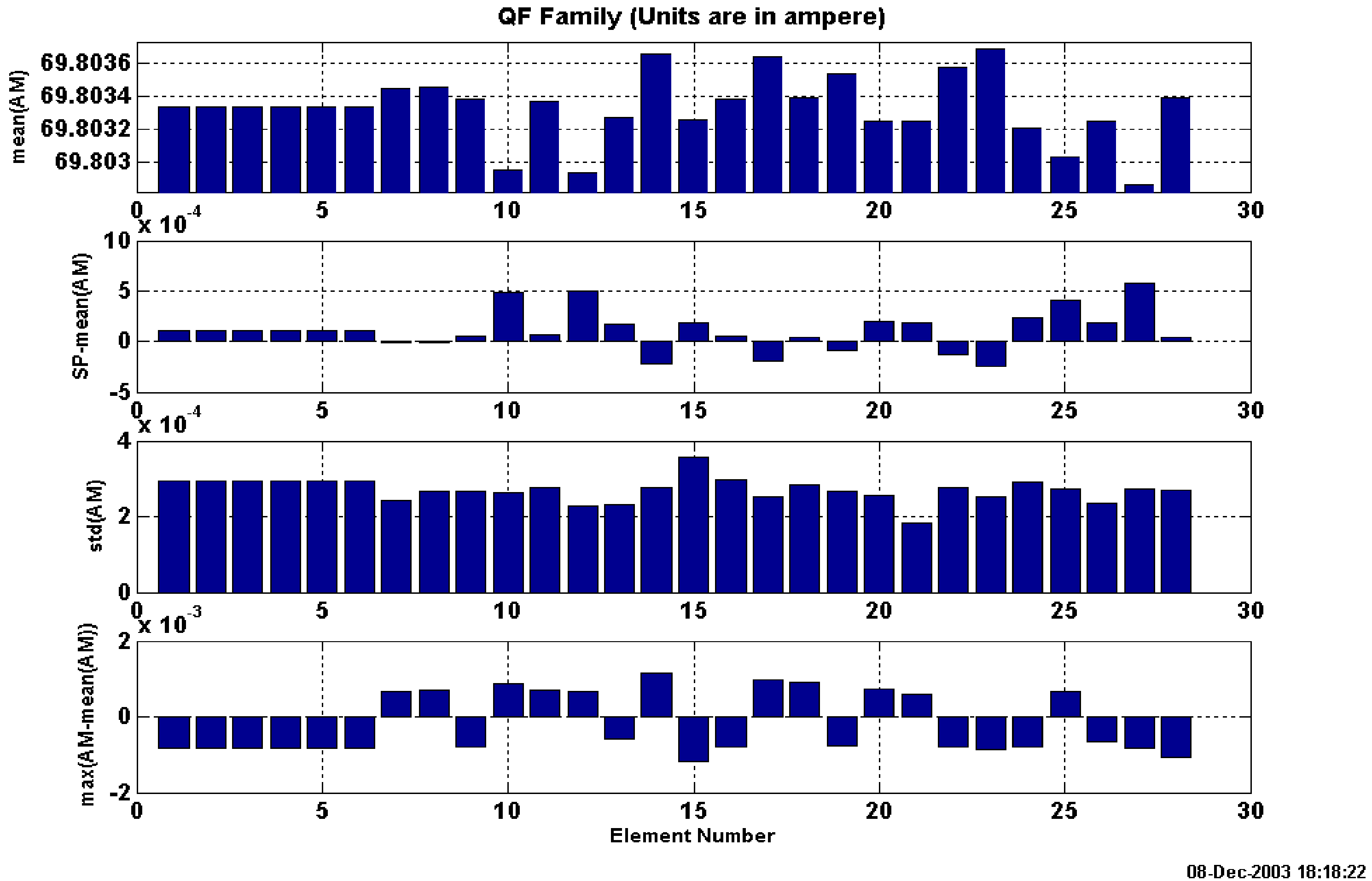
plotfamily application



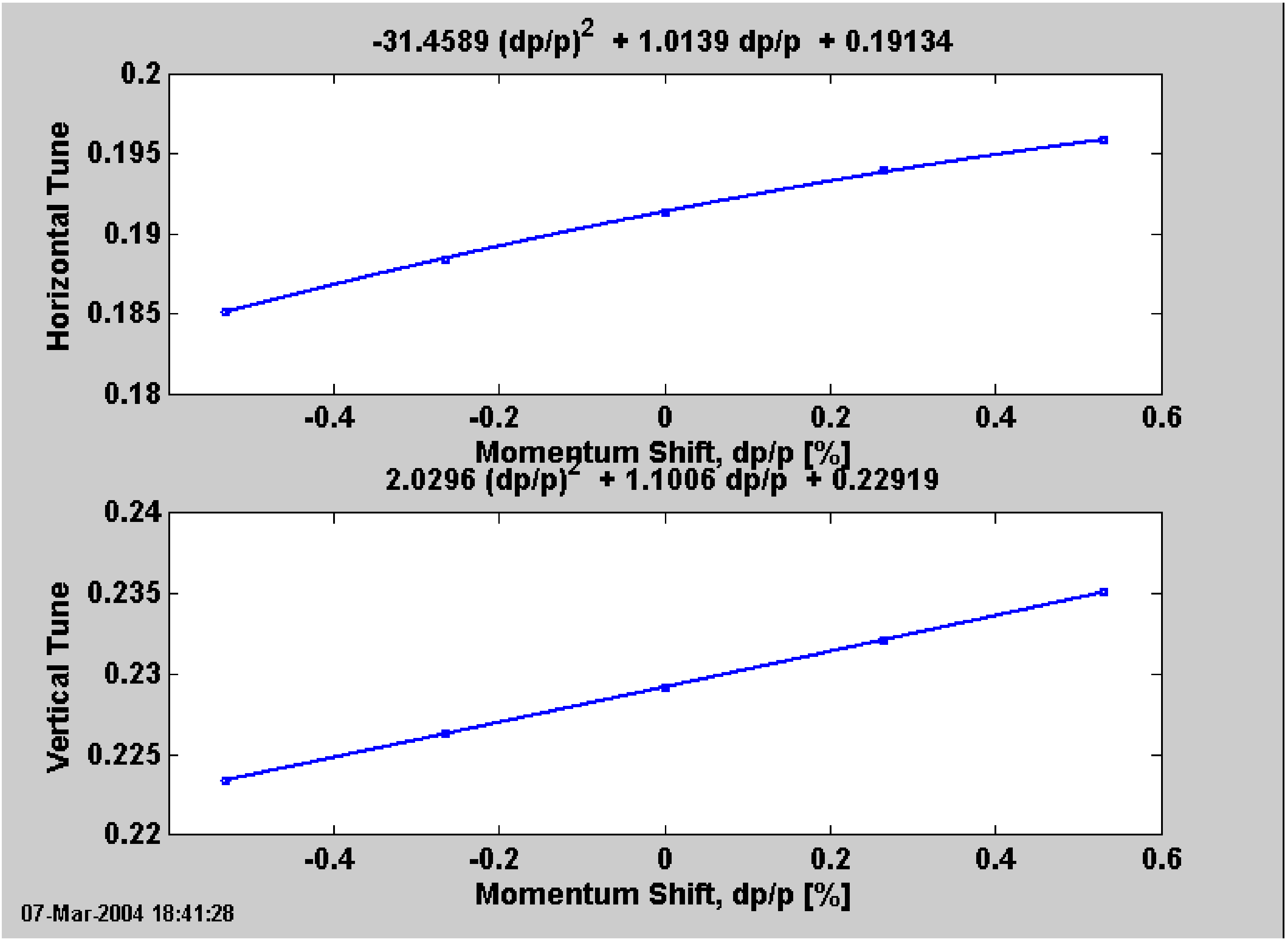
Scripting Example: Orbit Correction

```
% Create an Orbit Error
vcm = .25 * randn(73,1);     % 73 vertical correctors at the ALS
setsp('VCM', vcm);

% Get the vertical orbit
Y = getam('BPMy');

% Get the Vertical response matrix from the model
Ry = getrespmat('BPMy', 'VCM');       % 122x73 matrix

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

% Find the corrector changes use 48 singular values
DeltaAmps = -V(:,Ivec) * S(Ivec,Ivec)^-1 * U(:,Ivec)' *  Y;

% Changes the corrector strengths
stepsp('VCM', DeltaAmps);
```

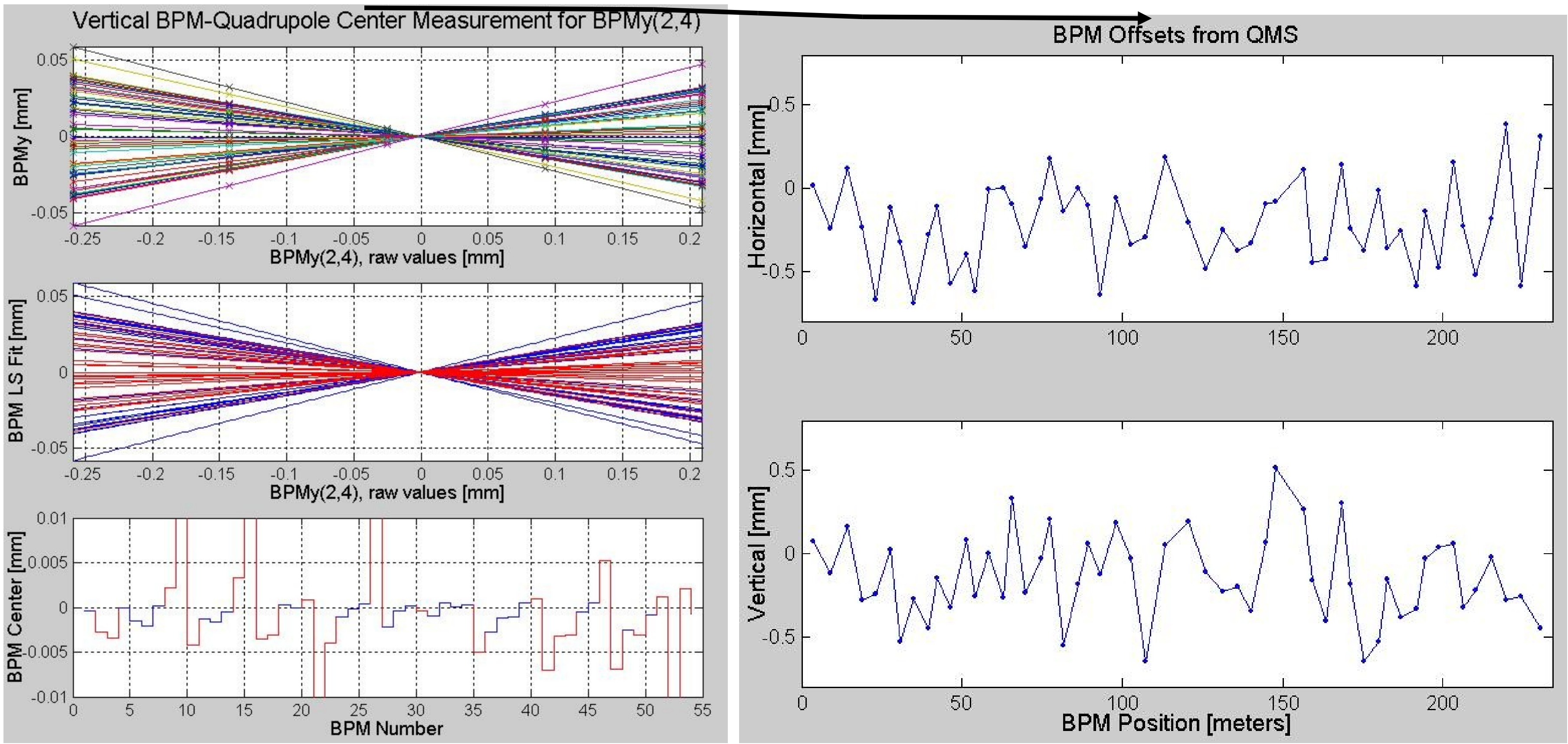
Finding Power Supply Errors (monmags)



Chromaticity Measurement



Beam-based alignment

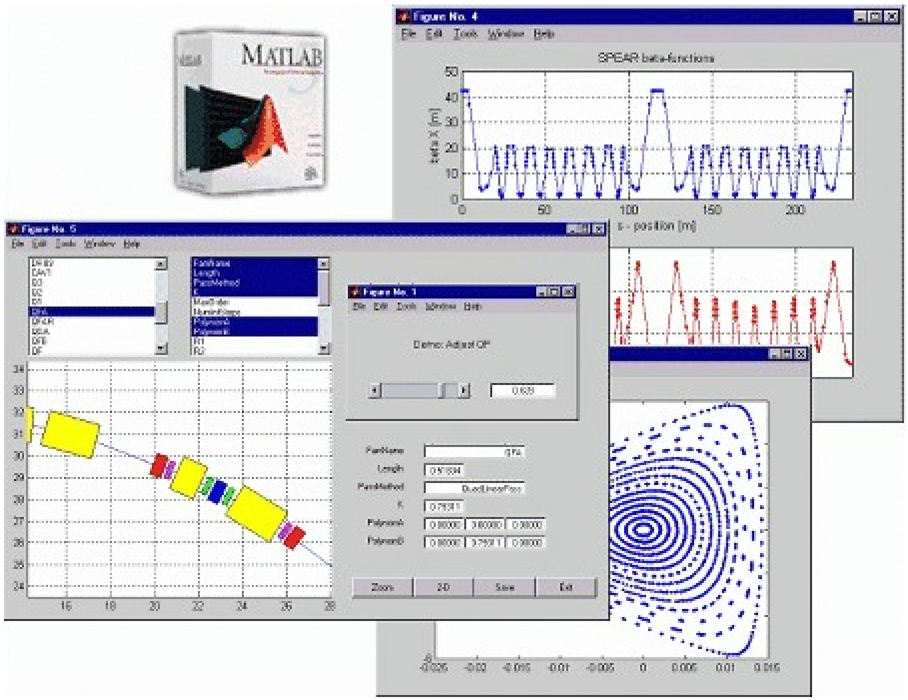


AT – Accelerator Toolbox

Andrei Terebilo

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.

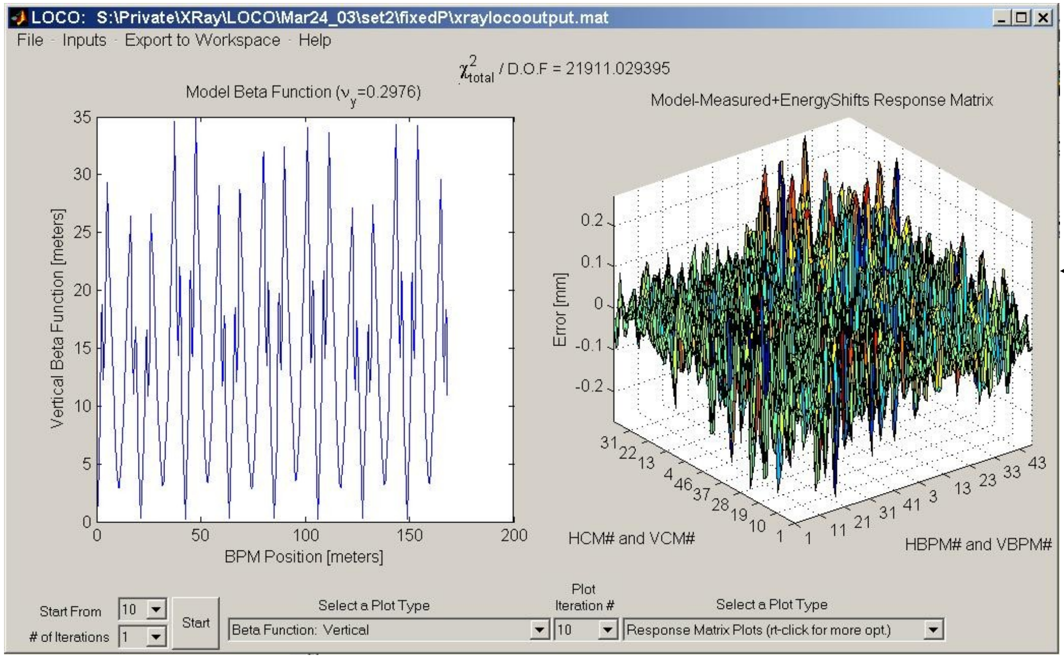


What is Accelerator Toolbox
New in AT version 1.2
Download and Installation
Get Started
Collaboration
Publications
e-mail AT
Links

www-ssrl.slac.stanford.edu/at/welcome.html

LOCO Optics Analysis

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



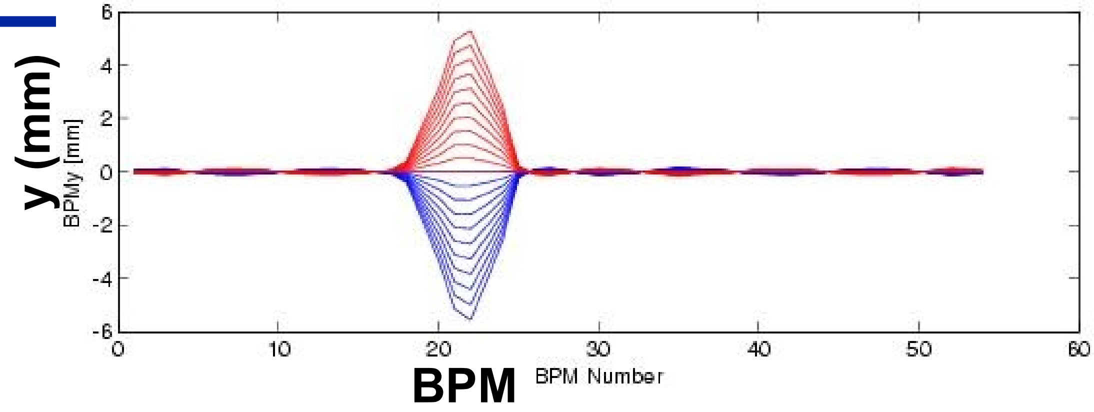
← New MATLAB version of code

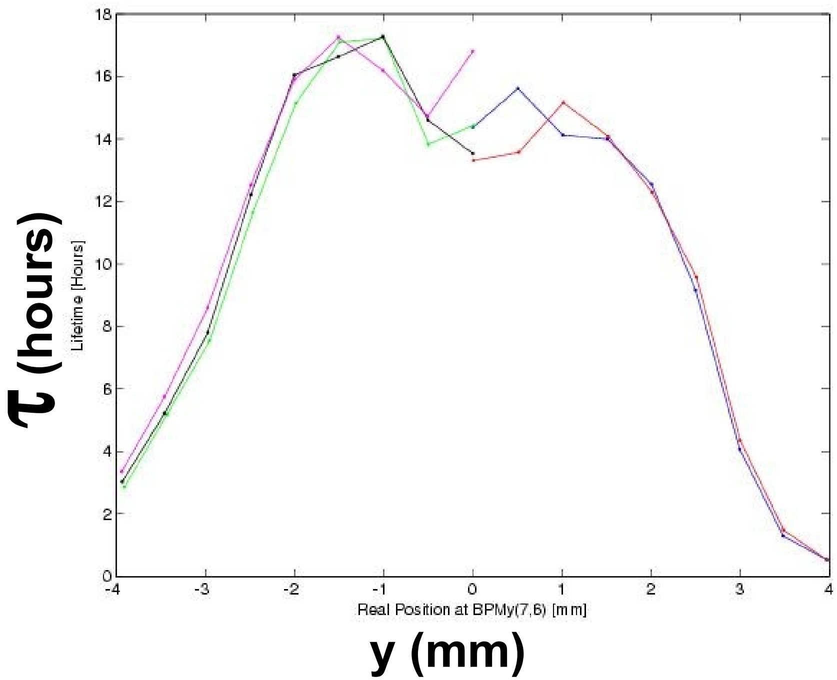
- rewritten from FORTRAN
- linked to control system
- linked to AT simulator

Aperture scans

- ❖ τ vs y in small gap ID
  - Poor man's scraper
- ❖ Determine minimum gap for future IDs
  - 12 mm ⟶ 8 mm
  - Smaller still for reduced β_y
- ❖ Complications
  - Vacuum degrades with beam bump
  - Coupling, η_y degrades with beam bump
  - Need scraper

Vertical beam bump in ID chamber



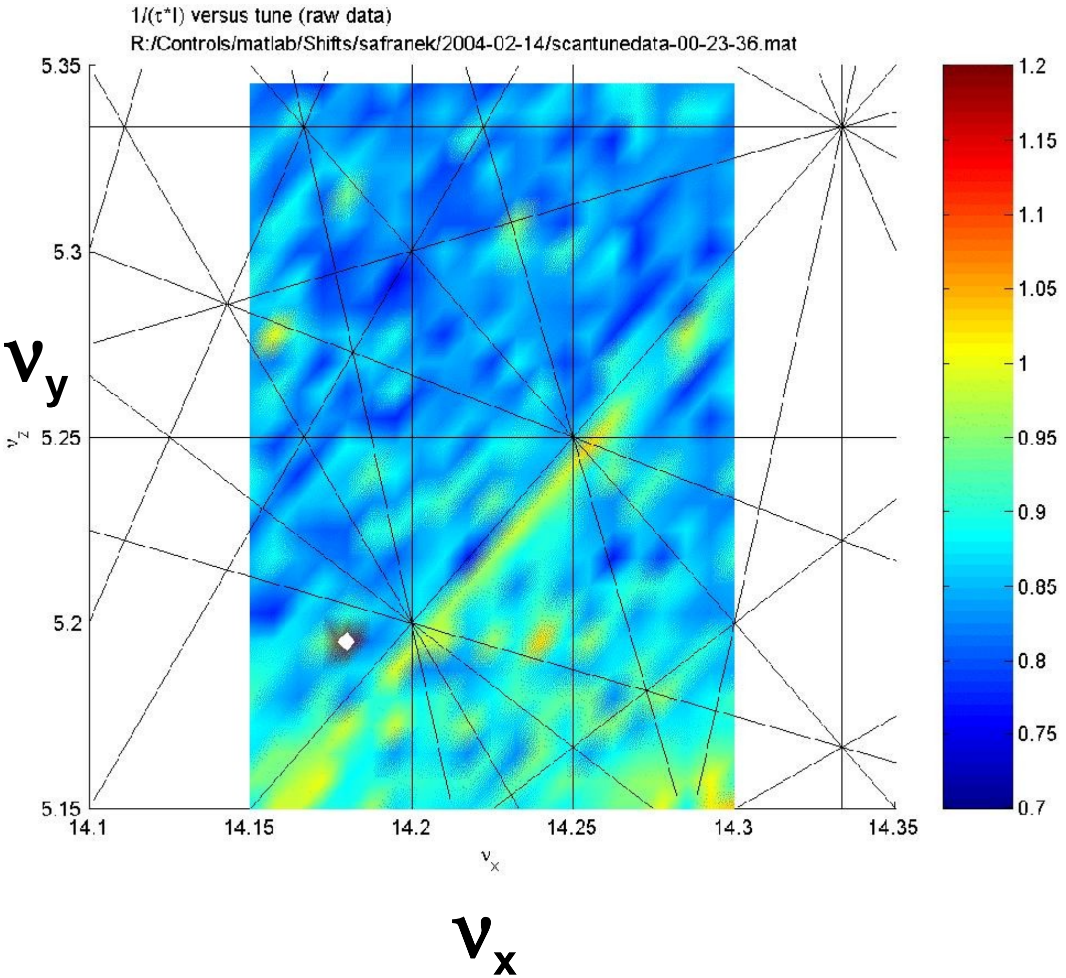


Lifetime vs. Tunes

- Resonant line:

  $\forall\ \nu_x - \nu_y = 9$

- Operating tunes (5.19, 6.23)
- Data gathered automatically on owl shift.



Conclusion

- Relatively easy to use. Most people start writing useful scripts in a few hours.
- MiddleLayer + LOCO + AT + MCA/LabCA/SCAIII cover many of the high level software concerns for storage rings. 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.
- >10 light sources are active or semi-active MiddleLayer users -- ALS, Spear, BNL (vuv and x-ray ring), CLS, Duke, FEL, PLS, Australian light source, DIAMOND, and Soleil. DESY, ALBA, and SSRF are experimenting with it (maybe more).
- The semi-machine independence software has fostered collaboration and code sharing between the laboratories.