

SNS Application Programming Infrastructure and Physics Applications (*XAL*)

Paul Chu

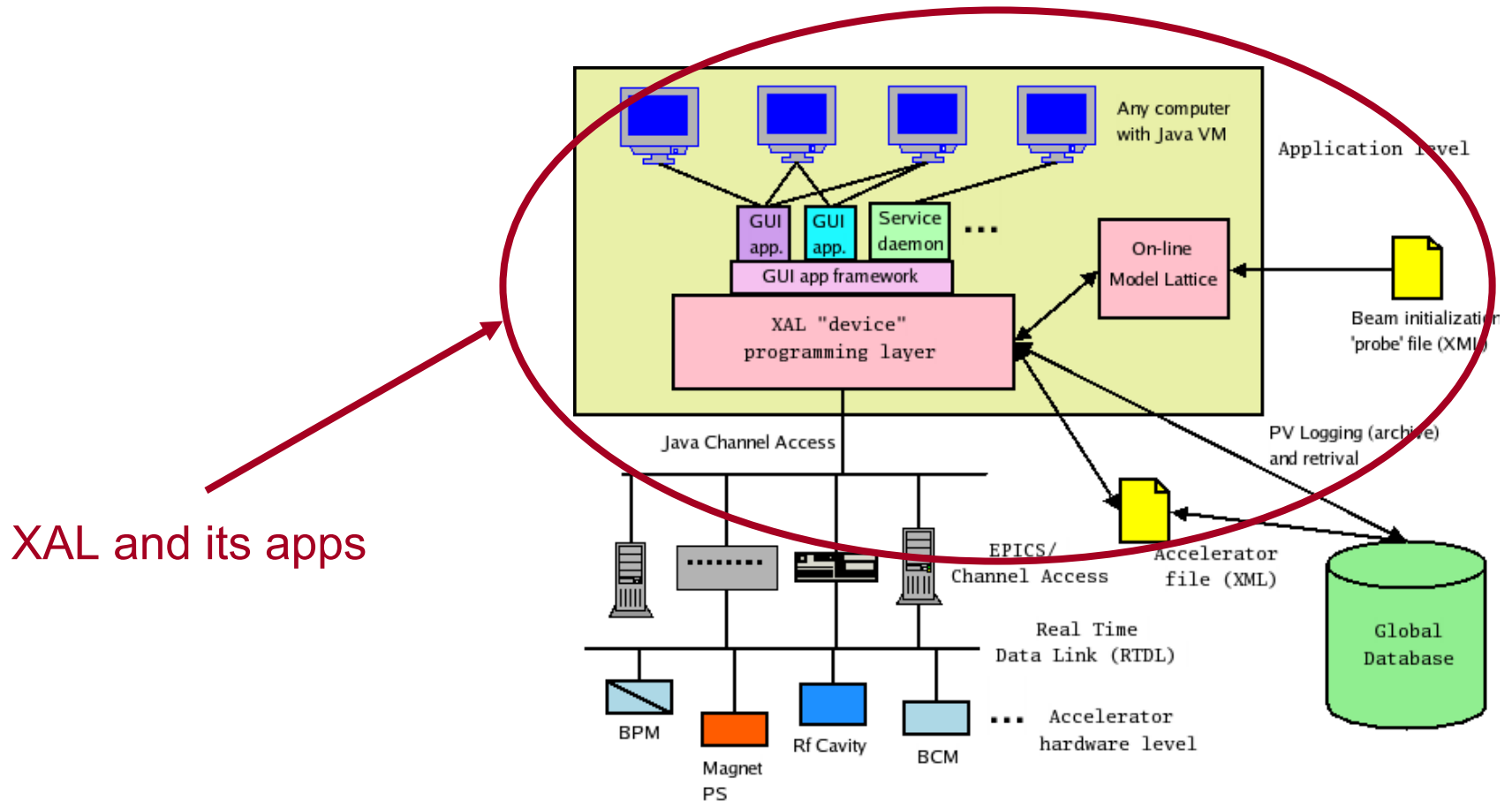
March 24, 2006

- SNS Application Programming Team: P. Chu, J. Galambos, T. Pelaia, A. Shishlo
- Active application programmers at SNS: D. N. Armstrong, S. Cousineau, V. Danilov, Y. Zhang + database team
- Other contributors: A. Aleksandrov, C.K. Allen, S. Bunch, I. Campisi, S. Chevtsov, R. Dalesio, K. Danilova, A. Feshenko, D. Gurd, S. Henderson, J. Holmes, D. Jeon, R. Kennedy, W.D. Klotz, Y. Kiselev, I. Kriznar, A. Leahman, C. McChesney, N. Malitsky, D. Ottavio, N. Pattengale, M. Plesko, M. Plum, A. Pucelj, H. Sako, C. Sibley, E. Tanke, J. Wei, E. Williams, and A. Zupanc...

- SNS application infrastructure (XAL)
 - Overview
 - Accelerator hierarchy
 - Database
 - Connections
 - Data Correlation
 - Online Model
 - GUI framework
 - Utility tools
- Applications

XAL Overview

- High level physics application software is for modeling, integrated operation and accelerator physics studies
- SNS application software environment --



XAL Accelerator Hierarchy

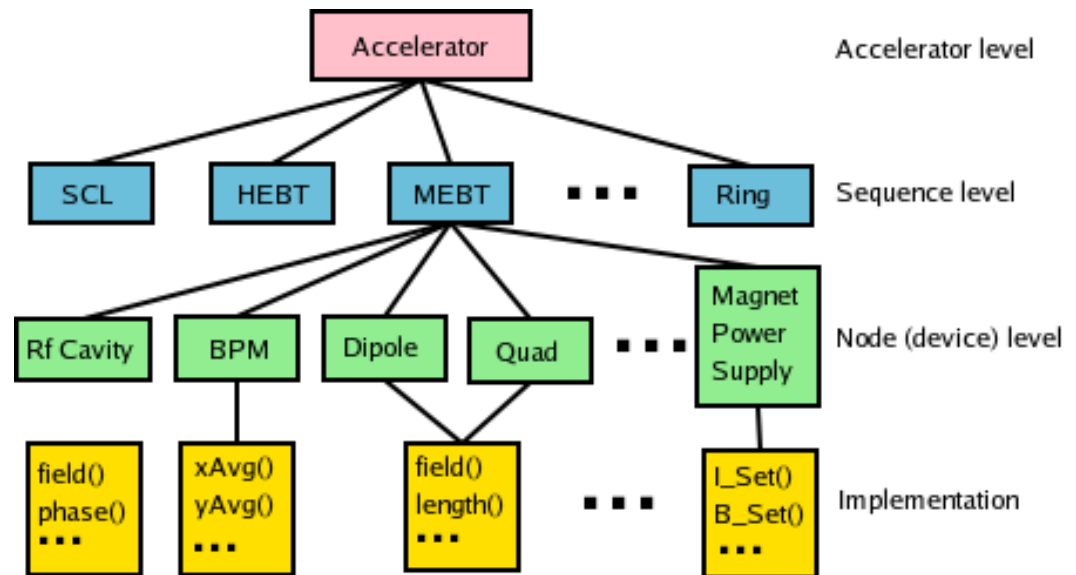
- XAL is Java based
- Includes a class hierarchy describing the accelerator structure
- Methods exist to directly work with accelerator devices

Java.lang.Object

AcceleratorNode
(base class)

AcceleratorSeq

Accelerator



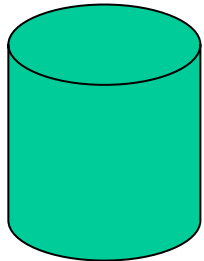
The Database Connection

- XAL uses “Beamline Device”, “Device Setting” and other hardware related tables in the global SNS database
- Creates the accelerator hierarchy
- Serves as a common configuration for all applications
- Provides the “map” from the flat list of EPICS signals to the accelerator hierarchy
- Use an intermediate XML file

Database



XML configuration files



```
<node type="QH" id="MEBT_Mag:QH01" pos="0.135" len="0.061">
  <attributes>
    <magnet len=".061" polarity="-1" dfltMagFld="-34.636"/>
    <align x="10000" y="2000.046" z="19546.440844" pitch="0" yaw="0" .../>
  </attributes>
  <ps main="MEBT_Mag:PS_QH01" />
  <channelsuite name="magnetsuite">
    <channel handle="fieldSet" signal="MEBT_Mag:QH01:fieldSet" .../>
  </channelsuite>
</node>
```

- On-the-fly editing for quick across the board fixes

The Control System Connection

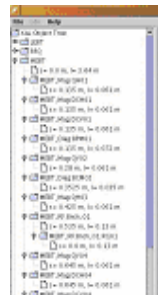
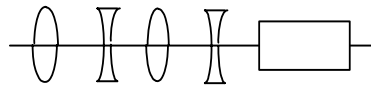


- Channel Class
 - An abstract class that provides an interface to the control system.
 - Includes convenience functions to hide details of the connection mechanism.
- JcaChannel
 - A concrete class that uses the Java Channel Access (JCA) interface to EPICS channel access protocol -- each EPICS signal is called a *process variable (PV)*.
- Transformations: allows modification of the value of the signal coming from the control system
 - Can map one signal to > one “Channel” object, with and without the transformation.
 - Facilitates quick fixes, changes to signals
 - Magnetic field calculated from power supply current.
 - BPM polarity fixes.

The Online Model

- Calculate beam parameters.
- A lattice view of the machine is constructed from the “device” structure (via a set of rules).
 - Drifts are added, elements are split.
 - Device view -> intermediate lattice -> online model lattice
- Lattice element values can be updated from the **machine**, **design** or **logged** values.
- Can do ‘**what-if**’ with any one of the above data sources.
- Mostly use an envelope model for single-pass linac tracking or closed orbit for ring.

Device View
(Stored in XAL)



- Single entry per element

- Only physical devices

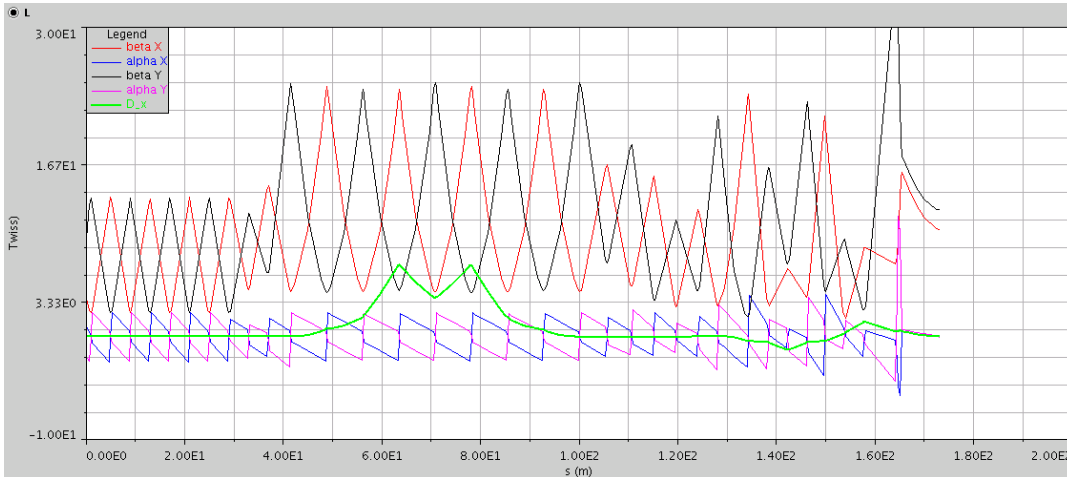
Lattice View
(used in modeling)



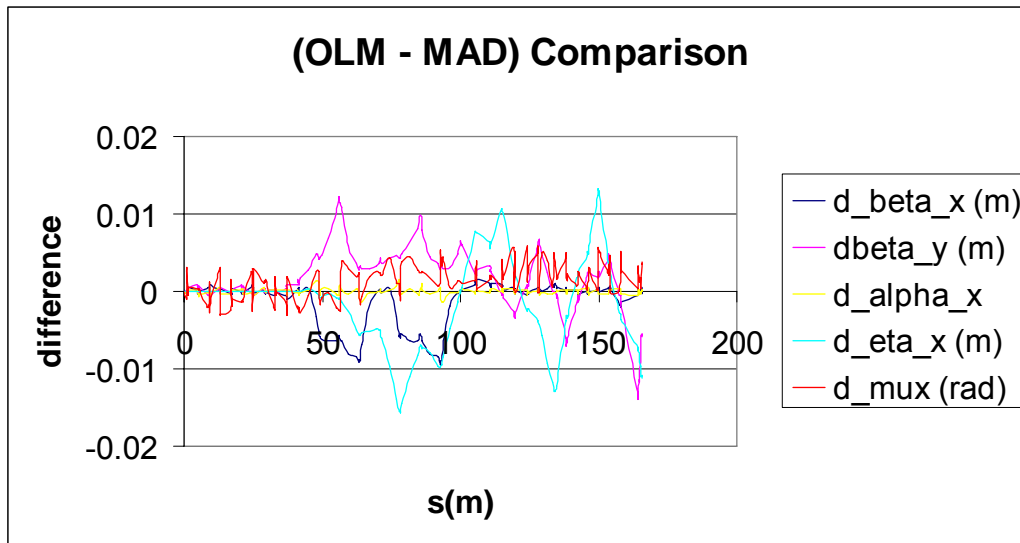
- Elements may be split

- Includes drifts

Online Model vs. MAD



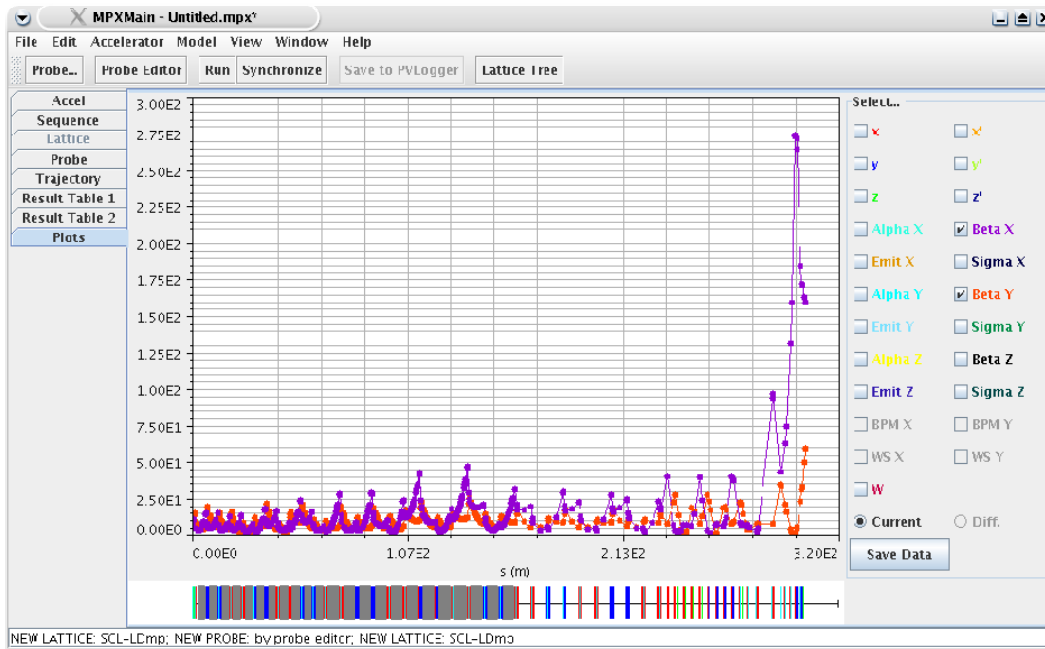
Online model
HEBT results
(design setting)



Comparison of
online model and
MAD results

Online Model (cont.)

- Many applications based on the online model.
- A 'virtual accelerator' application using online model and a portable channel access server (PCAS) as the engines.



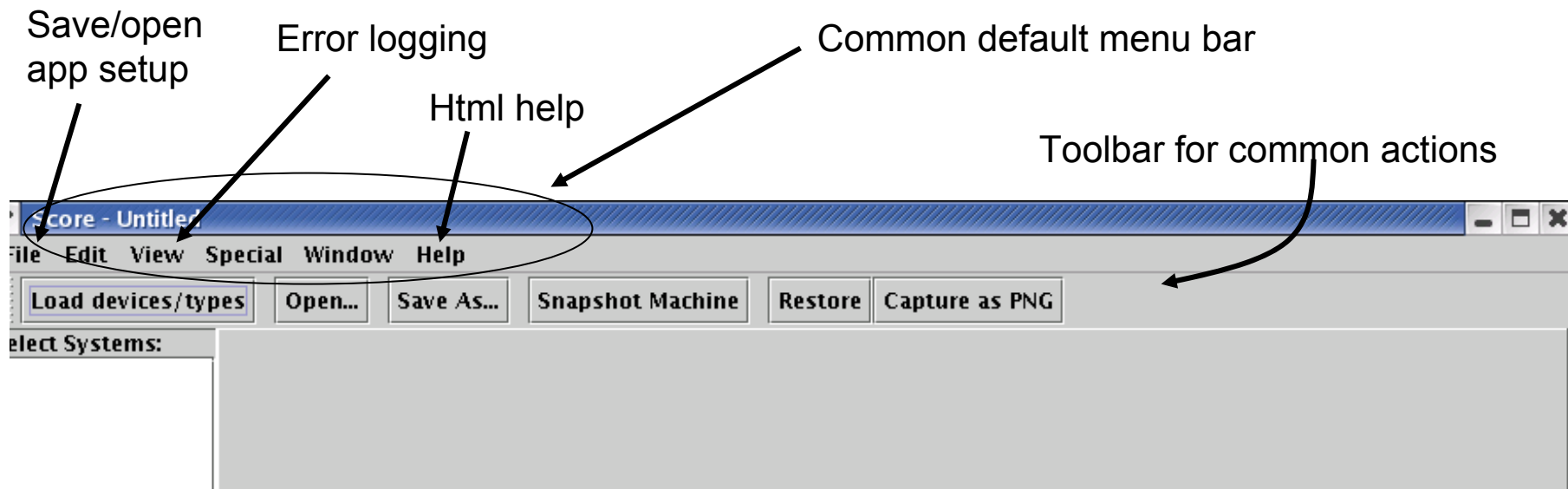
•Beta functions through the SNS SCL and Linac dump line for a machine snapshot during Sep. 05 commissioning.

•The app can server as orbit difference tool with live BPM and profile monitor data.

- Orbit difference
- Orbit correction (also has empirical algorithm)
- Online model application
- Linac tuning apps
 - PASTA (for linac phase/amplitude)
 - SLACS (for superconducting phase setting)
 - Delta-T procedure
- Ring tuning apps
 - Injection
 - Ring measurement and optics tuning (beta-function, tune, etc.)

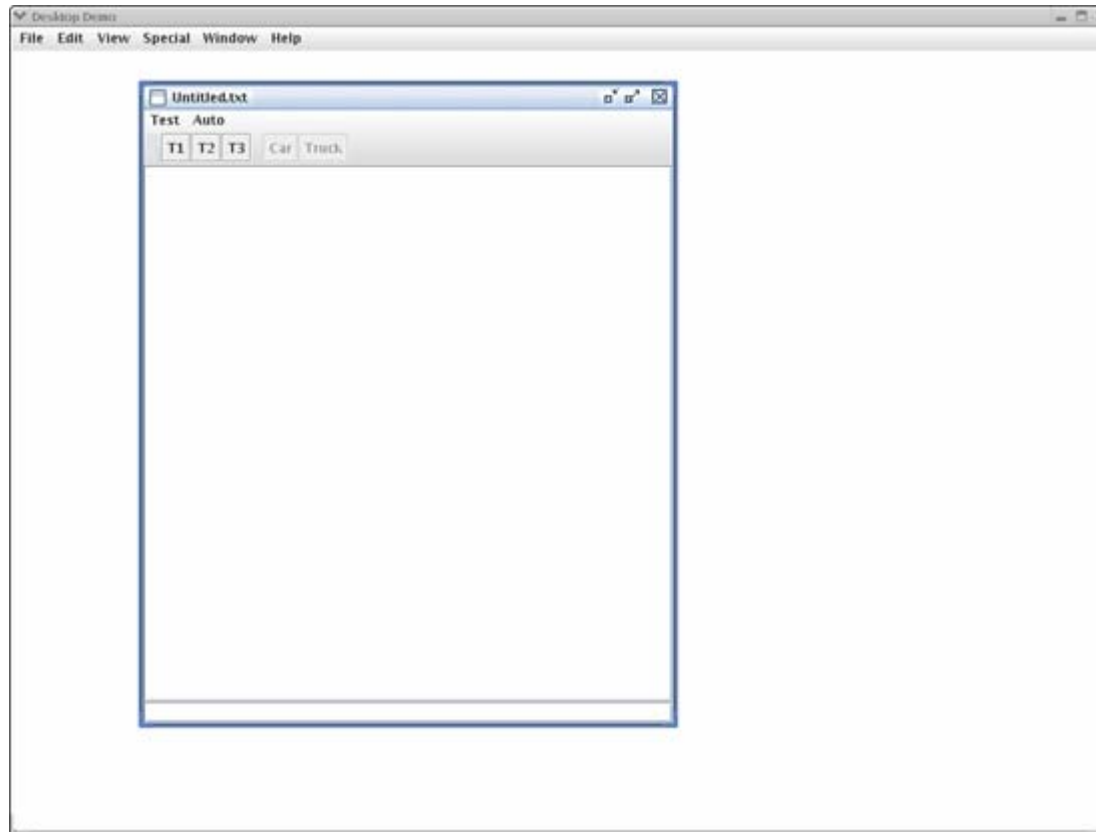
Application Programming Framework

- A GUI Application Framework is developed and used as a common starting point for application programs
 - Provides a common look feel for all apps
 - Quick jump-start for application development
 - Easy retro-fixes across many apps
 - Use familiar “windows” look feel paradigm



Application Programming Framework (cont.)

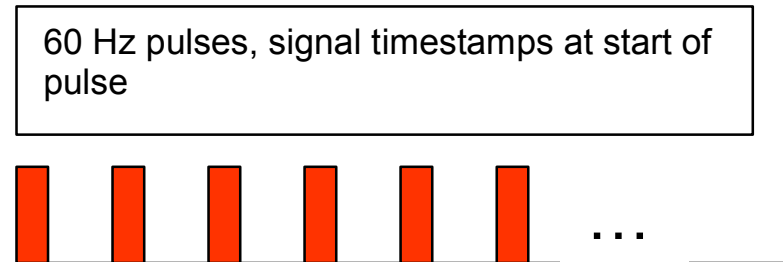
- A new “desktop” type GUI frame is developed (request from ANL).
 - Documents are within a big frame
 - Require a different way of initialization



Tools (Channel Correlator)

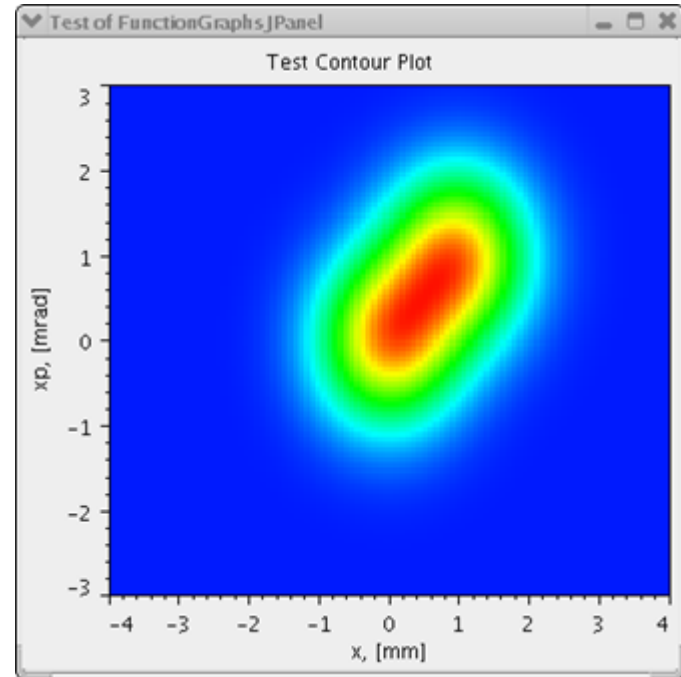
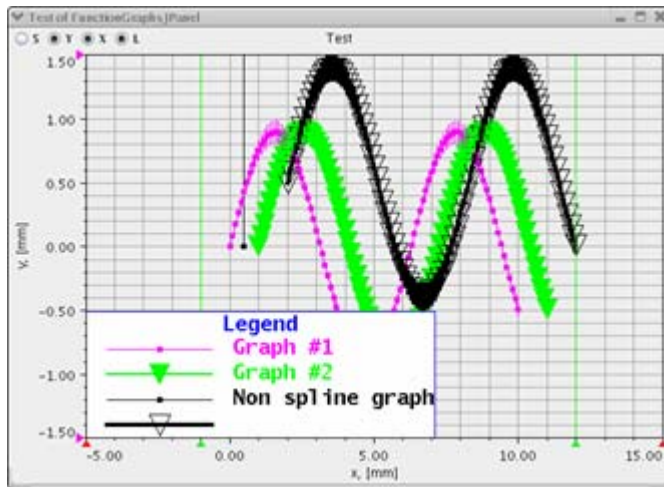


- Channel Correlator
 - SNS is a pulsed machine (up to 60Hz of rep rate)
 - Correlator gathers sets of signals from the same pulse
 - Different modes: stream correlation sets to listeners or periodic posting
 - Can use filters (triggered data acquisition)
 - RTDL (Real Time Data Link) ensure correct time-stamps for PV correlation



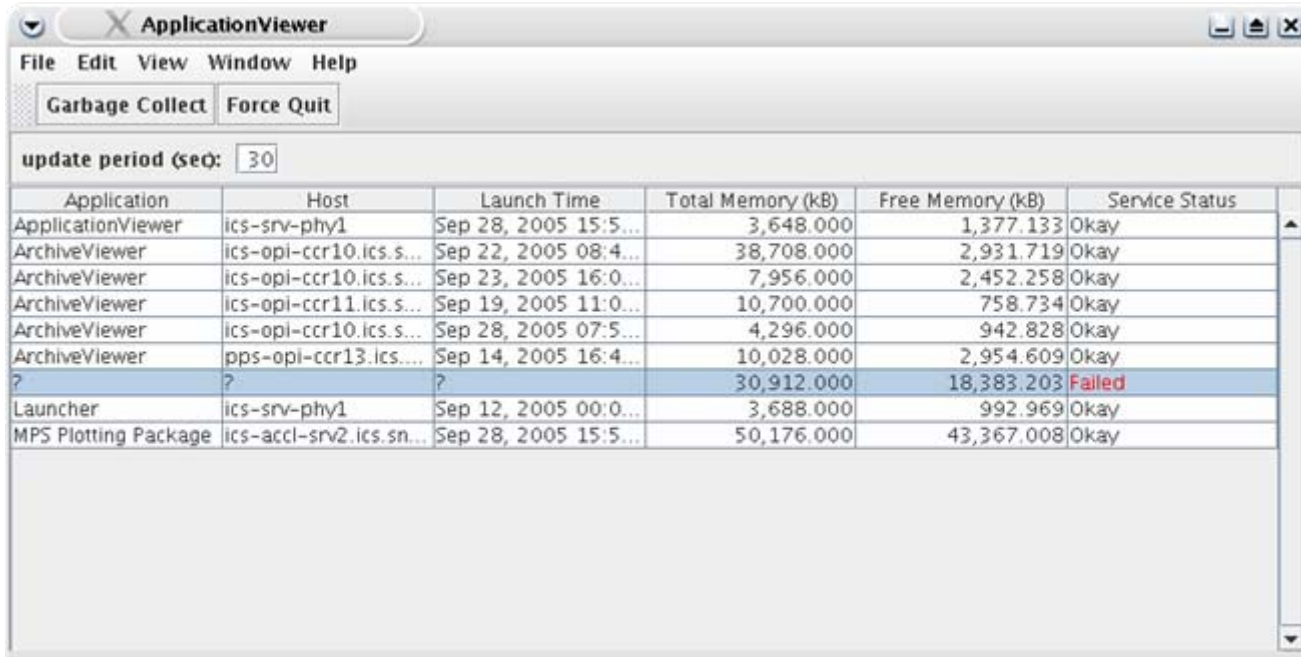
Tools (data plot package)

- Charting package
 - Designed for scientific purpose.
 - Can be light-weight and fast (e.g. for digital scope application).
 - Feature highlights:
 - Fast update option
 - Remove data points
 - 3D support
 - Range selection
 - Can be a stand-alone package



Tools (client-service)

- Communication tool (based on XML-RPC) ready for client-service implementation.



The screenshot shows a window titled "ApplicationViewer" with a menu bar (File, Edit, View, Window, Help) and two buttons: "Garbage Collect" and "Force Quit". Below the buttons is a text field for "update period (sec)" with the value "30". The main area contains a table with the following data:

Application	Host	Launch Time	Total Memory (kB)	Free Memory (kB)	Service Status
ApplicationViewer	ics-srv-phy1	Sep 28, 2005 15:5...	3,648.000	1,377.133	Okay
ArchiveViewer	ics-opi-ccr10.ics.s...	Sep 22, 2005 08:4...	38,708.000	2,931.719	Okay
ArchiveViewer	ics-opi-ccr10.ics.s...	Sep 23, 2005 16:0...	7,956.000	2,452.258	Okay
ArchiveViewer	ics-opi-ccr11.ics.s...	Sep 19, 2005 11:0...	10,700.000	758.734	Okay
ArchiveViewer	ics-opi-ccr10.ics.s...	Sep 28, 2005 07:5...	4,296.000	942.828	Okay
ArchiveViewer	pps-opi-ccr13.ics....	Sep 14, 2005 16:4...	10,028.000	2,954.609	Okay
?	?	?	30,912.000	18,383.203	Failed
Launcher	ics-srv-phy1	Sep 12, 2005 00:0...	3,688.000	992.969	Okay
MPS Plotting Package	ics-accl-srv2.ics.sn...	Sep 28, 2005 15:5...	50,176.000	43,367.008	Okay

- DataTable
 - More powerful than the standard Java collections
 - Provides database like functionality with easy interface
- PV Logger
 - “baby” archiver
- XML DataAdaptor
 - Easy file i.o. in XML format, for any hierarchy
 - Used in applications for saving settings etc.
- Math Tools
 - Optimization
 - Matrix
 - Fitting
 - ...
- Can export other modeling input files (via intermediate lattice)
 - Trace-3D, MAD
 - DYNAC, IMPACT are in progress

Scripting Interface to XAL



- Scripting interfaces are available with Jython (www.jython.org) and Matlab
 - No glue code, or extra compile steps etc needed!!! Mix XAL java classes seamlessly with scripting language
- Providing simple code examples.
- Coding up on-the-fly experiments.
- <http://www.sns.gov/APGroup/appProg/xal/scripts/jythonScripts.html>

Jython

```
# read the accelerator
#acc_xml = "file:/user1/chu/xaldev/xal_xmls/sns.xml"
acc_xml = "file:./sns.xml"
acc = XmlDataSource.parseUriAt(acc_xml, 0)

# get the some primary sequences from the accelerator

mebt = acc.getSequence("MEBT")
dtl1 = acc.getSequence("DTL1")
dtl2 = acc.getSequence("DTL2")

print " There are ", mebt.getAllNodes().size(), "nodes in the
sequence", mebt.getId()
```

MatLab

```
% scan the first quad
% monitor beam positions in the last MEBT BPM

for i =1:10
    fld(i) = field;
        quad.setField(field);
        va_chan.putVal(1); % for virtual accelerator
        pause(1); % for virtual accelerator
        xpos(i) = bpm.getXAvg;
        ypos(i) = bpm.getYAvg;
        field = field * 1.015; % increment field value
end

% Plot results
plot(fld, ypos)
```

LCLS, March, 2006

- Accelerator physics apps
 - Accelerator tuning
 - Physics experiment
- General purpose apps
 - Hardware monitor or diagnosis
- Service daemons
 - Minimize network traffic as multiple apps accessing the same PVs
 - Always running in the background

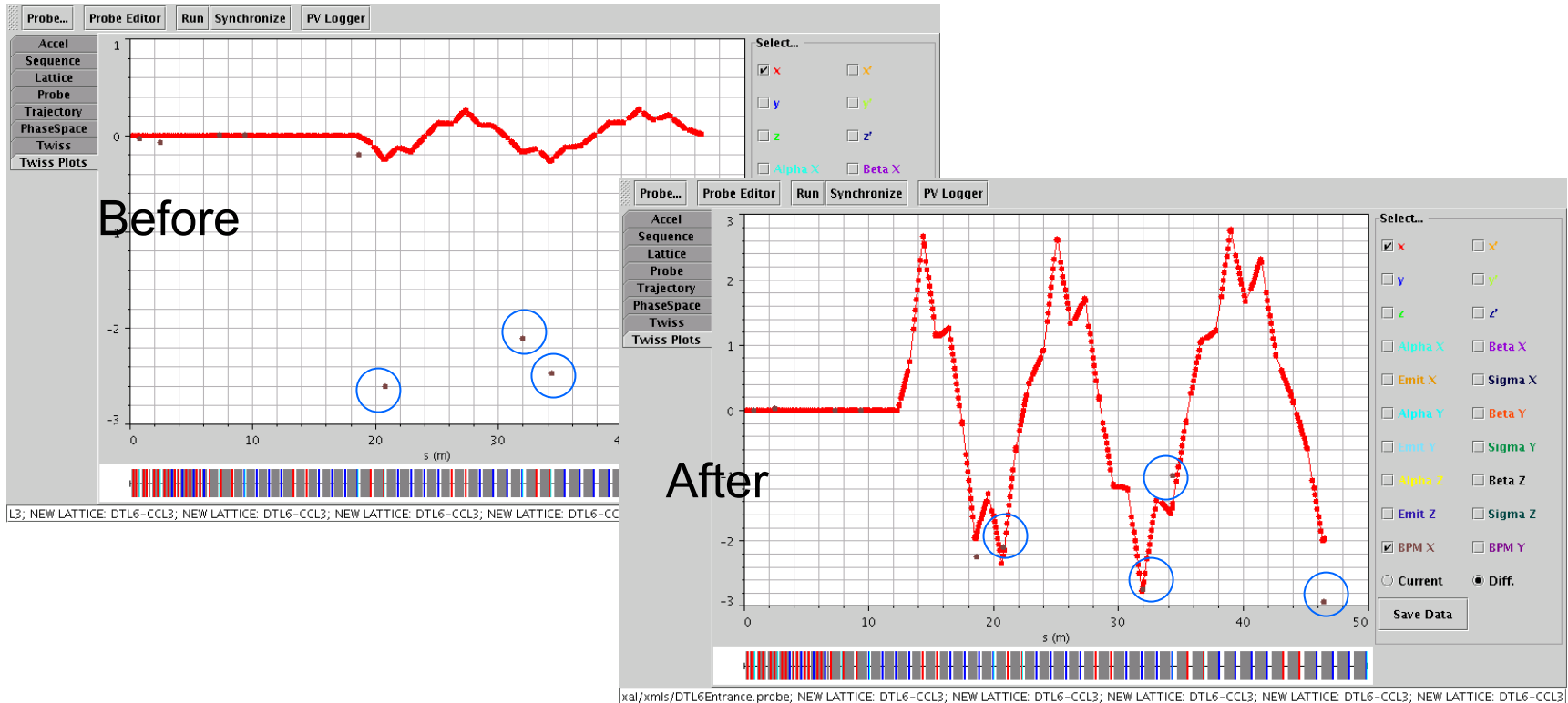
Accelerator Physics App Examples



- Orbit difference – check steer and BPM polarities with online model.
- Orbit correction – flatten trajectory.
- Energy manager – find a good lattice when energy changed.
- Linac RF phase and amplitude tuning app.
- Emittance analysis – process raw data from emittance device.
- Wire scanner analysis – process raw data from wire scanner and compare with online model.
- Ring injection – phase space painting.
- Ring measurement – measure tune, β -function, dispersion chromaticity.
- Virtual Accelerator – machine simulator.

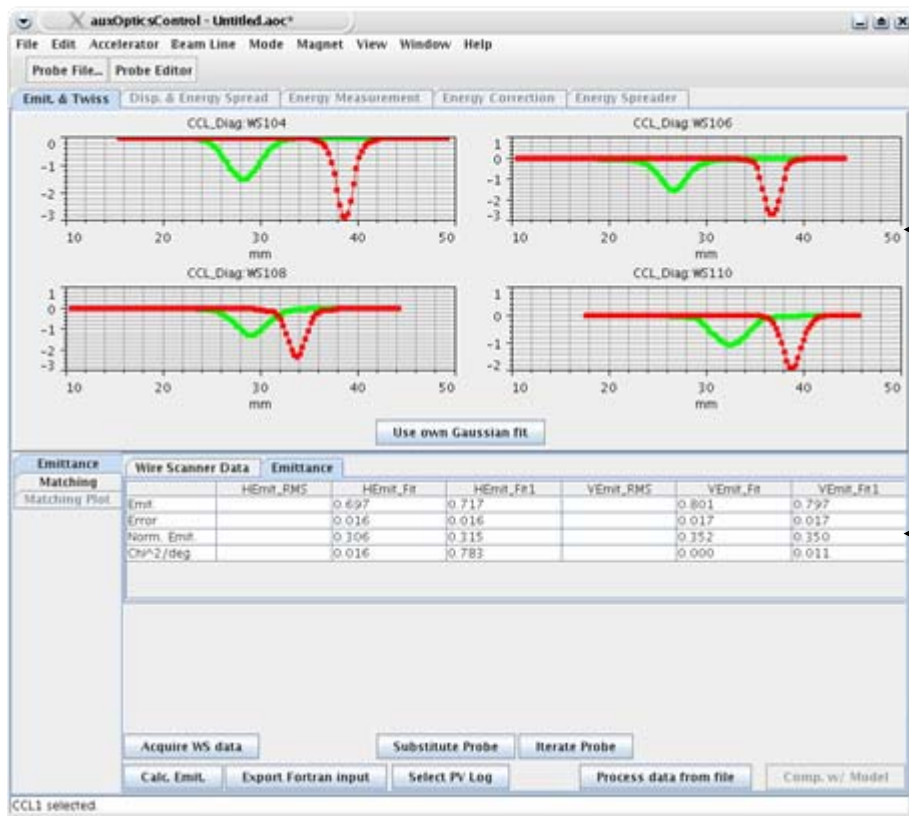
Orbit Difference App

- Use the online model app for orbit difference purpose (run it twice with a steer changed).
- Found CCL steer field wrong.



Linac Transverse Tuning Apps

- Use 3+ wire scanners for emittance measurement.
- Perform transverse matching using online model.
- Can work both on-line and off-line.

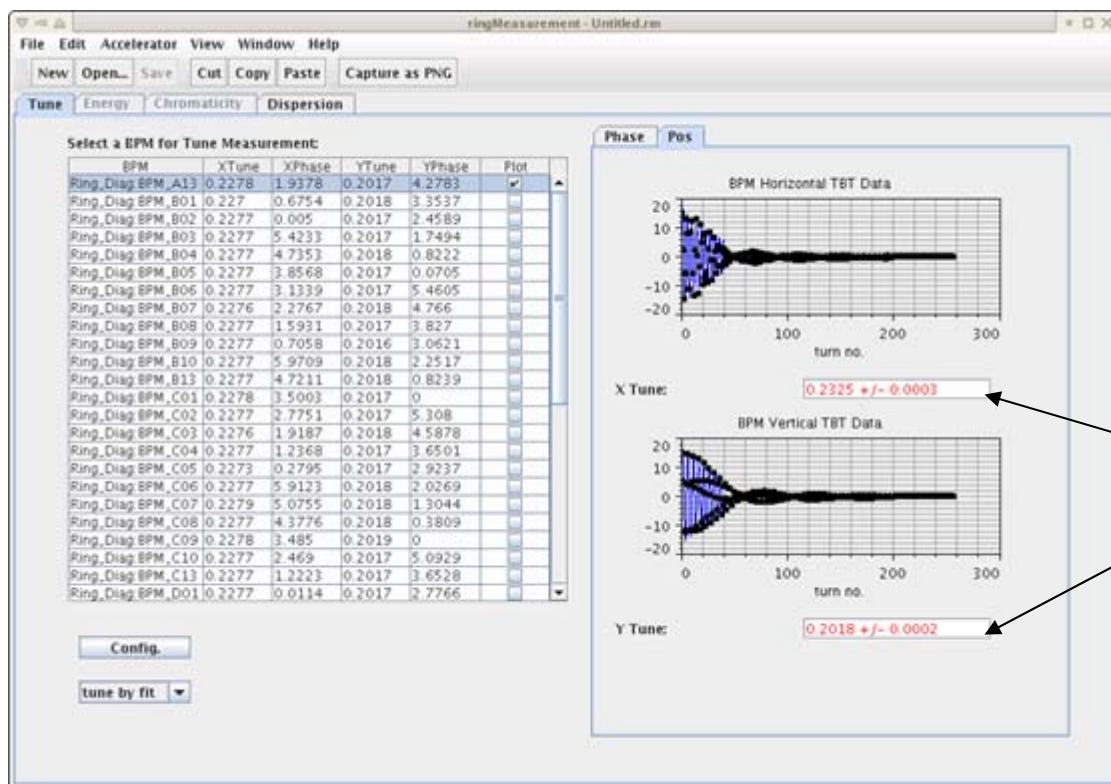


Wire profiles

emittances

Ring Measurement

- Measure tunes with BPM turn-by-turn data.
- Measure dispersion
- Perform quad tuning with betatron phase advance (in progress).



Superconducting Linac Tuning

- Use drifting beam (RF cavity off) induced field for RF phase setup.

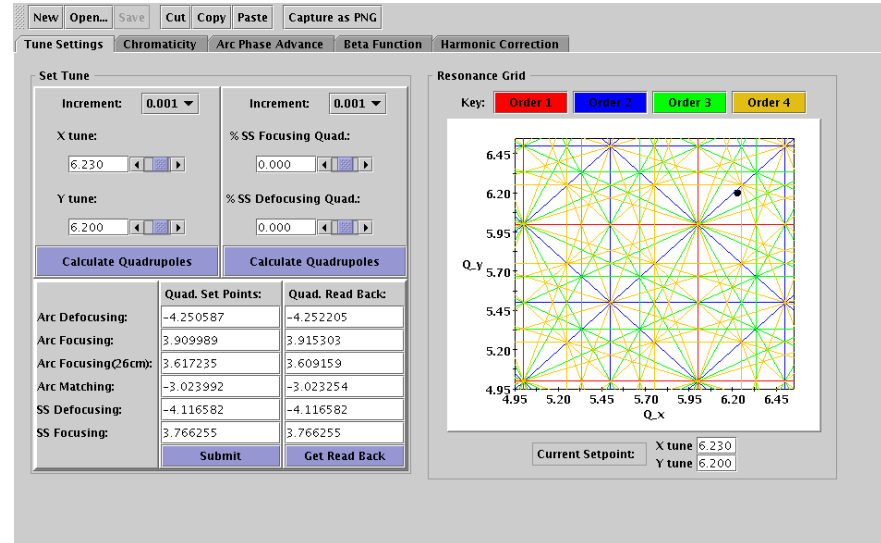


Calculated phase set point

Calculated output energy

Ring Apps

- Ring applications using XAL tools.
- HEBT matching algorithms.
- Ring Optics settings. →
- Injection.



The screenshot displays the XAL software interface. The 'Set Tune' panel on the left contains the following controls and data:

Parameter	Value
Increment	0.001
X tune	6.230
Y tune	6.200
% SS Focusing Quad	0.000
% SS Defocusing Quad	0.000

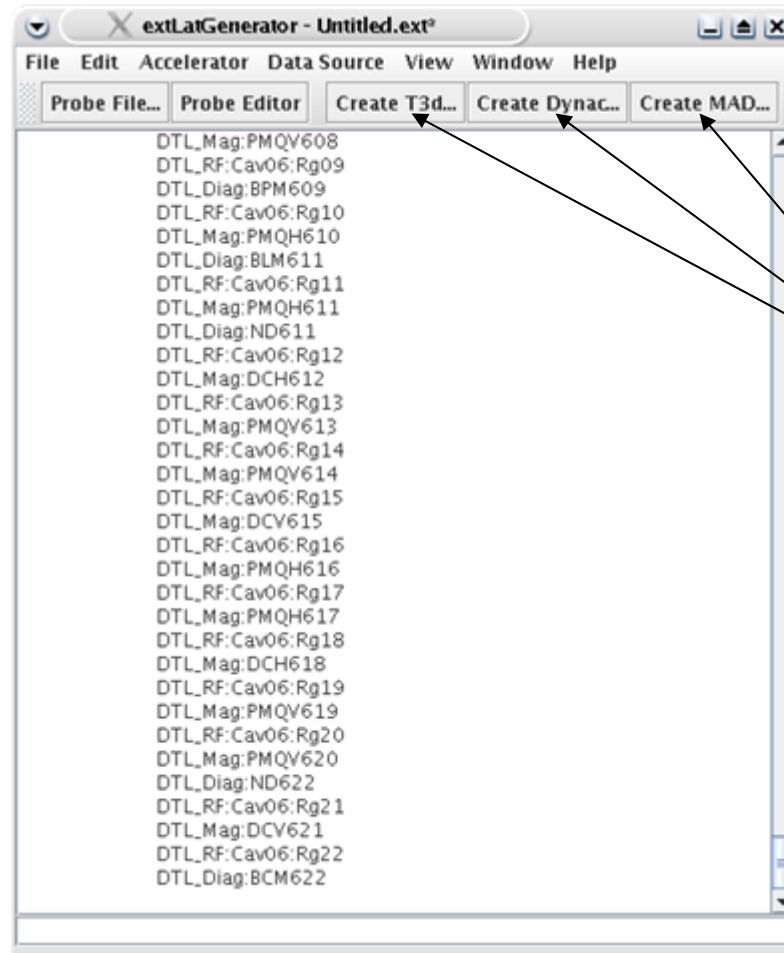
Below these are two 'Calculate Quadrupoles' buttons. A table shows the current quadrupole settings:

Parameter	Quad. Set Points	Quad. Read Back
Arc Defocusing:	-4.250587	-4.252205
Arc Focusing:	3.909989	3.915303
Arc Focusing(26cm):	3.617235	3.609159
Arc Matching:	-3.023992	-3.023254
SS Defocusing:	-4.116582	-4.116582
SS Focusing:	3.766255	3.766255

The 'Resonance Grid' panel on the right shows a plot of Q_x vs Q_y with a key for Order 1 (red), Order 2 (blue), Order 3 (green), and Order 4 (yellow). The current setpoint is X tune 6.230 and Y tune 6.200.

External (to XAL) Lattice Generator

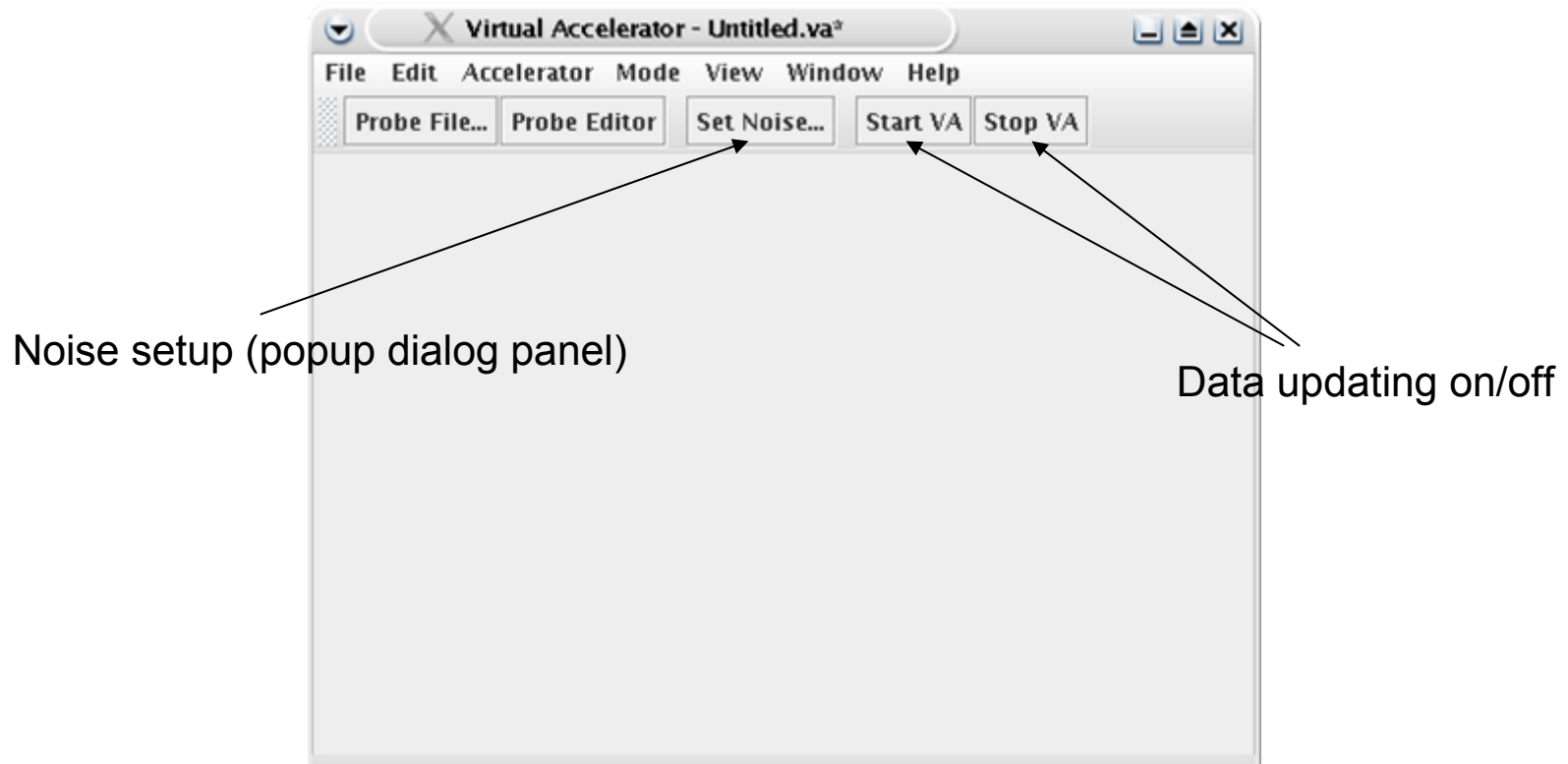
- Can generate MAD, Trace-3D, DYNAC input files on the fly.



Select the file format

Virtual Accelerator

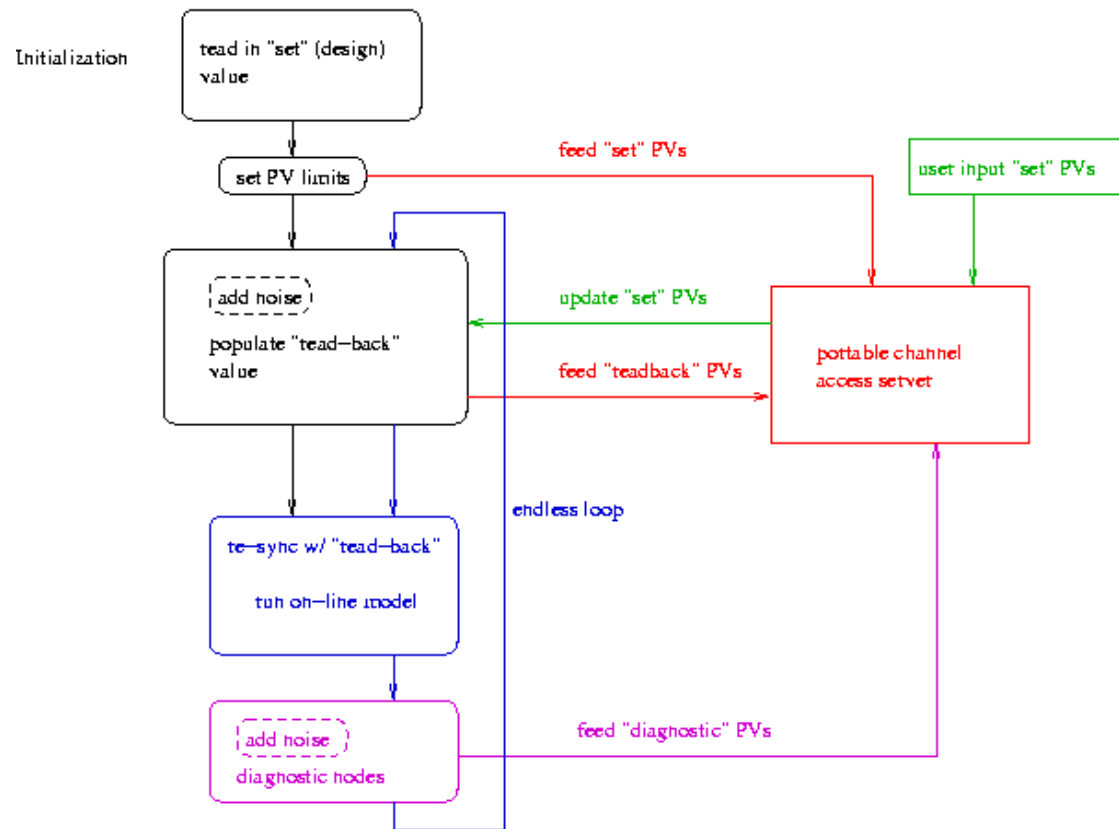
- Application test tool when the machine is not running.
- Based on online model.
- Can set noise, off-set.



Virtual Accelerator (cont.)

- VA control diagram

On-line Model Virtual Accelerator



General Purpose App Examples



- XIO – XAL I/O diagnostic tool, monitoring and plotting any beam-line device PVs.
- PV correlator – display 2 or 3 PVs from the same beam pulse.
- Beam loss viewer – display beam loss monitor readings.
- Scan apps – scan 1 or 2 PVs and monitor some other PV(s).
- Scope – a digital scope-like app.
- PV timestamp test – display PV timestamps and plot them.
- Diagnostics device timing – display and set timing for diagnostics devices.
- Score – save, compare and restore PVs.
- Knobs – can combine arbitrary PVs to a single knob, e.g. 3- or 4-bump app.

Save-Compare-Restore (Score) Application



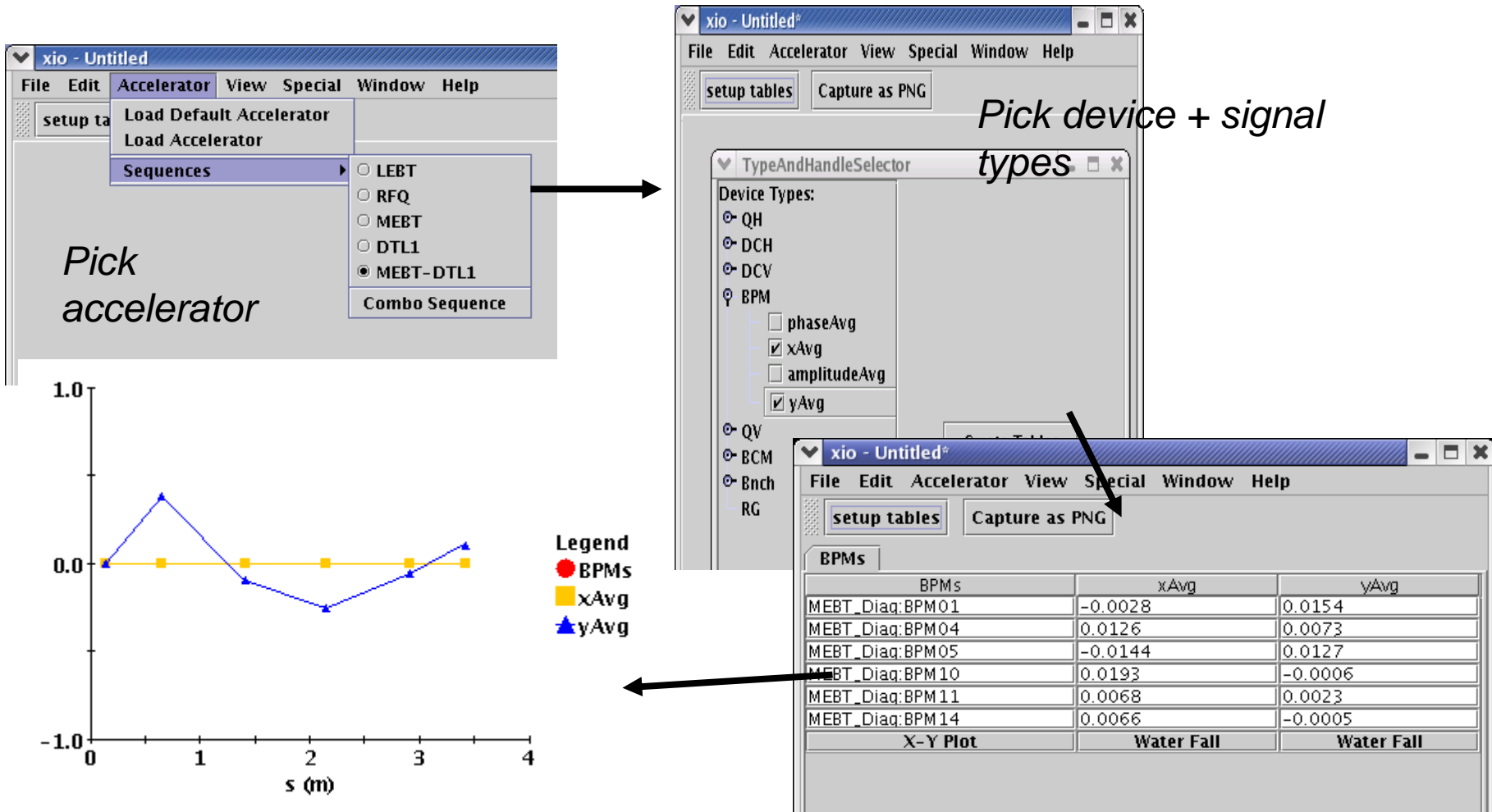
- Provides a means to capture machine setup, compare live values to a saved set, and to restore values to a saved set.
 - Grabs settable + readback signals.
- Can sort by system and device type.
 - Uses DataTable classes for querying.

Load devices/types Open... Save As... Snapshot Machine Restore Capture as PNG							
Select Systems	RFQ	MEBT	DTL	Timing	FE	DPlate	
	Type	Setpoint name	SP Save Val	SP live Val	Readback Name	RB Save Val	RB live Val
DPlate	RF						
DTL							
FE		RFQ:RF:Gain	0.3500	0.3500			
MEBT		RFQ:RF:Gain_Rot	116.9083	116.9083			
RFQ		RFQ:RF:InL_scale	7000.0000	7000.0000			
Timing		RFQ:RF:Loop	1.0000	1.0000			
		RFQ:RF:cavAmpSet	0.5512	0.5512	RFQ:RF:cavAmpAvg	0.5488	0.5493
		RFQ:RF:cavPhaseSet	24.3920	24.3920	RFQ:RF:cavPhaseAvg	24.3090	24.2417
		RFQ_HPRF:Mod1:VCTL_Set	130.0000	130.0000	RFQ_HPRF:Mod1:V_Mon	100.3780	100.3750
					RFQ:RF:FwdPower	686.1238	673.1348
					RFQ:RF:RFIPower	9.4841	9.0523
					RFQ_HPRF:Mod1:I_Mon	48.8328	48.8791
	Temp						
Sys Set		RFQ:Chllr_2:T_Set	24.2000	24.2000	RFQ:Chllr_2:T	25.2192	25.3264
					RFQ:Chllr_2:T_LB	24.1819	24.1849
	Vac						
					RFQ_VacIG_2:P	2.710E-7	2.694E-7
					RFQ_VacXV:Sts	1.0000	1.0000

Machine data saved at Sat Aug 30 19:36:59 EDT 2003

Xio Application

- General purpose value displayer (tables, and or plots).
- Browser the accelerator hierarchy to select what you are interested in.



Pick accelerator

Pick device + signal types

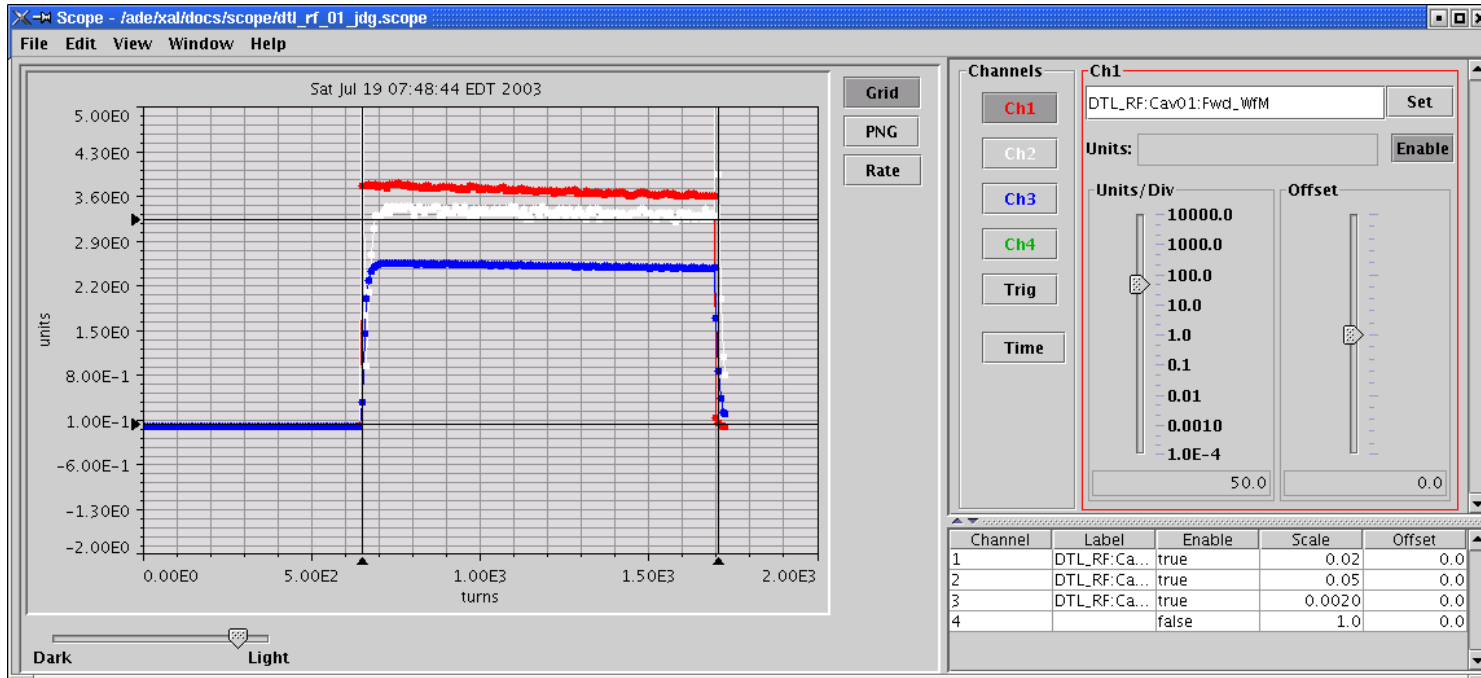
Legend
● BPMs
■ xAvg
▲ yAvg

BPMs	xAvg	yAvg
MEBT_Diaq:BPM01	-0.0028	0.0154
MEBT_Diaq:BPM04	0.0126	0.0073
MEBT_Diaq:BPM05	-0.0144	0.0127
MEBT_Diaq:BPM10	0.0193	-0.0006
MEBT_Diaq:BPM11	0.0068	0.0023
MEBT_Diaq:BPM14	0.0066	-0.0005

X-Y Plot Water Fall Water Fall

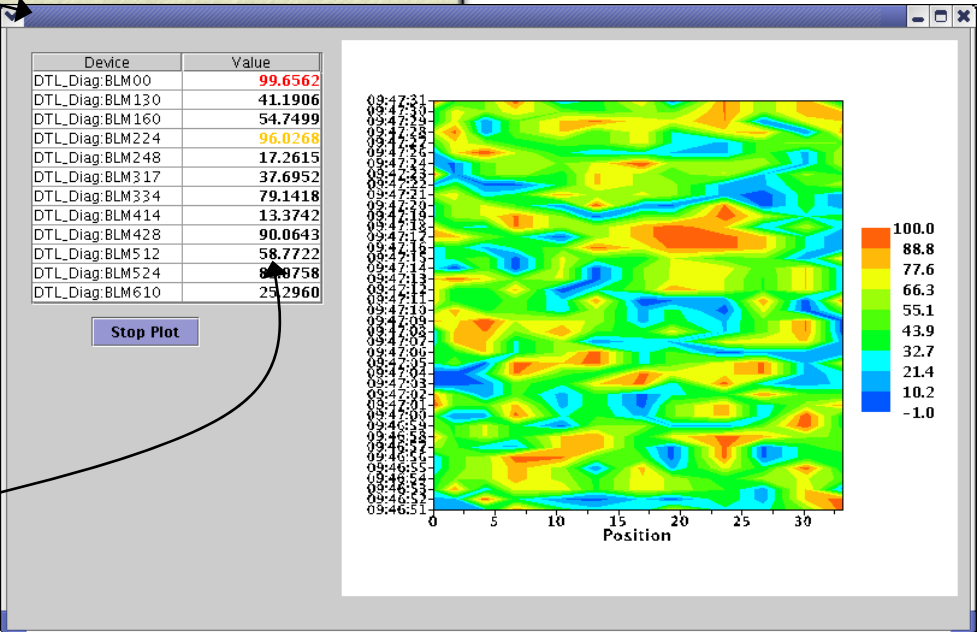
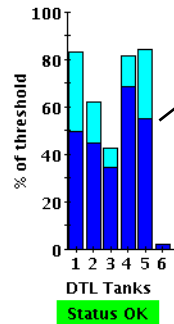
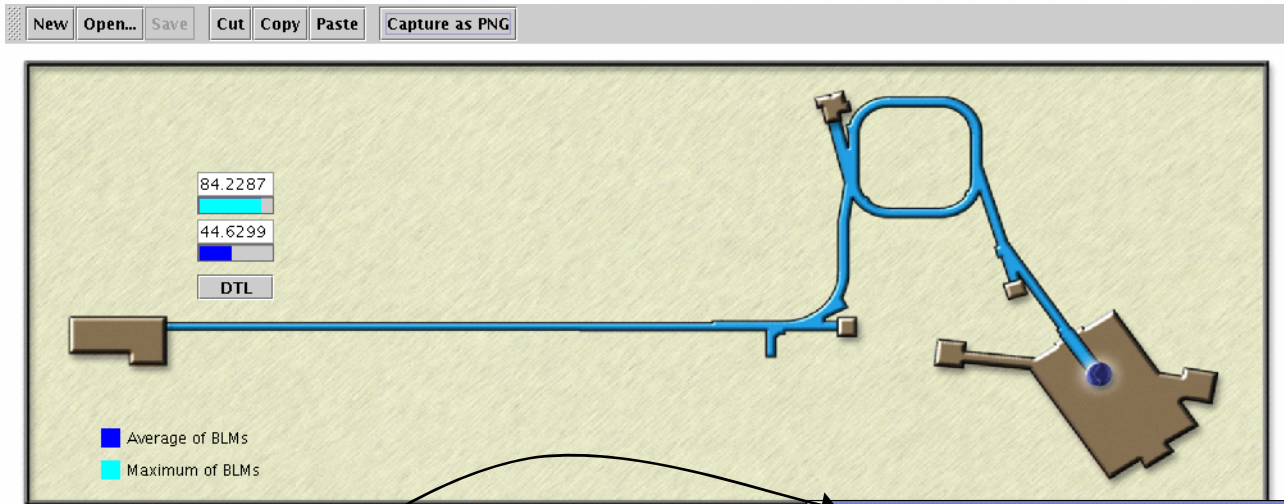
Scope Application

- A Digital Oscilloscope – with a similar user interface as analog scopes.
- Displays array waveforms vs. time (NOT vs arbitrary units).
- Uses the time correlator, has built-in math capability, triggered data acquisition+ many other features.
- Will use this for comparing waveforms from RF, diagnostics, etc.
- Requires input from signal providers describing how the array information is packaged, and offset from the cycle start.



DTL1 RF
waveforms

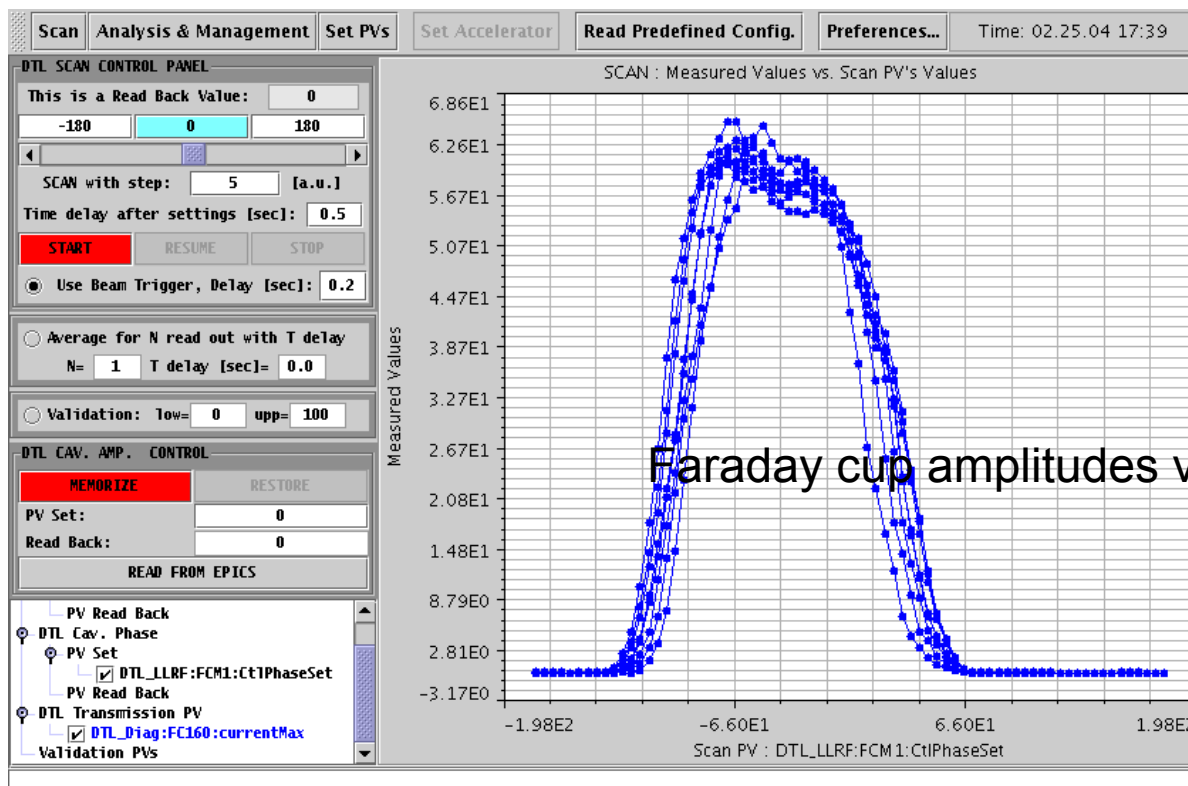
Loss Viewer Application



- View a summary of beam loss by machine section.
 - “Zoomable” to specific BLMs.
 - Viewable as fraction of permissible loss.

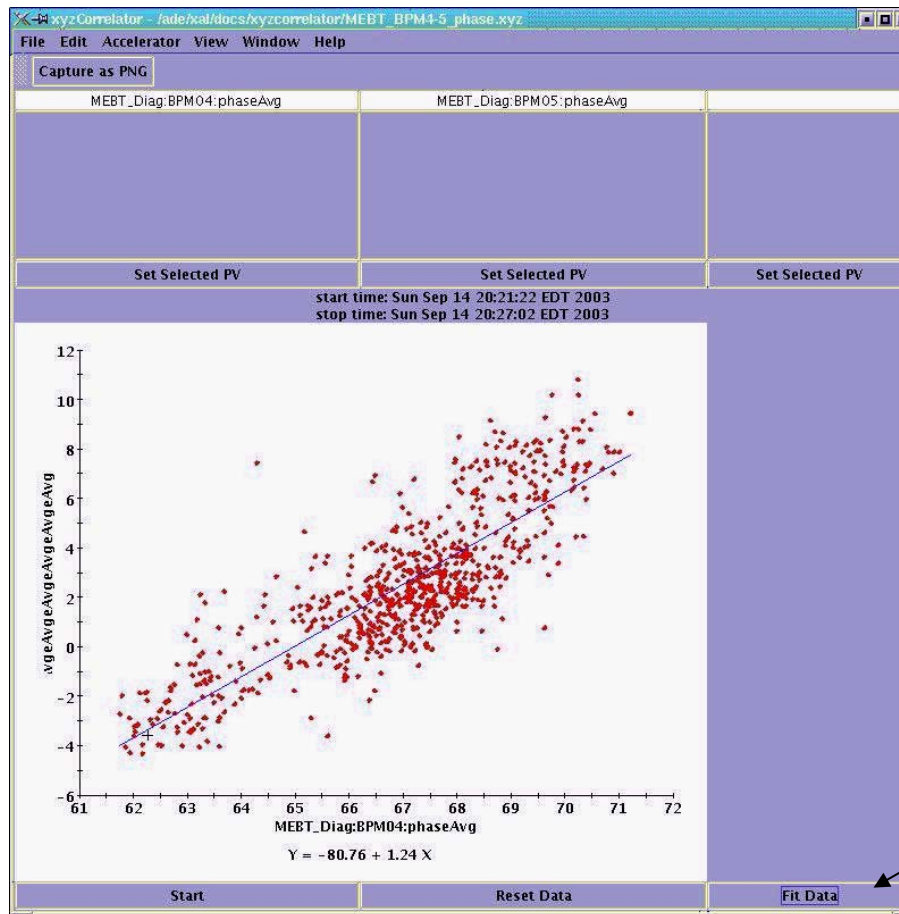
1-D Scan Application

- Provides an easy way to scan one quantity and monitor others.
- Can average over pulses, scan multiple times, pause.
- Analysis includes fitting, intersection finding, min/max, etc.
- Easy way to do a quick unanticipated experiment.
- Predefined scans with specialized analysis are possible.
 - DTL and MEBT phase + amplitude setting applications.



XYZ Correlator Application

- Pick 2 (or three) signals and monitor them together.
- Uses the time correlator to ensure signals are from the same pulse.
- Can export or fit the acquired data.

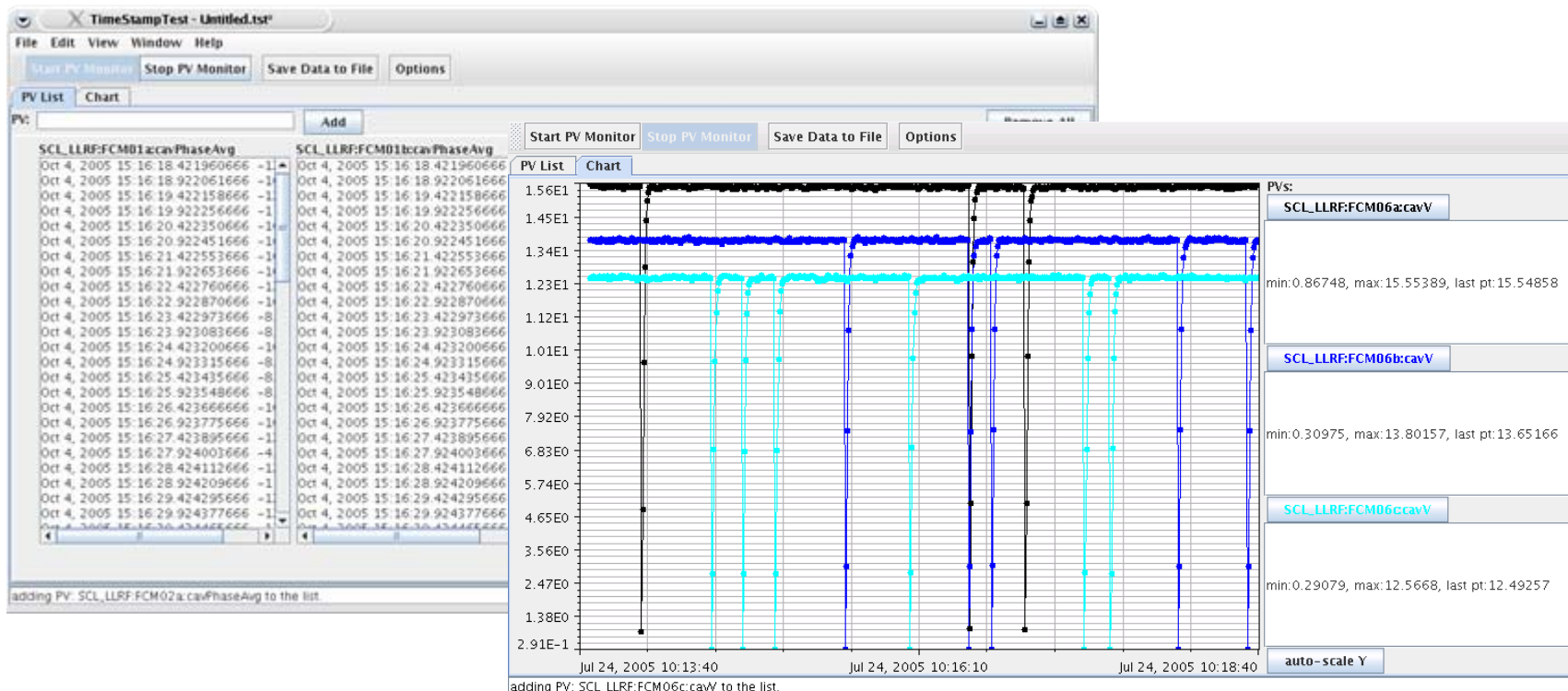


Correlation between phase measurements of 2 BPMs.

Can fit the data to a polynomial

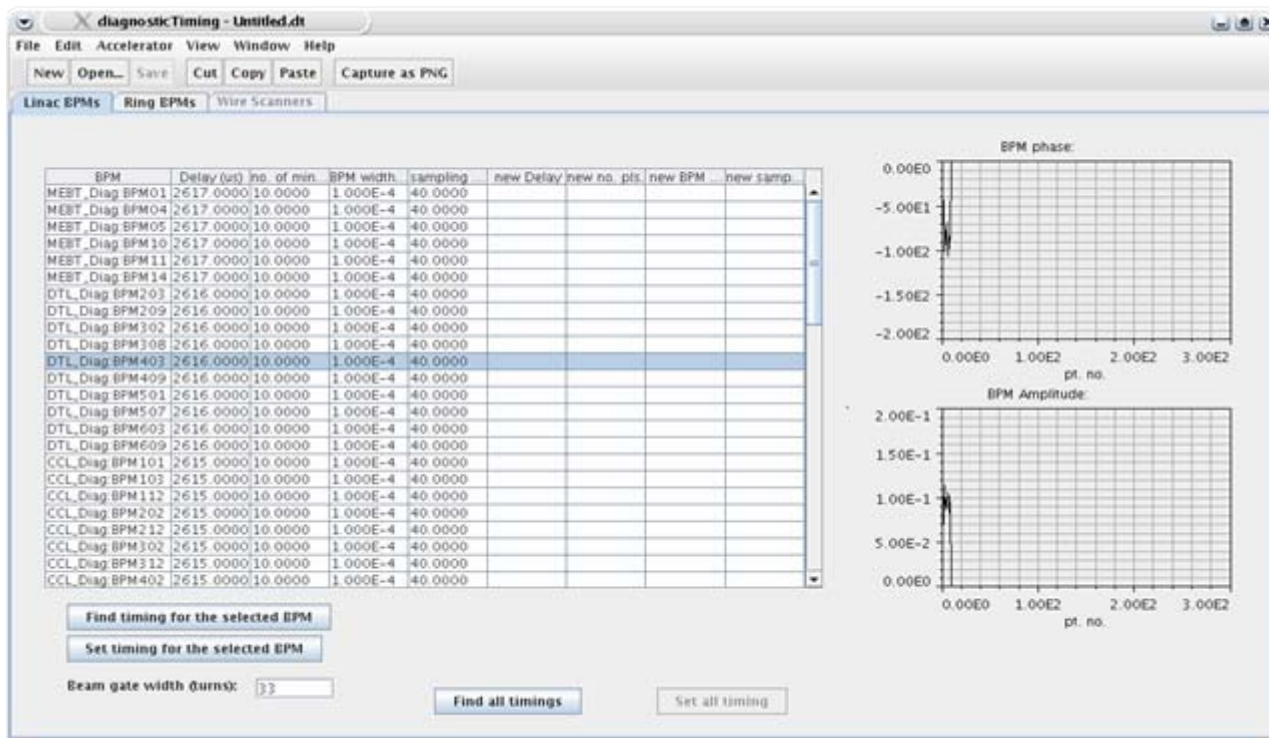
PV Timestamp Test

- Show multiple PV timestamps side by side.
- Can serve as a strip tool.
- Use Channel Access monitor.



Diagnostics Device Timing

- Display multiple diagnostics device timings in one place.
- Can set timing automatically or manually.



- XAL is a Java based hierarchal framework SNS is using to write high level applications.
 - Online modeling is available for both linac and ring.
 - The framework is matured: > 800 classes so far.
- Uses a database to initialize the hierarchy.
- Many tools are available.
- Applications are written and being used in the SNS commissioning.
 - > 40 applications written.
- Directions:
 - Adding more Ring applications.
 - Adding more ring online model features.
 - Adding more detailed information to the database.
 - Moving towards service applications.
 - ...

- Possible collaboration:
 - rearrange the xal structure to make the non-swing stuff (tools, smf, model, ...) somehow more “standalone” so that they could be shared as “plugins”.
- Comments from John Galambos:
 - Some of their stuff could perhaps benefit from looking at the way we did things (tested under fire) in areas where there is already an overlap. For example the Gumtree “scan” program looks like it may not have all the features the xal scan1D, scan2D does (filter PV, averaging, delays, analysis layer, ...). Another area is our Application Framework – even though they seem set on an Eclipse solution, there are probably lessons learned from our Framework. E.g. the ability to kill wayward apps (with an appViewer), see process size, shared features like creen capure, ...
- Comments from Tom Pelaia:
 - A visit to GumTree makes a great point in favor of XAL's pure Java foundation. It isn't platform independent. Immediately you notice that they have a Win32 download for GumTree M7 and Win32 and Linux GTK **separate** downloads for the latest build. Where are the Mac and Solaris downloads? XAL provides universal J2SE 5 support.
 - Among the advantages of pure Java over Eclipse:
 - - universal Java platform support without having to ship separate binaries
 - - consistent behavior across platforms
 - - support for more platforms than Eclipse
 - - more Swing utilities (plotting, etc.)
 - - more developer options: developers use the tools they prefer including but not restricted to Eclipse