Review of the OpenXAL Framework

Yngve Inntjore Levinsen
on behalf of M. Munoz and the Beam Physics Group

AD and ICS Retreat mtg 2014

11. December, 2014

Introduction

Motivation

- An API layer or framework is needed for control room applications for the accelerator
  - Maintainability
  - Speed up software development
  - Aid during commissioning
- OpenXAL proposed by ICS as a suitable framework

Requirements

Beam Physics Requirements - Major Points

- Need to have an online model which performs reasonably accurate, **fast** calculations of the beam dynamics for ESS
- Need an structured access to the accelerator components
- Previous deployment in working accelerator considered very valuable

Requirements

Beam Physics Requirements - Major Points

- Need to have an online model which performs reasonably accurate, **fast** calculations of the beam dynamics for ESS
- Need an structured access to the accelerator components
- Previous deployment in working accelerator considered very valuable
- Reasonably easy API (for physicists) and good documentation is important
- Should support both interactive scripting and standalone programs

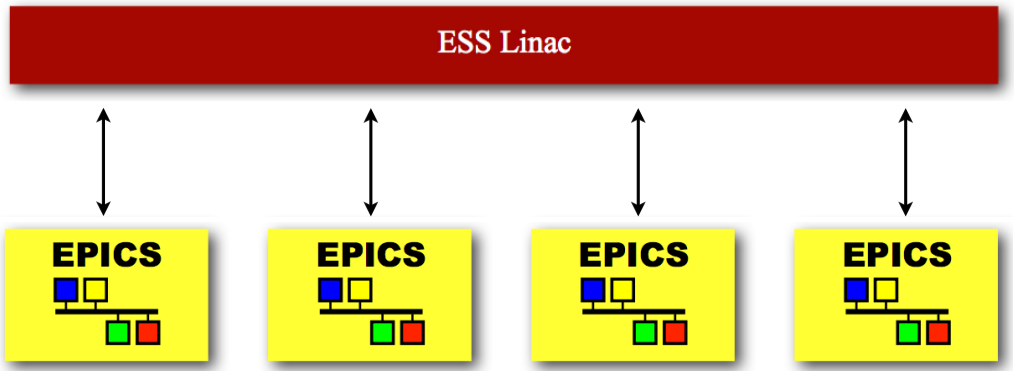
Requirements

Beam Physics Requirements - Major Points

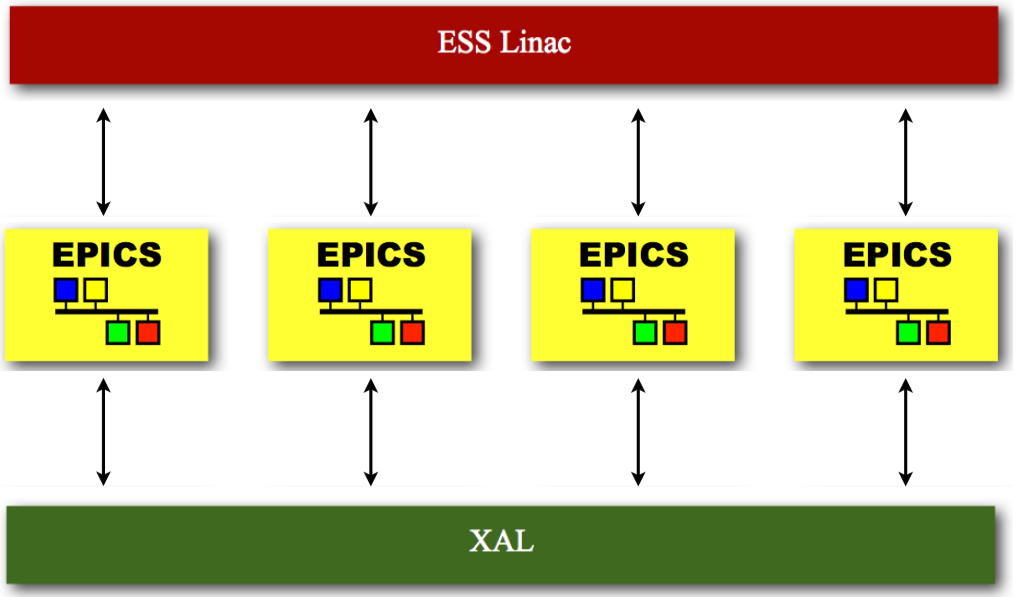
- Need to have an online model which performs reasonably accurate, **fast** calculations of the beam dynamics for ESS
- Need an structured access to the accelerator components
- Previous deployment in working accelerator considered very valuable
- Reasonably easy API (for physicists) and good documentation is important
- Should support both interactive scripting and standalone programs
- Full access to source code and possibility of extending code is essential
- Need to have access to other services provided by ICS

Features

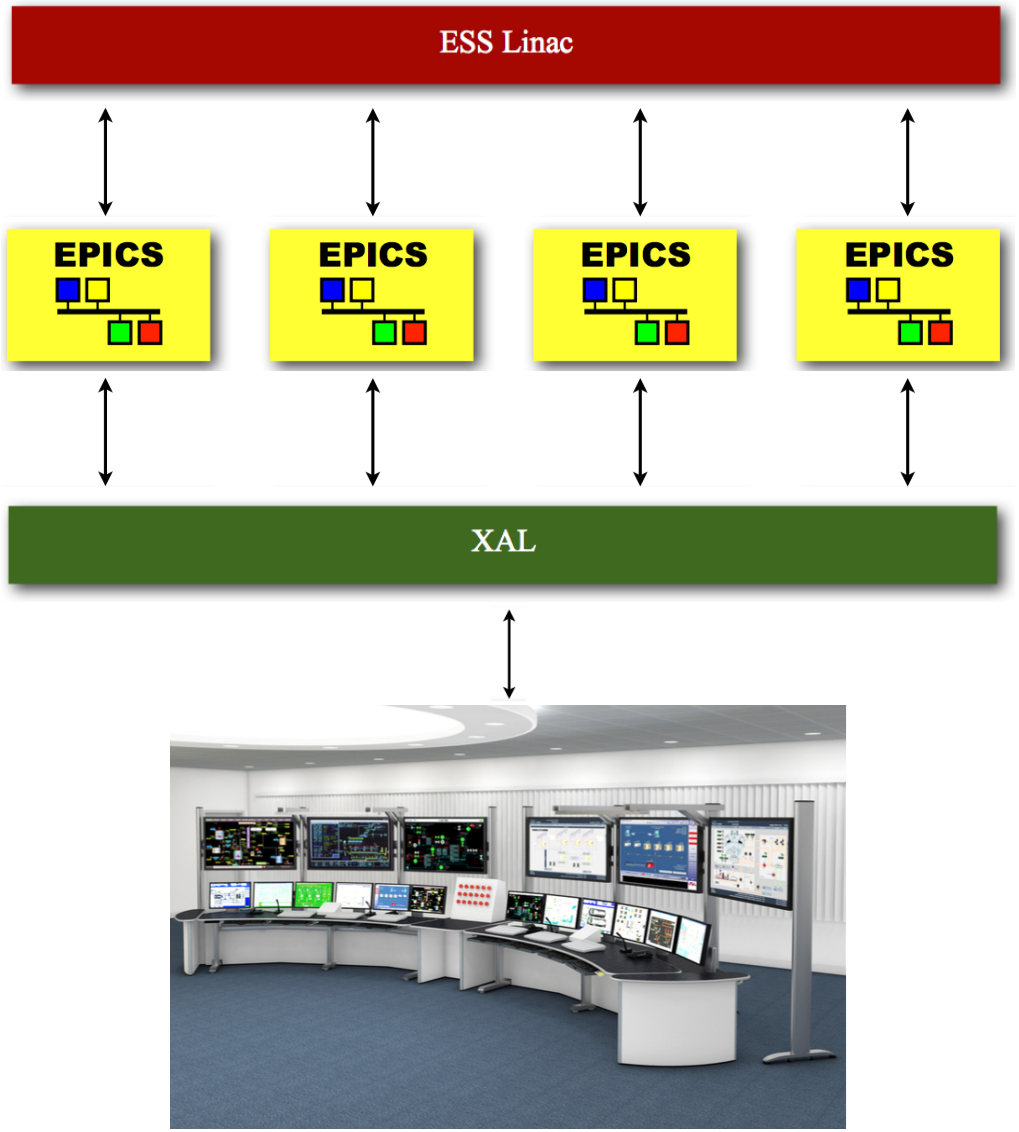
Features



Features



Features



Features

Open XAL is an open source development environment used for creating accelerator physics applications, scripts and services. This project is a collaboration among SNS, CSNS, ESS, GANIL, TRIUMF and FRIB.

- OpenXAL originates from SNS
- Main developers are T. Pelaia and C. Allen
- Additions for ESS developed by ICS for some time already (E. Laface, I. List)

Features

Main Features

- Pure Java Framework
- Online Modeling
- Application Collection
- Structural representation of the accelerator components
- Structured access to EPICS

Features

ESS Specific Features

- Java ESS Linac Simulator (main area of development at the moment)
- Import from database (deprecated)
- Import from LinacLego

Features

ELS/JELS

- ELS - ESS Linac Simulator
- JELS = Java version
- Fast envelope calculations
- Space charge model included
- Thorough benchmarking performed

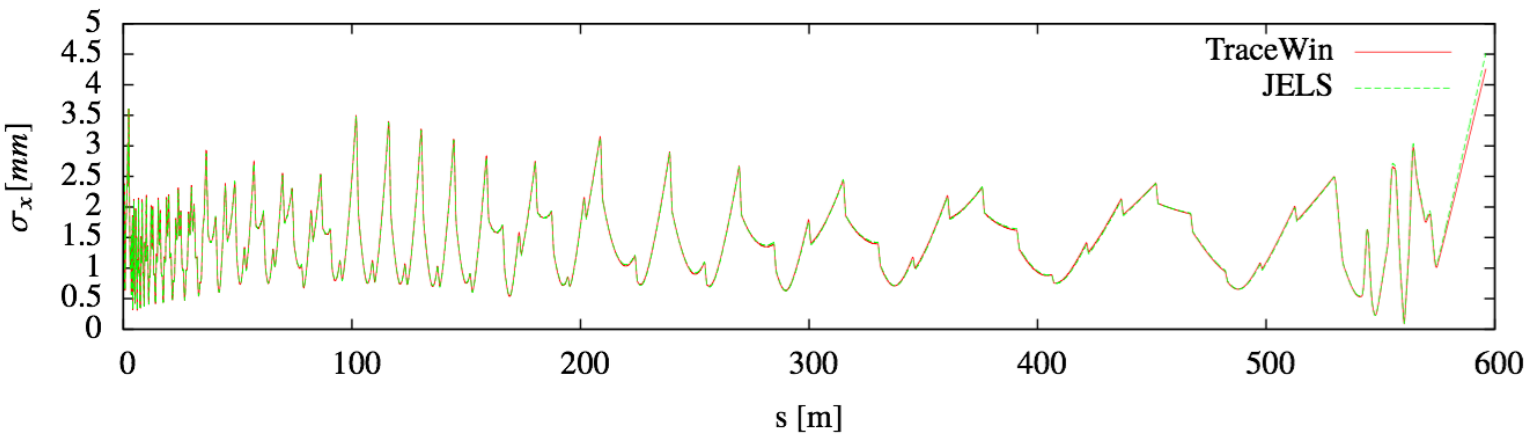
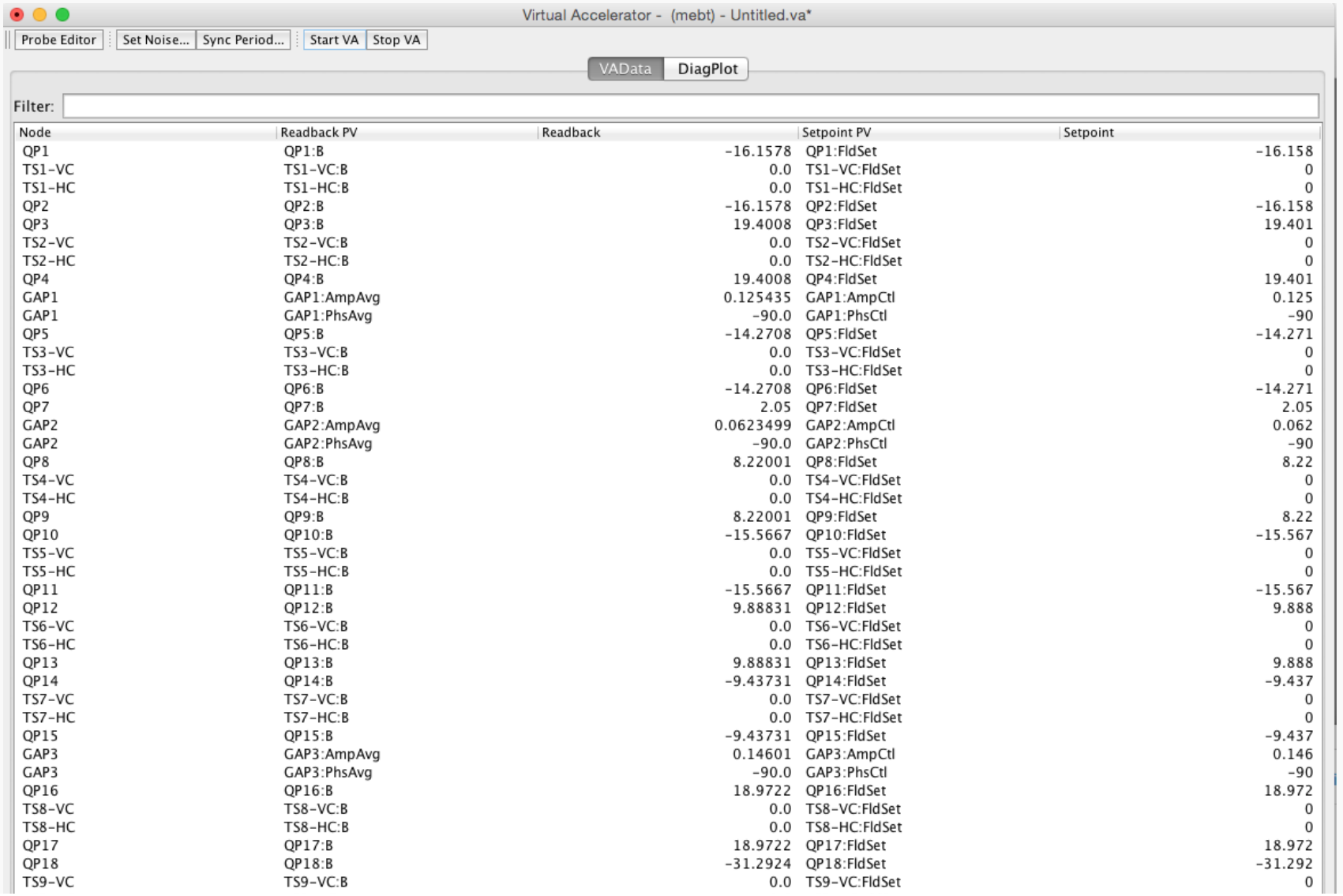


Figure 1: Comparison of JELS and TraceWin simulation on ESS beam-line starting with MEBT section with initial energy 3 MeV and without space-charge.

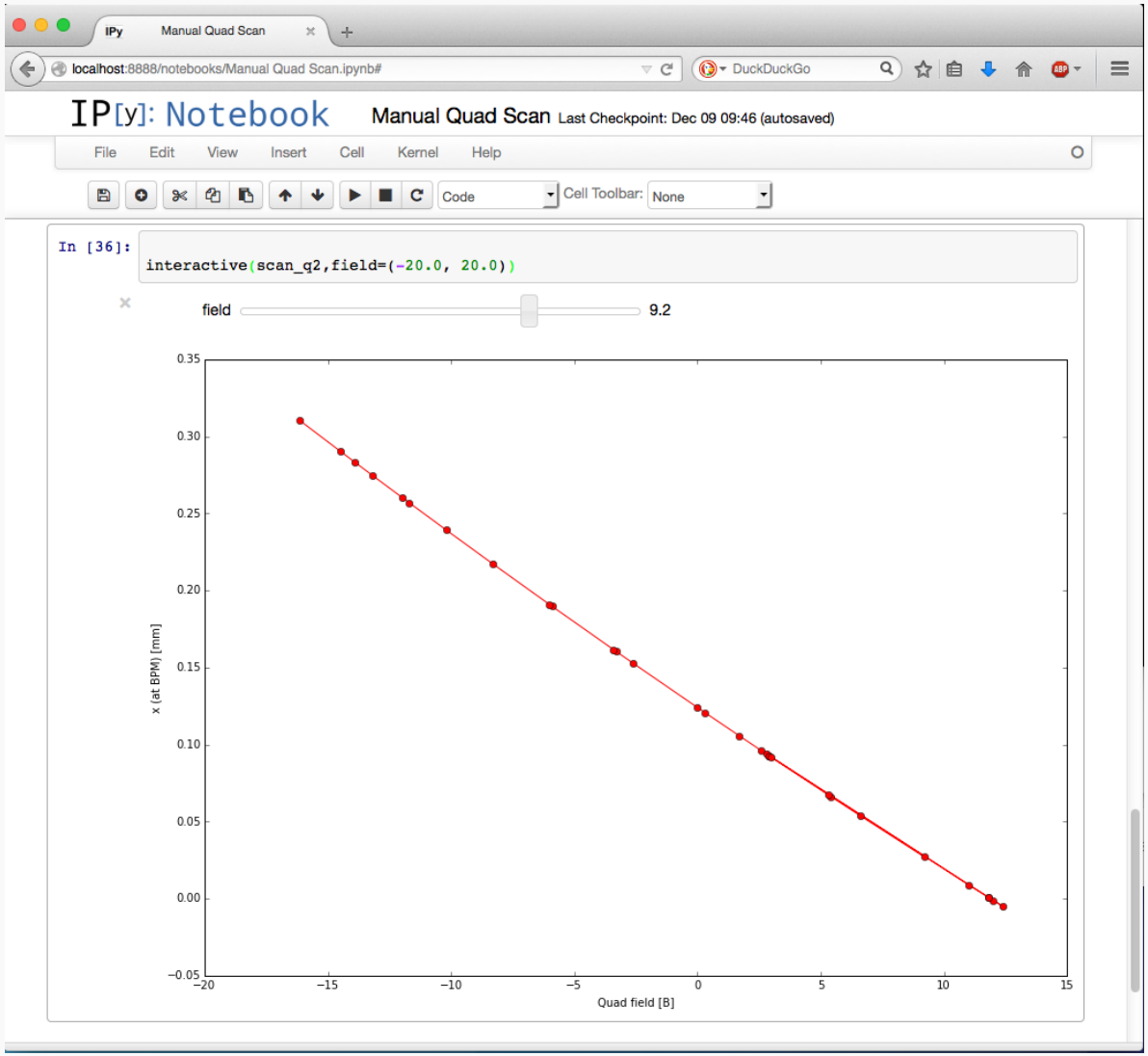
I. List, E. Laface, IPAC'14

Example Applications - Virtual Accelerator



Node	Readback PV	Readback	Setpoint PV	Setpoint
QP1	QP1:B	-16.1578	QP1:FldSet	-16.158
TS1-VC	TS1-VC:B	0.0	TS1-VC:FldSet	0
TS1-HC	TS1-HC:B	0.0	TS1-HC:FldSet	0
QP2	QP2:B	-16.1578	QP2:FldSet	-16.158
QP3	QP3:B	19.4008	QP3:FldSet	19.401
TS2-VC	TS2-VC:B	0.0	TS2-VC:FldSet	0
TS2-HC	TS2-HC:B	0.0	TS2-HC:FldSet	0
QP4	QP4:B	19.4008	QP4:FldSet	19.401
GAP1	GAP1:AmpAvg	0.125435	GAP1:AmpCtl	0.125
GAP1	GAP1:PhsAvg	-90.0	GAP1:PhsCtl	-90
QP5	QP5:B	-14.2708	QP5:FldSet	-14.271
TS3-VC	TS3-VC:B	0.0	TS3-VC:FldSet	0
TS3-HC	TS3-HC:B	0.0	TS3-HC:FldSet	0
QP6	QP6:B	-14.2708	QP6:FldSet	-14.271
QP7	QP7:B	2.05	QP7:FldSet	2.05
GAP2	GAP2:AmpAvg	0.0623499	GAP2:AmpCtl	0.062
GAP2	GAP2:PhsAvg	-90.0	GAP2:PhsCtl	-90
QP8	QP8:B	8.22001	QP8:FldSet	8.22
TS4-VC	TS4-VC:B	0.0	TS4-VC:FldSet	0
TS4-HC	TS4-HC:B	0.0	TS4-HC:FldSet	0
QP9	QP9:B	8.22001	QP9:FldSet	8.22
QP10	QP10:B	-15.5667	QP10:FldSet	-15.567
TS5-VC	TS5-VC:B	0.0	TS5-VC:FldSet	0
TS5-HC	TS5-HC:B	0.0	TS5-HC:FldSet	0
QP11	QP11:B	-15.5667	QP11:FldSet	-15.567
QP12	QP12:B	9.88831	QP12:FldSet	9.888
TS6-VC	TS6-VC:B	0.0	TS6-VC:FldSet	0
TS6-HC	TS6-HC:B	0.0	TS6-HC:FldSet	0
QP13	QP13:B	9.88831	QP13:FldSet	9.888
QP14	QP14:B	-9.43731	QP14:FldSet	-9.437
TS7-VC	TS7-VC:B	0.0	TS7-VC:FldSet	0
TS7-HC	TS7-HC:B	0.0	TS7-HC:FldSet	0
QP15	QP15:B	-9.43731	QP15:FldSet	-9.437
GAP3	GAP3:AmpAvg	0.14601	GAP3:AmpCtl	0.146
GAP3	GAP3:PhsAvg	-90.0	GAP3:PhsCtl	-90
QP16	QP16:B	18.9722	QP16:FldSet	18.972
TS8-VC	TS8-VC:B	0.0	TS8-VC:FldSet	0
TS8-HC	TS8-HC:B	0.0	TS8-HC:FldSet	0
QP17	QP17:B	18.9722	QP17:FldSet	18.972
QP18	QP18:B	-31.2924	QP18:FldSet	-31.292
TS9-VC	TS9-VC:B	0.0	TS9-VC:FldSet	0

Example Applications - Quad Scan in Python



Example Applications - RF Phase Scan in Python

IPy jpype openxal rf

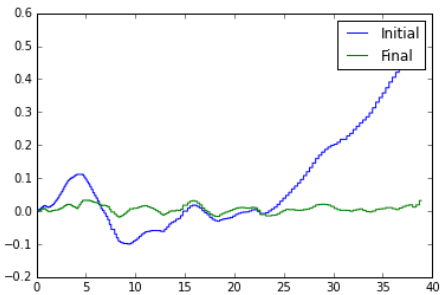
localhost:8888/notebooks/jpype openxal rf.ipynb

IP[y]: Notebook jpype openxal rf (autosaved)

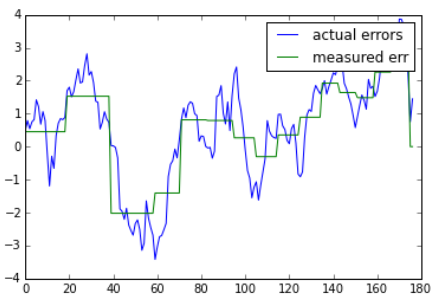
File Edit View Insert Cell Kernel Help

Code Cell Toolbar: None

In [15]:
```
# Plot the result...
traj=get_trajectory(rf_phase.tolist())
final_energy_meas=get_kinetic_energy(traj)
plt.plot(positions,kinetic_energy_2-kinetic_energy,  label="Initial")
plt.plot(positions,final_energy_meas-kinetic_energy, label="Final")
legend=plt.legend()
```



In [18]:
```
# plot phase errors and corrections..
plt.plot(errors, label="actual errors")
plt.plot(-rf_phase, label="measured err")
legend=plt.legend()
```



Conclusions

OpenXAL

- . . . has a good and fast model already implemented (ELS)
- . . . has defined an organized view of the accelerator (XML)
- . . . will soon be deployed at SNS (XAL has already been tested)

Conclusions

OpenXAL

- . . . has a good and fast model already implemented (ELS)
- . . . has defined an organized view of the accelerator (XML)
- . . . will soon be deployed at SNS (XAL has already been tested)
- . . . supports both Java applications and scripting in e.g. Python, Ruby, Matlab
- . . . has an acceptable API, decent documentation (could be improved)
- . . . is open source, we can extend at will
- . . . will be interfaced with other tools provided by ICS

Further Documentation

- OpenXAL Homepage
- ESS Wiki
- CHESS Link to BP Review (to be published)
- OpenXAL Status Report 2013