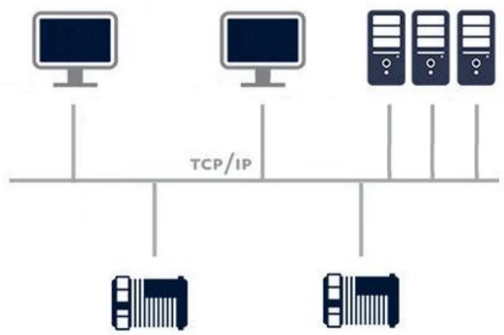
Accelerator Control System

from BPM, magnets and power supplies software to orbit correction



TANGO controls

Jean-Luc PONS

ASD lecture

ESRF, 2nd May 2024

OUTLINE

- Brief Tango overview
- BPM software
- Magnet software
- Orbit correction software
- Applications

DISTRIBUTED CONTROL SYSTEM (TANGO)

- Tango is a free open source device-oriented controls toolkit for controlling any kind of hardware or software and building **SCADA** (**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition) systems

- Based on **CORBA** (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture) and **ZeroMQ** standard

- Tango consortium includes several institutes

DISTRIBUTED CONTROL SYSTEM (TANGO)

- 862 Tango classes shared by the community (www.tango-controls.org)

There are device classes for the following families:

All (862 device classes) AbstractClasses(9). AcceleratorComponents(4). Acquisition(114). Application(1). Archiving(2). BeamDiagnostics(17). BeamlineComponents(21). Calculation(30). Communication(42). Controllers(6). CounterTimer(22). InputOutput(39). Instrumentation(96). Interlock(1). Laser(1). MagneticDevices(5). MeasureInstruments(59). Measurement(1). Miscellaneous(30). Monitor(1). Motion(159). Motors(2). OtherInstruments(40). PLC(1). PowerSupply(39). REST(1). RadioProtection(3). SampleEnvironment(5). Security(2). Simulators(15). SoftwareSystem(23). StandardInterfaces(20). System(3). Temperature(23). Training(2). Vacuum(28).

SEARCH CATALOGUE

DISTRIBUTED CONTROL SYSTEM (TANGO@ESRF)

Code:

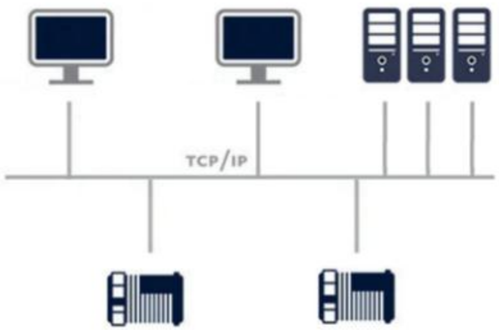
~ 500 classes (C++/Java/Python Tango classes)
~ 100 graphical applications (Java Tango **A**pplication **T**ool**K**it)

Instances:

~ 3.000 servers (on ~400 hosts)
~ 26.000 devices

Configuration database:

~ 620.000 lines

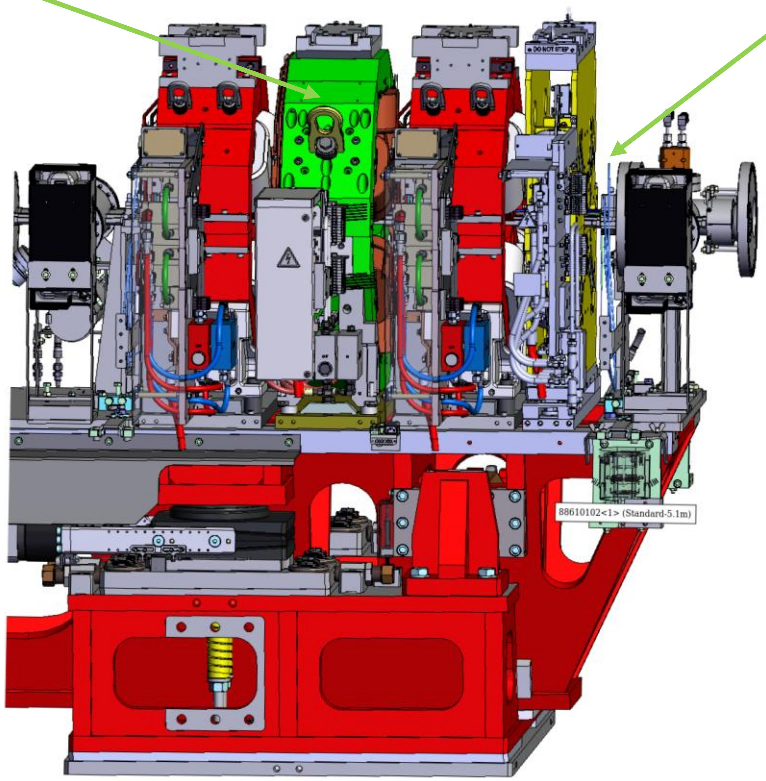


SR ORBIT CORRECTION OVERVIEW

Compute beam position correction for the whole ring

Apply corresponding coil currents on magnets to correct the orbit

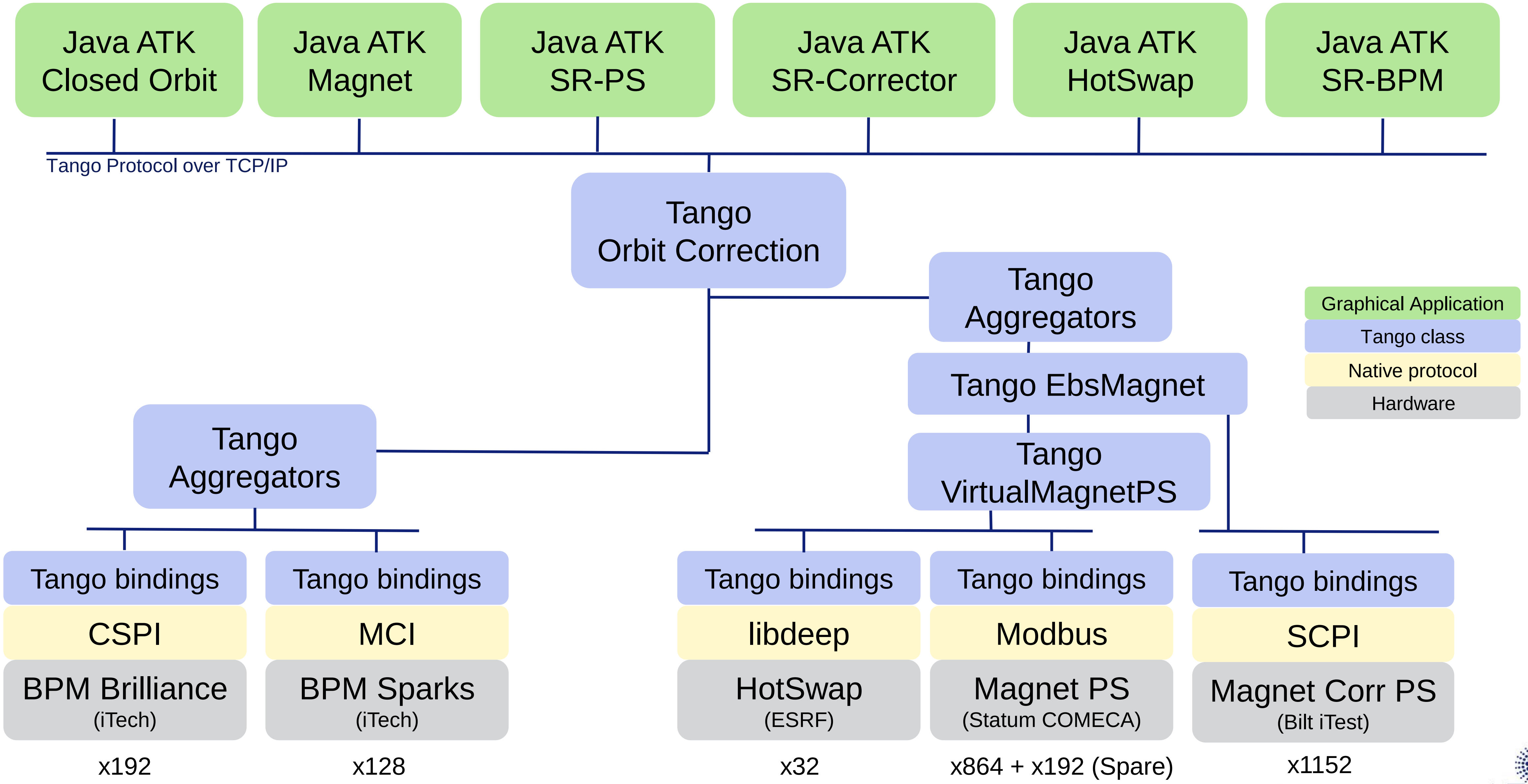
Get the beam position from the BPMs



Software refurbishment in 2022

- Improve maintainability
- Fix conflicts when applying/loading strengths
- Improve reactivity
- Allow non standard configuration (mini-beta)
- Integrate HotSwap system
- Digital twin
- Simulate magnet cycling and electrical power overhead
- …

SR ORBIT CORRECTION GLOBAL SOFTWARE ARCHITECTURE



BPM SOFTWARE

Hybrid system

- 192 Brilliances
- 128 Sparks

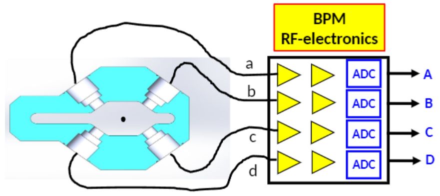
Different conventions
Different protocols
Different configuration and parameters

→

Standardization needed using
Object Oriented Interface / Tango

Brilliance: CSPI protocol (iTech)
Native DCC for Fast Orbit Feedback

```
// Positions and Incoherency [m] (DoS)
p.sum = a + b + c + d;
…
p.h = (KH * (((a + d) - (b + c)) / p.sum) + OH)*1e-9;
p.v = (KV * (((a + b) - (c + d)) / p.sum) + OV)*1e-9;
p.q = (KQ * (((a + c) - (b + d)) / p.sum) )*1e-9;
```



Sparks: MCI protocol (iTech)
No Native DCC for Fast Orbit Feedback

```
// Positions and Incoherency [m] (Dos)
p.sum = a + b + c + d;
…
p.h = (KH * (((a + d) - (b + c)) / p.sum) - OH)*1e-9;
p.v = (KV * (((a + b) - (c + d)) / p.sum) - OV)*1e-9;
p.q = (KQ * (((a + c) - (b + d)) / p.sum) - OQ)*1e-9;
```

For further details: Diagnostics with Beam Position Monitor System lecture from Kees Scheidt

HSM POWER SUPPLIES SOFTWARE (MAIN MAGNET CURRENT)

Power supplies (COMECA STATUM)

- 32 cubicles
- 33 power supplies (27 + 6 spares) per cubicle
- Power supplies can be swapped while beam is stored

Procotol

- **Modbus** protocol over TCP/IP for direct STATUM access
- **Libdeep** protocol over TCP/IP for HSM access

Tango Power Supply virtualization

- Handle low level connection between magnet and PS
- Allow to work with a unique name to access a magnet
- Allow standalone power supply (QF1E QF1A for injection)
- Allow spare usage for mini-beta tests

Orbit correction

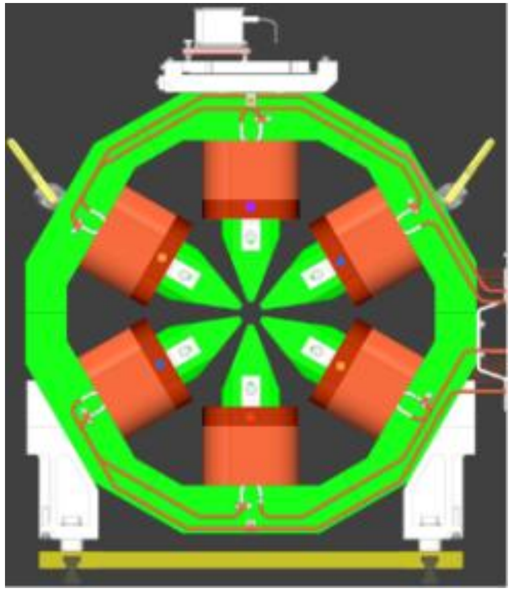
- DQ and sextupole main current changes are required for H & V steering

For further details: Power Supplies Part 2 lecture from Laurent Jolly

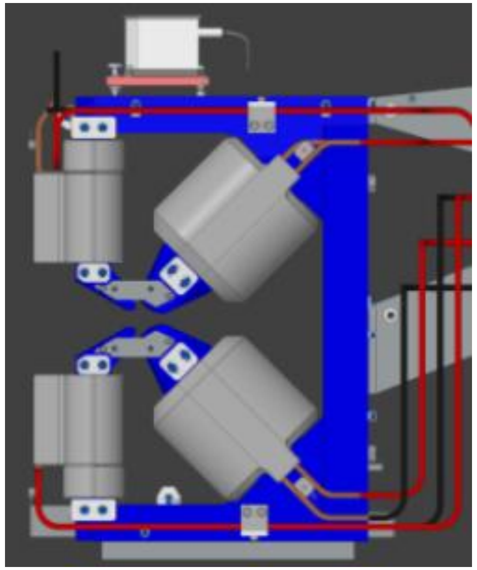
CORRECTOR POWER SUPPLIES SOFTWARE

Corrector coil power supplies (Bilt FOFB and **Bilt EBS** from iTest**)**

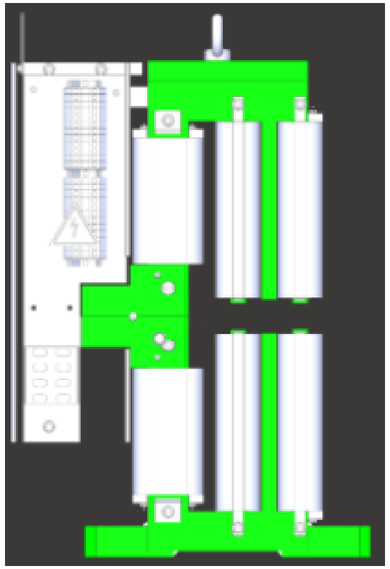
- **768** corrector coil currents for sextupoles,
- **96** corrector coil currents for DQ,
- **292** (9x31 + 2x5 + 3) corrector coil currents for SH,
- **SCPI** protocol over TCP/IP



Sexutpole
1 MainPS, 4 CorrPS
Horz, Vert, SkewQuad, Sext



Dipole Quadrupole
1 MainPS, 1 CorrPS
Horz, Quad



SH
3 or 5 CorrPS
Horz, Vert, SkewQuad (,Sext)

For further details: FOFB Orbit correction lecture from Benoit Roche, IDM Confluence pages

EBSMAGNET SERVER (MAGNET MODEL #1)

Object Oriented Programming

- Avoid code duplication
- Prevent calculation conflicts
- Database is used for configuration
- Based on Eigen C++ linear algebra library

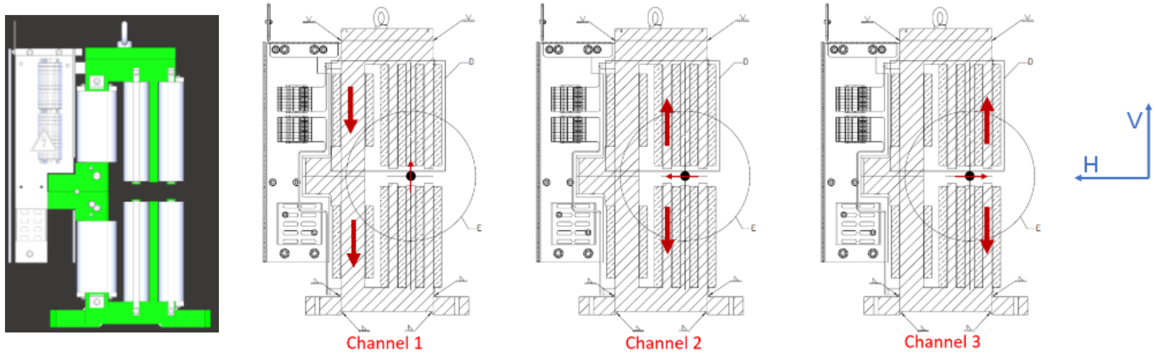
MagnetModel *SuperClass*

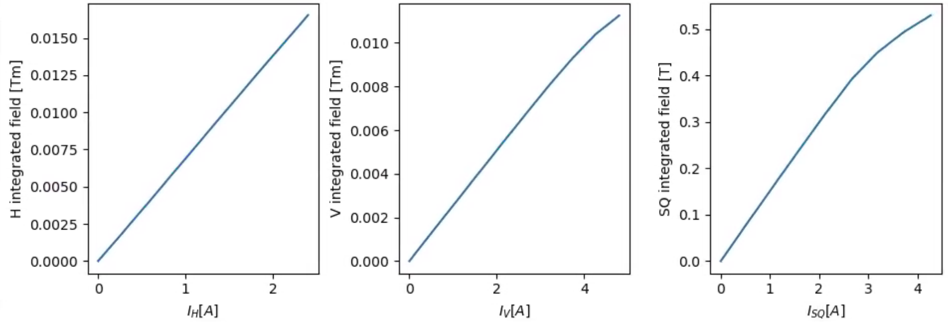
```
bool has_main_current();
void get_strength_names(std::vector<std::string> &names);
void get_strength_units(std::vector<std::string> &units);
void compute_strengths(double magnet_rigidity_inv, std::vector<double> &in_currents, std::vector<double> &out_strengths);
void compute_currents(double magnet_rigidity, std::vector<double> &in_strengths, std::vector<double> &out_currents);
void compute_pseudo_currents_from_currents(std::vector<double>& in_currents, std::vector<double>& out_currents) {};  // For FOFB AC offset
void compute_strength_from_pseudo(double magnet_rigidity_inv, int idx, double& in_current, double& out_strength) {};  // For FOFB AC offset
… // SuperClass methods are not listed here
```

Quadrupole | DQuadrupole | Sextupole | Octupole | SH3 | SH5

EBSMAGNET SERVER (MAGNET MODEL #2)

- Compute coil currents from strength setpoints
- Calibration data comes from magnetic measurements (IDM)

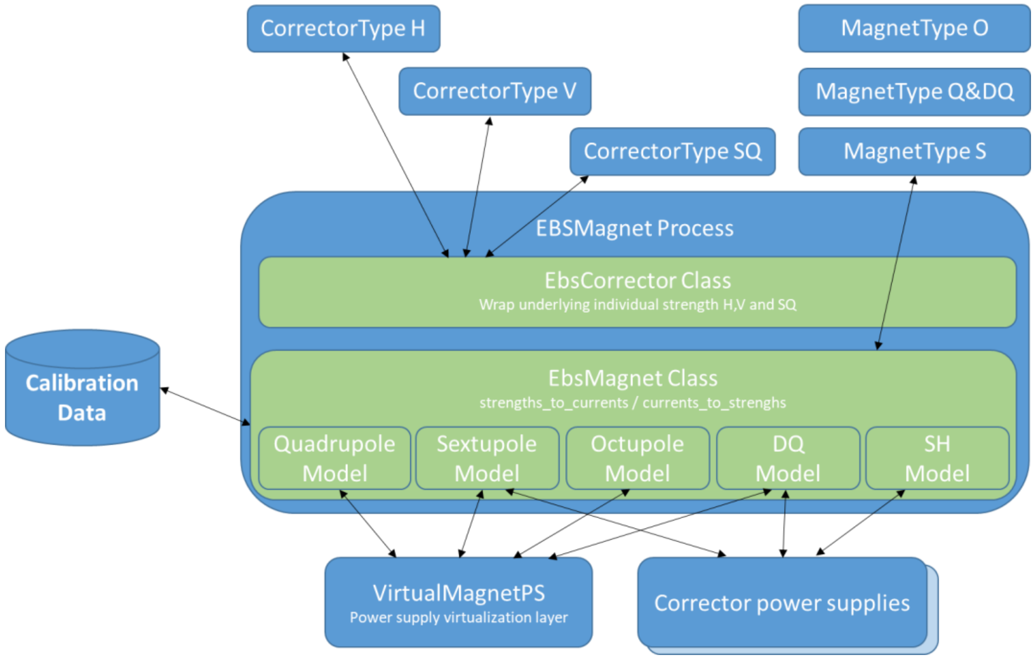


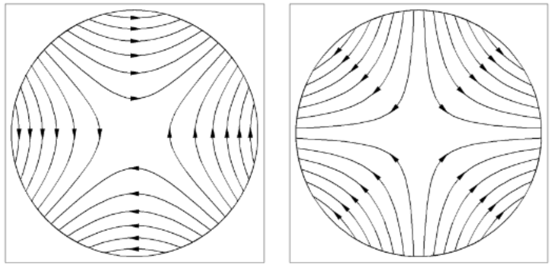


Excitation curves H,V and skew quad

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1.061 \\ 0 & 0.962 & 1 \end{pmatrix}$$

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = M^{+} \begin{pmatrix} I_H \\ I_V \\ I_{SQ} \end{pmatrix} \qquad \begin{pmatrix} I_H \\ I_V \\ I_{SQ} \end{pmatrix} = M \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix}$$

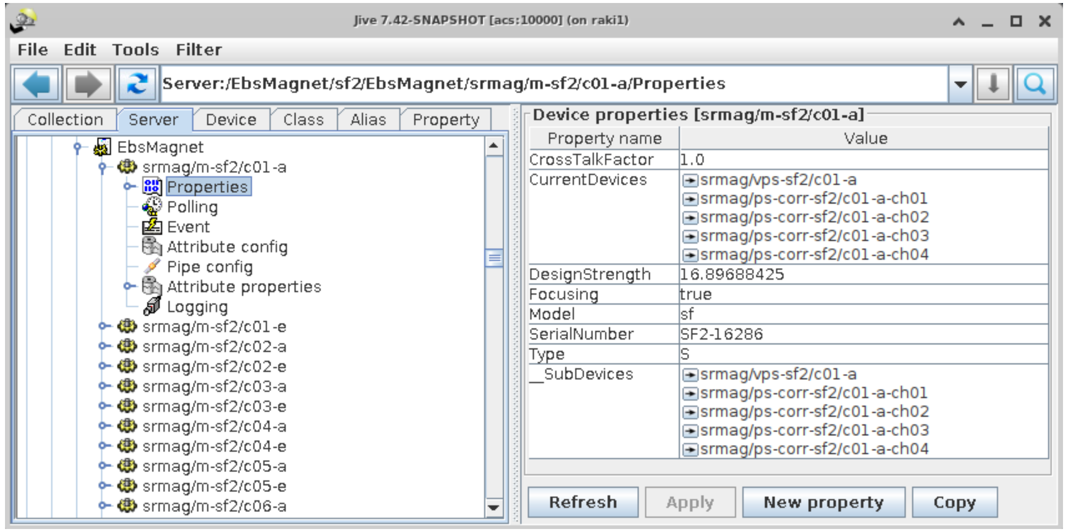


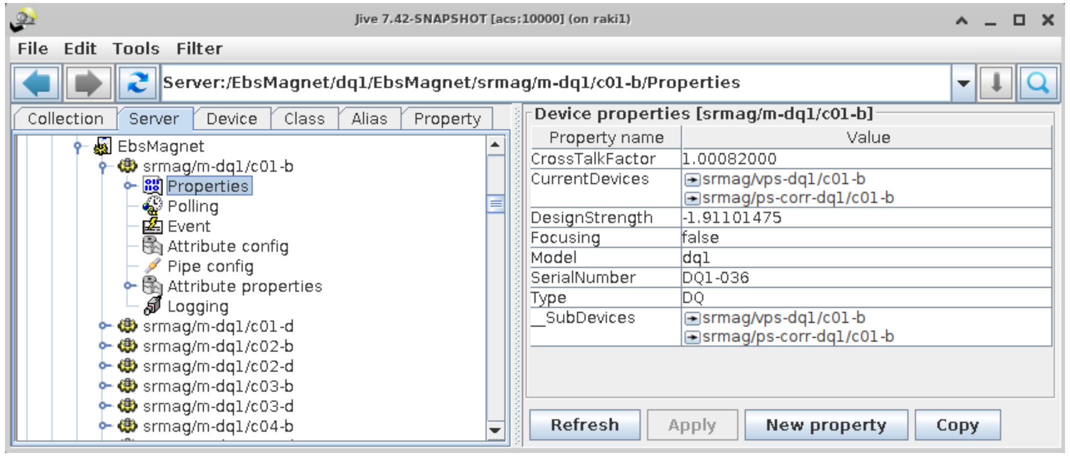


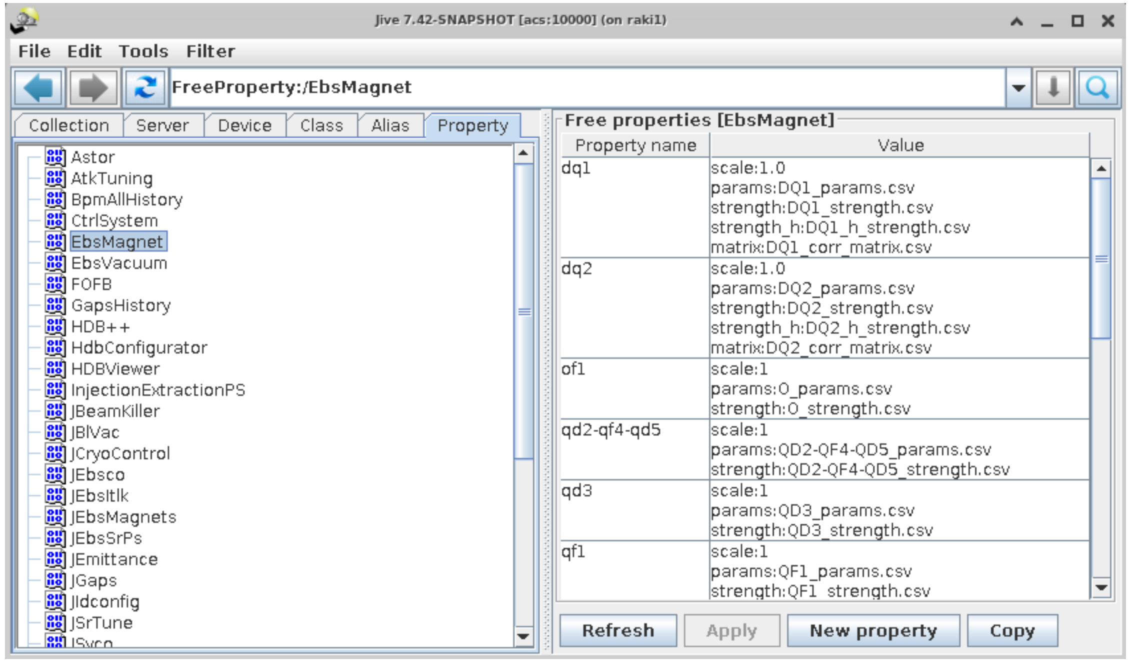
Normal Quad   Skew quad

To get strengths, integrated fields are normalized by the magnet rigidity Bρ

For further details: IDM confluence pages







Magnet configuration

- All specific settings are in the database and on file system
- Only one Tango class (EbsMagnet) handle all magnet types.

EBSMAGNET SERVER (OBJET ORIENTED SOURCE)

```
// ------------------------------------------------------------------------------------------
DipoleQuadrupole::DipoleQuadrupole(EbsMagnet *ds, bool focusing,double crossTalk,string &serial,string &magModel,string& type):
        Magnet(ds,type,serial,magModel) {

  // Init model
  auto *dq = new MagnetModel::DipoleQuadrupole();
  try {
    dq->init(focusing,
             crossTalk*getConfigItemDouble("scale"),
             getConfigItem("_path")+getConfigItem("strength"),
             getConfigItem("_path")+getConfigItem("params"),
             getConfigItem("_path")+getConfigItem("strength_h"),
             getConfigItem("_path")+getConfigItem("matrix"),
             serial);
  } catch(std::invalid_argument& e) {
    Tango::Except::throw_exception("DipoleQuadrupoleConfigError",
                                   e.what(),
                                   "Magnet::parseConfig");
  }
  model = dq;

  // Create attributes
  createAttributes();

}
```

```
void Magnet::writeStrengths(vector<double>& strengths) {

  setpointCheck(strengths);
  vector<double> newCurrents;
  model->compute_currents(MAGNET_RIGIDITY, strengths, newCurrents);
  for(size_t i=0;i<nbCurrents;i++)
    ds->setCurrents[i] = newCurrents[i];
  send_currents();

}
```

Gitlab repo: https://gitlab.esrf.fr/accelerators/Magnets

```
// ------------------------------------------------------------------------------------------
SH3Magnet::SH3Magnet(EbsMagnet *ds, bool focusing,double crossTalk,string &serial,string &magModel,string& type):
        Magnet(ds,type,serial,magModel) {

  // Init model
  auto *sh3 = new MagnetModel::SH3Magnet();
  try {
    sh3->init(crossTalk*getConfigItemDouble("scale"),
              getConfigItem("_path")+getConfigItem("strength_h"),
              getConfigItem("_path")+getConfigItem("strength_v"),
              getConfigItem("_path")+getConfigItem("strength_sq"),
              getConfigItem("_path")+getConfigItem("matrix"));
  } catch(std::invalid_argument& e) {
    Tango::Except::throw_exception("SH3MagnetConfigError",
                                   e.what(),
                                   "Magnet::parseConfig");
  }
  model = sh3;

  // Create attributes
  createAttributes();

}
```

```
void Magnet::send_currents() {

  ds->attr_RequireCycling_read[0] = true;

  vector<Tango::DeviceAttribute> attributes;
  for(double setCurrent : ds->setCurrents) {
    Tango::DeviceAttribute attribute("Current", setCurrent);
    attributes.push_back(attribute);
  }

  Tango::GroupReplyList arl = ds->psGroup->write_attribute(attributes);

  if (arl.has_failed()) {
```

EBSMAGNET SERVER (USAGE)

Lorentz force and centripetal acceleration (v≈c): $F_{\perp} = qcB = \gamma m \frac{c^2}{\rho} = \frac{E_{[Joule]}}{\rho}$

Magnet rigidity: $B\rho = \frac{10^9}{c} E_{[GeV]} \approx 3.3356 E_{[GeV]} [Tm]$

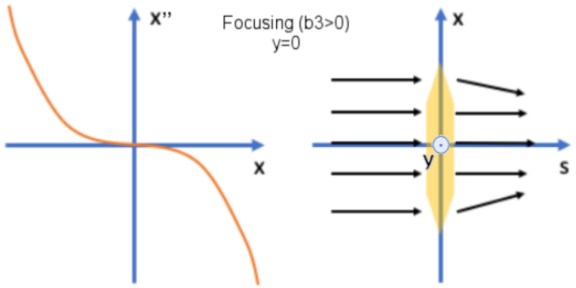
Field expansion: $\frac{b_y + ib_x}{B\rho} = \frac{q}{\gamma_0 mc}(b_y + ib_x) = \Sigma_{n=0}^{k}(ia_n + b_n)(x+iy)^n$

- ρ is the radius of curvature of the particle trajectory for a given perpendicular field B at the nominal energy (Bρ normalize the tracking code according to the nominal momentum)
- L is the magnet length
- (b_x,b_y) is the transverse field at a specified position (x,y) in [T]
- (a_n,b_n) the multipole field strengths (skew and normal) expressed in $[m^{-(n+1)}]$
- k is the multipole order, k=0 for a dipole, k=1 for a quadrupole, k=2 for a sextupole, etc…

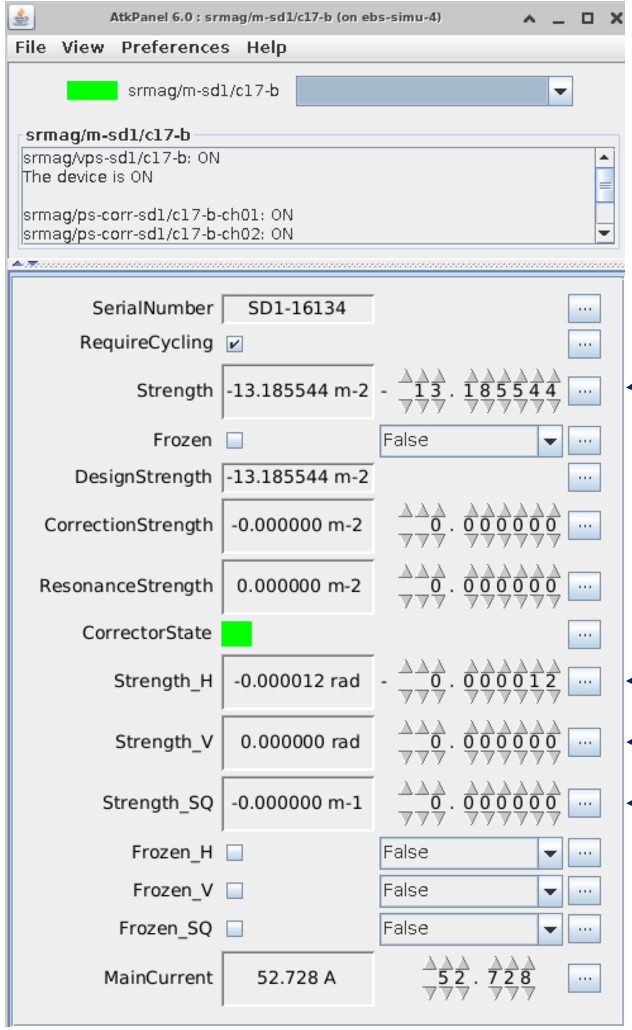
Example of transverse motion for a thin octupole (k=3):
(On-energy 4D tracking with small angle approximation)

$x_1 = x_0 + x'_0 \Delta l \qquad x'_1 = x'_0 - b_3(x^3 - 3xy^2)\Delta l$

$y_1 = y_0 + y'_0 \Delta l \qquad y'_1 = y'_0 + b_3(3x^2y - y^3)\Delta l$



Note: x' and y' has no unit, an octupole focus (or defocus $b_3<0$) in both planes



$L.b_2$

$\arcsin(L.b_0)$

$\arcsin(L.a_0)$

$L.a_1$

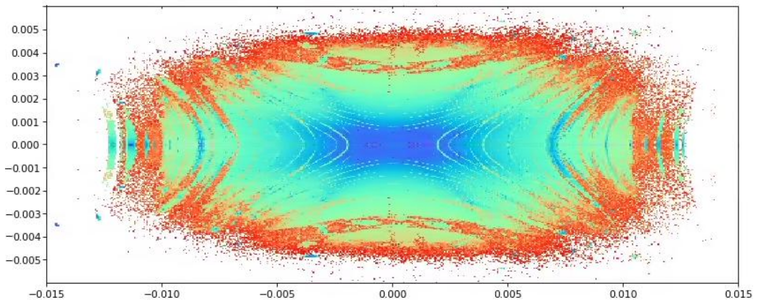
For further details: Magnetic measurements of accelerator magnets and insertion devices lecture from Gael Le Bec
Accelerator Magnets lecture from Chamsedine Benabderrahmane
Beam Dynamics confluence pages

Orbit Response Matrix

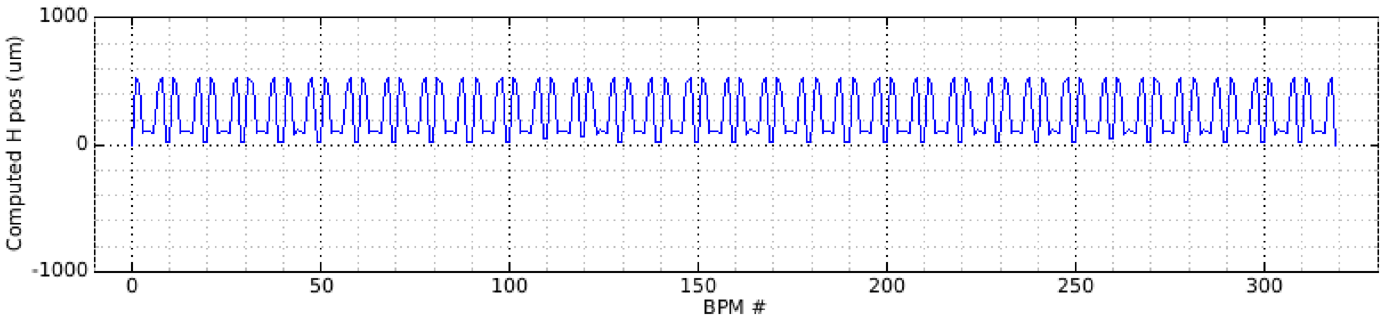
- For orbit correction, we compute the ORM from the model using tracking
- Horizontal: **320** BPM x **385** H Steerer (96 SH + 192 Sextupole + 96 DQ + 1RF)

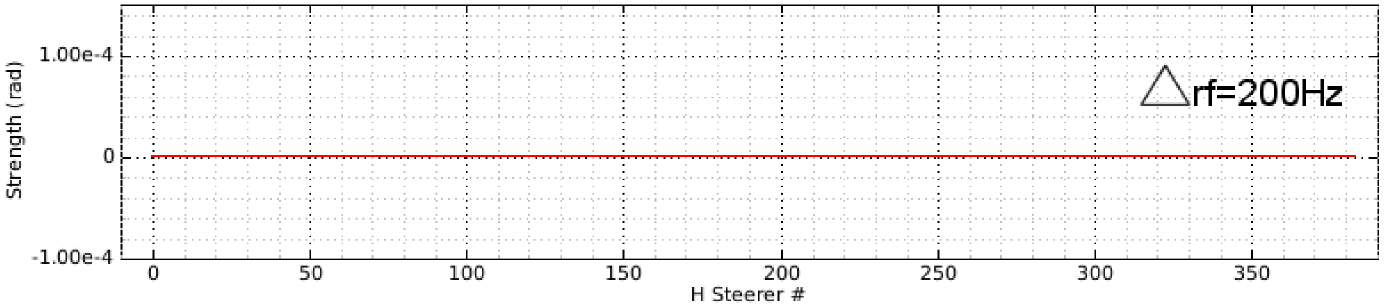
$$M = \begin{bmatrix} \frac{\partial x_{BPM1}}{\partial k_{ST1}} & \cdots & \frac{\partial x_{BPM1}}{\partial k_{STn}} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_{BPMn}}{\partial k_{ST1}} & \cdots & \frac{\partial x_{BPMn}}{\partial k_{STn}} \end{bmatrix}$$

$$\delta x = M \ \delta K$$



EBS Dynamic Aperture at injection (no error, no radiation)



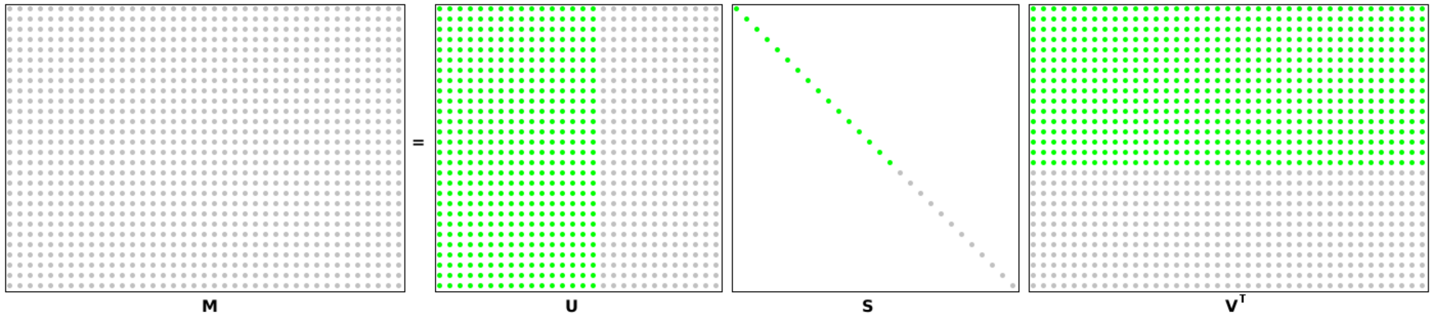


For further details: Beam measurements and Injection lecture from Simon White
EBS design and commissioning lecture from Simone M.Luizzo

ORBIT CORRECTION (SVD)

Pseudo Inverse of M using SVD

- Singular values are always positive reals and are sorted in descending order
- We can select a subset of singular values (to smooth the correction, to avoid too large steerer strengths)
- We often call them “eigen” values
- Selecting a subset of “eigen” values also select a subset of “eigen” vectors



$$M^{+} = V \, . \, S^{-1} \, . \, U^{T}$$

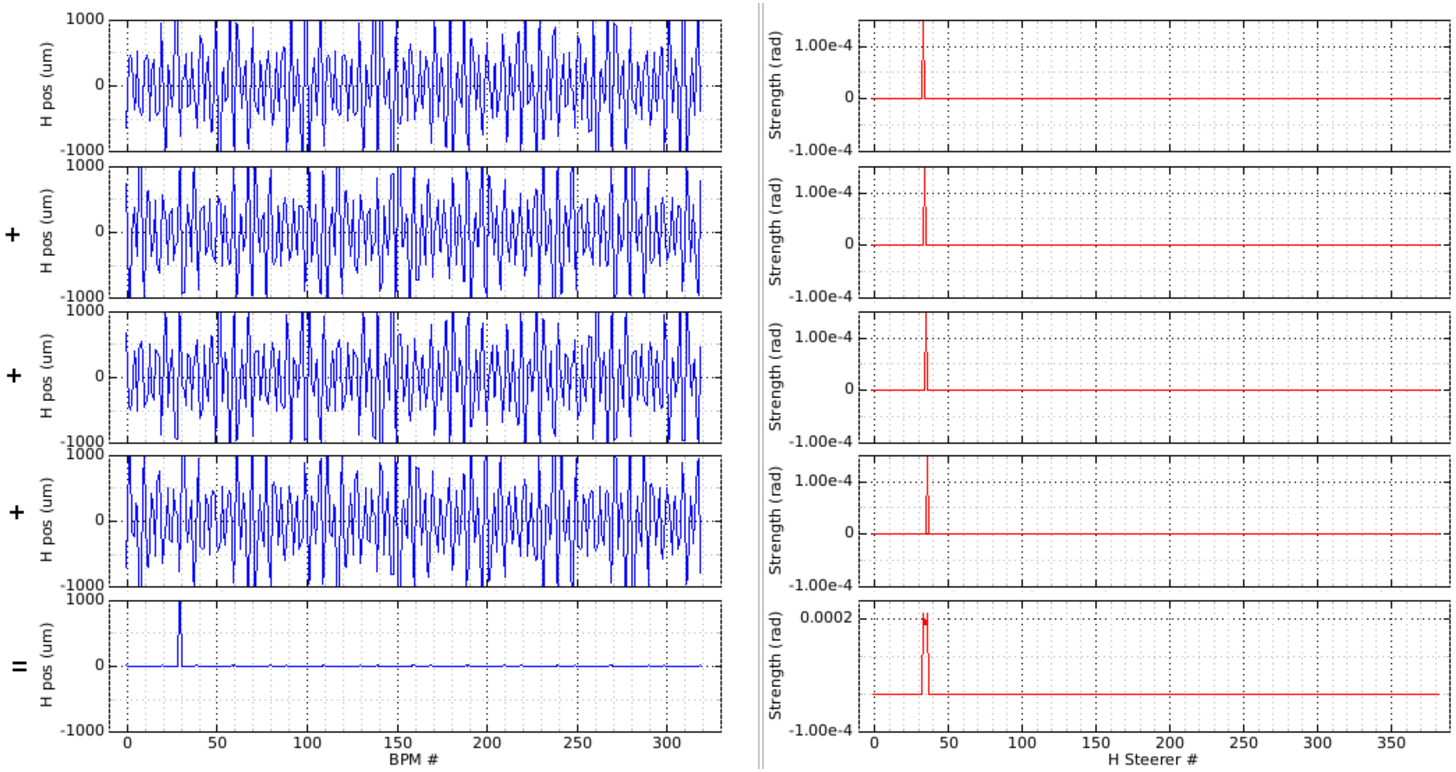
- More advanced singular value selection (or regularization) can be performed such as Tikonov filtering

The European Synchrotron | ESRF

ORBIT CORRECTION (CORRECTION ALGORITHM)

Find a linear combination of steerer responses to get the desired closed orbit using M^+

$$(\Delta k_{ST1}, \dots, \Delta k_{STn}) = M^+ \begin{pmatrix} \Delta x_{BPM1} \\ \vdots \\ \Delta x_{BPMn} \end{pmatrix}$$



SR AutoCorr loop

- 1 correction / minute in USM
- 5 correction / minute in MDT

Next objective

- Correct both planes at once
- 1 correction / 3 sec in MDT

For further details: Beam measurements and Injection lecture from Simon White
Single particle dynamics and synchrotron radiations lecture from Nicola Carmignani

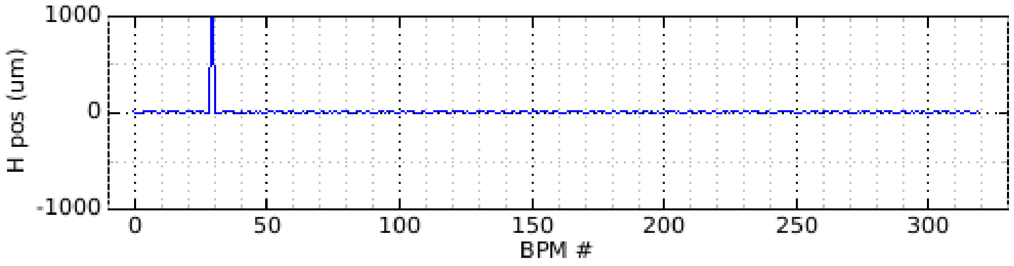
ORBIT CORRECTION (STEERER SUM)

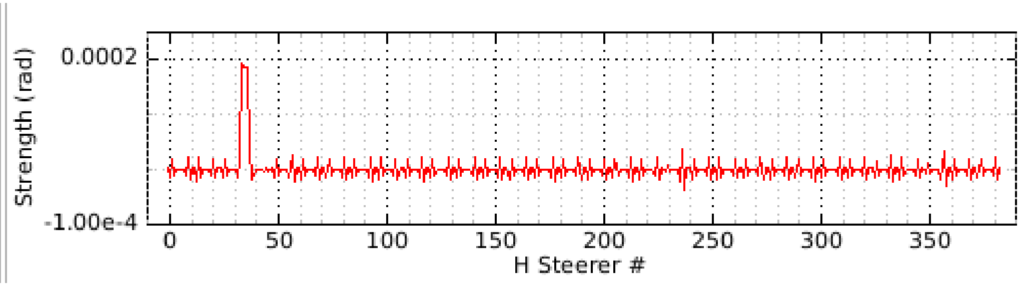
Avoid energy change

- In horizontal, we have to maintain a steerer sum equal to zero (don't add dipole) to avoid energy drift
- We trick the ORM by adding a virtual BPM

$$\begin{pmatrix} bpm_1 \\ bpm_2 \\ bpm_3 \\ \vdots \\ bpm_{virt} \end{pmatrix} = \begin{pmatrix} \frac{\delta bpm_1}{\delta k1} & \frac{\delta bpm_1}{\delta k2} & \cdots & \frac{\delta bpm_1}{\delta rf} \\ \frac{\delta bpm_2}{\delta k1} & \frac{\delta bpm_2}{\delta k2} & \cdots & \frac{\delta bpm_2}{\delta rf} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_{rf} \end{pmatrix} \qquad bpm_{virt} = \Sigma k_i$$

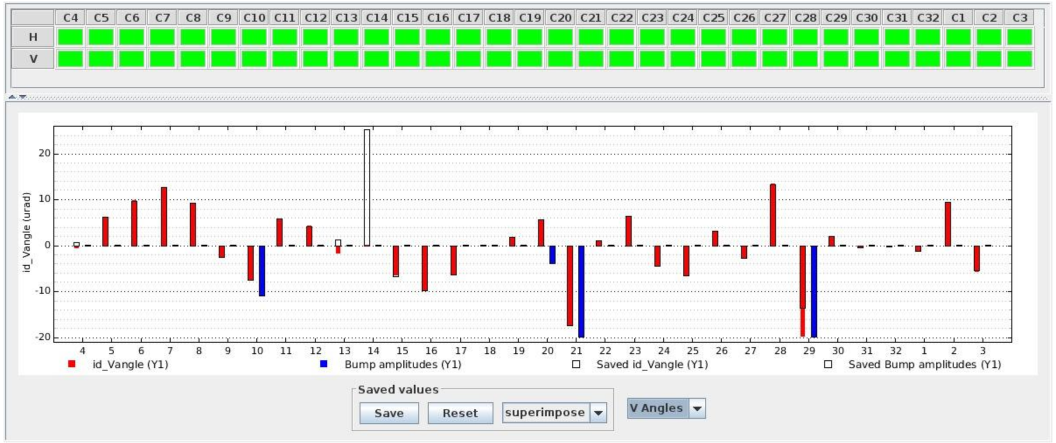
- As for a position we want to reach, using the pseudo inverse of the above ORM, we can add a steerer sum to target
- Problem: When performing a parallel bump, an unwanted steerer pattern appears, need to be improved



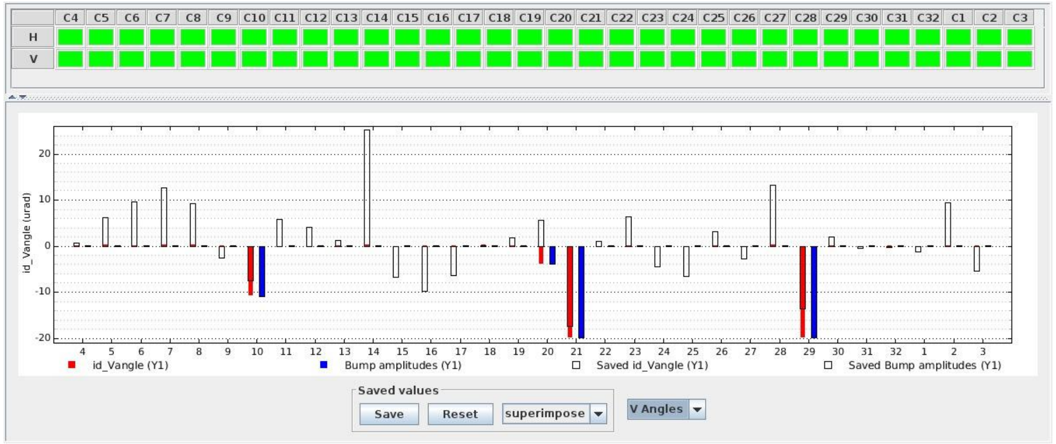


ORBIT CORRECTION (BPM WEIGHTS)

- In order to improve the correction in specific area (typically to improve bump accuracy) we can add BPM weights.



All weights to 1



Weights of all BPM around straight sections to 10 (MDT 25/04/2022)

Weights are applied on the ORM
W is a diagonal matrix containing BPM weights

$$U, S, V = SVD(W \cdot M)$$

The correction has to done on weighted errors

$$\delta K = (W \cdot M)^{+} \cdot W \cdot \delta x$$

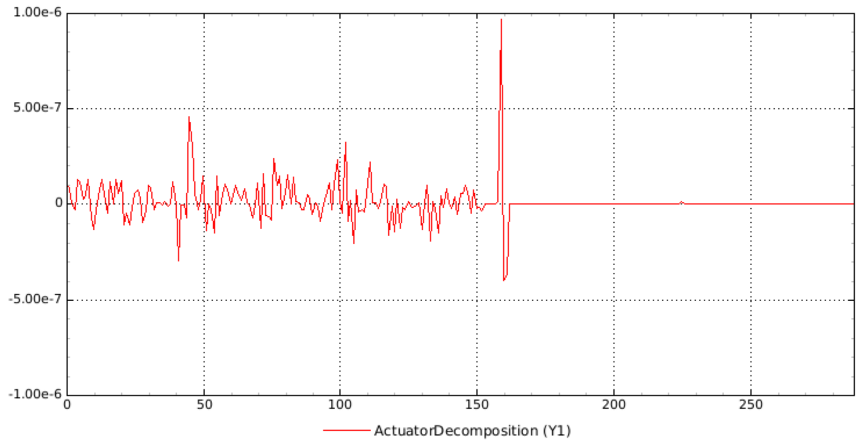
ORBIT CORRECTION (TRUNCATION)

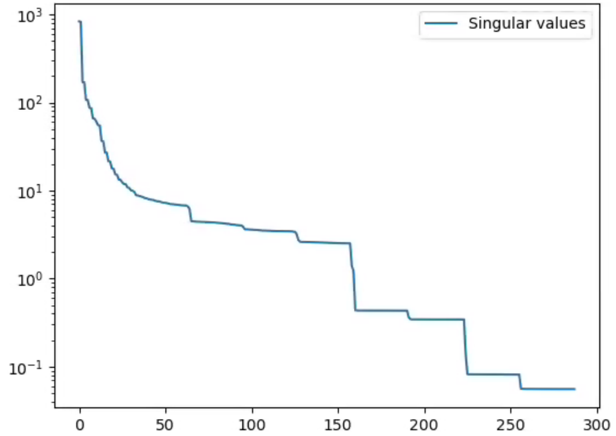
Truncation of correction (bump contribution is not truncated)

- As S^{-1} is truncated (we usually use only 162 vectors for orbit correction), the vector K_{vec} must have last components equal to 0
- K_{bump} is computed using more vectors

$$K_{corr} = M^{+}\epsilon_x = V.(S^{-1}.U^T.\epsilon_x) = V.K_{vec}$$

$$K = K_{corr} + K_{bump}$$





- The Fast Orbit feedback works with an other matrix (96 steerers), when AC correction offsets are reinjected into DC correction (Bilt FOFB), the above rule is broken. We perform a cleaning of the steerer vector:

$$K_{vec} = V^T(K - K_{bump})$$

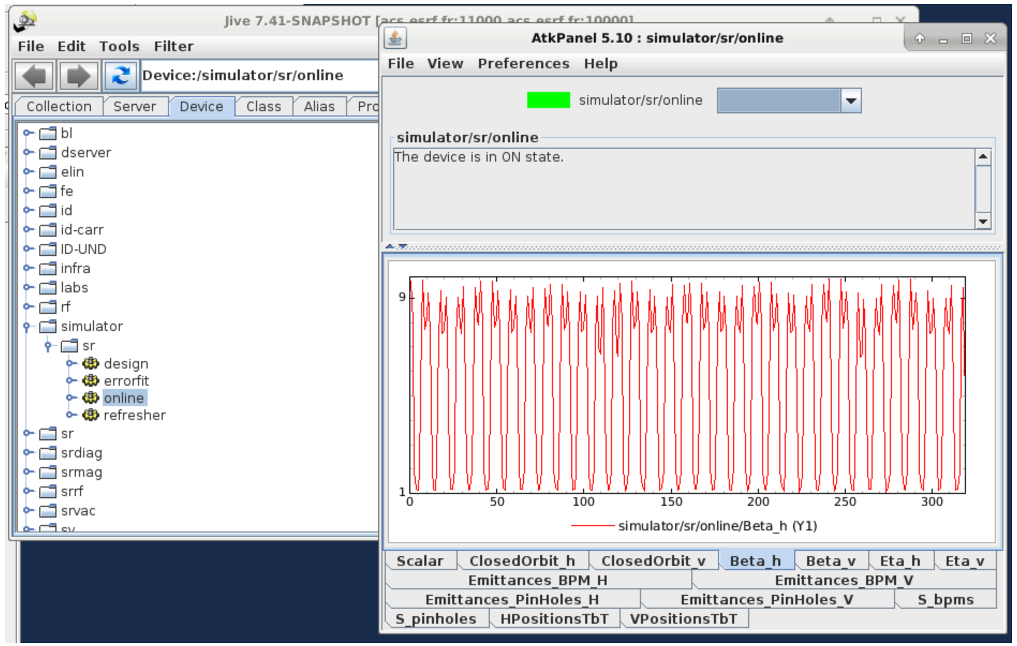
$$K_{vec}[162:] = 0$$

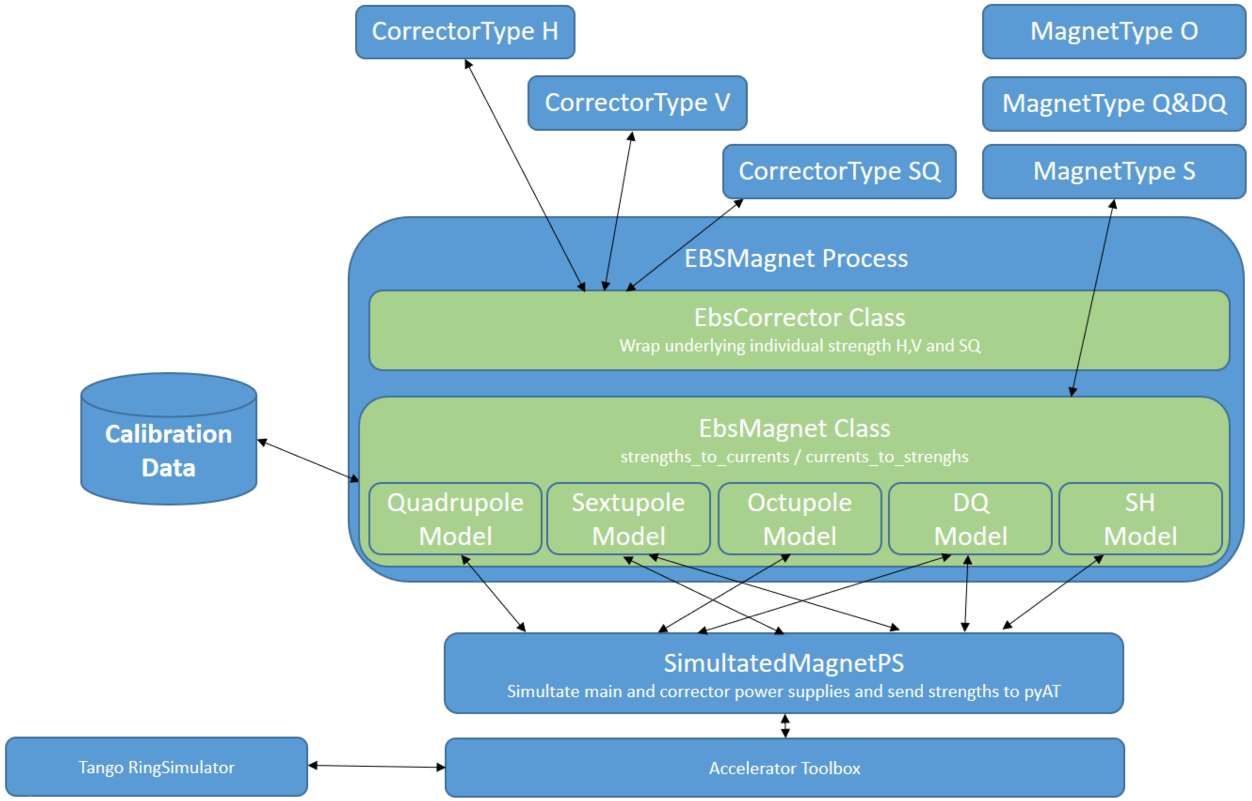
$$K = V.K_{vec} + K_{bump}$$

EBS DIGITAL TWIN (ARCHITECTURE)

- EbsMagnet server code is the same on both machine and simulator instances
  (no #ifdef SIMULATOR directive in the code)

- Allow to run same servers/applications on all systems
  (no specific code for EBS simulators)

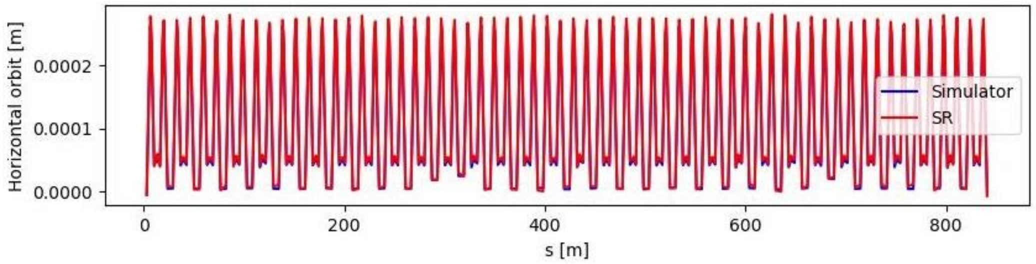


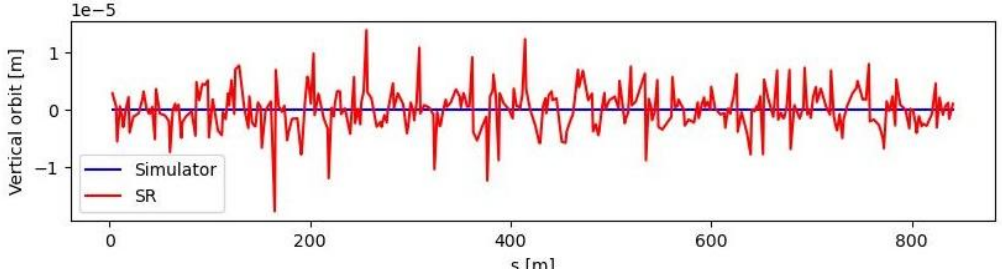


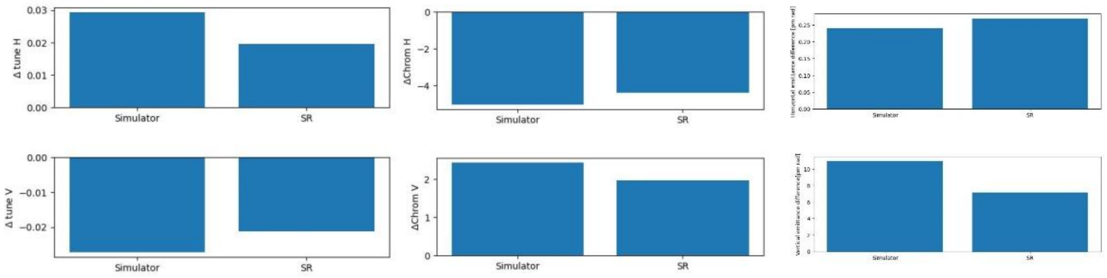
- DigtalTwin has been shared with Elettra and Soleil

EBS DIGITAL TWIN (TESTS IN MDT)

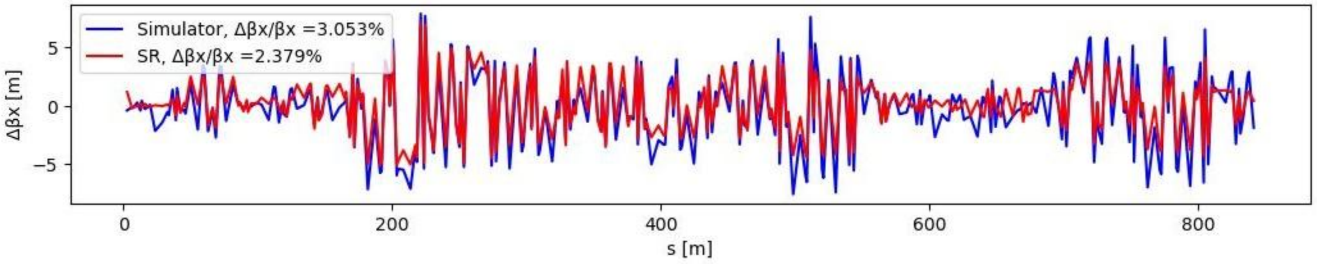
The simulators device-server is adapted to work as online “digital twin” of the EBS beam dynamics properties.
Provides **relative** optics change compared to an initial set of magnets strengths.

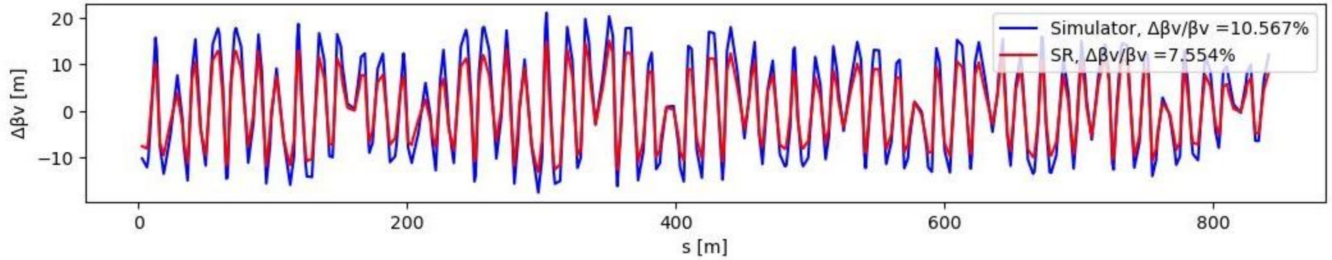


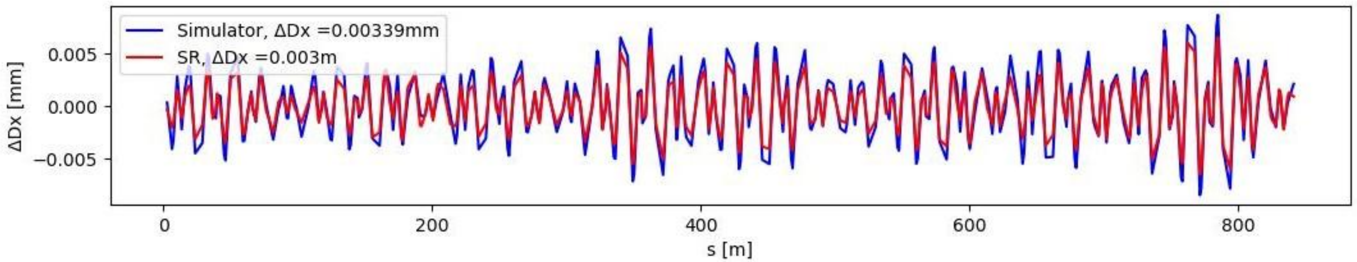




Tune Chromaticity Emittances



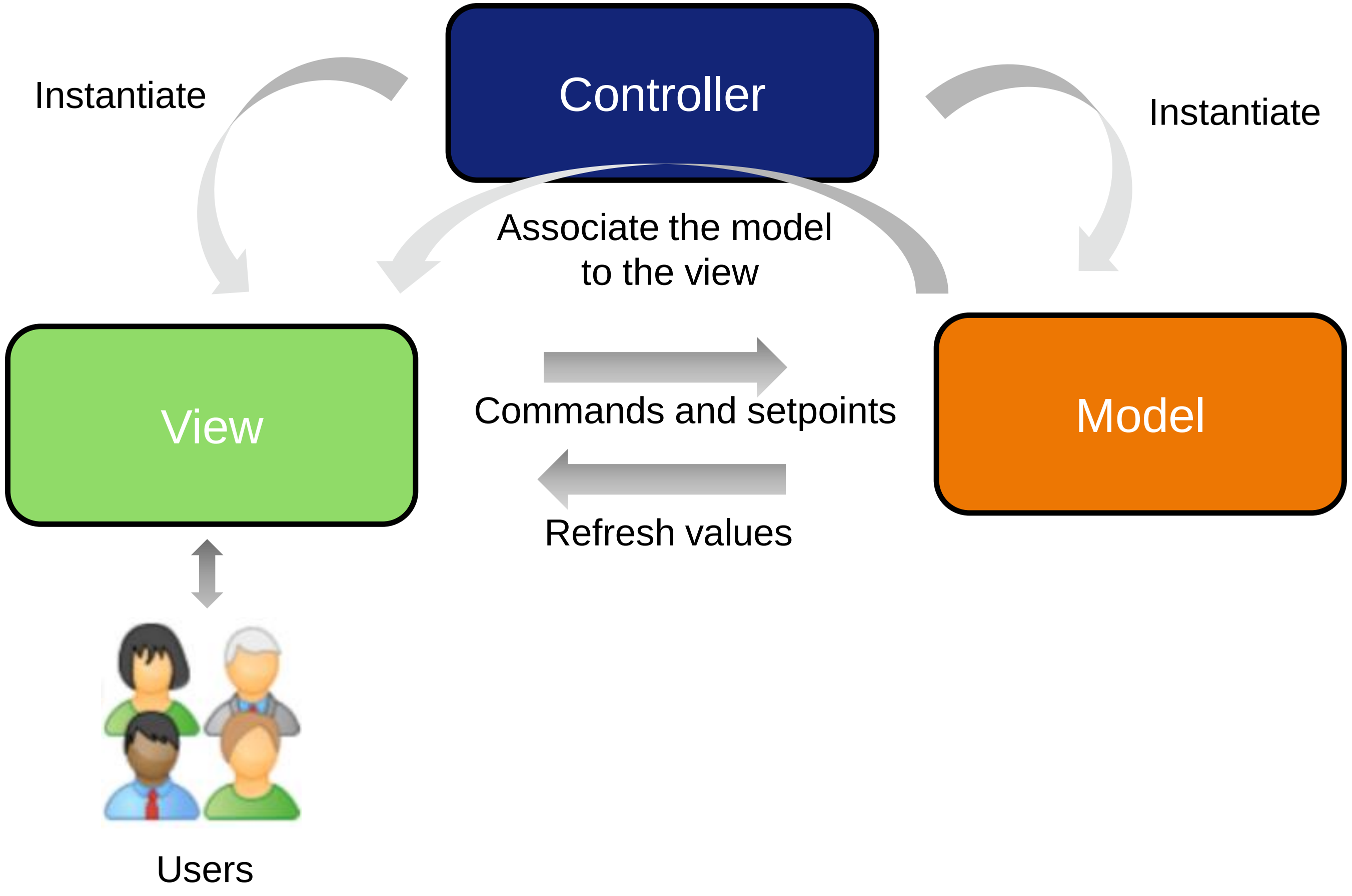




All artificially introduced variations in the SR are observed in the digital twin (MDT 26/08/2023)

TANGO ATK APPLICATION FRAMEWORK (MVC)

Tango Application ToolKit (ATK) is a java framework based on MVC design pattern on top of Java Swing



ATK Core

Classes implementing Tango models

ATK Widget

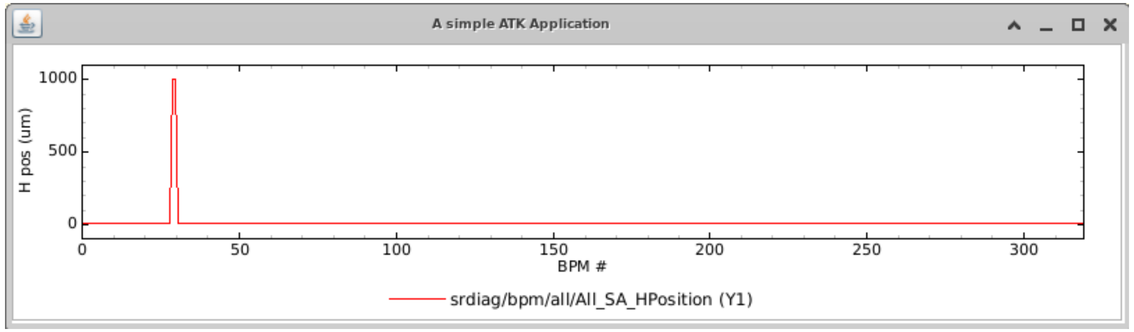
~ 80 attribute viewers/editors
~ 30 command handlers

Advanced image viewers (used for emittance)
Synoptic editor (JDraw)
2D plot (JLChart)
Advanced error handling

JAVA TANGO ATK FRAMEWORK (EXAMPLE)

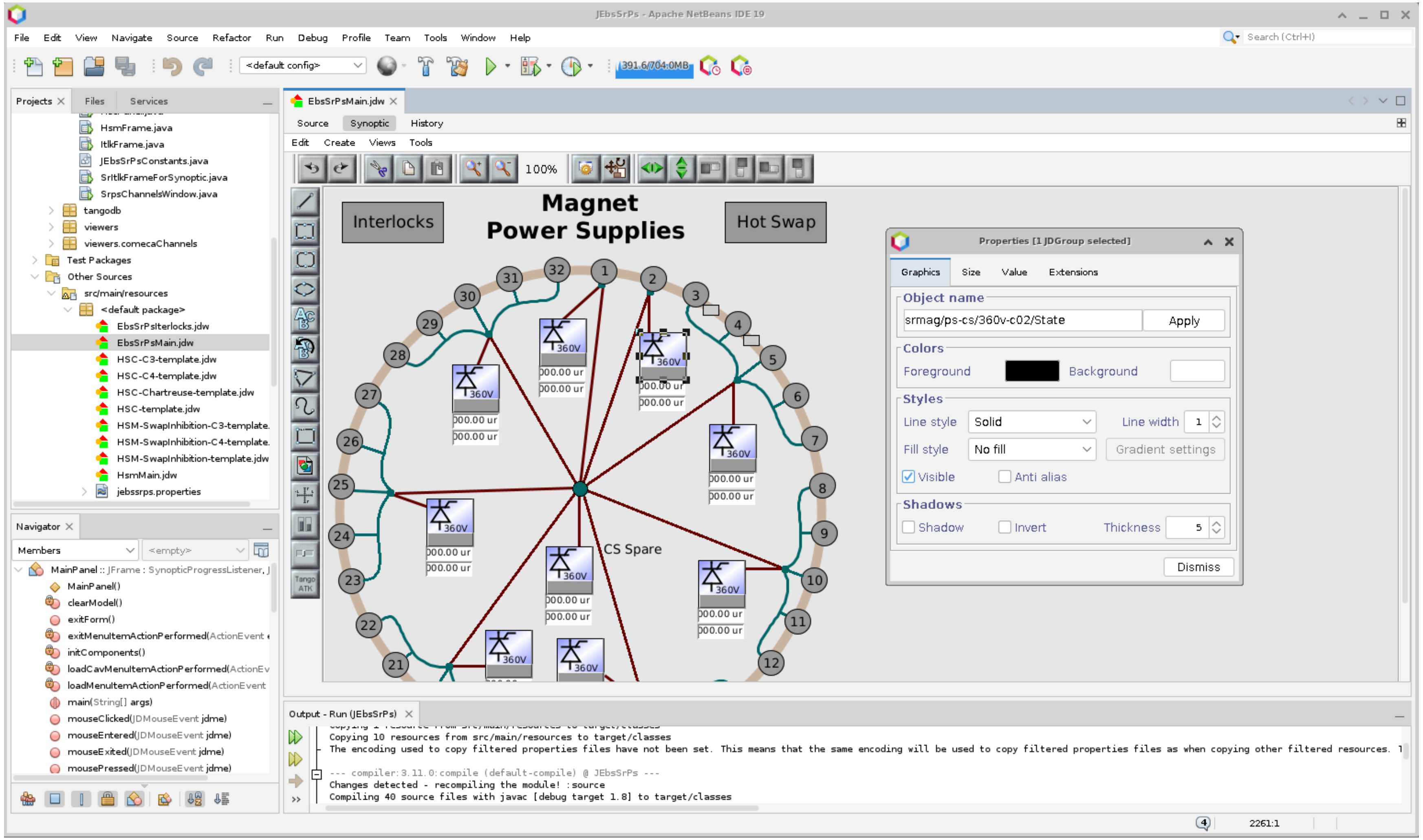
```java
import fr.esrf.tangoatk.core.*;
import fr.esrf.tangoatk.core.attribute.*;
import fr.esrf.tangoatk.widget.attribute.*;
import fr.esrf.tangoatk.widget.util.ATKGraphicsUtils;
import javax.swing.*;
import java.awt.*;

public class TestATK extends JFrame {

  private TestATK() {

    AttributeList attList = new AttributeList(); // List of models
    attList.setRefreshInterval(1000);

    NumberSpectrumViewer orbitView = new NumberSpectrumViewer(); // Instantiate the viewer
    try {
      // Instantiate the model
      NumberSpectrum orbitModel = (NumberSpectrum) attList.add("srdiag/bpm/all/All_SA_HPosition");
      // Associate the model to the viewer
      orbitView.setModel(orbitModel);
    } catch (ConnectionException e) {
      System.out.println("Fail: " + e.getMessage());
      System.exit( status: -1);
    }
    orbitView.setBackground(Color.WHITE);
    orbitView.getY1Axis().setName("H pos (um)");
    orbitView.getXAxis().setName("BPM #");

    attList.startRefresher();
    orbitView.setPreferredSize(new Dimension( width: 800, height: 200));
    setContentPane(orbitView);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    ATKGraphicsUtils.centerFrameOnScreen(this);
    setTitle("A simple ATK Application");
    setVisible(true);

  }

  public static void main(String[] args) { new TestATK(); }

}
```

A simple ATK application example



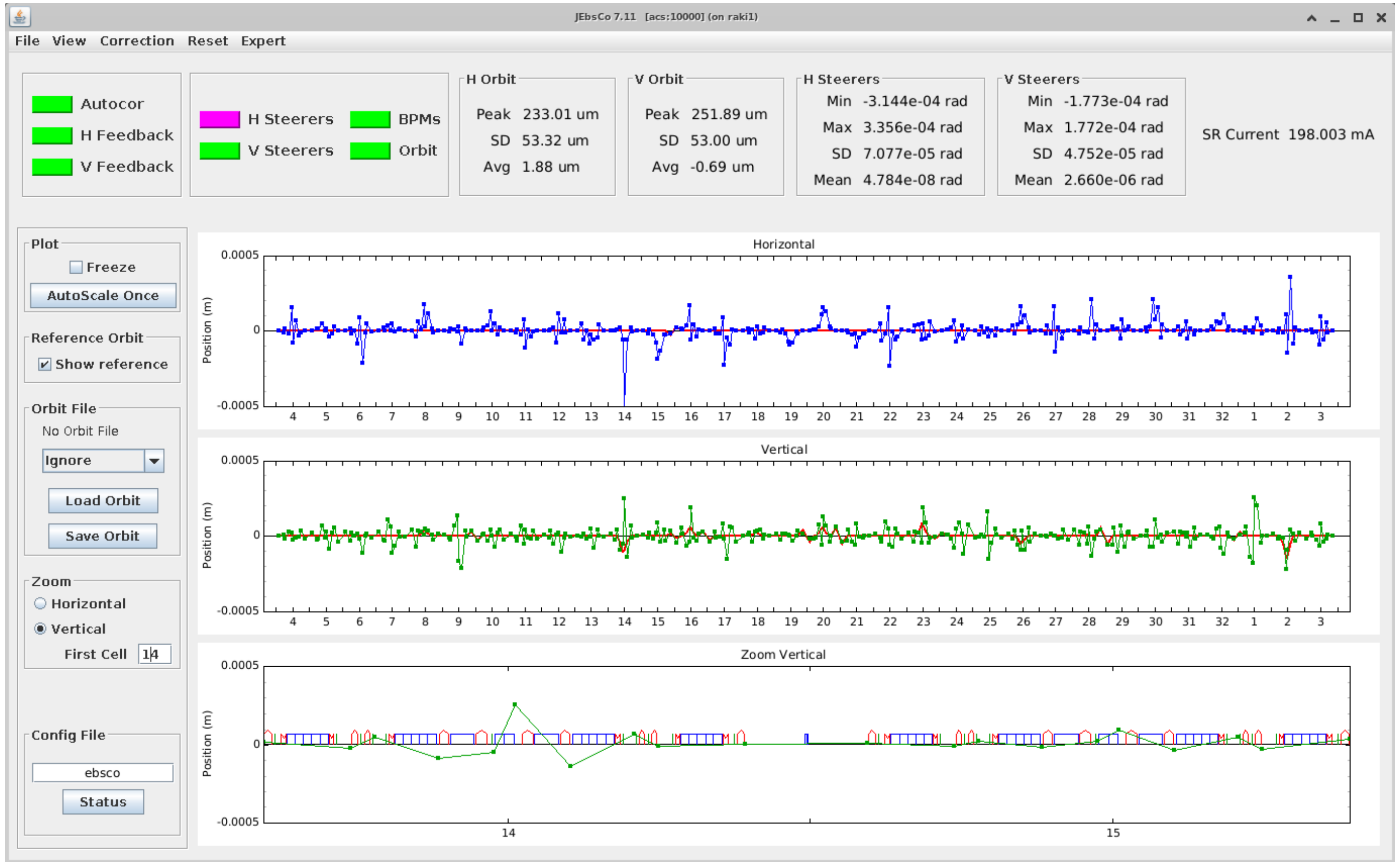
JAVA TANGO ATK FRAMEWORK (IDE)

ATK is fully integrated in Netbeans IDE (including JDraw synoptic edition)



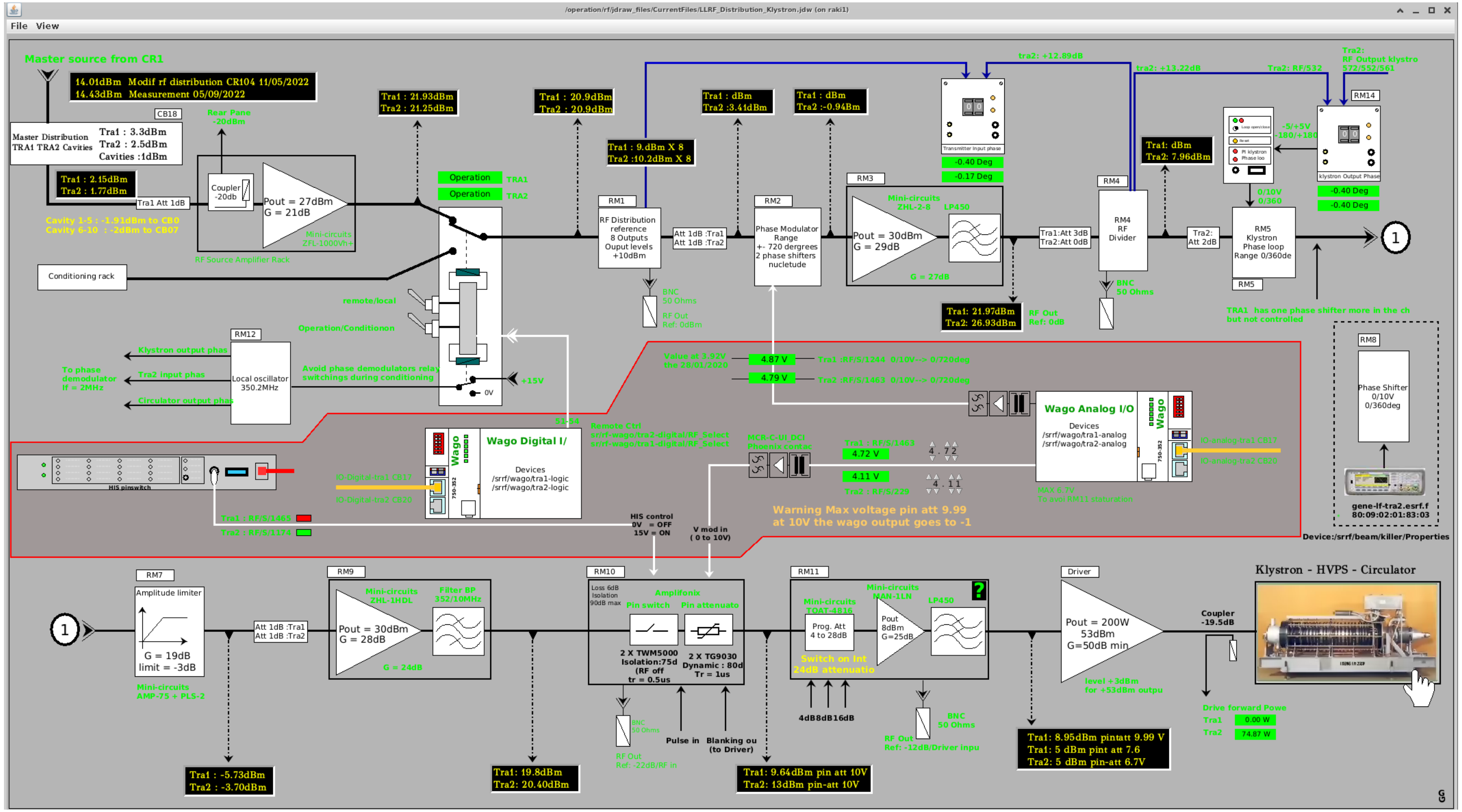
JAVA TANGO ATK FRAMEWORK (SYNOPTIC #1)

Expert application: Java (Closed orbit)



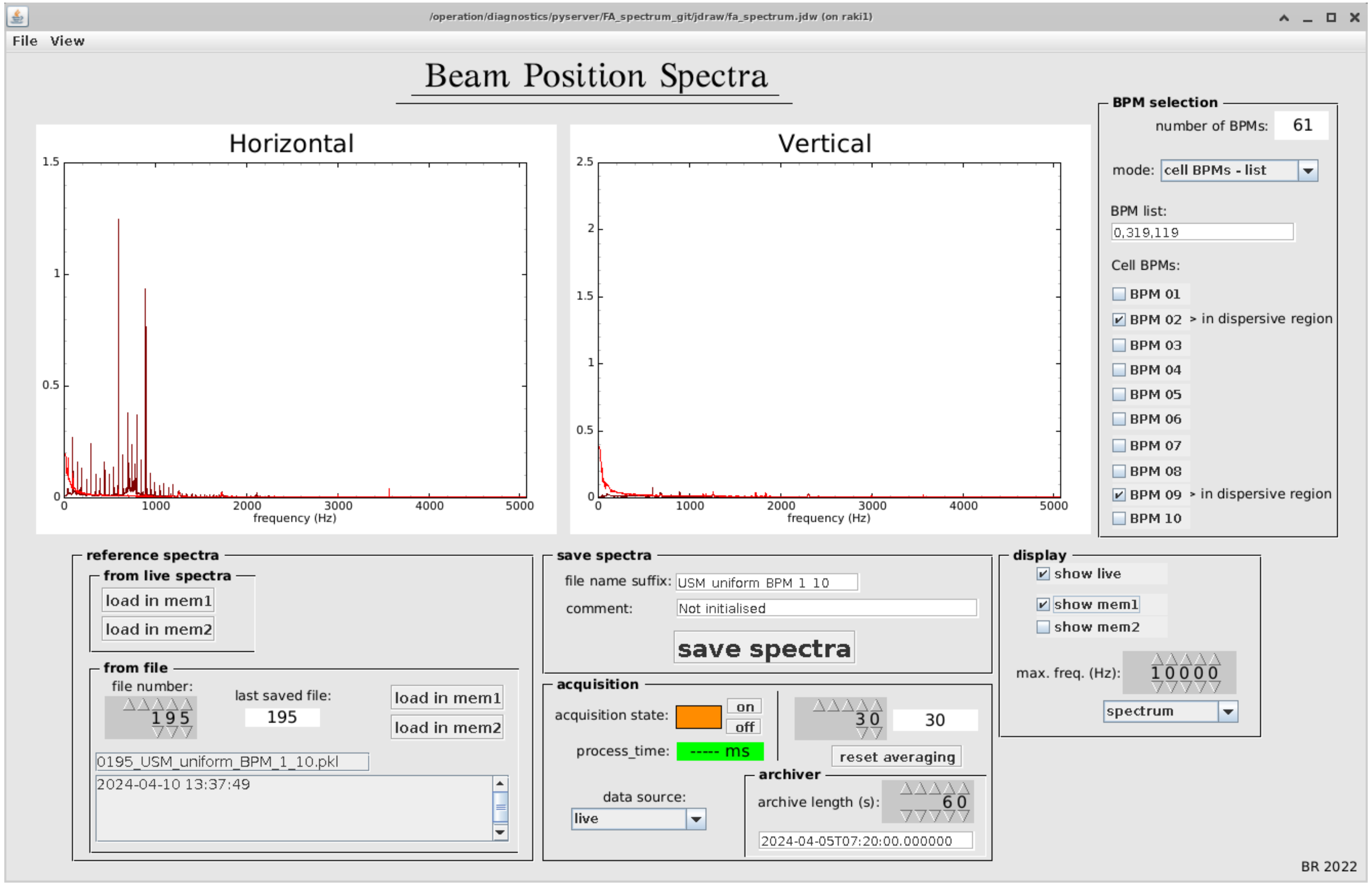
JAVA TANGO ATK FRAMEWORK (SYNOPTIC #1)

Synoptic application: No code written for the GUI (Detailed RF)



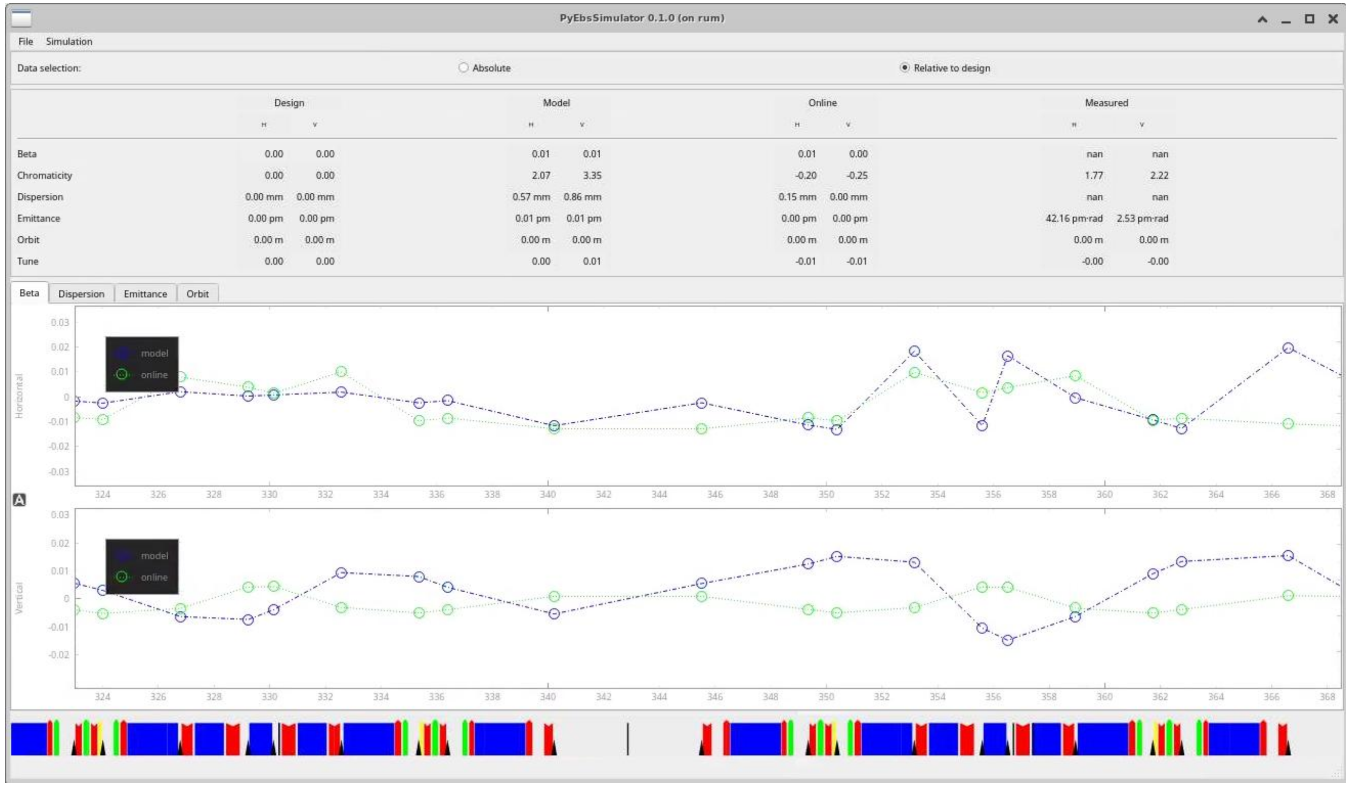
JAVA TANGO ATK FRAMEWORK (SYNOPTIC #2)

Synoptic application: No code written for the GUI (Fast Orbit Correction stability)



JAVA TANGO ATK ALTERNATIVE

Expert application: EBS Digital Twin (Taurus pyQt evaluation / On going development)



CONCLUSION

- EbsMagnet and OrbitCorrection Tango servers are good example that show how “underlying logic” is integrated in servers
- No logic in application which allow several applications to run simultaneously
- Java ATK framework also allow application standardization and intuitive usage in CTRM

- ACU is in charge of software and support for (on going developments):
  - Power Supplies (**TL2 corrector power supplies refurbishment**)
  - Radio frequency (**New SSA**, data logging, sequencing, **Linac diagnostics**)
  - Diagnostics (digit500, FDA improvements, **320 BPM for FOFB**)
  - Beam Dynamics (**Orbit correction improvement**, FILO refurbishment, Digital Twin)
  - Insertion Device and Magnets (**Magnet Bench/NLK**, ID – Monochromator synchronization)
  - Timing and others Digital Electronics control (Tango event on White Rabbit, **HSM updates**)
  - Alignment (Integrate girder motion in simulator)
  - Vacuum (Monitoring improvement of vacuum trends & RGA)
  - Operation (ID Modes fix, **JYSE replacement**)
  - Safety (**New neutrons monitors and improvements**)
  - Front End ()
  - ACU infrastructure (**CI/CD deployment**, Cyber Security constraints, OS upgrade, Tango 10, HDB++, Obsolescence: 10Mbits Half Duplex, Evaluate new GUI technologies)

Still lots of work for ACU !

Greetings go to

ASD lecture organizers
ASD speakers

Thank you