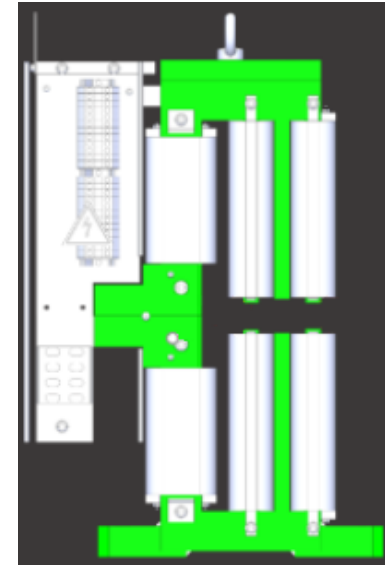
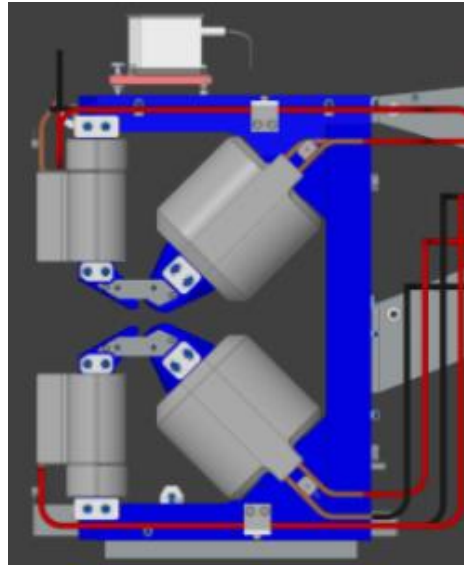
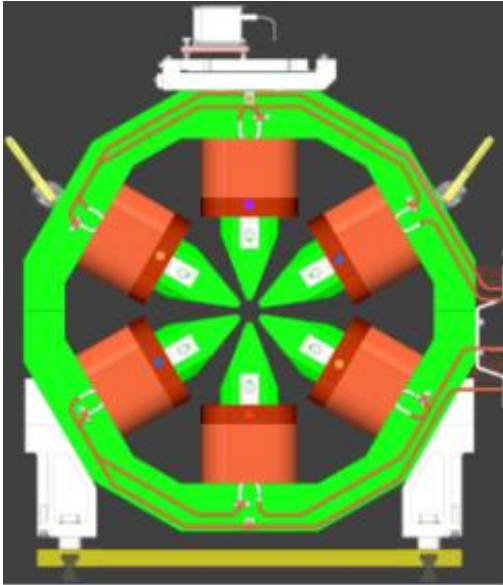


Magnet calibrations architecture at ESRF



Second Accelerator Middle Layer Workshop, 12-14 February 2025
Jean-Luc Pons

EBS MAGNETS SOFTWARE UPDATE (2022)

- **Improve maintainability/reactivity**

- Centralize magnet calibration data and source code in a single library
- Fix conflicts when applying strengths on combined function magnets
- No `#ifdef SIMULATOR` directive in the code
- Minimize hardware access, allow to set H/V at once

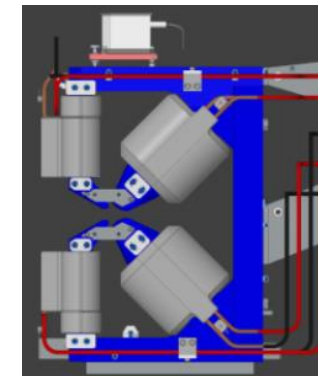
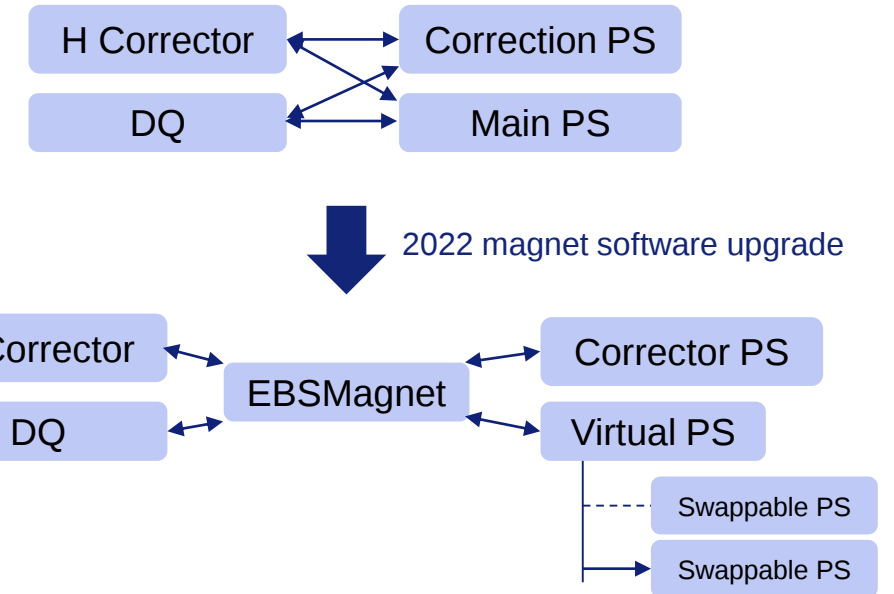
- **Power supply virtualization layer**

- Integrate HotSwap system
- Handle low level connection between magnet and PS
- Use a unique device name to access a magnet
- Allow standalone power supply (QF1E QF1A for injection)
- Allow HotSwap spare usage for mini-beta optics evaluation (using additional quads)

- **Digital shadow / Simulator**

- Simulate magnet cycling and electrical power overhead
- Implement digital twin application

- **EBS: 1056 swappable PS / 1156 corrector PS**



- <https://gitlab.esrf.fr/accelerators/Magnets/MagnetModel>
- 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

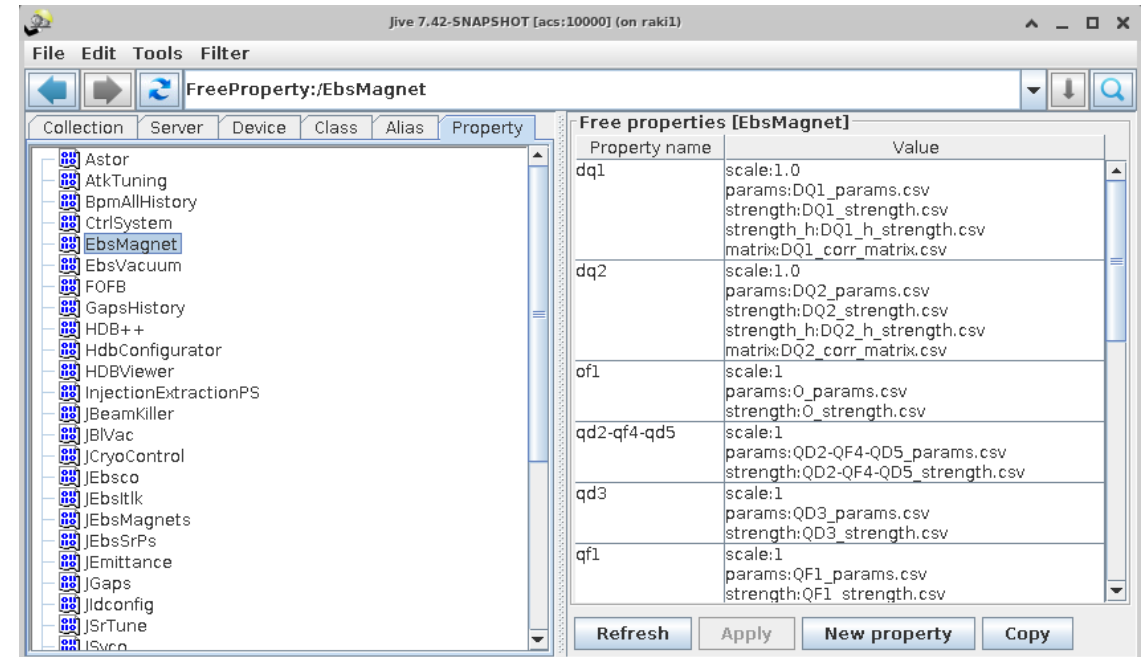
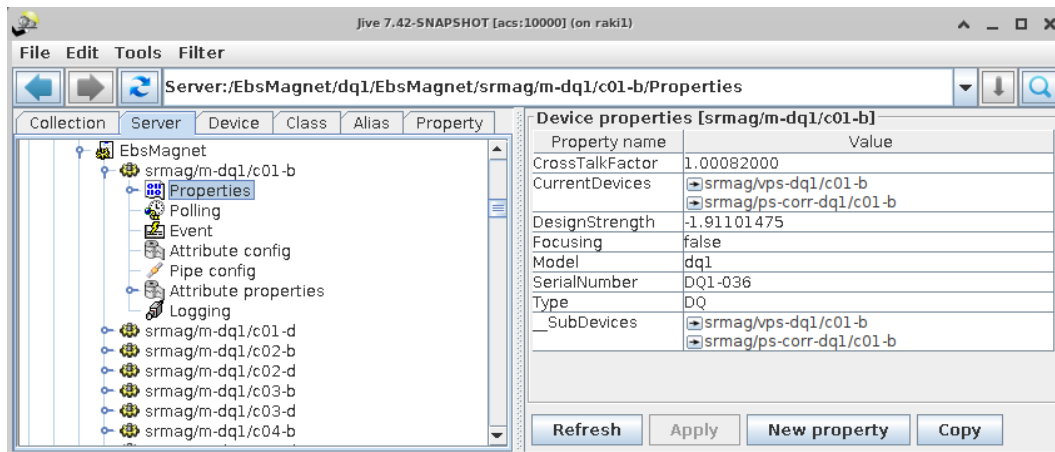
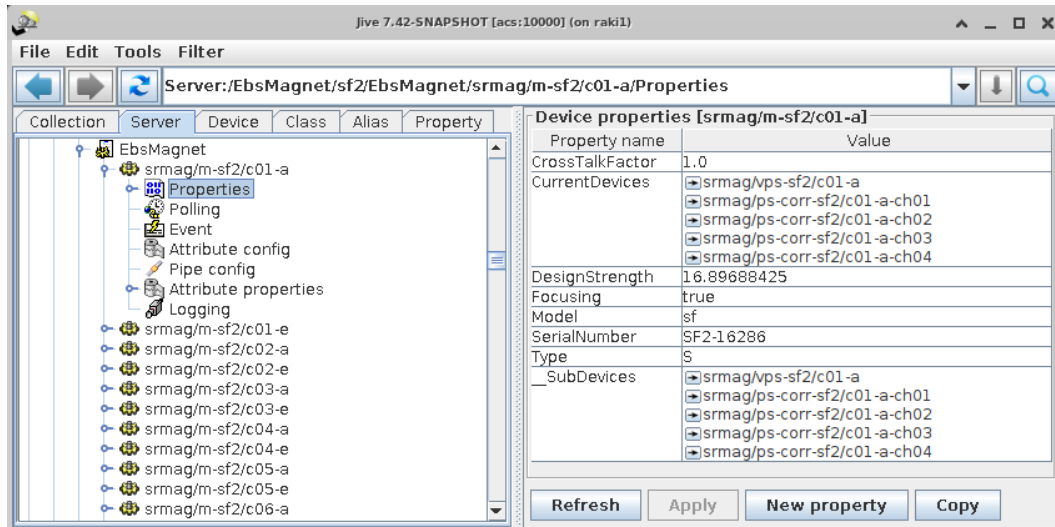
Sextupole

Octupole

SH3

SH5

EBS MAGNETS CALIBRATION CONFIGURATION

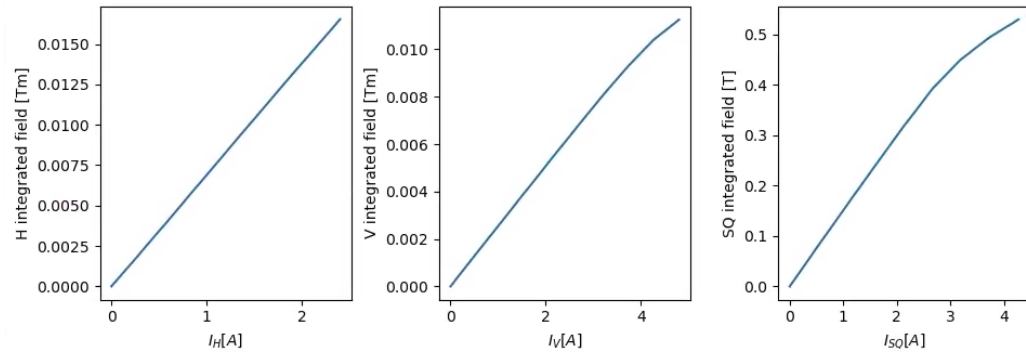


Magnet configuration

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

EBS MAGNETS CALIBRATION EXAMPLE (SH MAGNET)

- Compute coil currents from strength setpoints and reverse transform
- Calibration data (calibration curves and matrix) issue from magnetic measurements

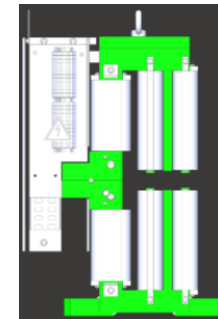
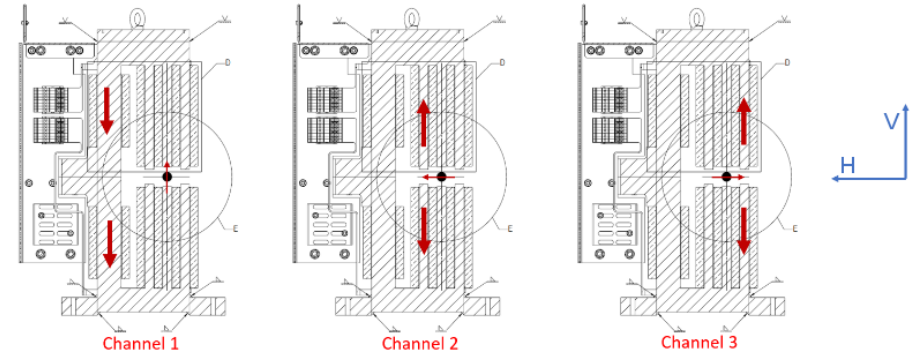


Excitation curves H,V and skew quad

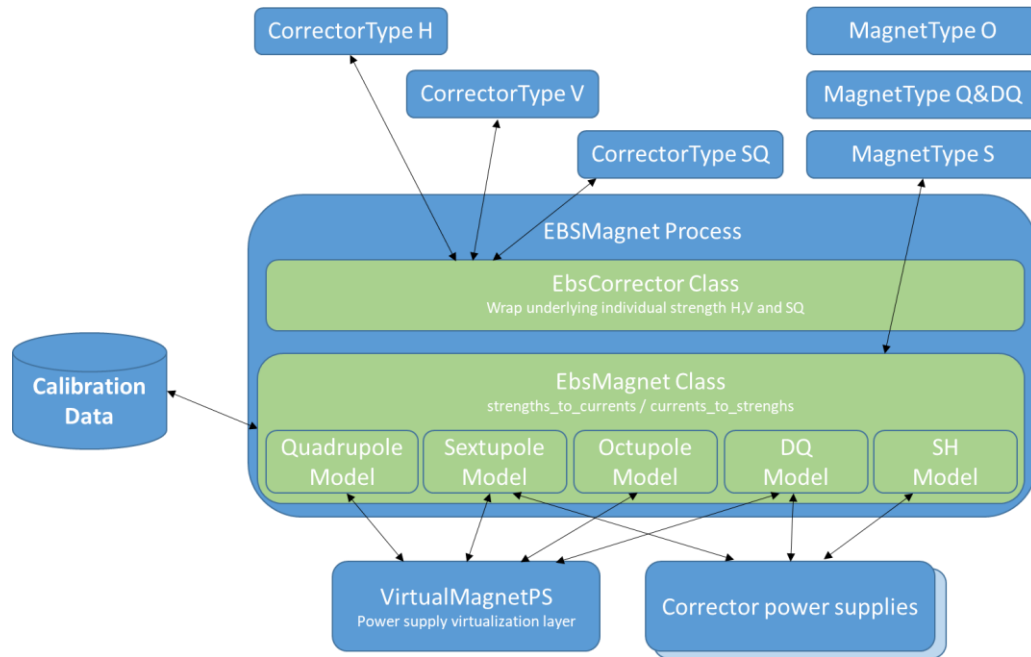
$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1.061 \\ 0 & 0.962 & 1 \end{pmatrix} \quad \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = M^+ \begin{pmatrix} I_H \\ I_V \\ I_{SQ} \end{pmatrix} \quad \begin{pmatrix} I_H \\ I_V \\ I_{SQ} \end{pmatrix} = M \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix}$$

I_H , I_V and I_{SQ} are named “pseudo current”. Each one can be mapped to a single multipole strength.

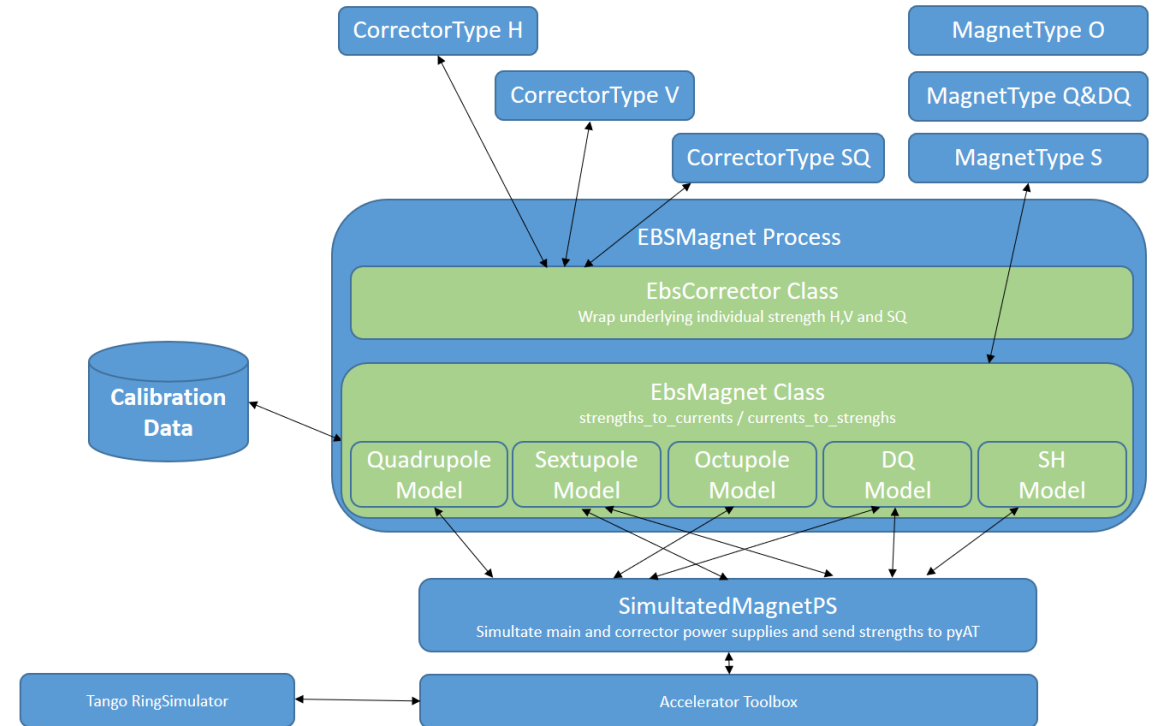
To get strengths, integrated fields are then normalized by the magnet rigidity $B\rho$ [Tm].



EBS MAGNET GLOBAL ARCHITECTURE



Live Machine



Simulated Machine

- EbsMagnet class prevents from conflicts when applying/loading strengths on combined function magnets.
- No `#ifdef SIMULATOR` directive in the code.
- Allow to change directly PS current and simulate cycling on simulators (useful for software development).
- Simulate electrical power overhead.