Diffractometer in Sardana

Teresa Núñez
DESY Photon Science

- Current status
- Details of implementation
- Initialization and use
- Dedicated macros
- Dedicated GUIs
- Next steps

Sardana Workshop
ELI beamlines, 05-06-18

Current status

- Complete diffractometer control from Sardana
- Used at three Petra beamlines (DESY)
- Dedicated GUIs developed in PyQt using Taurus
- Dedicated SEP with status description (not documentation)

Details of the implementation

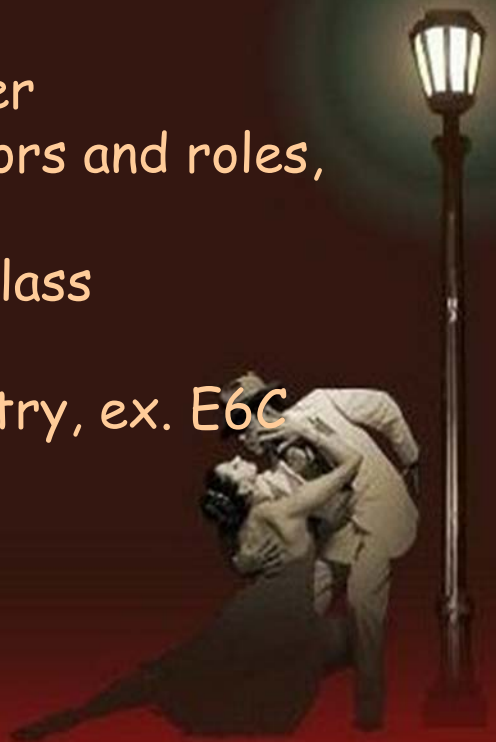
Diffractometer as a controller device of type PseudoMotor

- Use of the hkl library exclusively done in hkl controller code, not in Sardana core
- Base diffractometer class with all calculations
- Derived diffractometer classes:
  - differing in number of motors and pseudomotors
  - not contains explicit use of hkl library
- Dedicated macros

Details of the implementation (ctd.)

Creating diffractometer controller in Pool:

- Controller type: PseudoMotor
- Controller library: HklPseudoMotorController
- Controller class: depends on number of motors and roles, ex. DiffracE6C
- Motors and Roles: depending on controller class
- Properties:
  - DiffractometerType, depending on geometry, ex. E6C



Details of the implementation (ctd.)

Current diffractometer controller classes:

- Diffrac6C (types: "PETRA3 P09 EH2")
  - motor roles: mu, omega, chi, phi, delta, gamma
  - pseudomotor roles: h, k, l
- DiffracE6C (types: "E6C", "SOLEIL SIXS MED2+2"
  - motor roles: mu, omega, chi, phi, gamma, delta
  - pseudomotor roles: h, k, l, psi, q, alpha, qper, qpar

Details of the implementation (ctd.)

Current diffractometer classes (ctd.):

- DiffracE4C (types: "E4CV", "E4CH", "SOLEIL MARS")
  - motor roles: omega, chi, phi, tthh
  - pseudomotor roles: h, k, l, psi, q

- Diffrac4Cp23 (types: "PETRA3 P23 4C")
  - motor roles: omega_t, mu, gamma, delta
  - pseudomotor roles: h, k, l, q, alpha, qper, qpar, tth2, alpha_tth2, incidence, emergence

Details of the implementation (ctd.)

Controller device interface:

- Properties:
  - DiffractometerType
- Attributes:
  - parameters describing current diffractometer status
  - commands performing actions, calculations …
    (not extra commands can be defined in controllers)

- A: a parameter of the current lattice.
- AddCrystal: add a new sample.
- AddReflection: add reflection to current sample.
- AdjustAnglesToReflection: changes the angles associated to the selected reflection.
- AffineCrystal: creates a new sample with '(affine)' attached to the name of the current one and performs the affine. This affine sample is set as the current one.
- Afit: fit value of the a parameter of the current lattice.
- Alpha: alpha parameter of the current lattice.
- AlphaFit: fit value of the alpha parameter of the current lattice.
- B: b parameter of the current lattice.
- Beta: beta parameter of the current lattice.
- BetaFit: fit value of the beta parameter of the current lattice.
- Bfit: fit value of the b parameter of the current lattice.
- C: c parameter of the current lattice.
- Cfit: fit value of the c parameter of the current lattice.
- ComputeTrajectoriesSim: computes the list of trajectories for the current engine and engine mode corresponding to the values of the pseudo axes written in this attribute. The number of arguments has to correspond to the number of pseudo axes for the current engine. The computed trajectories are shown in the TrajectoryList attribute.
- ComputeU: computes UB matrix using the reflections corresponding to the indexes given as arguments.
- Crystal: sample.
- CrystalList: list of samples.
- DeleteCrystal: delete the crystal given in the argument.
- Engine: selected engine. It is taken into account for computing the physical positions corresponding to a movement of a pseudo axis.
- EngineList: list of engines for the diffractometer type corresponding to this controller.
- EngineMode: selected mode for the current.
- EngineModeList: list of the modes corresponding to the current engine.
- HKLModeList: list of the modes corresponding to the hkl engine.
- HKLPseudoMotorList: list of the hkl motor names.
- Gamma: gamma parameter of the current lattice.
- GammaFit: fit value of the gamma parameter of the current lattice.
- LoadReflections: loads reflections for current crystal from ascii file. Reflections will be loaded from the last used file when the Pool is started.
- LatticeReciprocal: reads the values of the reciprocal lattice.
- ModeParametersNames: name of the parameters assotiated to the current engine mode (if any).
- ModeParametersValues: get/set the value of the parameters assotiated to the current engine mode (if any).
- MotorList: names of the physical motors associated to the diffractometer.
- ReflectionAngles: angles (computed and theoretical) between the reflections of the current sample.
- ReflectionList: list of reflections for current sample.
- RemoveReflection: remove reflection with given index.
- SaveReflections: saves reflections from current crystal to ascii file. The value written to this attribute is the path to the file, the name of the file is the name of the sample with the termination .ref. If this file already exists a copy will be created adding to the name the current time in seconds.
- SelectedTrajectory: index of the trajectory you want to take when you perform a movement for a given set of pseudo axes positions.
- TrajectoryList: list of trajectories for the current engine and engine mode corresponding to the pseudo axes values written in the ComputeTrajectoriesSim attribute and the engine and engine mode when the calculation was performed. It gives the possibility of checking the trajectories before performing a movement.
- UBMatrix: reads current UB matrix values.
- Ux: reads/writes current ux value.
- Uy: reads/writes current uy value.
- Uz: reads/writes current uz value.
- Wavelength.

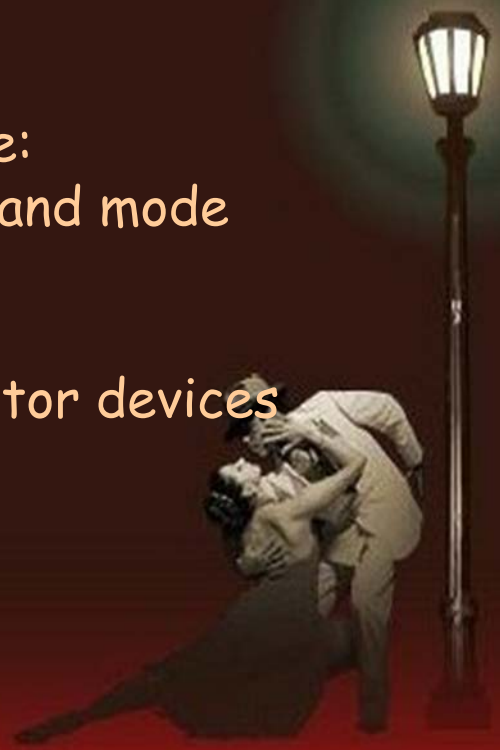
Initialization and use

- Initialization:
  - sample set to last value or 'default crystal'
  - lattice parameters set to default values
  - geometry created according to DiffractometerType
  - engine set to 'hkl'

- Initial settings can be read from config file:
  - set crystal, ub matrix, reflections, engine and mode

- Movements:
  - using the corresponding motor and role motor devices

- Settings and actions:
  - writing to the controller attributes



emacs@haspp09.desy.de (on haspp09)

File Edit Options Buffers Tools Text Help

Save Undo

```
Created at 2018-04-29 00:39

DiffractometerType E6C

Crystal    srru2o6

Wavelength 4.36871767044

A 5.20795754769 B 5.2079575476 C 5.23299798961
Alpha 90.0 Beta 90.0 Gamma 120.0

R0 0 0.0 0.0 1.0 0 1 0.0 28.90655 91.934 282.357825 -1.52587888991e-08 49.347775
R1 1 0.333333333333 0.0 1.0 0 1 0.0 12.39615 102.452375 282.357825 -1.52587888991e-08 53.17195

Engine hkl

Mode constant_phi_vertical

PsiRef not available in current engine mode

AutoEnergyUpdate 0

U00 0.410 U01 -0.946 U02 0.058
U10 0.067 U11 0.112 U12 1.197
U20 -1.330 U21 -1.016 U22 0.078

Ux -86.2748320555 Uy 2.78981647684 Uz 72.8523304456

SaveDirectory /home/p09user/crystals
```

-:--- **srru2o6.txt** All L1 (Text)

For information about GNU Emacs and the GNU system, type C-h C-a.

Configuration file

Dedicated diffractometer macros

Included in standard sardana macros

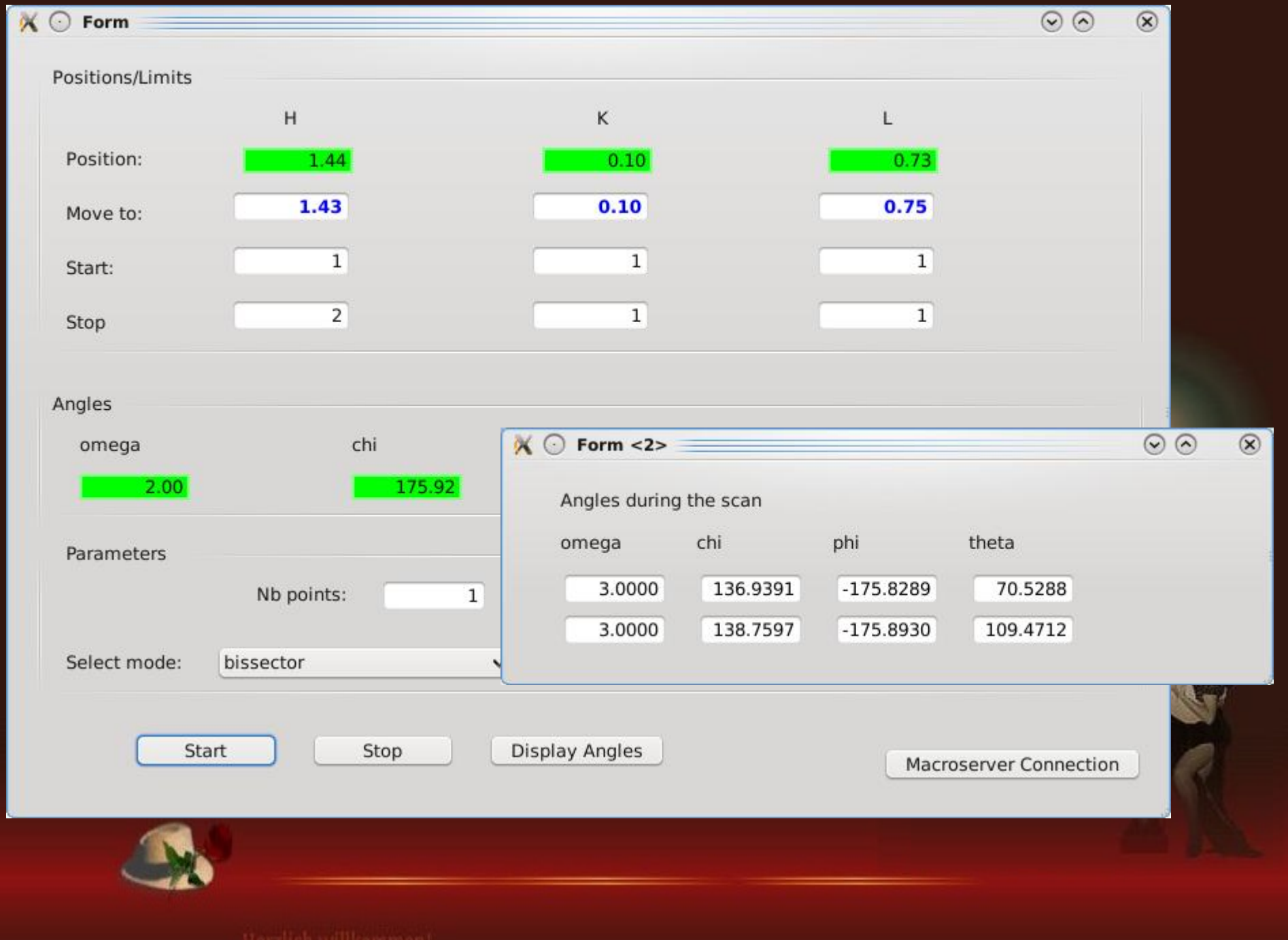
- Control diffractometer and display information
- Follow SPEC syntax as close as possible
- Not tested in all diffractometer types
- Examples:
  - br, ubr: move to hkl values
  - hklscan, hscan, kscan, lscan: scans in hkl space
  - setor0, setor1, setorn: set orientation reflections
  - ca, caa, ci: motor positions from hkl values or v/v
  - computeub, setlat, loadcrystal, …

Dedicated diffractometer GUIs

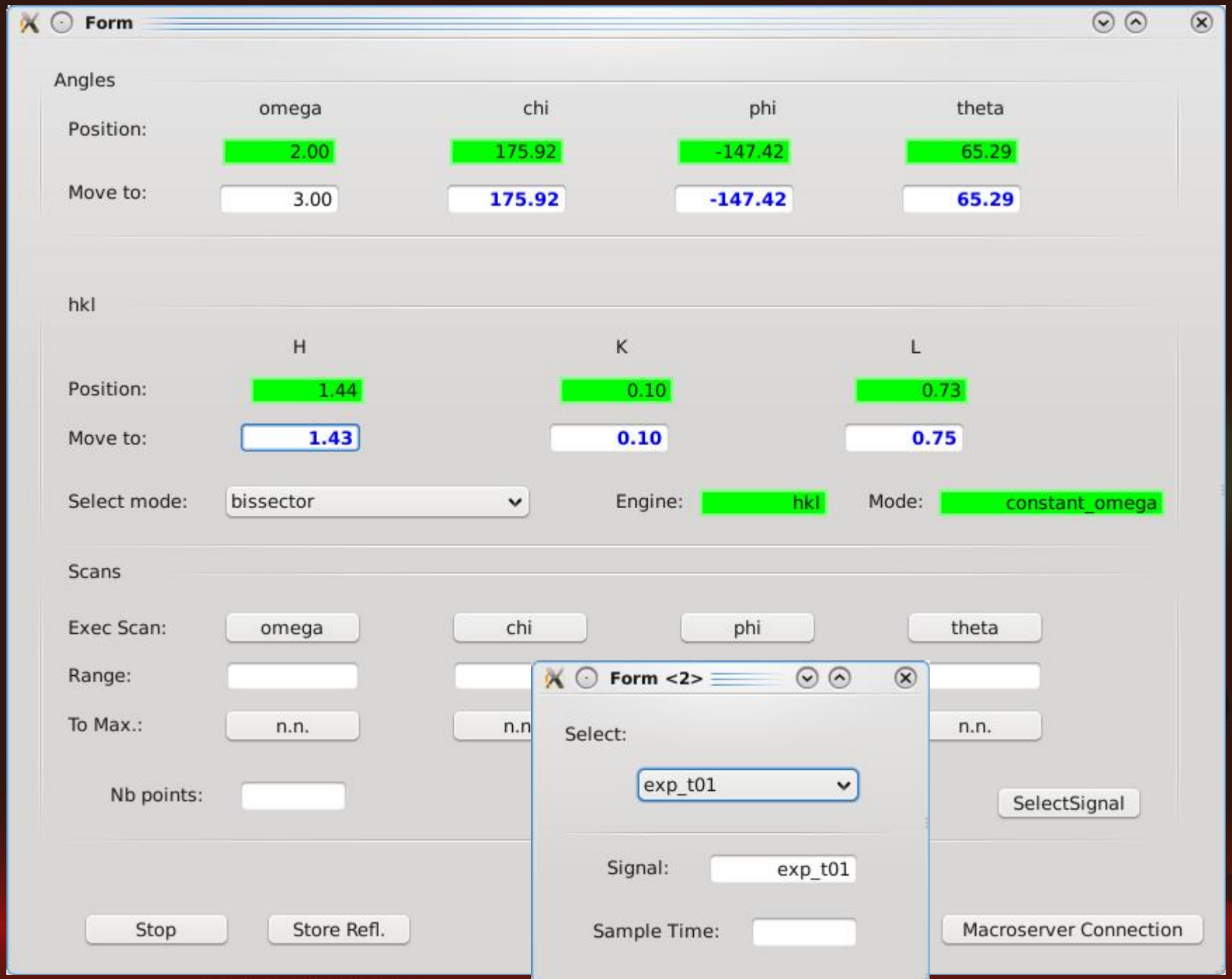
Implemented using PyQt and Taurus

- Three main GUIs:
  - hkl scans
  - diffractometer alignment
  - UB matrix/lattice parameters
- GUIs connects to:
  - diffractometer controller device
  - MacroServer via Door device

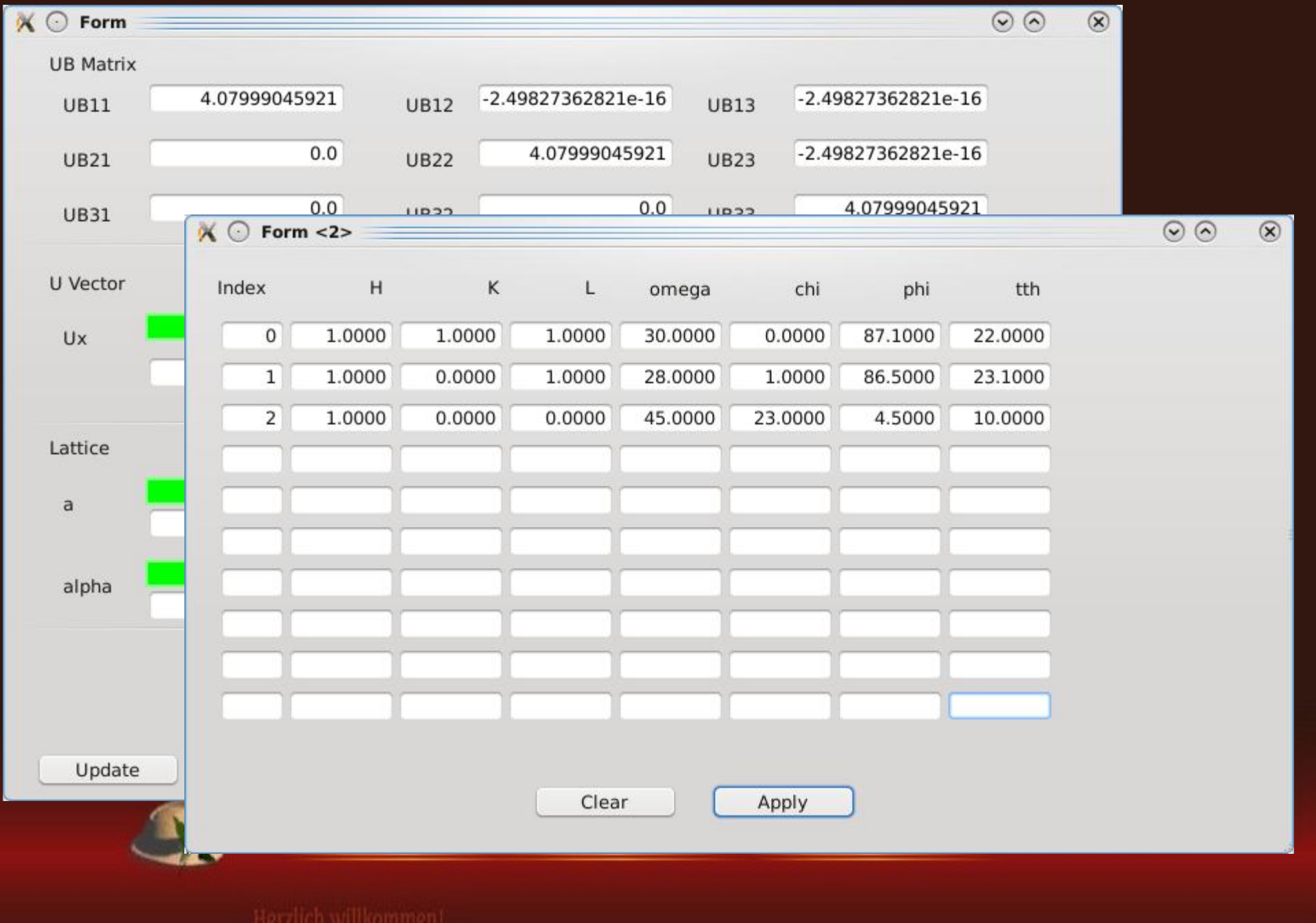

Form
Positions/Limits
H
K
L
Position:
1.44
0.10
0.73
Move to:
1.43
0.10
0.75
Start:
1
1
1
Stop
2
1
1
Angles
omega
chi
phi
theta
2.00
175.92
-147.42
65.29
Parameters
Nb points:
1
Sample time:
1
Select mode:
bissector
Engine:
hkl
Mode:
constant_omega
Start
Stop
Display Angles
Macroserver Connection


Form
Positions/Limits
H
K
L
Position:
1.44
0.10
0.73
Move to:
1.43
0.10
0.75
Start:
1
1
1
Stop
2
1
1
Angles
omega
chi
2.00
175.92
Parameters
Nb points:
1
Select mode:
bissector
Start
Stop
Display Angles
Macroserver Connection
Form <2>
Angles during the scan
omega
chi
phi
theta
3.0000
136.9391
-175.8289
70.5288
3.0000
138.7597
-175.8930
109.4712


Form
Angles
omega
chi
phi
theta
Position:
2.00
175.92
-147.42
65.29
Move to:
3.00
175.92
-147.42
65.29
hkl
H
K
L
Position:
1.44
0.10
0.73
Move to:
1.43
0.10
0.75
Select mode:
bissector
Engine:
hkl
Mode:
constant_omega
Scans
Exec Scan:
omega
chi
phi
theta
Range:
To Max.:
n.n.
n.n.
n.n.
n.n.
Nb points:
SelectSignal
Stop
Store Refl.
Macroserver Connection


Form
Angles
omega
chi
phi
theta
Position:
2.00
175.92
-147.42
65.29
Move to:
3.00
175.92
-147.42
65.29
hkl
H
K
L
Position:
1.44
0.10
0.73
Move to:
1.43
0.10
0.75
Select mode:
bissector
Engine:
hkl
Mode:
constant_omega
Scans
Exec Scan:
omega
chi
phi
theta
Range:
To Max.:
n.n.
n.n
n.n.
Nb points:
SelectSignal
Stop
Store Refl.
Macroserver Connection
Form <2>
Select:
exp_t01
Signal:
exp_t01
Sample Time:


Form
UB Matrix
UB11
4.07999045921
UB12
-2.49827362821e-16
UB13
-2.49827362821e-16
UB21
0.0
UB22
4.07999045921
UB23
-2.49827362821e-16
UB31
0.0
UB32
0.0
UB33
4.07999045921
U Vector
Ux
0.00
0.00
Uy
0.00
0.00
Uz
0.00
0.00
Lattice
a
1.54
1.54
b
1.54
1.54
c
1.54
1.54
alpha
90.00
90.00
beta
90.00
90.00
gamma
90.00
90.00
Update
ComputeU
Reflections List
Edit Reflections


Form
UB Matrix
UB11
4.07999045921
UB12
-2.49827362821e-16
UB13
-2.49827362821e-16
UB21
0.0
UB22
4.07999045921
UB23
-2.49827362821e-16
UB31
0.0
0.0
4.07999045921
U Vector
Ux
Lattice
a
alpha
Update
Form <2>
Index
H
K
L
omega
chi
phi
tth
0 1.0000 1.0000 1.0000 30.0000 0.0000 87.1000 22.0000
1 1.0000 0.0000 1.0000 28.0000 1.0000 86.5000 23.1000
2 1.0000 0.0000 0.0000 45.0000 23.0000 4.5000 10.0000
Clear
Apply

Next steps

- Test of current implementation:
  - performance
  - completeness
- Discuss about other possible implementations