pyScan

M. Aiba

Jan. 2016

Yet another scan tool

- Based on well established C.A. library
  - i.e. CAFE
  - All the complicate operations (monitor, set and match, etc.) are performed by CAFE
  - Thus, the scan tool code itself could be compact
- Implementation as a python class
  - Easy to maintain and extend
  - Can be wrapped (cython) to be C/C+ shared object
  - Can be extended to be a scan server
- Developer = Heavy user (namely myself)
  - Good for commissioning

Demand

- Generic and flexible scan tool
  - Easy integration into high level applications
  - Any kind of scan
    - Single-knob scan (SKS), multi-knob scan (MKS) and series scan (SS)
    - And combination of them
  - Pre action, post action, monitor, etc.
- Need a decent scan tool now for application development
  - The code is ready (found in PSI git together with User’s manual) and briefly tested

Single/Multi knob scan

- Schematic flow chart

```
PreAction
for Knob(s) in ScanValues:
        In-loopPreAction
        setAndMatch(Knob(s), Readback(s))
        for i_meas in NumberOfMeasurements:
                Measure observable(s)
                sleep(Waiting)
        In-loopPostAction
PostAction
```

Interrupt the measurement
If any problem

← Monitoring(Monitor)

Color code:
CAFE method
pyScan user defined

ScanValues can be given in different format

- List of absolute values,
- Start/End values + Number of measurements
- Additive to the present values

Single/Multi knob scan example: Quad scan

- Schematic flow chart

```
GunOnDelay->Q cycling ->GunOn
for Q:I-SET in ScanValues:
        In-loopPreAction = None
        setAndMatch(Q:I-SET, Q:I-READ)
        for i_meas in (0,5):
                Measure beam size
                sleep(1 sec)
        In-loopPostAction = None
GunOnDelay->Qcycling->Q:I-SET restore->GunOn
```

Monitoring(BPMvalid)

Pause the measurement if no beam

Series knob scan

- Schematic flow chart

```
PreAction=None
for Knob in Knobs:
        for Knob in ScanValues:
                In-loopPreAction
                setAndMatch(Knob, Readback)
                for i_meas in NumberOfMeasurements:
                        Measure observable(s)
                        sleep(Waiting)
                In-loopPostAction
PostAction
```

Interrupt the measurement
If any problem

← Monitoring(Monitor)

Series knob scan example:
Orbit response

- Schematic flow chart

```
PreAction=None
for Knob in Knobs(Corrector:I-SET):          Monitoring(BPMvalid)
        for Knob in (0,+$\Delta$I):
                In-loopPreAction=None
                setAndMatch(Knob, Readback)
                for i_meas in 5:
                        Beam position at many BPMs
                        sleep(1 sec)
                In-loopPostAction=None
PostAction=None
```

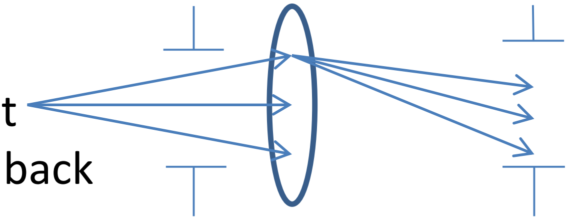
Combination

- SKS over MKS, for example, Multi Quad scans over various Gun solenoid setting

```
for Knob in ScanValues:
          setAndMatch(Knob, Readback)
         for Knobs in ScanValues:
                   setAndMatch(Knobs, Readbacks)
                  for i_meas in NumberOfMeasurements:
                           Measure observable(s)
                           sleep(Waiting)
```

- Pre and post actions are omitted in the above flow-chart but they can be configured also in a combined scan at any level
- Almost all kinds of scan can be build by combining SKS, MKS and SS

Combination example: BBA to find Q centre



Knob:
Quad current

Orbit
feedback

Knob:
BPM offset

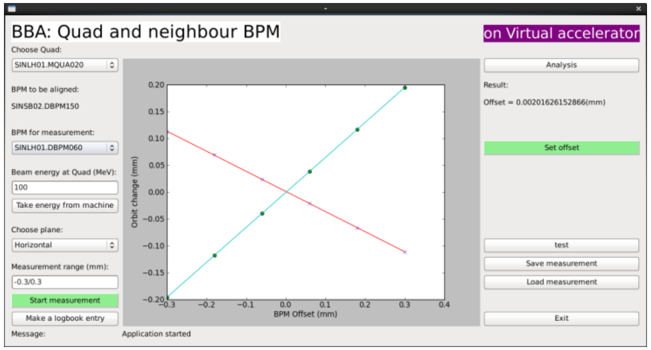
Observable:
Beam position

Input definition (only important fields are shown)

```
indict0={}
indict0['Knob']=[self.EC.prefix+BPM+':OFFS-X'] # Knob: BPM offset
indict0['ScanRange']=[-0.3, 0.3]
indict0['Nstep']=6

indict={}
indict['Knob']=[self.EC.prefix+Quad+':I-SET'] # Knob: Quad current
indict['Additive']=1 # Knob is varied with respect to the present value.
indict['ScanValues']=[-dI,0,dI] # dI is computed previously using on-line model
indict['NumberOfMeasurements']=5
# Wait until the orbit feedback (another application) brings the beam to the shifted BPM centre
indict['PreAction']=[[self.EC.prefix+'DUMMY:NUMBER',self.EC.prefix+BPM+':X1',0.0,0.005,60]]
# Monitor if BPMs and Q power supply are OK
indict['Monitor']=[self.EC.prefix+BPM+':X1-VALID',self.EC.prefix+BPMobs+':X1-VALID']
indict['Monitor'].append(self.EC.prefix+Quad+':ONOFF')

self.Measurement.indict=[indict0,indict]
```



BBA: Quad and neighbour BPM

on Virtual accelerator

Choose Quad: SINLH01.MQUA020

BPM to be aligned:
SINSB02.DBPM150

BPM for measurement: SINLH01.DBPM060

Beam energy at Quad (MeV): 100

Choose plane: Horizontal

Measurement range (mm): -0.3/0.3

Result:

Offset = 0.00201626152866(mm)

Message: Application started

Non Epics action

- Pre/post-actions can be defined by a user defined function (so far, only in python application)
- Example: close and open a feedback loop by talking to the feedback server during the scan