PINK beamline

PShell custom functions

Table of Contents

scan.

continuous

scan.continuous(detector, det_exposure, sample_exposure, X0, X1, dX, Y0, Y1, passes, sample, linedelay)

detector:	Detector selection
det_exposure:	Detector's exposure time
sample_exposure:	Sample's exposure time per pass
X0:	start location on X axis
X1:	End location on X axis
dX:	Distance between vertical lines
Y0:	Start location on Y axis
Y1:	End location on Y axis
passes:	Number of passes
sample:	Name of the sample
Linedelay:	Time delay between vertical lines

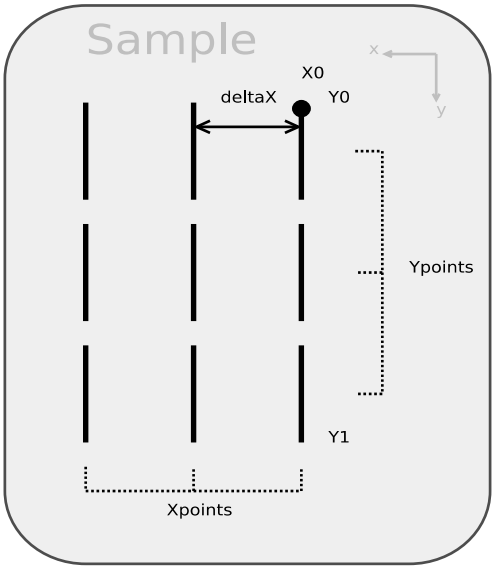

Fig. 1 - Continuous scan movement diagram (beam perspective looking downstream).

Performs a scan on a moving sample. The goal is to maximize detector's exposure time and minimize sample degradation. The speed of the sample motion is calculated using the defined "sample_exposure" time and a vertical beam size of 50 micrometers.

---

filter_scan

scan.filter_scan(filters, start, end, step, exposure)

Filters:	Filter selection
start:	Start position on filter
end:	End position on filter
Step:	Distance between point (float) or number of steps in interval (int)
Exposure:	Electrometer's reading time

Performs a filter position scan against Izero. The beam intensity readout comes from the sum of 4 quadrant diodes used for beam positioning monitoring.

---

gap_scan

scan.gap_scan(start, end, step, exposure, fit)

start:	starting undulator gap
end:	end undulator gap
step:	Step (float) or number of points to acquire (int)
exposure:	Electrometer acquisition time
fit:	If True, fits data to a gaussian curve with a exponential background

Performs an Undulator gap scan. The beam intensity readout comes from the sum of 4 quadrant diodes used for beam positioning monitoring.

---

line

scan.line(detector, exposure, Y0, dY, Ypoints, passes, sample)

Detector:	Detector selection
exposure:	detector's exposure time
Y0:	Start location on Y axis
dY:	distance between points
Ypoints:	number of points
passes:	number of passes
sample:	name of the sample

Performs multiple exposures on a sample along a vertical line.

---

spot

scan.spot(detector,exposure, images, sample)

detector:	detector selection
exposure:	detector's exposure time
images:	number of exposures
sample:	name of the sample

Performs multiple exposures on a single location on sample.

---

zigzag

scan.zigzag(detector, exposure, X0, dX, Xpoints, Y0, dY, Ypoints, passes, sample, linedelay)

detector:	detector selection
exposure:	detector’s exposure time
X0:	Start location on X axis
dX:	distance between columns in the X axis
Xpoints:	number of columns in the X axis
Y0:	Start location on Y axis
dY:	distance between rows in the Y axis
Ypoints:	number of rows in the Y axis
passes:	number of passes
sample:	name of the sample
linedelay:	time delay between each line exposure

Performs multiple exposures with a grid pattern on a sample along X and Y lines.

---

sample_scan

scan.sample_scan(axis, detector, start, end, step, exposure)

axis:	axis selection: vertical or horizontal
detector:	detector selection
start:	Start position on sample
end:	End position on sample
step:	Distance between point (float) or number of steps in interval (int)
exposure:	detector’s exposure time

Performs a sample position scan against the detector’s peak-count-per-second value. Useful to find sample edge positions.

---

pink.

bl_snapshot_print

pink.bl_snapshot_print()

Prints on screen the position, temperature and pressure of relevant components for the Pink beamline.

---

bl_snapshot_save

pink.bl_snapshot_save()

Prints on screen the position, temperature and pressure of relevant components for the Pink beamline.

---

filter<n>

pink.filter1(pos), filter2(pos), filter3(pos)

pos:	position of the filter

Moves filter to the requested position. All filters are located inside the diagnostic chamber.

---

gap

pink.gap(value)

value:	no value or undulator gap (mm)

Prints the gap position if no value is given; Or sets the gap on the undulator.

---

ml_<energy>ev

pink.ml_2300ev(), ml_3000ev(), ml_4000ev(), ml_5000ev(), ml_6300ev(), ml_6800ev(), ml_7300ev(), ml_8000ev(), ml_9500ev()

Sets up the correct stripe on the multilayer mirror based on the desired beam energy.

---

rename_sample

pink.rename_sample(path, newstring)

path:	variable path (can be retrieved by right clicking on the variable using the data explorer tab)
newstring:	new string

Rename the sample on a acquire data file (.h5 data file)

---

shutter_hard_CLOSE

shutter_hard_OPEN

pink.shutter_hard_OPEN(), shutter_hard_CLOSE()

Open/close hard beam shutter.

---

valveCLOSE

valveOPEN

pink.valveOPEN(vnum), valveCLOSE(vnum)

vnum:	Valve number according to diagram on control panel

Open/close the vacuum valve.

AU1

pink.AU1(top, bottom, wall, ring):

top,bottom,wall,ring:	Aperture defined in mm.

Set the specified aperture on AU1.

---

eiger_set_background

pink.eiger_set_background():

:	no argument

Execute sequence to set the correct background information on eiger detector after reboot of detector's IOC.

---

plot

pink.plot(filepath)

filepath:	Copied file location from the data tab

Plots relevant data from h5 file from the data tab. To copy the filepath, click first on the desired file on the left window, then right-click the name of the file on the middle window. Click on “Copy link to clipboard”, then past it as the filepath.

---

scatter

pink.scatter(pos)

pos:	position of the scatter filter

Moves scatter filter to the requested position. The scatter filter is located inside the diagnostic chamber.

---

spec_atm_go2_stdby_position

pink.spec_atm_go2_stdby_position()

:	no argument

Moves ATM spectrometer to the standby position for detector replacement.

---

bpm.

BPM<n>_Horizontal_Profile

BPM<n>_Vertical_Profile

bpm.BPM<n>_Vertical_Profile()

Provides information about the beam dimensions on the selected BPM (Beam Position Monitor)

---

blade.

diagnostic_chamber_blade_scan

blade.Diagnostic_Chamber_Blade_Scan(start, end, step, exposure)

start:	Start position on filter's blade
end:	End position on filter's blade
step:	Distance between points (float) or number of steps in interval (int)
exposure:	detector's exposure time

Perform a blade scan with blade inside the diagnostic chamber. After scan, the script attempts to perform a gaussian fit to derivative of the acquired data.

Sample_Env_Blade_Scan

blade.Sample_Env_Blade_Scan(start, end, step, exposure)

start:	Start position on sample holder's blade
end:	End position on sample holder's blade
step:	Distance between points (float) or number of steps in interval (int)
exposure:	detector's exposure time

Perform a blade scan with blade on the sample holder. After scan, the script attempts to perform a gaussian fit to derivative of the acquired data.

Slit_Scan

blade.Slit_Scan(slit, start, end, step, exposure)

slit:	slit selection
start:	Start position on sample holder's blade
end:	End position on sample holder's blade
step:	Distance between points (float) or number of steps in interval (int)
exposure:	detector's exposure time

Perform a blade scan with the selected slit. After scan, the script attempts to perform a gaussian fit to derivative of the acquired data.