Science & Technology Facilities Council
ISIS

Thomas Löhnert, Aaron James Long (ISIS, STFC)

John Holt (Tessella)

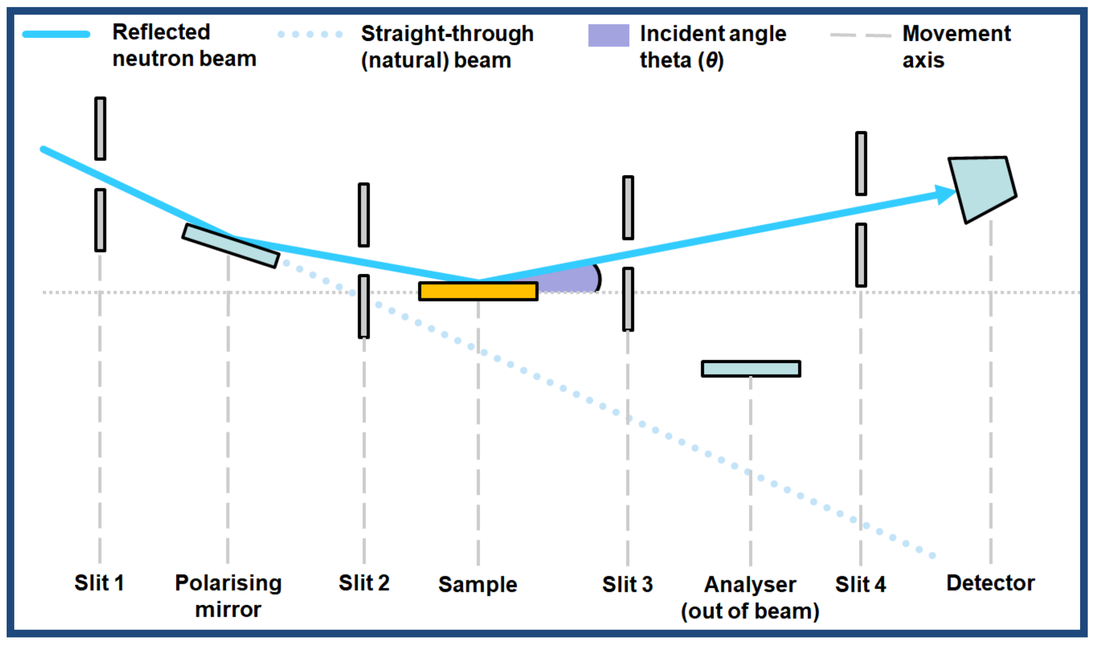
Generalising the High-Level Geometry System for Reflectometry Instruments at ISIS

Introduction

Reflectometers have complex motion requirements as equipment needs to maintain its position and angle to a neutron beam with a changing beam path.

As part of the ongoing migration to our new EPICS-based control system, we have re-implemented the beamline-level geometry layer used to hide this complexity from the user.

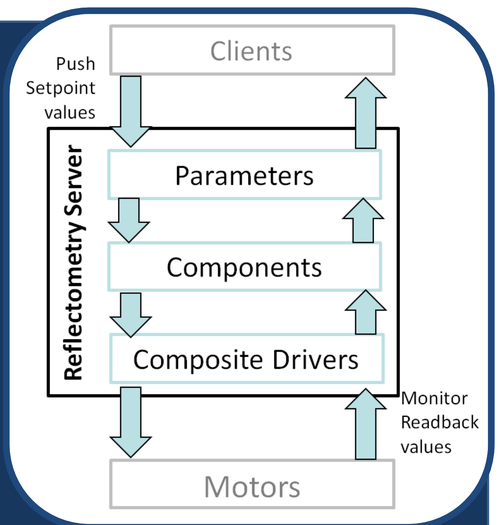
We have used this opportunity to redesign the architecture of the system with the goal of being easily reconfigurable and extensible for future beamline developments.



Schematic of a typical reflectometry beamline

Server Architecture

- IBEX is client-server
- Reflectometry driver is a python service running locally on beamline server
- Uses PCASpy to interface to channel access – acts like any EPICS IOC to the outside



Server architecture overview

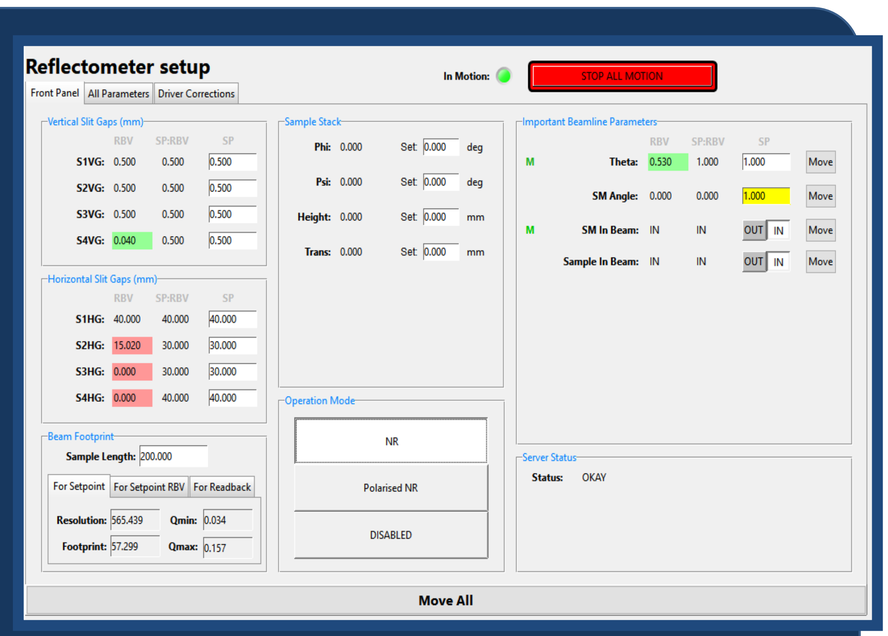
Layers

- **Parameters:** relevant beamline parameters, relative to the current beam path
- **Components:** Building blocks of the beam path model. Responsible for conversion between beamline parameters and (absolute) motor values
- **Drivers:** Push and read values to/from low-level motor PVs, handle (simple) move synchronization, apply engineering corrections

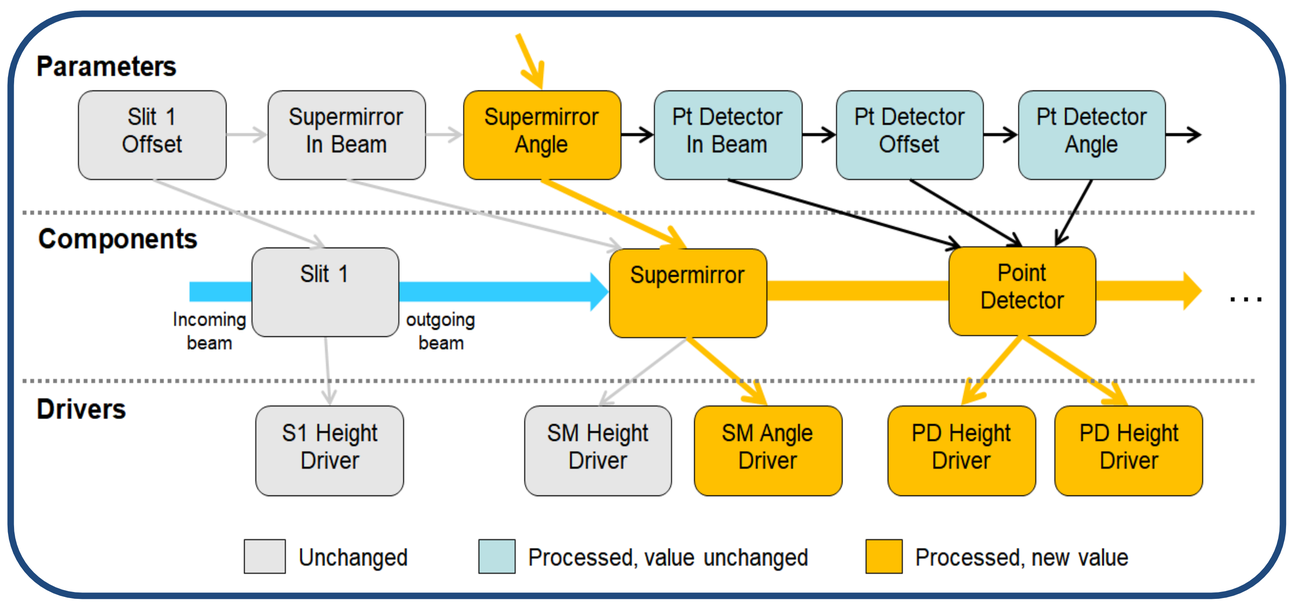
Interaction between items is coordinated by a top-level **Beamline** object. Parameters can be unlinked from beam model depending on active **Mode**.

Interaction

- Front Panel is implemented as CSS OPI
- Scripting: values are exposed like any other device variable in IBEX



The reflectometry front panel



Example: Processing a move of the supermirror angle

Future Work: Continuous Scanning

Experiment data is stitched from datasets at various θ (incident angle Theta). More datasets means better data, but the overhead of starting/stopping data collection currently limits us to few data points

Ideally, we would like to take data while performing a continuous sweep over θ instead, producing many datasets annotated with real-time positions. θ is a compound axis → this requires truly synchronous motion, i.e. axes know and correct where they are in relation to each other in real time.

This is currently limited by our motion controllers, but we are planning to roll-out a more sophisticated Beckhoff-based motion control system at ISIS. These controllers can run synchronization logic on an embedded real-time OS, instead of in Composite Driver layer of Reflectometry Server.