Control System of the ForMAX Beamline at the MAX IV Synchrotron

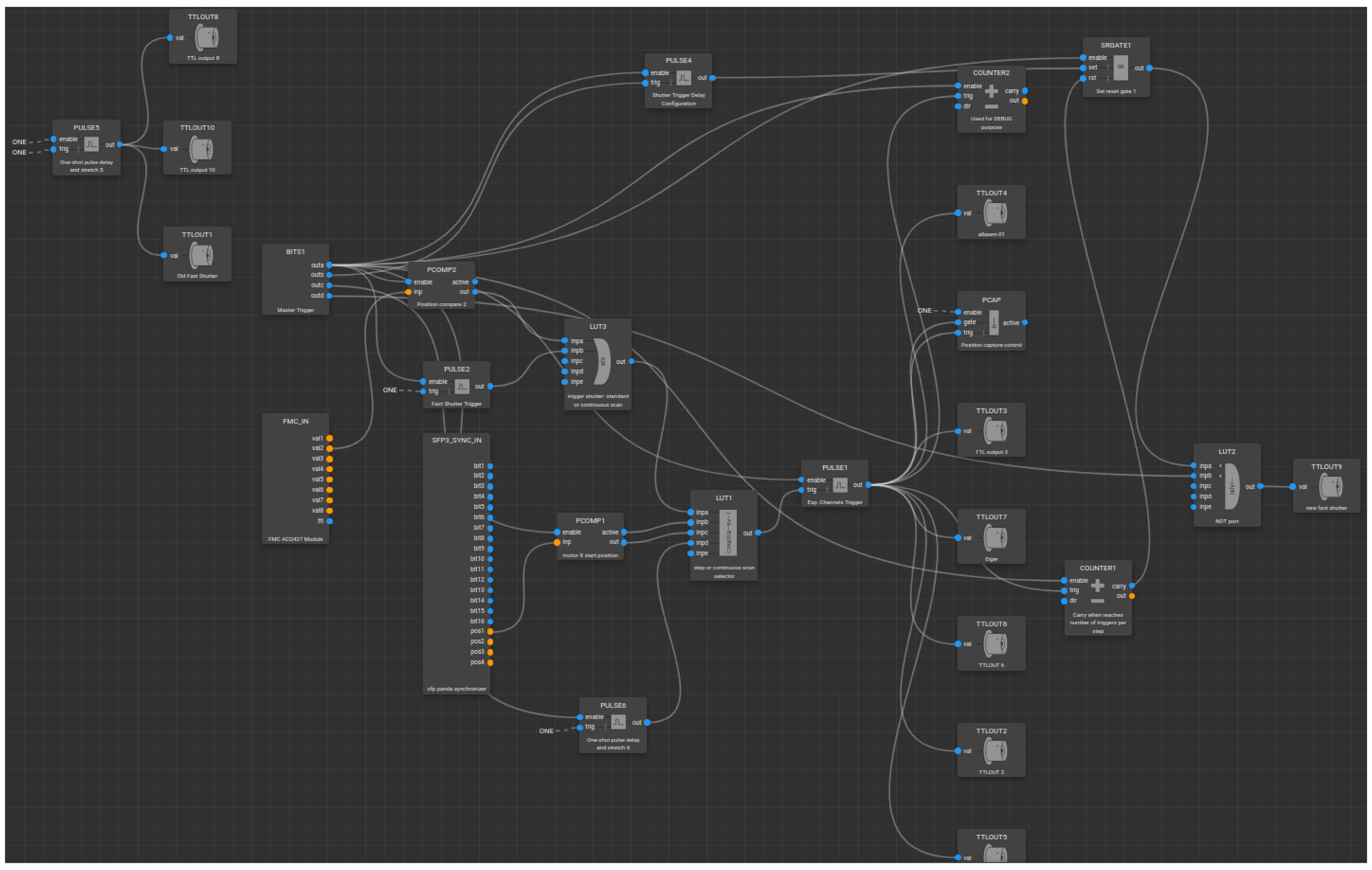
W. T. Kitka∗, S2Innovation, Kraków, Poland
V. Da Silva, V. Haghighat, Y. Li, J. Lidon-Simon, M. Lindberg, S. Malki,
K. Nygard, E. Rosendahl, MAX IV, Lund, Sweden

Conclusion

The amalgamation of TANGO Controls, Sardana, PandABox, and IcePAPs ensures the comprehensive and robust nature of the control system, ultimately optimizing the ForMAX beamline's performance for a diverse range of experiments. After making the scans improvements, scans have become considerably quicker. The next task involves enhancing the way the two motors work together in meshct scanning to reduce the time spent at the end of each scan line. In a meshct scan, the slower (step) motor begins moving only after the faster (continuous) motor reaches the end position. This setup slows down the scan between two lines more than anticipated.

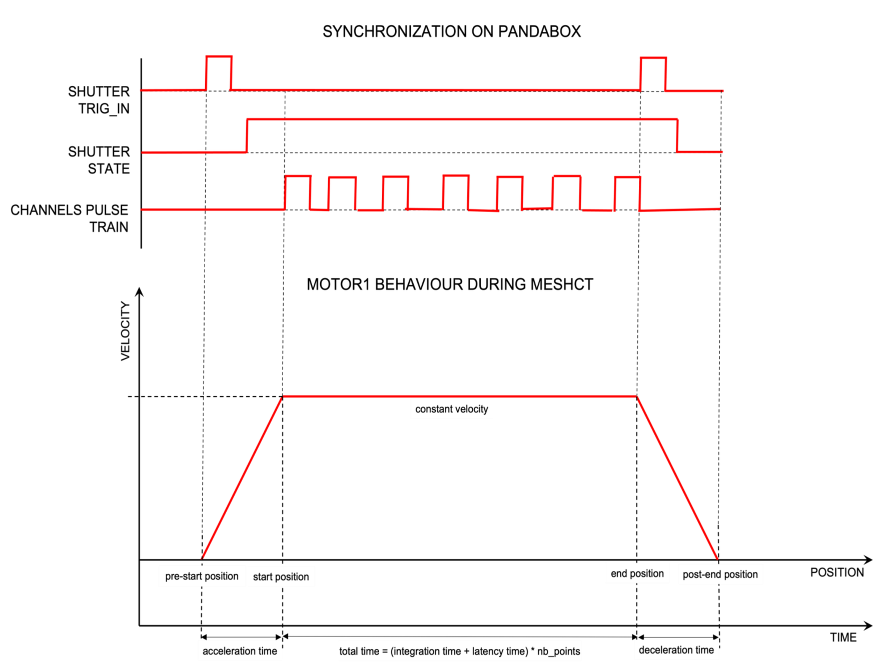
ForMAX introduction

ForMAX, located at achromat 9 of the MAX IV 3 GeV ring, is a hard X-ray beamline focused on versatile structural characterization. With an emphasis on efficiency, the beamline seamlessly switches between full-field X-ray microtomography, small- and wide-angle X-ray scattering (SWAXS), and scanning SWAXS imaging. The microtomography provides non-destructive 3D mapping in the microscale range (1 μm to 5 mm), enabling studies such as porosity characterization in forest-based materials with a temporal resolution of 1 s. SWAXS explores nanoscale structures (1 to 500 nm) for understanding biobased nanomaterials with a temporal resolution in the ms regime. Scanning SWAXS imaging generates 2D or 3D images of fibril orientation within samples, but its temporal resolution is limited due to the potential need for ≈ 106 individual SWAXS images for 3D reconstruction.



ForMAX control system

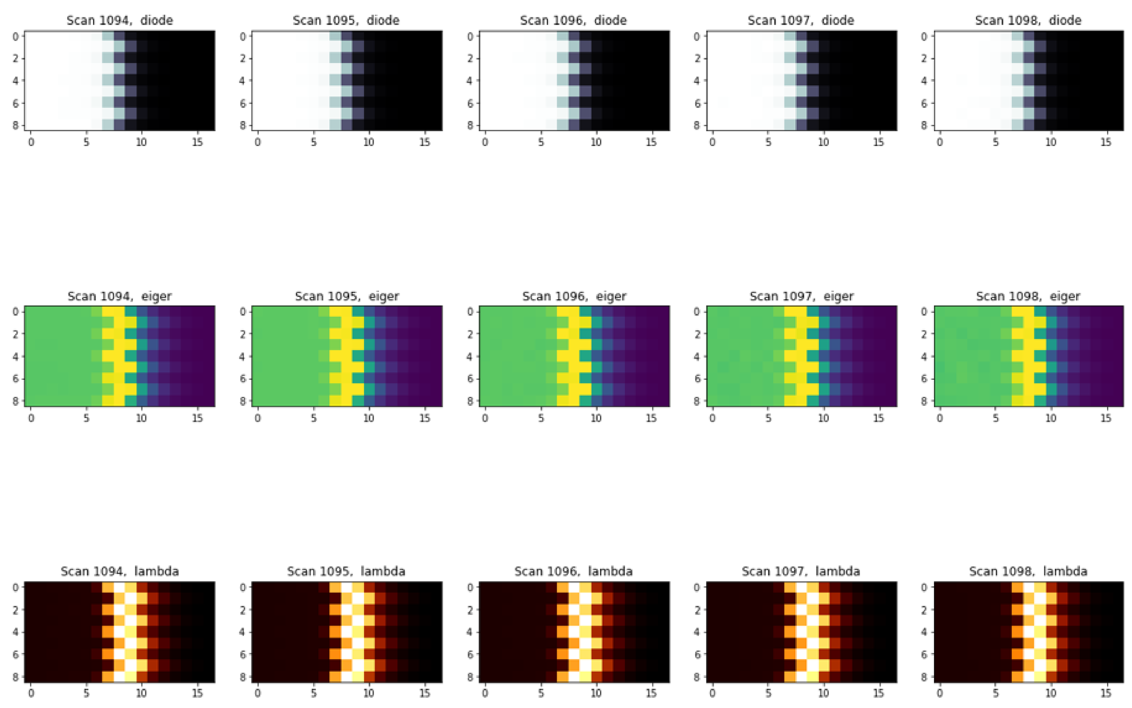
The control system governing the ForMAX beamline is a sophisticated infrastructure designed for precise and efficient experimental control. Making use of the TANGO Controls framework, it establishes a distributed architecture for seamless communication between diverse hardware components and software modules. Sardana, a Python-based software suite, assumes a pivotal role in orchestrating the control of ForMAX's instrumentation, ensuring streamlined data acquisition and processing. The integration of PandABox augments system versatility, enabling adaptable control of diverse devices. IcePAPs, utilized as motor controllers, further amplify the system's capabilities, ensuring high-precision positioning and effective movement control.



* Contact:
wojciech.kitka@s2innovation.com

Scans

At ForMAX, both step scans and continuous scans are supported. In a step scan, the motors move to specific points, stop at each point, and then data is collected for one or more channels. Only after the data collection is complete at one point do the motors move to the next point. In a continuous scan, the motors don't stop during data collection. Instead, data is gathered while the motors are still in motion. Continuous scans are usually used when one want to optimize data acquisition time by avoiding the delays associated with motor acceleration and deceleration between data points. In continuous scan setup at ForMAX, there is a scanning system that involves two motors. For each step of the second motor, the first motor scans continuously, effectively tracing out a grid pattern.



Optimizing the scans

While performing experiments, two problems with continuous scans were identified:

- the drifting issue,
- a overhead per line related to arming of detectors on each line of a scan.

The problem of drifting occurred because the PandABox sent an extra pulse for each line. This happened because of how Sardana calculates the velocity and the total time for the meshct scan.

Sardana formulas:

$$velocity = \frac{(pos\_final\ -\ pos\_initial)}{nb\_points\ *\ total\_time}$$

$$total\_time = nb\_points * (int\_time + latency\_time)$$

Proper formulas:

$$velocity = \frac{(pos\_final\ -\ pos\_initial)}{(nb\_points - 1)\ *\ total\_time}$$

$$total_{time} = (nb\_points * int\_time) + (nb\_points\ - 1) * latency\_time$$

The issue was resolved by using a meshct scan with overwritten formulas.

Normally, in a meshct scan, Sardana arms the detectors for every step of the slower motor. This adds a lot of extra work for each line. To fix this problem, additional changes to the meshct class were made so that the detectors are armed only once for the entire scan, reducing unnecessary overhead.

References

[1] MAX IV Laboratory home page, https://www.maxiv.lu.se/

[2] TANGO Controls, https://www.tango-controls.org/

[3] Sardana, https://sardana-controls.org/index.html

[4] PandABox, https://www.ohwr.org/project/pandabox/wikis/home

[5] H. Enquist, A. Bartalesi, B. Bertrand, J. Forsberg, A. Freitas, V. Hardion, M. Lindberg, and C. Takahashi, "Continous scans with position based hardware triggers", in Proc. 18th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'21), Shanghai, China, Oct. 2021, pp. 1069-1073. doi:10.18429/JACoW-ICALEPCS2021-FRBR04