12th Int. Conf. Mech. Eng. Design Synchrotron Radiat. Equip. Instrum. MEDSI2023, Beijing, China JACoW Publishing
ISBN: 978-3-95450-250-9 ISSN: 2673-5520 doi:10.18429/JACoW-MEDSI2023-THOAM02

SmarGon MCS2: AN ENHANCED MULTI-AXIS GONIOMETER WITH A NEW CONTROL SYSTEM

W. Glettig, D. Buntschu, E. Panepucci, M. Wang, Paul Scherrer Institute, Villigen PSI, Switzerland
A. Omelcenko, SmarAct GmbH, Oldenburg, Germany

Abstract

As an improvement on the commercially available SmarGon multi-axis goniometer (SmarAct GmbH), the Macromolecular Crystallography (MX) Group at the Paul Scherrer Institute (PSI) has been pursuing a further development of the system. In addition to suggesting mechanical improvements to SmarAct for improved ruggedness and reliability, PSI has developed a brand-new and flexible control system for better customization, reliability, and control. Calibration routines were implemented to reduce systematic errors, and the system has been tailored for practical beamline usage. SmarGon is a six degree-of-freedom positioning device, allowing positioning of a sample and orientation around any given point, with < 5 µm sphere of confusion diameter. It was purpose-built for protein-crystallography experiments but, as will be presented here, was also re-purposed for other applications. Two devices have been in continuous 24/7 use for two years at the MX Beamlines PXI & PXII at SLS.

INTRODUCTION

Initially developed based upon PSI's 6-axis-goniometer for protein crystallography PRIGo [1], SmarAct GmbH's SmarGon is a further developed and commercially available positioning device [2] allowing 4 mm translational XYZ motion and three angles of rotation ω: [−∞, +∞], χ: [0, 90 °], φ: [−∞, +∞] around any arbitrary point in space (Fig. 1). Positioning resolution is 1 nm and spheres of confusions are achievable of below 1 µm for ω, and well below 7 µm for χ & φ. [2]

Figure 1: SmarGon with a representation of the rotations ω: [−∞,+∞], χ: [0, 90 °], φ: [−∞,+∞].

SmarGon is used to position and orientate a sample with respect to the X-ray beam, and is one of the central components of a macromolecular crystallography (MX) beamline setup [3]. It was purpose built for MX experiments, but has also been used in other applications.

INITIAL RELIABILITY ISSUES

While offering advantages over the PRIGo Goniometer in terms of compactness and build simplicity, a big challenge in daily operation was to preserve SmarGon's reliability over extended periods of time. MX beamlines are often set up for high throughput and can process hundreds of samples a day. They are often controlled remotely, and now increasingly in unattended automatic operation [4]. In such cases, downtime must be avoided, and all systems must be as remotely monitorable and controllable as possible, without any need for physical human intervention.

The initial version of SmarGon posed problems: Due to its fine mechanical structure it was prone to mechanical damage caused by rough human manipulation during manual sample mounting, or unforeseen collisions during robotic sample mounting [5]. User-prepared samples can sometimes present unpredictable defects, leading to misgripped samples and ice-related slipping and sticking issues.

Another limitation was the inability to customise the control system for different modes of operation, like permitting flexible recovery in case of problems during remote access. Or in tweaking the calibration routine. Or modifying interfaces, to extend remote system diagnostics or to collect usage statistics.

Improvements both on the mechanical side as well as on the control side were strongly requested.

MECHANICAL IMPROVEMENTS

Over the initial design of the SmarAct goniometer, several simple mechanical improvements were implemented by SmarAct GmbH, primarily to increase robustness and reliability of the mechanism, and by improving processes for more repeatable assembly tolerances. These improvements are now standard in the latest SmarGon devices. Revisions to the pivot and ball joints were made and hard stops were added to prevent dislodging. Both during manual interaction and during robotic sample mounting, misaligned or mis-gripped samples can cause large forces on the goniometer, which can lead to plastic deformation of the mechanical structure. The design of a critical area of high stress, and the choices of materials were revised. Holding forces of the SmarAct piezo stick-slip positioners were taken into account, so that during a physical interaction the compliance in the structure would be in the sliders, and not as a plastic deformation of the structure.

ISBN: 978-3-95450-250-9 ISSN: 2673-5520 doi:10.18429/JACoW-MEDSI2023-THOAM02

SmarAct MCS2 CONTROLLER

SmarAct's initial controller architecture (Delta Tau PPMAC + SmarAct SDC2) proved to be rather cumbersome when customizing for reliable beamline operation. The decision was made to replace the old system with a new one, using SmarAct's state-of-the-art MCS2 controller. Main advantages included: 1) Use of distance coded reference marks, so (re-)referencing could be performed in less than two seconds, with minimal sample movement. 2) Higher resilience to knocks: MCS2 uses a higher encoder readout and interpolation frequency over previous versions (~8 kHz –> 50 kHz), and therefore it can sustain a 'faster' knock, in case of unintended collisions with the sample mounting robot system, so the loss of encoder counts is no longer an issue, and the encoder can be trusted.

Using MCS2 provides a robust foundation for the control of the individual positioners. But a high-level system is required for geometrical model calculation, trajectory generation and user interfacing, and for more complex operations like calibration and active correction.

ROS & SMARGOPOLO

MCS2 can be interfaced via USB or Ethernet. An API for Windows and Linux is provided by SmarAct. For more predictable latency, it was decided to do high-level control on a PC connected via USB.

ROS (Robot Operating System) [6] was selected as a framework for the high-level controller. ROS is an open-source set of software libraries and tools to build robot applications. It has widespread use in robotics research and has strong community support. ROS was installed on Ubuntu Linux 20.04 LTS on a moderately performing Desktop PC (HP Z240, i7, 8GB RAM, 256GB SSD). At the time of development, ROS 1 LTS version "Noetic Ninjemys" was widespread and therefore used. Today ROS2 exists, but has a significantly different system architecture, and porting requires significant refactoring of the code.

A ROS package named "smargopolo" was written to control SmarGon MCS2 (Fig. 2). Smargopolo follows a microservices architecture, with several independent modules (executables) processing parts of the control loop. Trajectory generation, geometrical models, active correction, device interfacing, user input parsing are all individual modules. The package provides a RESTful interface for user and beamline software control, and handles communication with the hardware (MCS2 & Aerotech).

Between modules, ROS offers a lightweight and fast publisher-subscriber communication protocol, ROS messages. This also allows introspection of a running system to observe system behaviour and detect faults.

ROS comes with visualisation tools for signal plotting, and for 3D visualisation. A 3D model of SmarGon was implemented, which can be driven from a running loop and used as a digital twin to test functionality during development. ROS also provides a tool to record streams (rosbag), where motion can be recorded, and later analysed or replayed.

Hardware interfacing can be done whenever there is a device driver/API available for Linux. For instance, ω rotation of SmarGon is controlled with an Aerotech A3200 controller. A fast and pragmatic way to get the absolute position from the Aerotech controller to smargopolo, was to get the Aerotech controller to convert its ω angle into to a sine and cosine signal on two free analog outputs on the controller, and to read this signal by two analogue inputs on a LabJack UE9 I/O card on smargopolo. The advantage of this is an absolute position readout, and no need for coordination if one of the devices needs to be restarted.

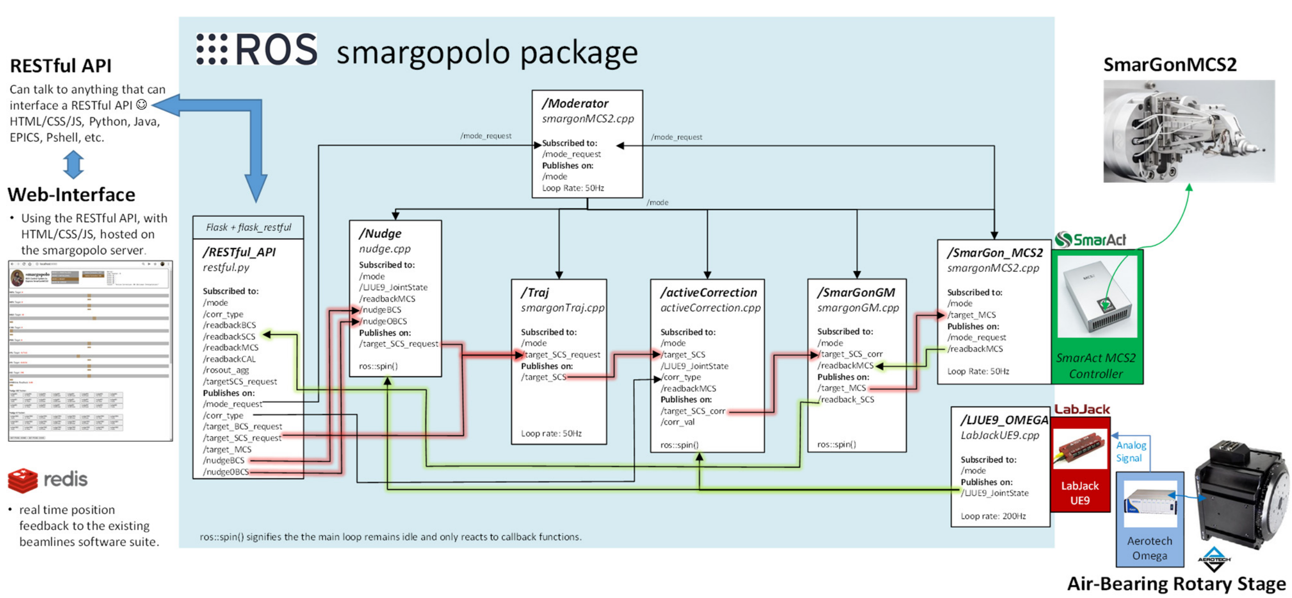

Figure 2: Smargopolo ROS package system architecture. Each box represents a module in the microservice architecture, and the lines represent message flow. Most modules are C/C++ compiled executables, /RESTful_API is a Python script. Red lines show the message flow in actuation direction, green lines show readback message flow.

12th Int. Conf. Mech. Eng. Design Synchrotron Radiat. Equip. Instrum. MEDSI2023, Beijing, China JACoW Publishing
ISBN: 978-3-95450-250-9 ISSN: 2673-5520 doi:10.18429/JACoW-MEDSI2023-THOAM02

Interfacing can be extended with flexiblity. ROS is bilingual and allows modules to be written in Python or in C/C++, depending on the required performance. The RESTful API was implemented in Python using flask & flask_restful. Utilising this API, a user friendly web-interface (HTML/CSS/JavaScript) was written, and is in frequent use. The beamline control software [7] interfaces the API from code written in Java & Python. The SLS beamline software suite also uses a redis database for real time status & position tracking of devices. There was no issue adding this into the Python code of smargopolo.

CALIBRATION

The conversion from user coordinates into motor coordinate is done with an inverse kinematics model (in the opposite direction, it's the direct kinematics model which transforms motor positions into user coordinates). While these geometrical models can be implemented with whatever complexity is required, it is often more efficient to model the geometry with easily verifiable lengths and angles, while making simplifications regarding orthogonality and parallelism. Non-linear effects, like gravitational sag, mounting and bearing inaccuracies, etc. are more difficult to model geometrically, and may vary even amongst devices of the same build.

The concept is to measure the remaining residual systemic error, and to actively correct for it in operation. We rely on the system's repeatability to improve accuracy.

In theory, for every point in the entire workspace, a correction vector must be known. For SmarGon this means, in 6 dimensional space (3 linear + 3 rotational coordinates) corrections in 6 dimensions need to be known.

In practice we can concentrate on a sub-portion of the workspace which is of high importance, and deduce the rest. This is driven by a time vs. performance trade-off of the calibration procedure. Re-calibration might be necessary if there was collision incident, or if a spare part has needed to be exchanged. In such cases, the calibration procedure must be swift, easy to perform and reliable, to be able to resume beamline operation quickly.

The workspace coordinates of main concern for the MX application are the ω and the χ (and φ) axes. Radial and axial runout of these rotations must be kept to minimum. Calibration routines were therefore focused on minimizing these runout errors. In MX, this performance metric is referred to as the sphere of confusion (SoC).

ω is used for every MX data acquisition run. The main contributing factor to runout error is gravitational sag of the mechanical structure. This is rather repeatable. Calibration typically reduces +/− 5 µm errors to +/− 1 µm.

The χ angle is primarily used to change orientation of the crystal. It has the largest run-out error, due to the translation-to-rotation linkage, where the geometrical model doesn't perfectly match with reality. Errors in the region of 30 µm have been observed. With calibration this can be reduced to < 7 µm for χ: [0, 90°], or < 5 um for the commonly used χ: [0, 40°].

φ is more rarely used, and for practical reasons is not calibrated.

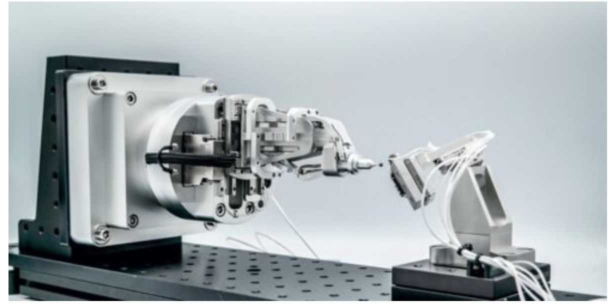

Figure 3: The calibration setup involves rotating a calibration sample, a sapphire sphere (D=2mm) around each of the rotation axes individually and measuring the runout in three dimensions (XYZ). This is negatively fed back into smargopolo as a translation.

In a practical calibration scene (Fig. 3), for ω, points (XYZ positioning errors) over the 360 range are collected every 0.1 degrees and averaged over 10 degrees. These are saved in a lookup table (LUT). χ can be stepped in increments of 10 degrees from 0 to 90, and at each step, a ω rotation is performed like above, and new points are added to the LUT.

This results LUT with 10*36 = 360 points, where for each point correction values in OX, OY & OZ directions are recorded.

In regular operation, this LUT serves as the basis for the active correction. Intermediate points are calculated with bilinear interpolation. With 360 points, this does not put much strain on the main control loop in smargopolo.

USER COORDINATES

SmarGon's coordinate system (SCS) (Fig. 4) corresponds to the coordinate system of PRIGo, where the X & Y axes rotate on the omega stage. The coordinates [OX, OY, OZ, CHI, PHI, OMEGA] describe position and orientation target of the sample. The vector [SHX, SHY, SHZ] describes a sample holder pin from the pin base to the centre of the crystal to be measured at the tip. Smargopolo continuously tries to fulfil the vector equation, so that the sample holder [SHX, SHY, SHZ] is orientated along [CHI, PHI, OMEGA], and positioned, so that the sample at the tip of the sample holder lies on the point [OX, OY, OZ].

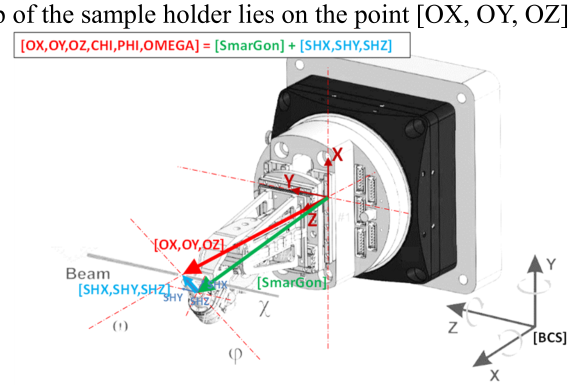

Figure 4: SmarGon's coordinate system (SCS) and vector equation that smargopolo constantly strives to fulfil. SCS rotates with ω, while the beamline coordinate system (BCS) does not.

ISBN: 978-3-95450-250-9 ISSN: 2673-5520 doi:10.18429/JACoW-MEDSI2023-THOAM02

SmarGon was designed for the following use case:

During installation of SmarGon at the beamline, technical staff calibrate it, so its rotation axes (ω, χ, φ) all intersect in one point (within tolerated runout errors). This intersection point is referred to as O in the SCS. O is set to lie on the ω axis (with OX & OY) at a given distance from the base (OZ = 190 mm).

Local beamline staff can then set up the beamline and translate the whole ω stage incl. SmarGon, so the ω axis intersects with the X-Ray at the given distance of 190 mm. O will then be in the centre of interest.

Now when users mount their samples, sample centring is done only via SHX, SHY & SHZ. Similar to a tool vector on an industrial robot, SH represents the Sample Holder vector, which will be different for every newly mounted sample. With SHX, SHY & SHZ, different parts of the sample can be brought into the centre of interest O. With the nudge function, relative motion along any given axis can be performed, so you can nudge the sample along the beamline axes (BCS), or along the axes of a viewing camera, to get it well centred.

Hence, when rotation around CHI, PHI & OMEGA is performed, the sample will stay in the point of interest O.

INTERFACING

Using the RESTful API, readback position can be called using the HTTP verb GET:

```
GET: smargopolo:3000/readbackSCS
```

A JSON object is returned, with the form:

{"SHX": 0.1, "SHY": 0.2, "SHZ": 18.3, "OMEGA": 90.0, "CHI": 20.0, "PHI": 5.0, "OX": -0.04, "OY": -0.18, "OZ": 189.30, seq": 765.0, "secs": 6543.0, "nsecs": 43210.0}

Setting a position can be done with the PUT verb:

```
PUT: smargopolo:3000/targetSCS?SHX=0.123&SHY=0.456
```

This will return the current set values as a JSON object.

There are also other commands for status readback, nudging (relative motion in BCS), referencing, and active correction. For practical use, a web-interface allows easy control of the system (Fig. 5). It uses the RESTful API.

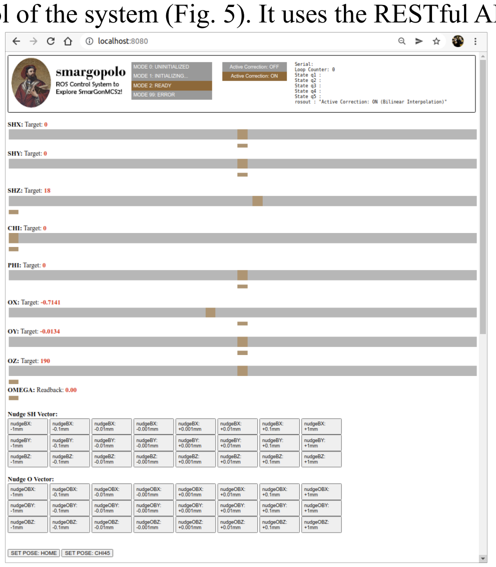

Figure 5: Smargopolo web-interface with double-sliders: the upper part is the move command, the thinner bar below is the current readback value. Nudge buttons below allow relative movement of SH & O in BCS.

APPLICATIONS OUTSIDE MX

A feature of SmarGon is that rotation around an arbitrary point can be performed, due to the use of a geometrical model in the control system. Compared to hexapod architectures, typically, larger rotation angles can be covered. Two rotations can even rotate indefinitely; ω: $[-\infty, +\infty]$, χ: $[0, 90°]$, φ: $[-\infty, +\infty]$. The system is also well adapted to orientation tasks where self-shadowing must be kept minimal.

An application outside the field of MX was for Small Angle Scattering Tensor Tomography (SAS-TT) experiments at the PXI beamline. A tomographic setup and multi-axis sample orientation was required. SmarGon could be re-purposed to allow orientation of the sample along axes commonly used in tomography experiments.

In another spontaneous setup for an experiment at beamline PXI, it was necessary to thread the 50 µm X-ray beam through a capillary of 100 µm inner diameter and 24 mm length. SmarGon was used to position the entrance of the capillary into the beam, and orientation could be scanned until the most intense signal could be detected.

CONCLUSION

With these improvements, SmarGon is now reliable enough to be constantly operated at all MX beamlines at SLS. Two devices have been in continuous 24/7 use for two years at beamlines PXI & PXII at SLS, with very satisfactory results. The replacement of PRIGo at PXIII, after 13 years of operation, is planned during the SLS 2.0 upgrade.

Thanks to the open platform, we are confident to be able to further optimise and adapt the design to accommodate emerging applications in the future.

ACKNOWLEDGEMENTS

The SLS MX Group would like to acknowledge the ongoing and fruitful collaboration between SLS and SmarAct GmbH.

REFERENCES

[1] S. Waltersperger *et al.*, "PRIGo: a new multi-axis goniometer for macromolecular crystallography," *J. Synchrotron Radiat.*, vol. 22, part 4, pp. 895-900, Jul. 2015. doi:10.1107/S1600577515005354

[2] SmarAct GmbH, http://www.smaract.com/en/smargon

[3] M. R. Fuchs *et al.*, "D3, the new diffractometer for the macromolecular crystallography beamlines of the Swiss Light Source," *J. Synchrotron Radiat.*, vol. 21, no. 2, pp. 340-351, 2014. doi:10.1107/S160057751400006X

[4] K. M. L. Smith *et al.*, "SDU – software for high-throughput automated data collection at the Swiss Light Source," *J. Synchrotron Radiat.*, vol. 30, part 3, pp. 538-545, May 2023. doi:10.1107/S1600577523002631

[5] I. Martiel *et al.*, "The TELL automatic sample changer for macromolecular crystallography," *J. Synchrotron Radiat.*, vol. 27, part 3, pp. 860-863, May 2020. doi:10.1107/S1600577520002416

ISBN: 978-3-95450-250-9 ISSN: 2673-5520 doi:10.18429/JACoW-MEDSI2023-THOAM02

[6] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," *ICRA workshop on open source software*, vol. 3. no. 2, 2009.

[7] J. A. Wojdyla *et al.*, "DA+ data acquisition and analysis software at the Swiss Light Source macromolecular crystallography beamlines", *J. Synchrotron Radiat.*, vol. 25, part 1, pp. 293-303, Jan. 2018. doi:10.1107/S1600577517014503