

RAPID CONTROL PROTOTYPING TOOL FOR THE SIRIUS HIGH-DYNAMIC DCM CONTROL SYSTEM

G. B. Z. L. Moreno, R. M. Caliari, R. R. Gerales, M. A. L. Moraes, LNLS, Campinas, Brazil

Abstract

The monochromator is known to be one of the most critical optical elements of a synchrotron beamline. It directly affects the beam quality with respect to energy and position, demanding high stability performance and fine position control. The new high-dynamic double-crystal monochromator (HD-DCM) [1], prototyped at the Brazilian Synchrotron Light Laboratory (LNLS), was designed for the future X-ray undulator and superbend beamlines of Sirius, the new Brazilian 4th generation synchrotron [2]. At this kind of machine, the demand for stability is even higher and conflicts with factors such as high-power loads, power load variation, and vibration sources. To identify and ensure sufficient control of the dynamic behaviour of all subcomponents in the prototype, an implementation in MATLAB/Simulink Real-Time environment in a Speedgoat Real-Time Performance Machine was developed. This approach enables rapid prototyping, by allowing a shared environment for system modeling and testing. The tool was developed in a modular architecture aiming at practical model iteration and platform migration to standard beamline controllers, which can prove portability and scalability features.

INTRODUCTION

The first beam for Sirius, the new 4th generation synchrotron light source being built in Brazil, is planned for mid-2018. When fully operational, Sirius promises to deliver a beam with very low emittance (250 pm.rad) and high brilliance [3], putting this facility amongst the best in the world for its energy range. This will enable beamlines to perform time-resolved experiments or use nanobeams to increase the flux and resolution at the sample even further, pushing for new performance standards for all kinds of beamline instrumentation and optics [4]. At X-ray beamlines, key factors as photon flux, energy selection, and beam positioning at the sample may be directly affected by instabilities of a Double-Crystal Monochromator (DCM), resulting in a decrease of the beamline experiment performance [5]. This way, not only the mechanical systems and mechatronics concepts had to be reviewed for this new generation DCM, but also the control system had to be quickly adapted to a new demand level. The concept review resulted in the HD-DCM, a system with high noise rejection and trajectory tracking capability. Considering the upcoming tight deadlines for having not only the HD-DCM system, but also other instruments with similar complexity, fully commissioned at Sirius, a Rapid Control Prototyping (RCP) tool became necessary to speed up control design and testing phases.

This paper describes the implementation of such an RCP tool for the new HD-DCM. In the next sections, an introduction to the control system parameters of the HD-DCM is followed by a brief description of the RCP concept and the hardware chosen for this project. Then, a detailed description of the software architecture implemented in MATLAB/Simulink environment and a quick report of the system performance and advanced capabilities are presented.

It is important to say that this implementation is meant for prototyping purposes only and that its architecture and parametrization also focuses on a smooth migration to the standard platform of advanced applications control for Sirius beamlines [6].

HIGH-DYNAMIC DCM

The HD-DCM is a completely reviewed version of the usual DCM systems, designed to bring parallelism stability between crystals to the level of a few nanoradians. As depicted in Fig. 1a and more detailed in [5], [7], the core of the DCM comprises: the main rotating frame (GoF) for the Bragg angle selection, with the horizontal rotation on the optical surface of first crystal; the metrology reference frame (MeF1), which is fixed to the GoF and to which the first crystal is stiffly connected; the long-stroke frame (LoS) with one translational degree of freedom (coarse gap) with respect to the GoF; and the high-dynamic module of the second crystal, with the short-stroke frame (ShS), to which the second crystal is stiffly connected, and the balance mass (BM), both with three degrees of freedom (fine gap, pitch and roll) with respect to the LoS. The high-dynamic performance is achieved by using low-stiffness (voice-coil) actuators between the ShS and the BM, and an embedded interferometry system between the ShS and the MeF1.

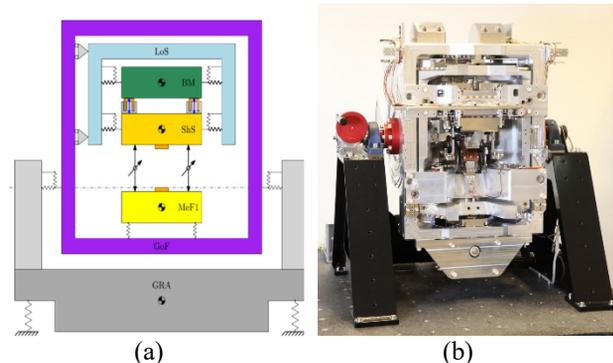


Figure 1: HD-DCM mechanical design schematics (left) and prototyp assembly (right).

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Partial characterization with the core of the HD-DCM mounted on dummy bearings, still in-air and at room temperature, as seen in Figure 1(b) resulted in 9.2 nrad pitch and roll, and 0.9 nm gap stability (RMS values integrated up to 2500 Hz). Results for the complete in-vacuum cryo-cooled system are expected by the end of 2017 [7].

To meet the goal of nanoradian stability between crystals, extensive error budgeting and modeling were made prior to the prototype construction. These analyses showed that a minimum feedback rate of 10 to 20 kHz was needed for controlling the position of the high-dynamic module within the required stability [8]. Considering the requirements for the Bragg angle, the gap and the 3 degrees of freedom coordinated motion, a selection of actuators, sensors, and feedback rates was made for the RCP tool. It is important to note that, in addition to precise motion control and active disturbance rejection, thermal gradients must be avoided to minimize geometric deformations that could compromise the system performance [9]. The updated control specifications for the HD-DCM prototype is revisited in Table 1.

Table 1: Main Control Specifications [5]

	Bragg Angle	Long Stroke	Short Stroke	Thermal Control
Actuator Type	Torque Motor	Stepper or Servo	Voice-Coils	Foil Heaters
Sensor Type	Rot. Incr. Encoder	Lin. Abs. Encoder	IFM	RTDs
Feedback Res.	50nrad	5nm	0.1nm/1nrad	<10mK
Stability target*	<0.8μrad	N/A	<10nrad	<50mK
Closed-Loop BW	35 Hz	20 Hz	>250 Hz	0.1 Hz
Feedback Sampling	10 kHz	10 kHz	20 kHz	20 Hz

RAPID CONTROL PROTOTYPING

Considering the upcoming high demand for designing and commissioning multiple high-end systems in a short time, and the earlier uncertainty about the standard control platform for Sirius, the decision in favor of an RCP tool was made in the beginning of the project.

Concept

Rapid Control Prototyping (RCP) is a process that allows real-time calibration of control algorithms, prior to the availability of an Industrial Electronic Control Unit (ECU) or any kind of hardware necessary for production [10]. In the context of a synchrotron beamline, it allows to test a new system and iterate control strategies without wasting beamtime, once it spares the use of physical hardware in almost every process phases. Furthermore, RCP can be allied to Model-Based Design, as shown in Figure 2, shortening development time and costs. Indeed, RCP may indicate corrections in early stages of the

process, avoiding the construction of several prototype versions. Combining simulation interfaces, control design tools, and target I/O drivers in the same platform unifies design and verification phases, enabling quick switching between model in the loop (MIL), software in the loop (SIL), processor in the loop (PIL), and hardware in the loop (HIL) testing, granting better flexibility and avoiding compatibility faults. The schematics in Figure 2 shows the system engineering V lifecycle used for designing and testing the HD-DCM. Note that only the last phase, Homologation Testing, should be performed at the beamline, with a good chance that most of the problems may already be solved in previous phases.

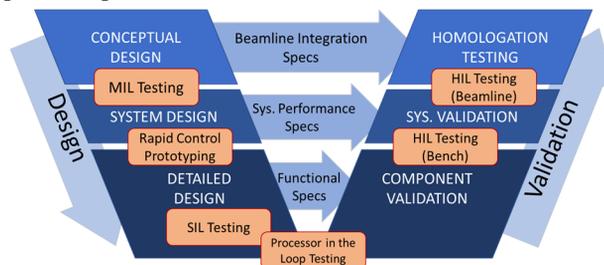


Figure 2: Model-Based Design Systems Engineering V Life Cycle for HD-DCM.

Hardware

The platform adopted as the RCP tool for the HD-DCM project was Speedgoat Performance Real-Time Target Machine, which is natively integrated with MATLAB/Simulink environment as an xPC target [11]. Using MATLAB resources, the control system could be prototyped directly from Simulink, where the analytical models were made during conceptual design. This hardware is Speedgoat's mainstay target machine, being able to replace the beamline ECU as the main controller until the final HD-DCM homologation tests.

The Performance Machine crate has an Intel Core i7 CPU, and includes 7 slots for I/O modules: 4 slots were occupied with Analog and Digital I/O, and dedicated boards for thermal input and system identification, used for control and characterization; 3 slots are still free and may be occupied by other I/O modules on demand. It is also possible to add an I/O-expansion chassis to make even more slots available if necessary. Table 2 shows the specifications for the main boards composing this hardware:

Table 2: Performance Machine Boards Specifications

Hardware	Function	Maximum Rates
IO333	Digital I/O (with FPGA)	Applic. Dependent (75MHz Clock)
IO107	Analog Output	~450kHz update
IO171	RTD Sensing	~200kHz (+DMA)
IO104	System Identification	2MS/s (AD+DMA) 1MS/s (DA+DMA)

Motor drivers can be controlled via Digital 5V PWM or Analog ±10V signals, and any feedback input such as

* In-position stability, RMS values integrated up to 2500 Hz.

quadrature or absolute encoders are configurable via specific drivers distributed within the Performance Machine bitstream. In the need for different pinout or new functionalities for the system, there is a chance of changing any configuration and even adding new drivers. This flexibility is an advantage granted by the programmable Real-Time controller and the IO333 FPGA board. This combination makes the platform scalable and flexible to control many other applications in the future.

SYSTEM ARCHITECTURE

The demand for quickly testing the HD-DCM system and future projects has put modularity and flexibility as key factors when structuring the RCP tool. This way, the software architecture for its implementation in Simulink Real-Time was made in blocks, and organized in preliminary layers, namely: low-level interface, mid-level, and high-level interface, as shown in Figure 3. Due to the serial nature of Simulink Real-Time, the blocks are connected via bus links. The hard-real-time MATLAB embedded operating system ensures all data processed in the entire model at each iteration. The tests were run using MATLAB scripts and direct command-line using Speedgoat libraries, allowing on-the-fly parameter tuning, signal monitoring, and data logging, in addition to versatile user interfacing options. Among these interfaces, one can use Simulink Real-Time Explorer, which connects directly with the Speedgoat target, or graphical user interfaces implemented in MATLAB tools as App Designer[12] or GUIDE[13], or even external languages as Python.

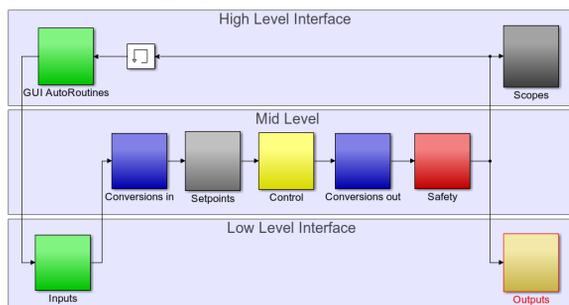


Figure 3: Modular architecture implementation.

Depending on the RCP tool application, or design phase, the high-level or low-level interface layers may be replaced by models. For instance, in a software-in-the-loop test, the low-level interface, meant for hardware-in-the-loop tests, may be replaced by plant models. Similarly, for homologation tests at the beamline, the blocks from the high-level interface layer may be replaced by other implementations integrated to the new control system. In all situations, however, the mid-level can be designed and tested independently from the configurations of the other architecture layers. This functional division also paves the way to the future migrations to Sirius standard beamline control system [6], in a way that the input parameters and modular divisions can be more straightforwardly converted or migrated to new platforms.

System Parametrization

To make a clear distinction between the different types of variables and signals used the system, each variable, parameter, or signal has a suffix, explained in Table 3. The intention of having such parameter division is grouping them more comprehensively according to their function. It also eases establishing an access hierarchy in the beamline version of the control system, protecting critical system parameters, such as homing distances, controller gains, and coordinate transformation matrices.

Table 3: Variables Naming Convention

Suffix	Description	Access	Example
...R	Ref. / Setpoint	R/W	ShsYR
...S	Sensor Signal	R	ShsYS
...E	Position Error	R	ShsYE
...F	Open Loop Force	R/W	ShsYF
...P	User Parameter	R/W	ShsYMinP
...D	Design Parameter	R	ShsYHomeD

Block Modules

Next, a brief description of each block function is presented:

I/O Drivers (Low-Level): *Inputs* and *Outputs* in the low-level interface layer are composed of Speedgoat's library drivers, which interface with their correspondent I/O resources. Examples of these drivers are quadrature and absolute encoders, from the IO333 FPGA board specified in Table 2.

I/O Conversion: *Conversions in* and *Conversions out* are responsible for converting any data format coming from the sensors into engineering units and the other way around going to the system drivers, respectively. This way, the setpoint generation and control calculations have a standard representation, avoiding compatibility errors when tuning the controller. Any sensor combination or geometric transformation is also included in this module, in a way that rigid body motions can be quickly translated from the readings of a group of sensors, for example. The units used inside the model are adopted as SI units, for all variables. Figure 5 shows a Simulink code snippet with the signal and coordinate systems conversions, as well as the inclusion of design parameters (i.e. homing positions), for all the DCM motion axes.

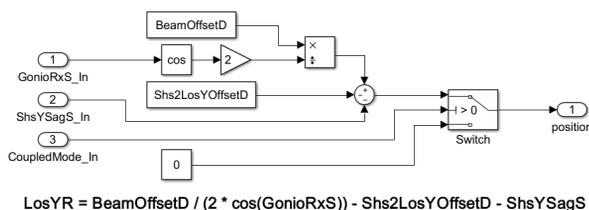


Figure 4: Coordinated-motion setpoint calculation.

Setpoints: This module merges user setpoints, sensor outputs, and calibrated corrections, passing the result to the

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

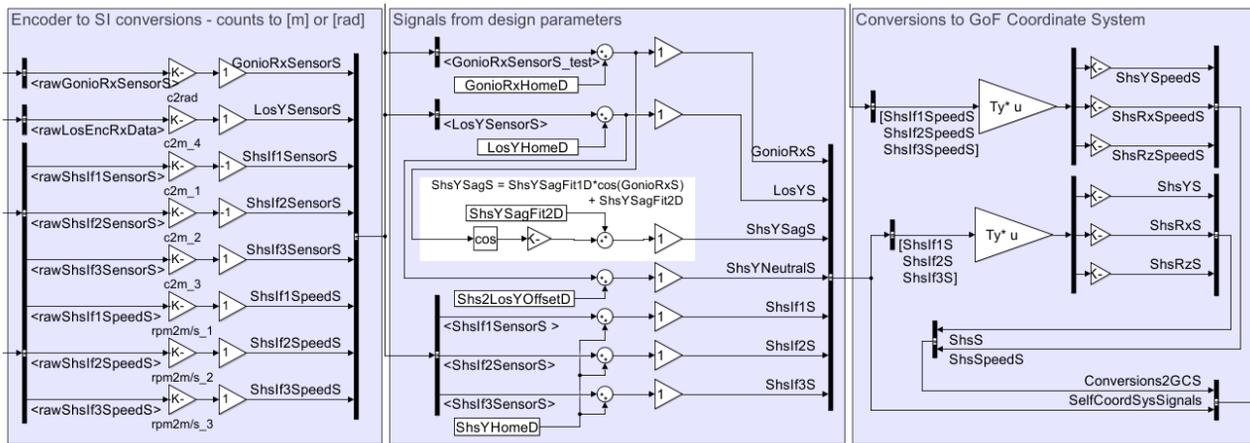


Figure 5: Example of input conversions for the HD-DCM.

controller input. Figure 4 shows an example of the setpoint calculation for the 2nd crystal long-stroke (LoS) motion: in coupled mode, the LoS changes its position according to the Bragg angle (GonioRxS) and additional design and calibration parameters.

Due to the high stability targets, an important requirement for the setpoint generation is smooth trajectory generation. Hence, a group of native MATLAB functions is included in this block to provide 3rd order trajectory generation. It is based on [14]. Although this package allows for 4th order trajectory generation, which imposes limitations until the derivative of jerk, a simpler approach (maximum jerk limitation) was judged to be sufficient to avoid vibration disturbances in the system.

Control: This module contains all the implementations regarding open-loop and closed-loop control for each axis. As Simulink offers a variety of control design representations, feedback, feedforward, and different control strength settings may be programmed. An LTI system block is used to allow for generic representation of controllers as a linear time-invariant model, accepting state-space, transfer function, or zero-pole gain formats. This also improves the flexibility and portability of the system, by allowing standard and well-known controller design representations.

Safety: To indicate dangerous events and prevent damaging the output drivers, a series of flags and error signals are generated and codified in this block. Alert messages and emergency procedures, such as gradually decreasing the speed of motors and shutting down, may be triggered according to a given error state.

Auto-Routines (High-Level): This high-level interface block is composed of Finite State Machines (FSMs) to perform automatic routines such as homing and initial functional tests. The use of FSMs may simplify the development of complicated routines and optimise system performance by minimizing communication between the target and the host PC, which may cause long delays.

Scopes (High-Level): This block in the high-level interface layer contains Simulink scopes, which directly extract relevant information from the main bus to workspace variables or a display at the target interface screen with

configurable frame rate. Another advantage of this structure is the capacity to quickly switch between scope layouts depending on the test mode being performed.

SYSTEM PERFORMANCE

The HD-DCM is considered one of the most challenging control systems being designed for Sirius so far, which is the reason why it was chosen as a model for advanced control systems in future applications. This granted the opportunity of testing some of the limits of the Speedgoat FPGA Board Drivers and giving some feedback for driver improvements. As the entire system loop was running without parallel processing, the maximum rates would be limited by the Simulink model size or the slowest driver in the model. During the RCP tests, with the full system running, the maximum achieved rate was 20 kHz, bottlenecked by the absolute encoder drivers for decoding BiSS-C protocol signals. Table 4 lists all the IO333 board drivers configured for the RCP system, indicating the bottlenecks that have already been met.

Table 4: Bottlenecks for Performance IO333 Drivers

Driver	Function	Bottleneck
QAD v3	Quad. Encoder	Limited to 75MHz clock (4x res. not possible at 20MHz)
BiSS Encoder	Biss-C Encoder	Minimum latency limited to 80µs (max. 12.5 kHz)
PWM Gen	PWM Output	Latency of a full PWM period to apply parameter changes
Dig. I/O	Trigger Output	13.3ns latency specification (max. 75MHz)

ADVANCED APPLICATIONS

For completing the V Life Cycle described in Figure 2, extensive modeling and system characterization was carried out in MATLAB environment. Two main functionalities present in the RCP tool are discussed in next sections.

System Identification

System identification is one of the most important functions for the system design and validation phases. For this project, analog drivers and electrical plants for the servo

motors were identified using the IO104 board [7]. Frequency domain identification was also performed imposing step sine profiles into the system outputs and collecting encoder data via the IO107 and IO333 boards respectively. Characterizing amplifiers and actuator dynamics is very important to confirm and, if necessary, correct the selection of components in early design phases. This functionality is essential for the system validation phase, once it can indicate unmodeled effects or ease error source identification, when matching diagrams of the modelled and real systems. Figure 6 shows Nyquist plots from the model system plant and the identified plant for the HD-DCM prototype. As it can be seen, performance and robustness can be quickly evaluated and compared, indicating the path for future improvements.

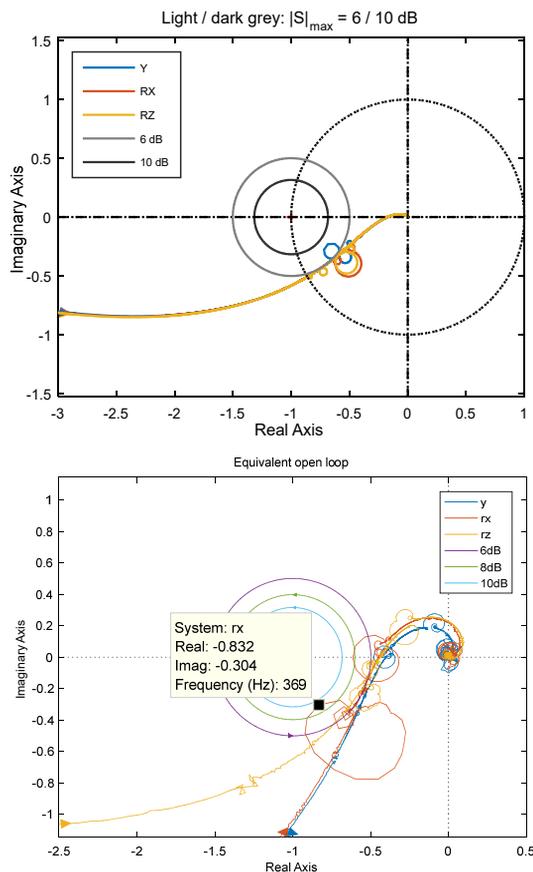


Figure 6: Comparison of Nyquist plots between the DCM model simulation (up) and system identification (down).

Virtual Instruments

The plant model, designed in MATLAB by LNLS and MI-Partners during the HD-DCM conceptual design phase, was implemented in Simulink, allowing for software-in-the-loop tests with no real instrumentation built. Expanding this to a full system model, considering beamline parameters and calibration, may speed up even the system homologation phase, saving precious commissioning time at the beamline. This strategy may be conveniently used in the upcoming Sirius beamline instruments.

Moving one step ahead, having recently performed the identification of the plant, sensors, actuators, and amplifiers, and having measured real disturbance signals, the analytical models and theoretical gains originally used in the Virtual HD-DCM can be replaced by their mathematical representations of the real system. This means that the RCP platform may recreate a nearly perfect copy of the real HD-DCM, which can be used, for instance, if the mechanical hardware is unavailable or for completely independent of-line tests and improvements.

FUTURE STEPS

The prototyping control platform will be used with the HD-DCM until all its subsystems are integrated and it is completely validated. These remaining steps include the main rotary stage, the cryocooling system, the vacuum system, safety interlocks and the thermal management. Regarding performance, the ultimate target is having the cryocooled system capable of running fly-scans within the desired stability levels in the end of 2017.

After that, the homologated controllers shall be migrated to the standard beamline control platform, in which its performance is expected to be optimized in minimum steps, as described in [6].

CONCLUSION

The RCP tool implemented in Speedgoat's xPC target accelerated the design and testing phases, permitting the model and hardware tests to coexist in the same platform. Amongst the future demand for micro and nanoprobe systems, high stability tomography stations, advanced mirror systems, and quick EXAFS monochromators, the new HD-DCM became a reference for future projects with similar complexity. It combined not only different motion control and metrology strategies, fast closed-loop control and instrument triggering, and coordinated motion, but also demands precise thermal control.

The proposed architecture modularity and flexibility, allows parameters and even complete structures to be directly exported from the Simulink model to the standard beamline control platform, which has a more advantageous cost-benefit relation. Indeed, once the development of the HD-DCM is finished, its final control platform will take over and the RCP tool will be modified for a new system.

ACKNOWLEDGEMENT

The authors would like to gratefully acknowledge the funding by the Brazilian Ministry of Science, Technology, Innovation and Communication and the contribution of the LNLS team, the MI-Partners team and those of the synchrotron community who directly or indirectly built the path to this development.

REFERENCES

- [1] R. R. Gerales, R. M. Caliar, and M. S. Silva, "Método de controle de grau de liberdade em sistemas mecânicos e monocromador de duplo cristal," INPI BR 10 2016 020900 5, 2016.

- [2] A. R. D. Rodrigues *et al.*, “Sirius Status Report,” *Proceedings of IPAC2016*, pp. 2811–2814, 2016.
- [3] L. Liu, F. H. de Sá, and X. R. Resende, “A new optics for Sirius,” *Proc. IPAC’2016*, pp. 3413–3415, 2016.
- [4] R. Hettel, “DLSR design and plans: An international overview,” *J. Synchrotron Radiat.*, vol. 21, no. 5, pp. 843–855, 2014.
- [5] R. R. Geraldes *et al.*, “The new high dynamics DCM for SIRIUS.”
- [6] M. A. L. Moraes, H. D. Almeida, R. M. Caliani, R. R. Geraldes, G. B. Z. L. Moreno, and J. R. Piton, “Control architecture standarization path for SIRIUS,” *ICALEPCS’17 Proc.*, 2017.
- [7] R. M. Caliani, M. A. L. Moraes, G. B. Z. L. Moreno, R. R. Geraldes, T. Ruijl, and R. Schneider, “System identification and control for the SIRIUS high dynamics DCM,” *ICALEPCS’17 Proc.*, 2017.
- [8] R. R. Geraldes, R. M. Caliani, G. B. Z. L. Moreno, M. J. C. Ronde, T. A. M. Ruijl, and R. M. Schneider, “Mechatronics concepts for the new high dynamics DCM for SIRIUS.”
- [9] M. S. Silva, R. R. Geraldes, A. Gilmour, T. A. M. Ruijl, R. M. Schneider, and M. I. Partners, “Thermal management and crystal clamping concepts for the new HIGH DYNAMICS DCM FOR SIRIUS,” no. c.
- [10] Speedgoat GmbH, “Speedgoat Solutions and Use,” 2011.
- [11] MathWorks, “Perform Hardware-in-the-Loop Simulation with MATLAB and Simulink to Test and Validate Control Algorithms,” p. 6, 2016.
- [12] W. Wilson, “What ’ s New in MATLAB,” 2016.
- [13] MathWorks Inc., “Creating Graphical User Interfaces,” *MATLAB User Guid.*, p. 502, 2015.
- [14] P. Lambrechts, M. Boerlage, and M. Steinbuch, “Trajectory planning and feedforward design for electromechanical motion systems,” *Control Eng. Pract.*, vol. 13, no. 2, pp. 145–157, 2005.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.