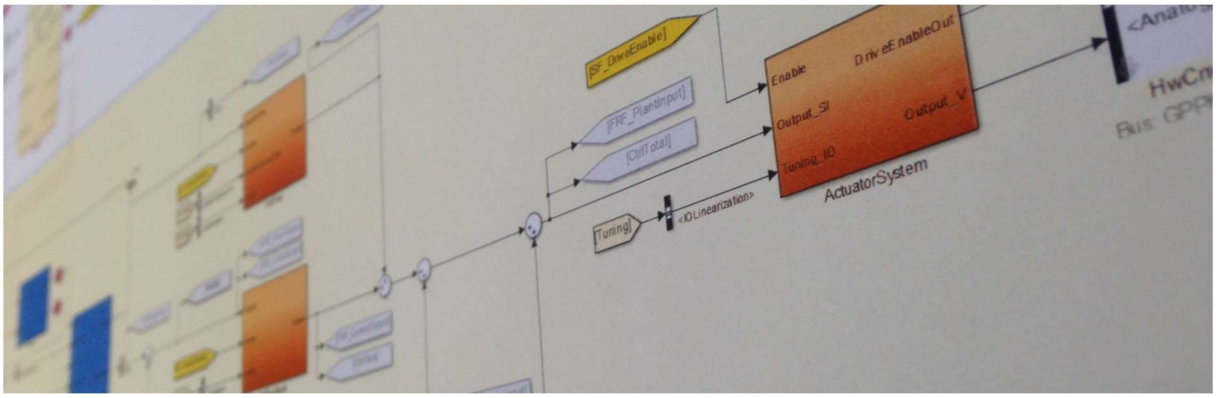
Domesticate your equipment physics with Sioux's Advanced fleXible Control Solution and outsmart the competition with an approach to unleash the full potential of state-of-the-art control engineering for industry.



Abstract

Today, much of the equipment used to perform routine tasks or to manufacture common products, are complex systems. These consist of complex computer-controlled mechanisms called cyber-physical systems. Delivering expected performance improvements, requires technological advancement of the whole chain of systems. It requires sophisticated methods to identify detailed behaviour down to the level of physics. This continuous quest nowadays gets assistance from tools that autonomously assess and analyse industrial machines, looking for potential improvements. Sioux technologies forged a workbench to concatenate multiple esteemed tools to bring solutions compatible with industry standards. This paper explains the challenge, how we faced it and what it brings.

Introduction

Sioux's drive and control experts can translate the functional requirements of e.g. a motion task into a tangible decomposition of the drive chain. By modelling a physical representation of those drive concepts, they can predict the performance at an early phase of the product creation. By doing so, they identify the mandatory *control algorithms* to meet the envision performance. We witness a continuous growing demand of more speed and accuracy. This requires multiple actuators (of different types) and multiple sensors that require a so-called MIMO control approach. A classical PID controller with some tuneable filters cannot deal sufficiently with repetitive disturbances. Reference setpoints must be shaped to avoid unwanted excitation of machine dynamics. High motion responsivity as in above 100Hz, requires sample times of at least 10 kHz. Low frequent thermal effect should be compensated for. This is a subset of features, often not facilitated in todays of-the-shelf control solution vendors. Although commercially available control platforms have become more mature and intelligent, they still focus on "standard solutions for standard problems". Test and integration more and more bring unexpected dynamical behaviour and/or interminglement of motion behaviour. With SAXCS our control engineers can instantly adapt the heart of the controller (impossible in of-the-shelf solutions!) and deploy it again in minutes.

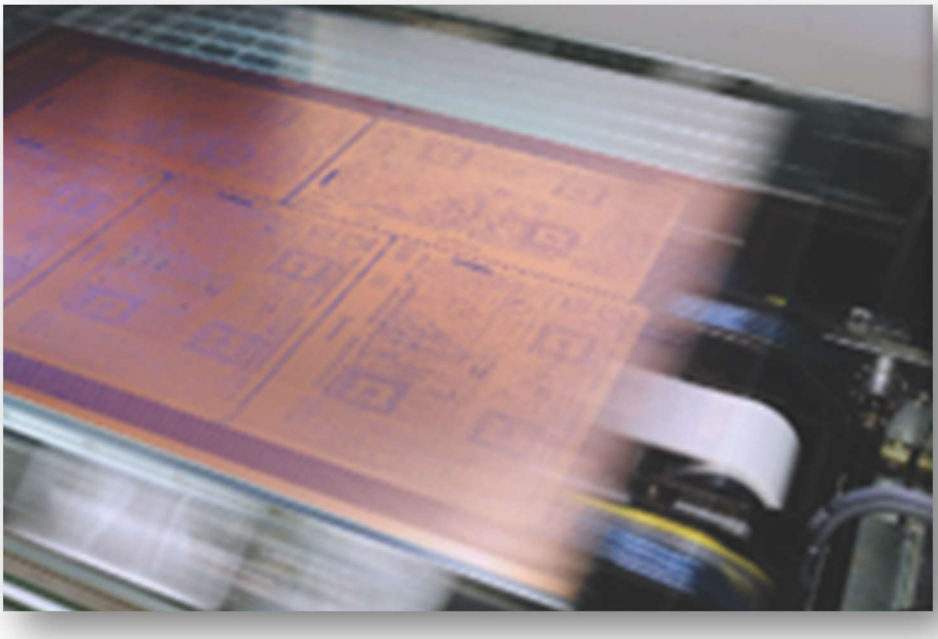


Figure 1 – Because of the ever-increasing demands for speed and accuracy, the need for a advanced and flexible control solutions is of paramount importance.

Why SAXCS Pays off

- **Flexible control design**
  Where of-the-shelfs solutions are rigid, SAXCS van be applied for any control architecture to serve more advanced functionality in your application.
- **Domain expert stays in control**
  The control expert can organize the whole solution from design to code.
- **Hardware independent**
  SAXCS fits any X86 CPU hardware architecture; thus, adaptable for customers supply chain.
- **Powerful test and integration framework**
  An open and widely used Python environment with unprecedented tracing features.

- **Faster development time**
  Model-based design methodology boosts internal communication and safes time, like explained at www.incose.org. The concept of a digital twin enables parallel and agile way of working. Generating code from this twin to the real-world hardware avoids laborious and error-prone transfer via SW engineers.
- **Less project risks**
  Easy reuse of control reference architectures and assurance the unexpected complexity of physical behavior can be mitigated with more advanced algorithms.

How did we do it?

To learn about the detailed motivation of Sioux's approach, you could check www.i-mech.eu and/or its successor www.imoco4e.eu. In figure 2, the layered approach reveals a EtherCAT communication bus to interface different types of I/O, usually close to the equipment's actuators and sensors. This esteemed bus protocol is applied by many vendors of I/O devices and an overview can be found through www.ethercat.org. This brings our customers flexibility to organize a supply chain with second sources for key components like e.g., their appropriate motor amplifier. Component obsolescence is something that OEM companies are increasingly facing today, often shortening the lifecycles of parts. With SAXCS risk full and costly redesigns can be avoided, since a second or third source can be organized easy. The central control layer (Layer 2) usually consists of high performing multicore X86 CPUs, but again this hardware can be tailored to the customers supply chain management strategy. The approach complies with the principles of model-based systems engineering, a.o. meaning that tailored solutions can run in simulation. The team responsible for the 'orchestration' of activities inside the equipment, can deploy their software without needing any equipment hardware. It speeds up development and improves software quality, by its model-based testing methods.

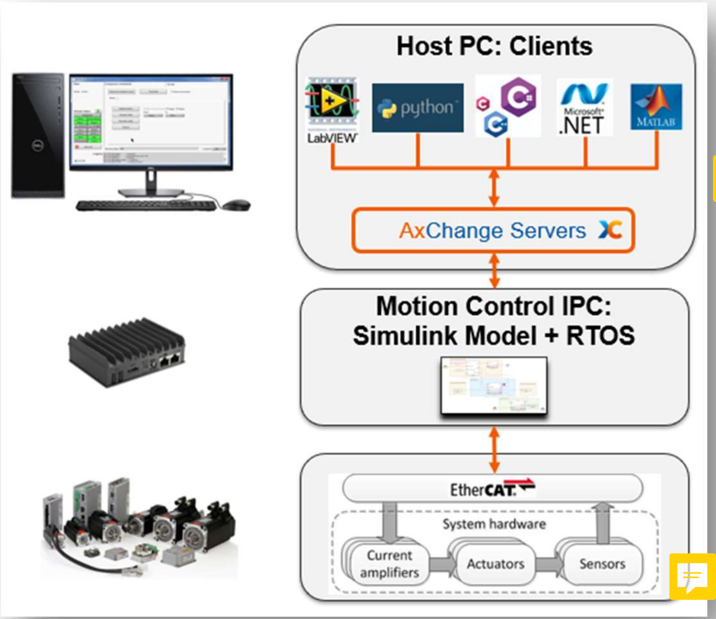


Figure 2 – SAXCS can be decomposed into 3 layers; The bottom layer 'Layer 1' is called 'instrumentation layer' to interact with your plant. The middle layer 'Layer 2' holds all control algorithms and execute the math real-time. The top layer 'Layer 3' kind of orchestrates the behavior of the equipment, including interaction with the factory and operators.

An integral part of the model-based design approach is the tailored controller mathematics as part of the 'Layer 2' as denoted in Figure 2. Control engineers can compose and verify this in MathWorks SIMULINK & Stateflow including the envisioned plant characteristics of 'Layer 1'. By using mature code generation tools from MathWorks, the controller can be deployed on the appropriate target, without interference of an embedded SW engineer. This method serves an agile way of working to boost the productivity. Furthermore, automatic code generation brings speed and high-quality code.

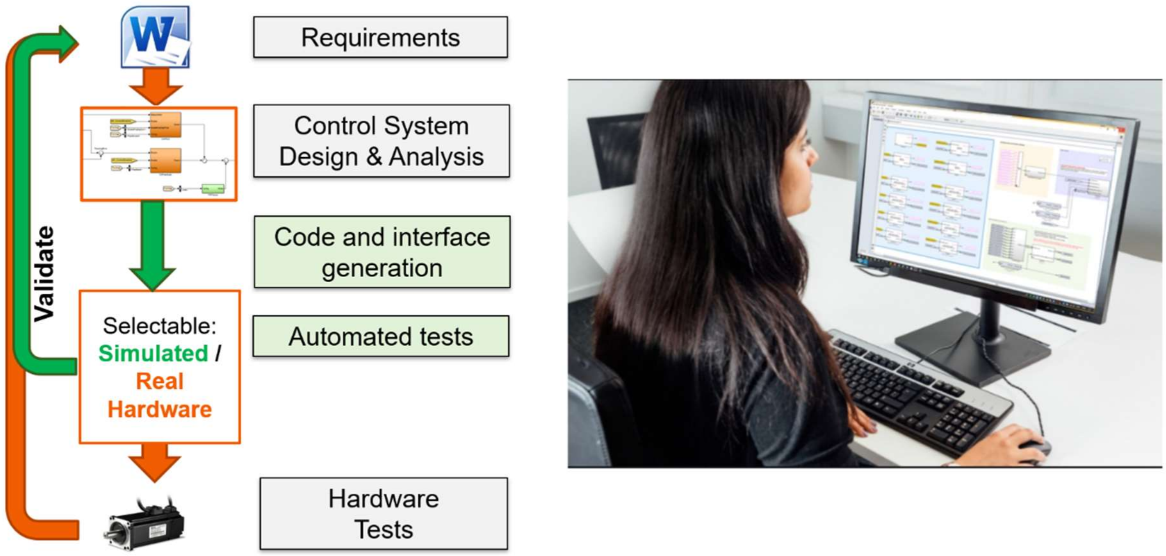


Figure 2 – SAXCS facilitates very fast design iterations and hardware integration because of code generation and integral hardware simulation

The behavior of equipment can be designed by a tool that fits the customers preference. Default, Sioux offers connectors with LabVIEW, Python, C#, .NET and MATLAB. The Python socket is particularly useful since many engineers are acquainted with intuitive programming languages. They can easily compose diagnostic scripts e.g., to engage automatic calibration. Examples can be found in this paper. This empowers mechatronics engineers to be even more efficient, flexible, and independent from software engineers. This again brings speed in the creation of advanced control solutions.

A closer look

Sioux created a solid software framework that allows the mechatronic engineers (!) to create high-quality motion solutions in a short time frame. This can be achieved by using many reusable components, thus preventing the continuous re-invention of the wheel. A modular architecture enables creation and interaction between any number of motion axes. The core of the framework consists of predeveloped and validated SAXCS building blocks (controllers, filters, setpoint generators, error detection, etc.), design patterns and templates for MATLAB/Simulink. Using these building blocks any control strategy can easily be designed and quickly validated on either the integrated system simulation model or the system hardware.

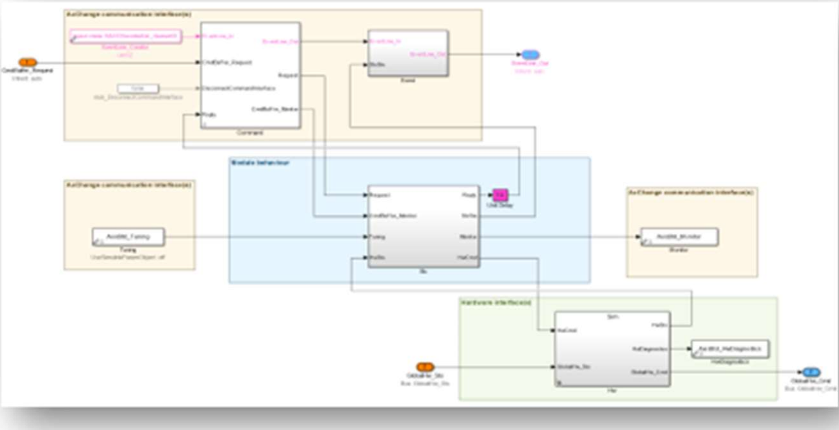


Figure 3 – The SAXCS platform core: Well-organized Simulink templates using well-tested building blocks.

Managing module behavior, configurability, tuning & diagnostics tools are seamlessly integrated in the platform and can be defined by the mechatronic engineer as well. Machine behavior is visually programmed using readable design patterns for Stateflow®.

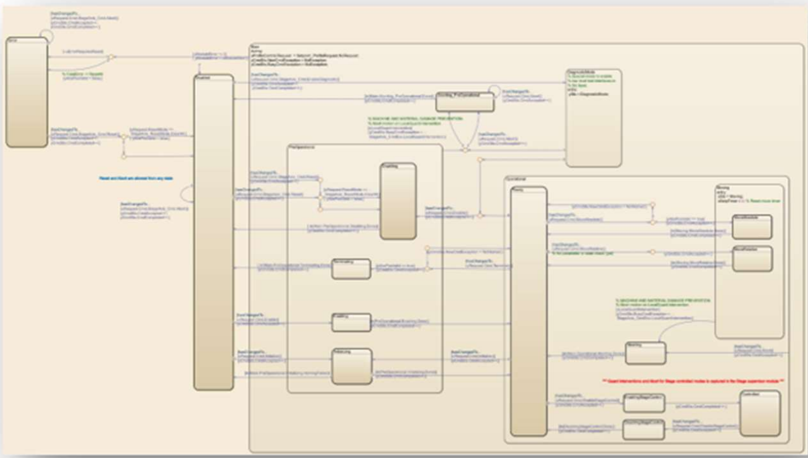


Figure 4 – Easily create readable machine behavior charts using SAXCS design patterns in Stateflow®.

The so-called *AxChange* servers (as depicted in Figure 1) provide multi-client communication between the motion modules running on the Motion Control IPC and the client applications running on the Host PC. This means that multiple clients, like a graphical user interface and Python automation interface can control and monitor motion modules at the same time, enabling parallel operation of different modules and powerful remote support. Motion commands, parameters and signals are generated from a text-based definition file for *all* layers of the software architecture and system documentation. This eliminates human error, produces self-documenting, human-readable interfaces and allows interface updates without help from a software engineer. The figure below shows a typical example of the hierarchical parameter structure of a basic SAXCS motion control axis and readable monitor signals in the configurable graphical user interface.

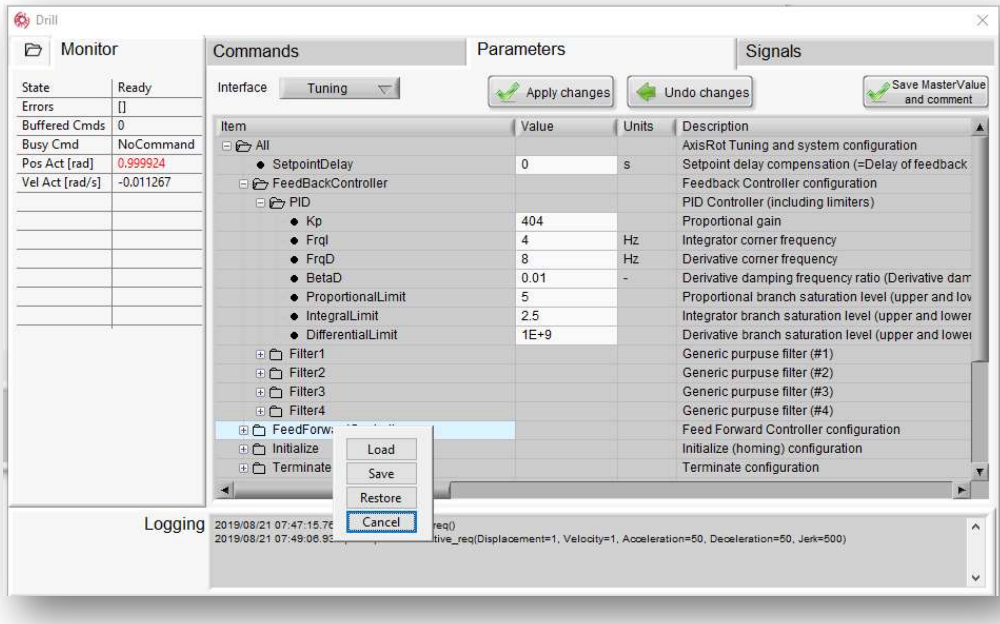


Figure 5 – The interface generator produces structured and readable commands, parameters, and signals

The interfaces for the Python automation language are also generated. Using the auto-complete and quick documentation functions of commonly used Python editors, one can easily create automation scripts with just basic knowledge of a system and its structure.

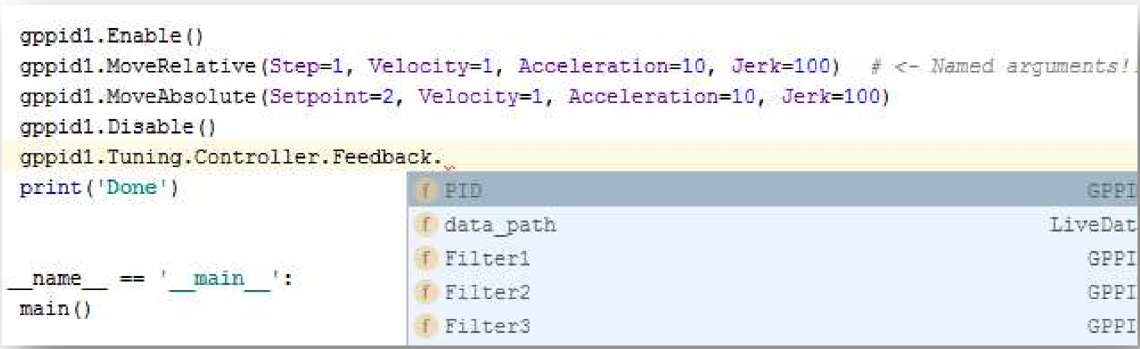


Figure 6 – Easily create machine automation scripts using code-completion and well-structured interfaces

The integrated application development suite simplifies the use of the SAXCS toolkit. The powerful combination of the built-in configurable user interface and the Python automation language provides limitless possibilities for integration, calibration, diagnostics, service, and maintenance of your application. Thanks to the solid platform architecture and generated interfaces, mechatronic engineers can now make what we like to call "quick & *clean*" solutions.

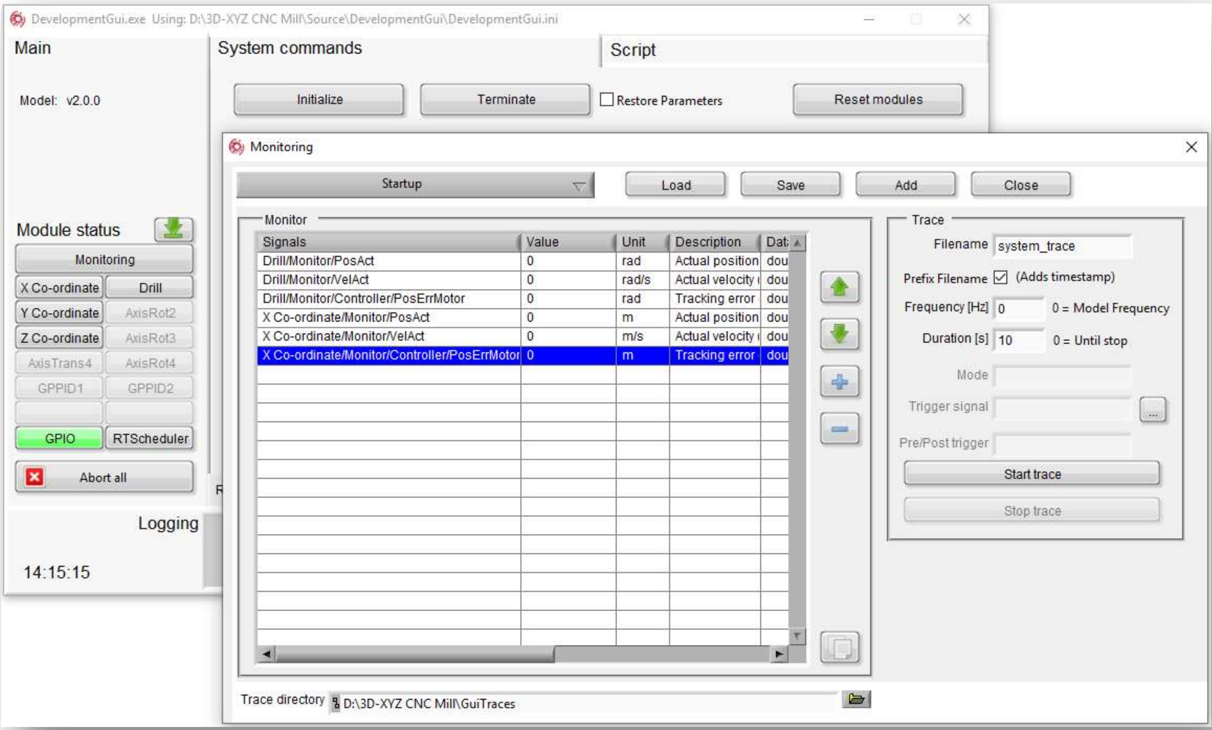
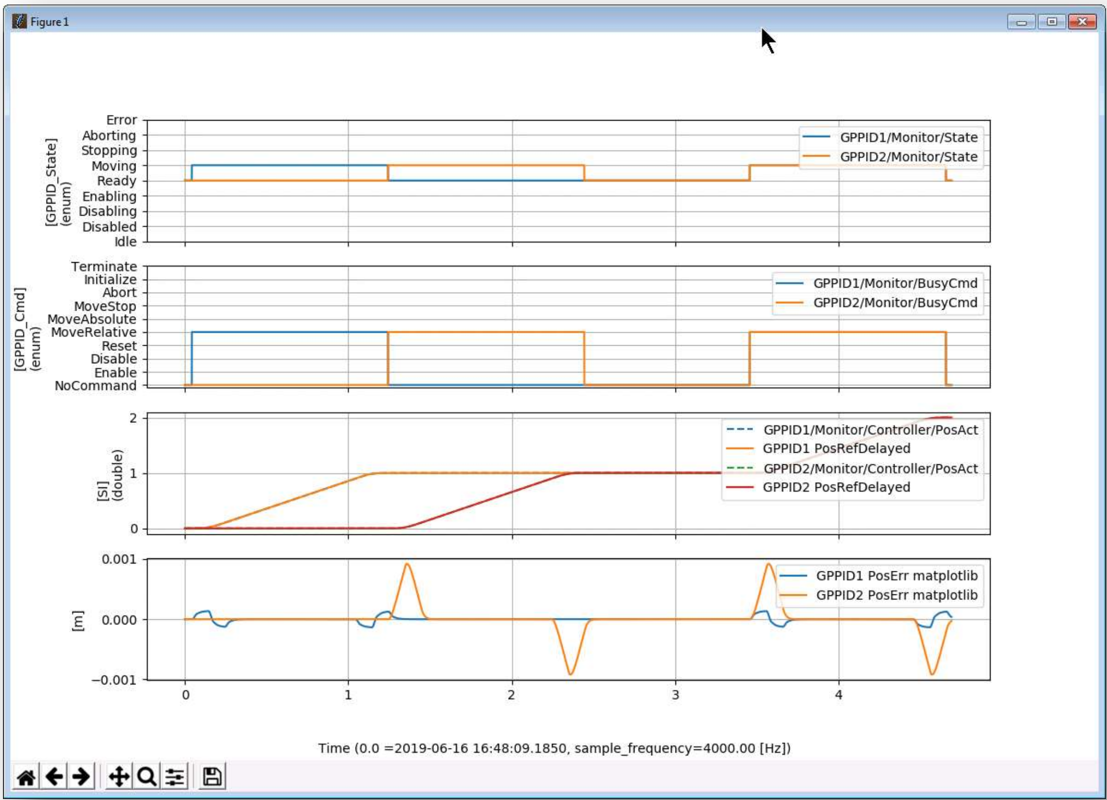


Figure 7 – SAXCS programmable development GUI and Python automation suite



Like explained earlier, all client applications can run simultaneously, including the production software. This allows control and monitoring of any application at any time, from anywhere. Smart data acquisition algorithms can trace and interpret data endlessly while the application is running at full spec. This facilitates predictive maintenance by making use of the continuously running freely programmable background traces and process monitors. In the section "Examples" you'll find evidence of this as part of the Sioux's VEXAR system.

Besides all the powerful tools that come standard with the platform, it is also possible to integrate the SAXCS system directly in your own application. Based on the interface files, API's are generated automatically for most modern programming languages (C++, C#, Python, LabView, Matlab, etc.)

Co-creation with TeamCube®

SAXCS is available for our customers via a secured online portal, called TeamCube® services. It was instantiated as part of www.i-mech.eu and allows our customers to tweak or develop SAXCS control architectures alone or together with the Sioux experts. All necessary tools (and their interconnections) are set and ready to use within a view mouse clicks. An explanatory video can be watched here.

Examples

Vexar is an intelligent substrate transport system that is integrated as a module in printing systems. The steel conveyor belt can transport a wide variety of media at high speed and with high accuracy under process modules such as laser and inkjet systems. High positioning accuracy of 5 microns is achieved by combining a unique patented steering mechanism and advanced motion control strategies. This guarantees the highest quality standards available in the industrial printing industry at 1200 dpi and more. By means of vacuum the substrate is fixed during transport, because of which the print quality becomes almost independent of the mechanical properties of the substrate.

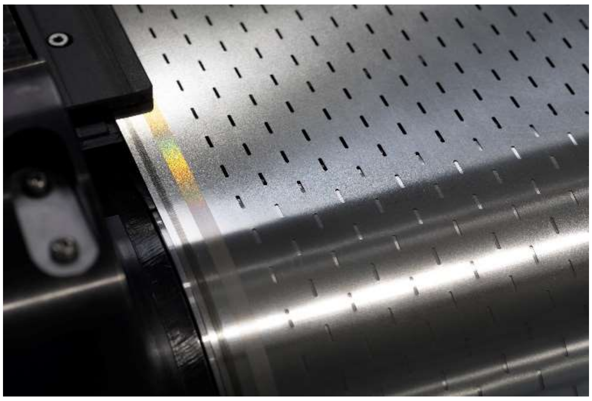


Figure 8 – VEXAR, the world's most intelligent substrate carrier. Made possible with SAXCS

To understand the detailed challenges of this example, consider reading *"Suppressing position-dependent disturbances in repetitive control: With application to a substrate carrier system"* by Noud Mooren, Gert Witvoet, Ibrahim Açan, Joep Kooijman, Tom Oomen.

Use-case 1: Advanced control techniques, in combination with COTS amplifiers

A position based repetitive controller was developed in collaboration with TU/e and successfully tested on VEXAR. The picture below shows the basic diagram of the repetitive controller, which can typically be used as an add-on module to an existing feedback controller.

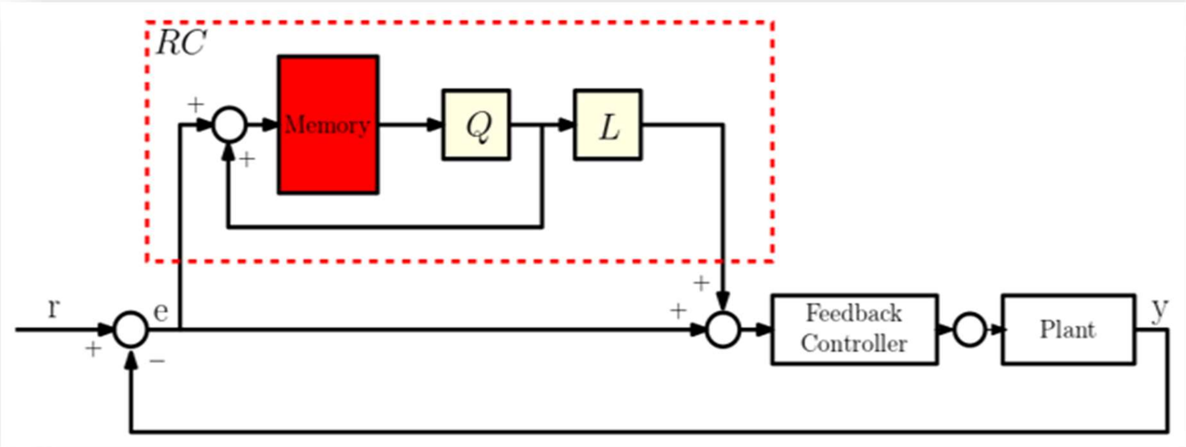


Figure 9 – Position based repetitive control diagram, developed for VEXAR

It effectively learns and suppresses periodic disturbances in the system, resulting in spectacular reduction of the controller tracking error. Even when changing rotation speed, which is typically an issue for existing time-based repetitive controller solutions. The figure below shows the typical behaviour of the repetitive controller, in which it is enabled at the red dotted line and starts learning.

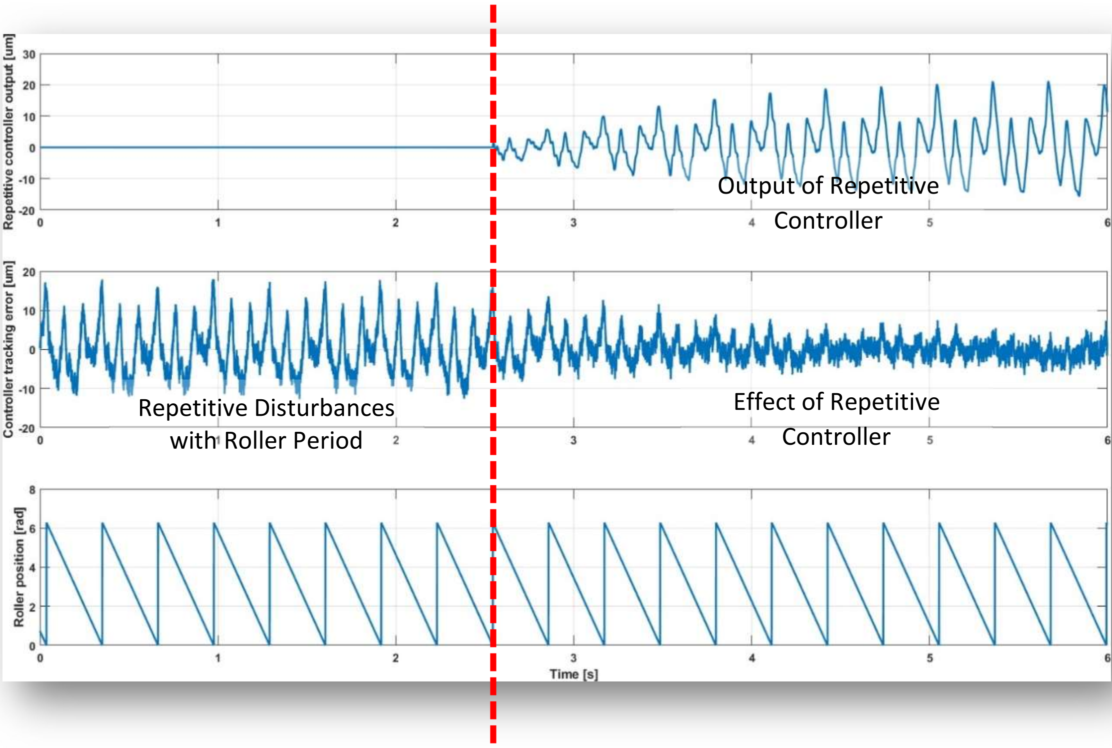


Figure 10 – Example of position based repetitive control behavior and performance

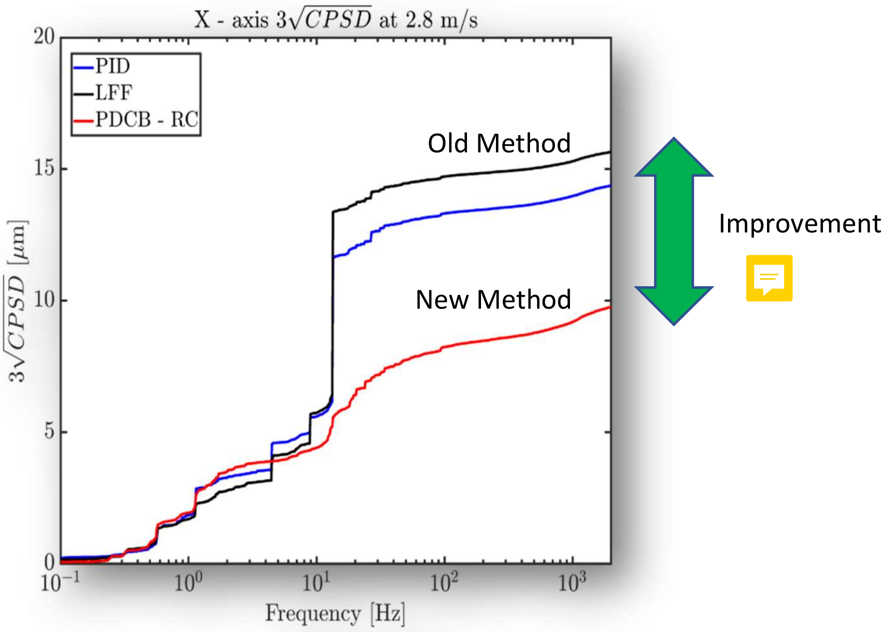


Figure 11 – VEXAR position based repetitive controller performance improvement at 2.8 m/s belt speed

Use-case 2: Reduce integration time by automatic calibration

Although Vexar is a complex system with many sensors and actuators, its integration is not time-consuming. This is facilitated by the versatile Python interface from SAXCS. Complex calibration procedures are scripted and can be executed with one mouse click. This includes a series of operations including data acquisition, post processing, results implementation, verification and logging. Predictive maintenance can be achieved by repeating the calibration procedure and comparing with history data. The calibration can be carried out by technicians instead of engineers since the execution does not require in-depth machine knowledge.

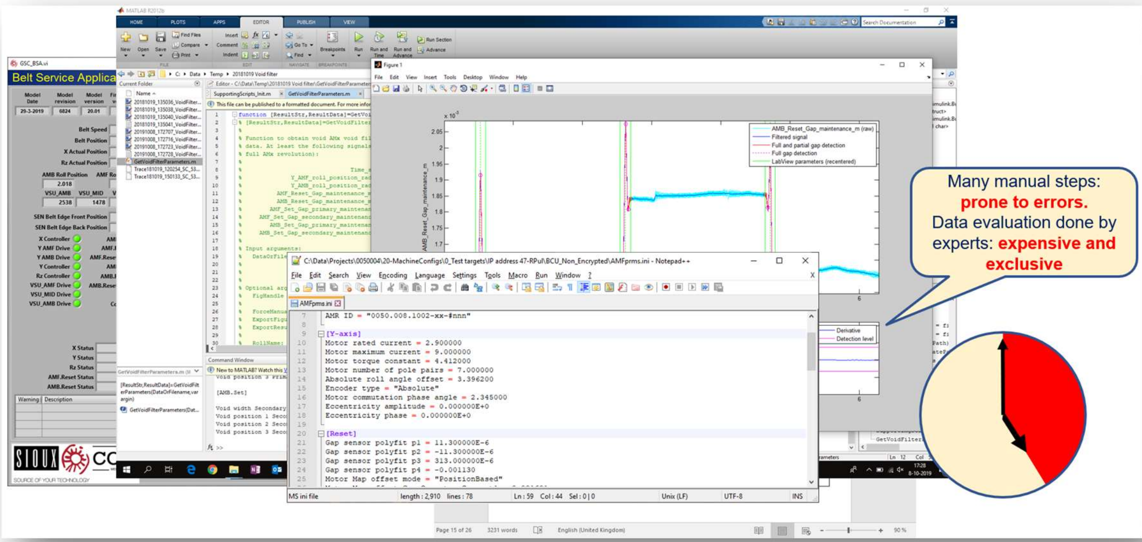


Figure 12 – Former calibration method using lots of different applications and written instructions

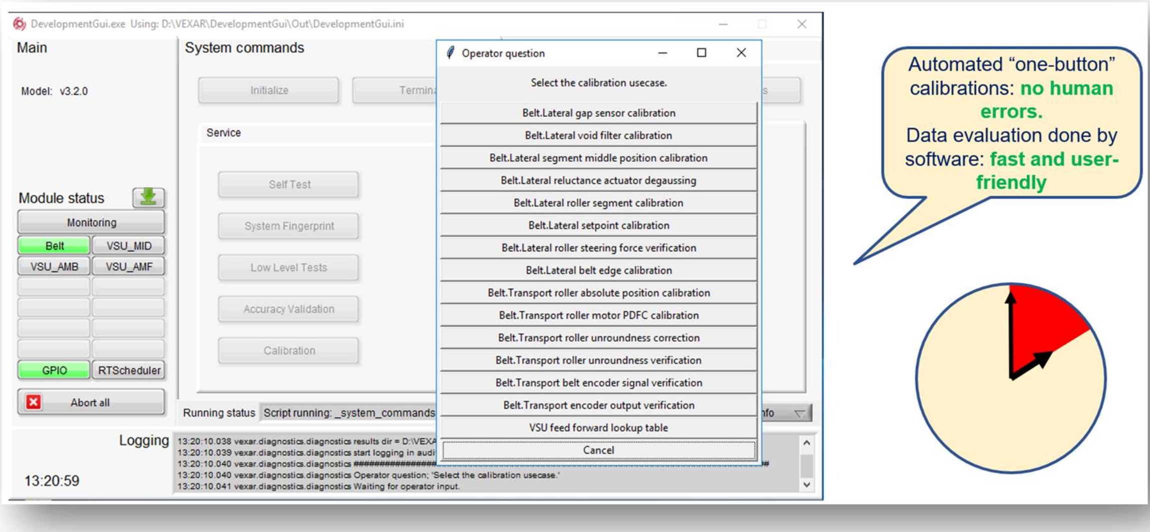
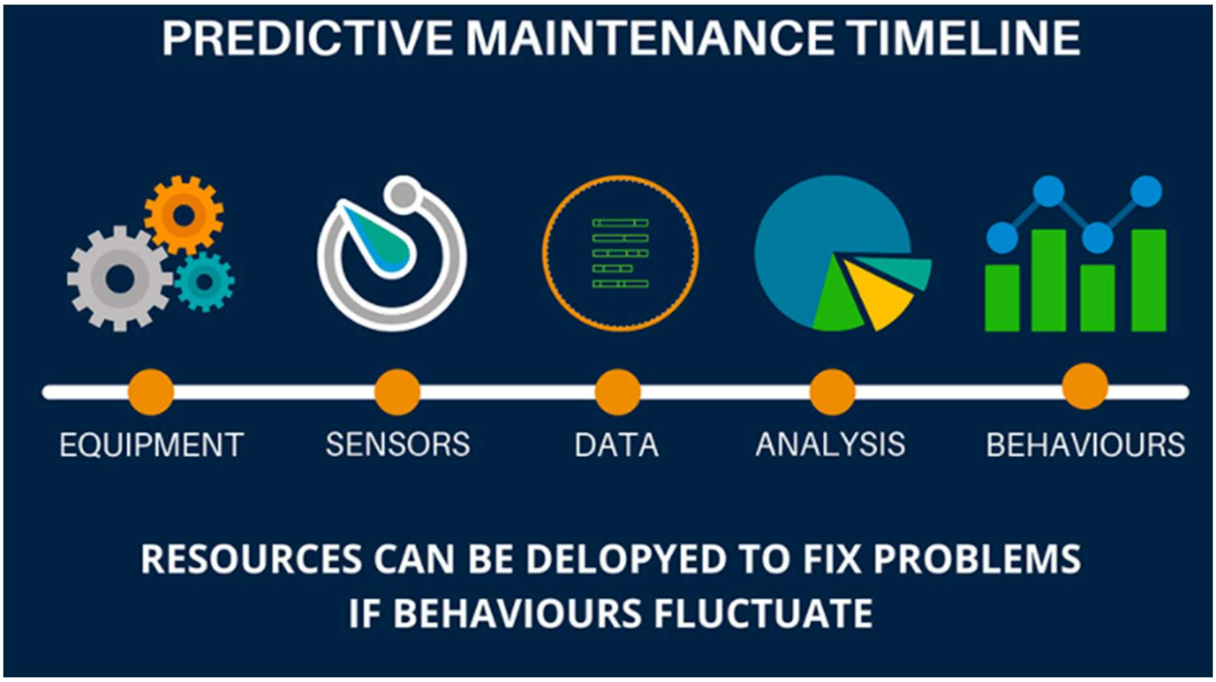


Figure 13 – Integrated SAXCS tools for automated calibration

These auto-calibration routines proved to be so powerful, timesaving and reducing human error to practically zero, that they were directly adopted in the commercial systems. Because of the ease of use of the Python automation language, it was quickly extended to more than 15 auto-calibration routines. Reducing the system integration/calibration time by almost 2 weeks.

Use-case 3: Predictive maintenance facilitated by background monitoring

The programmable "Single Write, Multiple Read" background trace file functionality enables continuous tracing and data viewing/processing in parallel (e.g. for trend analysis). This helps us to study some slow-changing effects, such as friction build up due to contamination. It reduces service intervals by predictive maintenance, short downtime by enabling remote service.



Who we are?

Sioux Technologies wants to contribute to a safe, healthy, and sustainable world through technological innovation. We aim for growth for our employees, our company, and our customers. Sioux creates success and adds value by learning, sharing, interacting, having fun, and working on innovative solutions based on transparency and trust.

Sioux supports high-tech customers in developing and building smart modules and software with impact. We add value by way of our strong focus on quality, functionality, reducing development times, integrity, and scalability. In doing so, we achieve sustainable competitive advantages for our customers at optimal prices.

We believe in flexible and learning organizations, long-term relationships, and the strength of cooperation within a system of open innovation. We operate as our clients' development and production departments or as an extension of their in-house departments. We take responsibility for turnkey projects from the conceptual phase up to and including series production.

Sioux is experienced with various collaborative models but distinguishes itself with its entrepreneurship. We dare to take financial risks in developing promising knowledge and innovative products for our customers. Moreover, we invest in start-ups and the development of new companies that are required for enriching and future-proofing the high-tech ecosystem in which Sioux operates.

Our strength lies in our committed and motivated employees. Sioux's engineers are part of the international top. Their personal development and the continuous enrichment of their knowledge and skills lie at the core of our success and that of our customers. The Sioux company culture is characterized by a sincere interest in people and a collective drive to get the best results every time.

How to contact us?

Sioux Headquarters | The Netherlands
Esp 130
5633 AA Eindhoven
The Netherlands

☎ +31 40 267 71 00
✉ info@sioux.eu
www.sioux.eu

References

[1] www.i-mech.eu (Summary via Projects & results | CORDIS | European Commission)
[2] www.imoco4e.eu
[3] http://www.incose.org
[4] www.MathWorks.com