**RESEARCH**                                                                **Open Access**

# CIRCUS: an autonomous control system for antimatter, atomic and quantum physics experiments

M. Volponi[1,2,3*], S. Huck[1,4*], R. Caravita[3,2], J. Zielinski[5], G. Kornakov[5], G. Kasprowicz[5], D. Nowicka[5], T. Rauschendorfer[1,6], B. Rienäcker[7,8], F. Prelz[9], M. Auzins[10], B. Bergmann[11], P. Burian[11], R.S. Brusa[3,2], A. Camper[12], F. Castelli[9,13], R. Ciuryło[14], G. Consolati[9,15], M. Doser[1], L.T. Glöggler[1], Ł. Graczykowski[5], M. Grosbart[1], F. Guatieri[3,2], N. Gusakova[1,16], F. Gustafsson[1], S. Haider[1], M. Janik[5], G. Khatri[1], Ł. Kłosowski[14], V. Krumins[1,10], L. Lappo[5], A. Linek[14], J. Malamant[1,12], S. Mariazzi[2,3], L. Penasa[3,2], V. Petracek[17], M. Piwiński[14], S. Pospisil[11], L. Povolo[2,3], S. Rangwala[18], B.S. Rawat[7,8], V. Rodin[7,8], O.M. Røhne[12], H. Sandaker[12], P. Smolyanskiy[11], T. Sowiński[19], D. Tefelski[5], T. Vafeiadis[1], C.P. Welsch[7,8], T. Wolz[1], M. Zawada[14] and N. Zurlo[20,21]

*Correspondence:
marco.volponi@cern.ch;
saiva.huck@cern.ch
[1]Physics Department, CERN, 1211 Geneva 23, Switzerland
[2]TIFPA/INFN Trento, via Sommarive 14, 38123 Povo Trento, Italy
Full list of author information is available at the end of the article

## Abstract

A powerful and robust control system is a crucial, often neglected, pillar of any modern, complex physics experiment that requires the management of a multitude of different devices and their precise time synchronisation. The AEḡIS collaboration presents CIRCUS, a novel, autonomous control system optimised for time-critical experiments such as those at CERN's Antiproton Decelerator and, more broadly, in atomic and quantum physics research. Its setup is based on Sinara/ARTIQ and TALOS, integrating the ALPACA analysis pipeline, the last two developed entirely in AEḡIS. It is suitable for strict synchronicity requirements and repeatable, automated operation of experiments, culminating in autonomous parameter optimisation via feedback from real-time data analysis. CIRCUS has been successfully deployed and tested in AEḡIS; being experiment-agnostic and released open-source, other experiments can leverage its capabilities.

**Keywords:** Antimatter; Antihydrogen; Aegis; Gravity; Control system; Automation; Quantum; Atomic; Physics

## 1 Introduction

Control systems are, generally speaking, combinations of hardware and software with the ability to modify the operation and/or configuration of other elements of a system and are in charge of the management of that system. *Autonomous* control systems are such that can operate with little to no human supervision. They are applied in any imaginable field, from satellites to dishwashers. Control systems for nuclear, atomic, and quantum physics experiments are a special category because they need to deal with systems that are continuously upgraded, fixed, and reshaped. For this reason, they need to maintain stability, re-

Springer

liability and reproducibility while allowing for the flexibility necessary for the experiment to mutate.[1] The nature of these experiments puts a range of constraints on the control system: nanosecond-precise execution, multiple computer synchronisation, interfacing with different hardware using multiple interfaces, and easy extendability, among others.

The experiments at CERN's Antiproton Decelerator (AD) complex [1–6], which investigate the asymmetry between matter and antimatter in the universe, are examples for such experiments. They rely on the combination of techniques from photonics, plasma, quantum, nuclear, and particle physics. For example, to be able to manipulate antimatter, it has to be isolated from ordinary matter to avoid annihilation. Antiprotons are typically trapped in ultra-high vacuum inside electromagnetic traps in the form of non-neutral plasmas [7, 8], often sympathetically cooled and manipulated using electrons [9, 10]. In combination with cold positron plasmas [11], they are used to form antihydrogen [12, 13], which can be trapped [14] and probed using techniques such as spectroscopy [15]. Manipulation and preparation of specific quantum states of anti-atoms is currently also explored in several experiments [16].

One of these experiments is AEḡIS (Antimatter Experiment: Gravity, Interferometry, Spectroscopy) [17], whose main aim is to measure the gravitational displacement of a horizontal pulsed beam of antihydrogen ($\overline{H}$) using a moiré deflectometer [18]. The experiment has developed a unique pulsed scheme which is able to provide precise knowledge of the $\overline{H}$ formation time, control the final antihydrogen temperature, and manipulate its excitation state, among others. The formation of antihydrogen is based on the charge-exchange reaction between Rydberg-excited positronium (Ps) atoms and trapped, cold antiprotons from the CERN decelerators [19, 20]. The AEḡIS apparatus [21] comprises two cylindrical cryostats containing superconducting magnets of 5 T and a 1 T, respectively. A Penning-Malmberg trap in the 5 T region is optimised for trapping and cooling antiprotons, while a second trap in the 1 T region is used to form antihydrogen. The axial confinement of charged particles is achieved by the more than 60 electrodes forming the two traps and, to minimise the losses of trapped antiprotons, an ultra-high vacuum of $10^{-13}$ mbar or better is maintained. Additionally to the electrodes, the manipulation of the accumulated particle plasmas and anti-atoms is done with a set of q-switched pulsed lasers, relevant for the excitation of positronium to efficiently produce $\overline{H}$. The apparatus is equipped with a Micro-Channel Plate (MCP) detector at the end of the two cryostats, a two-layer scintillator fibre tracker for detecting the annihilation [22, 23], plastic scintillators [24], and an optical fibre bundle to monitor the light from the lasers. The entry region of the antiproton beam from the AD also serves to bring in positrons from an accumulator, which are then converted to positronium in a dedicated silicon nano-channel target [25–27]. The complexity of the apparatus gives the possibility to investigate different phenomena: for example, attempts to laser-cool positronium atoms are currently ongoing, using the experience of positronium generation and the recently upgraded laser system. The installation of an additional trap for heavy ion generation is also ongoing, which will enable AEḡIS to perform studies on the formation processes of highly-charged antiprotonic heavy ions.

In the initial phase of the experiment, sequences of operations pre-defined by the users and executed by monolithic control systems developed progressively over the years, on top

---

[1]This is different from the demands of the control systems of big observational experiments (such as LHC main experiments, or neutrino telescopes), which are less prone to change.

of a custom-made electronics system with ns synchronisation capabilities, were adequate to successfully produce antihydrogen atoms in a pulsed modality [28]. In the process of establishing antihydrogen formation, however, the limits of this approach became evident: the lack of programming structures to tackle the increasing complexity of experimental sequences; the need of online procedure debugging capabilities; the limited re-usability of the written sequences. In other words, the necessity of an end-user interface providing the features of a standard programming language emerged, although still requiring arbitrary waveform generation and ns synchronisation capabilities to allow complex non-neutral plasma manipulations [10] as well as Ps formation and laser excitation [29].

In fact, as often occurring in complex experiments, the software infrastructure of the AEḡIS apparatus consisted of multiple independent subsystems (e.g. antiproton trap, positron apparatus, laser systems, detectors, etc.), managed by a set of computers running several control programs, all independently written and connected by pre-defined interfaces, which in turn had to be adapted to the changing needs of the experiment. While, with this approach, each single subsystem could be initially developed independently from the others, the performance of coordinated experiments (like antihydrogen production) required a significant human effort to operate the entire system as a whole, as the individual control programs needed constant monitoring during the data-taking periods.

Different examples of control systems for physics experiments exist [30–34], which share most of the concepts expressed above and propose different solutions to the aforementioned problems. Nevertheless, the interfacing capability is often limited, and, furthermore, none of them is envisaged with automation as the main driving force: the possibility of letting a control system run in full autonomy, especially with a feedback loop based on acquired data, relies on layers of self checks and self consistency, which are not straightforward to implement.

Furthermore, the size and complexity of experiments like AEḡIS renders impossible the entire control to be performed by a real-time code residing on a Field-Programmable Gate Array (FPGA). The multitude of interfaces required by the different instruments and the diverse time scales (nanoseconds for time-critical operations, minutes for an entire measurement sequence) cannot be provided by such a solution.

For these reasons, the AEḡIS collaboration has designed a generalised experiment control system that is customisable to individual experiments' specific requirements. This flexibility benefits the AEḡIS experiment (as it allows it to evolve smoothly to changing requirements), but equally importantly, the system was constructed with the needs of the much wider atomic and quantum physics community in mind. This control system incorporates a programmable end-user interface, providing advanced synchronisation, watchdog, error management, and online decision making features, re-enforced by an active feedback loop from the acquired data. This re-design specifically targeted complexity reduction of experimental procedures by standardising established sub-procedures into libraries, and by increasing stability, reliability, and autonomy. With this as the baseline, the subsequent implementation of increasing layers of automation and autonomy becomes feasible, strengthening the trust in the system by cycles of campaigns of implementation and debugging.

The implemented solution merges the capabilities of a real-time code with a distributed slow-control system that unifies the computer in a single entity and brings together all the features described above, so as to partially remove the operators' need to control the

running procedures. The control system itself is completely experiment-agnostic (technically, it could be used to control experiments outside the realm of physics as well), and it is released open-source so that other experiments can profit from the effort.

The high level of automation is a choice motivated further by the upgrade of the AD to the new Extra Low ENergy Antiproton (ELENA) ring [35]. ELENA is a small synchrotron with a 30 m circumference, used to further decelerate AD antiprotons from 5.3 MeV down to 100 keV and finally transfer them to the experiments present at the AD. This results in an increase of one to two orders of magnitude in the trapping capabilities of the experiments. With ELENA, the operation scheme and the share of the $\overline{p}$ beam has changed from experiment-specific allocated time slots of 8 hours to shared access and continuous 24/7 operation, increasing the shift personnel needs by a factor of three.

In this article, the new control system is presented, with the specific implementation in the AEḡIS experiment given as an example. It was designed around the Sinara/ARTIQ open hardware/software platform [36, 37], embedded within a LabVIEW™ [38]-based control framework called TALOS, providing the asynchronous high-level functionalities. The creation of experimental hardware procedures is done in the ARTIQ programming language (based on Python), which allows for ns-synchronous operation scheduling on the Sinara hardware. The new control system has been used in AEḡIS antiproton campaigns with ELENA and proved to be autonomous and reliable, while facilitating fast development of experimental procedures with version control, structured debugging, and agile development.

The article is structured as follows: general requirements imposed by scientific goals are outlined in Sect. 2.1. The new electronics setup is depicted in Sect. 2.2, explaining the functionalities of the Sinara ecosystem. The overall software control system is then introduced in the following Sect. 2.3, encompassing ARTIQ, the library for programming Sinara, and TALOS, the modular distributed slow-control system. An overview of the AEḡIS Data Acquisition System is offered in Sect. 2.4, as an example case. Similarly, in Sect. 2.5, the online and offline analysis system is shown, with feedback capabilities on the control system; a successful example of this application is described in Sect. 2.6. Subsequently, the CIRCUS (Computer Interface for Reliably Controlling, in an Unsupervised manner, Scientific experiments) control system validation is presented in Sect. 3. Last, the performance of the new setup is evaluated and foreseen future implementations are outlined.

## 2  Methods

### 2.1  Requirements for the autonomous control system

The design of the control system is driven by the requirements that this class of experiments has. A review of the literature was performed, to take some examples of atomic and quantum experiments [39–43], and relate their requirements to the ones derived from the experience of realising the first pulsed source of antihydrogen in AEḡIS [28]. The comparison showed that this class of experiments share similar requirements, which can be subdivided into four categories: interface requirements with the particle source; trap operations; particle and laser synchronisation; general slow control, data acquisition (DAQ) and networking.

Therefore, we decided to use the AEḡIS requirements as a base for the design of the control system: the generality of these requirements renders a system satisfying them applicable to a broad range of tasks. In the following, their rationale is exposed, and the requirements are then summarised in Table 1.

**Table 1** Summary of the different technical requirements set on the control system from experiments needs

| Particle source interfacing | |
| --- | --- |
| High-voltage catching potential range | 0–15 kV |
| High-voltage potential accuracy | <10 V |
| High-voltage gate raising edge duration | <100 ns |
| High-voltage gate temporal tuning accuracy | <10 ns |
| AD injection trigger synchronization | <5 s |
| ELENA injection trigger synchronization | <1 s |
| Bunch pre-arrival trigger synchronization | <1 $\mu$s |
| Bunch arrival trigger synchronization | <10 ns |
| **Trapped particle manipulation** | |
| Low-voltage potential range | $\pm$200 V |
| Low-voltage potential ramps duration | 100 $\mu$s–10 s |
| Maximum absolute calibration error | <5 mV |
| Maximum noise band amplitude | <1 mV$_{rms}$ |
| Fast pulse settling time | <30 ns |
| Fast pulse duration range | 100ns–100 $\mu$s |
| Fast pulse timing adjustment | <10 ns |
| Fast pulse timing jitter | <1 ns |
| RF frequency range | 1 kHz–100 MHz |
| RF amplitude range | 10 mV–5 V |
| **Particle and laser synchronisation** | |
| Positron/laser triggers time adjustment | <1 ns |
| Positron/laser trigger jitter | <0.5 ns |
| Detector arming timing accuracy | <100 $\mu$s |
| Detector triggering timing accuracy | <10 ns |
| **Slow control, DAQ and networking** | |
| PC–PC message delay | <100 $\mu$s |
| Real-time–PC message delay | <10 $\mu$s |
| DAQ run start/stop time | <10 s |
| Data availability for online analysis | <5 s |

*Requirements of the particle source interface:* AEḡIS obtains the antiprotons in bunches from the AD–ELENA complex. Consequently, the experiment is synchronised to the decelerator stack via a set of hardware triggers occurring at different times during each ≈120 s antiproton cycle: the AD injection trigger (occurring at the beginning of the cycle), the AD extraction/ELENA injection trigger (occurring ≈20 $\mu$s before antiproton delivery), a bunch pre-arrival trigger (occurring ≈20 $\mu$s before antiproton extraction from ELENA) and a bunch arrival trigger (synchronous with the extraction from ELENA). The bunch is approximately 150 ns (FWHM) long. Antiprotons are delivered to the experiment at 100 keV energy, which is further reduced by a thin foil (ca. 1500 nm of kapton) to about 10 keV . Antiprotons are subsequently caught by means of a pulsed high-voltage Malmberg-Penning trap operated up to 15 kV in a 5 T magnetic field. The timing of the trap has to be fine-tuned in ≈10 ns steps.

*Requirements for trapped particle manipulations:* a typical antihydrogen production sequence involves several manipulations steps of trapped particles (in the form of nonneutral plasmas), performed with low-voltage electrodes of the Malmberg-Penning trap in the 1 T region. These have to be controlled in the ±200 V range, by arbitrary function generators. An accuracy of 10 mV or better is needed to allow for accurate potential ramps

and thus enable measurements of the plasma space charge and temperature [44] as well as evaporative [45] and adiabatic cooling [46]. Standard manipulations in traps include both slow (several seconds) and fast (less than a millisecond) ramps, fast extraction of particles with ≈100 ns (≈100 $\mu$s) pulses for electron (antiproton) extraction respectively, as well as application of radiofrequencies (RF) in the 1 kHz–100 MHz range for plasma heating or cooling and density control with the Rotating Wall technique [10]. Often, these procedures are combined, and the ability to synchronise events with the accuracy of 1 ns over several hours is required.

*Requirements of particle and laser synchronisation:* antihydrogen formation via charge-exchange reactions with trapped antiprotons requires the control of the times of positronium formation and laser excitation to its Rydberg levels at the ns accuracy level, as well as triggering the diagnostic scintillation and Micro-Channel Plate (MCP) detectors, as detailed in [29, 47]. This is due to the fact that the excitation laser has to be carefully synchronised according to its beam shape and position to obtain efficient positronium excitation. Hardware trigger lines allowing time adjustment of 1 ns or better and jitters of <0.5 ns are required.

*Slow control, DAQ and networking requirements:* these include all the procedural sequences of trap initialisation, synchronisation on slow scales, computer responsiveness, data upload to the Data Acquisition System, etc., which admit a considerable jitter between the moment the command is issued and its execution, and must not interfere with the experimental sequence (typically in the order of 100 $\mu$s). Network communication has to guarantee a smooth control flow: the communication speed among the various machines needs to be at least an order of magnitude faster than the timescale of PC operations.

## 2.2 The control system hardware

For atomic and quantum physics experiments, the necessity to operate (parts of) the measurements with ns-precision is fundamental (as seen in 2.1). Hence, the control system electronics have a pivoting role in reaching the scientific objectives.

In AEḡIS, the main components of the control system electronics belong to the Sinara [36] ecosystem. Sinara features a versatile, open-source hardware portfolio which was originally developed for quantum information experiments utilising the ARTIQ control software [37] (see Sect. 2.3). The Sinara hardware provides compact, modular, reproducible and reliable electronics capable of controlling intricate, time-critical experiments. It is particularly optimised for experimental setups which are limited in space, as is the case inside the AD, and, thanks to its standardised and modular nature, assures the long-term maintainability of the control system.

While Sinara was chosen for the above reasons and is easily applicable to a multitude of very different procedures in quantum and atomic physics experiments, ARTIQ can be used in combination with hardware and peripherals from other manufacturers capable of nanosecond timing as well, if controlled by a dedicated FPGA.

As shown in Fig. 1, the hardware of the AEḡIS trap control system is organised in three rack-standard Eurocard 84 HP electronics crates with dimensions of 50 × 20 × 35 cm, which allow to connect a variety of modules.

The main controller is called Kasli (see Fig. 1). It comprises an Artix-7 Field-Programmable Gate Array (FPGA) and can be used as a stand-alone core device or in combination with additional carriers as a repeater or satellite of DRTIO (Distributed Real Time

**Figure 1** Photograph of one of three fully equipped Sinara electronics crates of the AEḡIS trap control system, including (from left to right) power module, Kasli carrier, digital I/O units, Fastino DAC, and four high-voltage amplifier boards

Input/Output) communication through optical fibre links, facilitating a stable, high-speed Gbps transfer of time and data information between the devices. This second option allows for a fast propagation of both a clock signal (internally generated or externally connected) and the control communication between controllers, thus offering straightforward adaptations and extensions of the experiment. Software communication with the Sinara electronics is facilitated via Kasli's high-speed Gigabit Ethernet port. Each Kasli is capable of controlling up to twelve extension modules with various purposes.
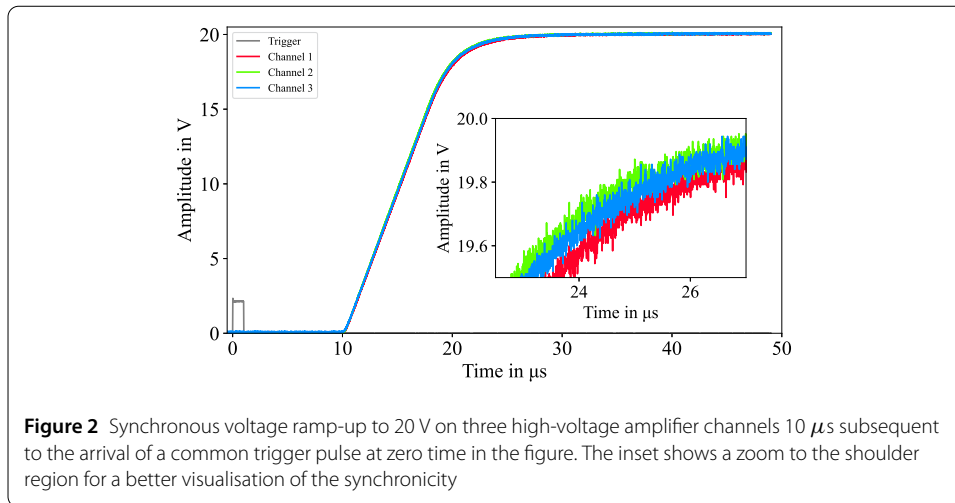
Each Sinara crate used in AEḡIS contains a Kasli carrier combined with digital I/O units and fast DAC modules, called Fastino, from the Sinara repertoire, as well as 1 MHz high-voltage amplifiers, which have been custom-designed for the requirements of the AEḡIS experiment.

The digital I/O cards are used for the reception and provision of high-speed ns TTL trigger signals between the sub-systems of the entire experimental setup. 16 MCX connectors are compactly arranged on each single, thin module and their direction of input or output can be individually configured in batches of four.

Each Fastino provides simultaneous 3 Mbits$^{-1}$ digital-to-analog conversion for 32 channels, yielding stable output voltages in the range of $\pm 10$ V with a 16 bit resolution. The Fastino DAC channels can either be used directly to supply low voltages in this range or be connected in batches of eight to the high-voltage amplifier modules.

One such amplifier unit comprises eight channels, each of which is capable of a 20-fold amplification of the output voltage of one Fastino channel respectively, i.e. yielding voltages of up to $\pm 200$ V. The high-voltage amplifiers are equipped with individual OptoMOS® relays, allowing to isolate the outputs and prevent the noise from the amplifiers from propagating to the connected load. The main control electronics of the AEḡIS setup are formed by three of the described Sinara crates: two (one Kasli acting as DRTIO repeater, the other as satellite) provide the high-voltage output channels for the synchronous potential steering of the electrodes of the 5 T trap region of the experiment, required for antiproton capture and electron cooling, while the third crate is used for the control of the 1 T antihydrogen production trap electrodes.

During the ELENA/AD antiproton run campaigns, the fast digital I/O units have demonstrated reliable acquisition and processing of the incoming trigger signals, essen-

**Figure 2** Synchronous voltage ramp-up to 20 V on three high-voltage amplifier channels 10 $\mu$s subsequent to the arrival of a common trigger pulse at zero time in the figure. The inset shows a zoom to the shoulder region for a better visualisation of the synchronicity

tially enabling the steering of the trap potentials with the required timing for the capture of antiprotons.

In Fig. 2, examples of output signals of three HV amplifier channels are shown. They are produced by sending an external trigger pulse to the digital I/O unit and subsequently setting a voltage on three of the Fastino channels. The voltage is amplified by the connected amplifier units. The final output is recorded using an oscilloscope and read out via LabVIEW™. The Sinara system is thus found to be able to satisfy the timing requirements of the AEḡIS experiment: reactions to triggers on the microsecond scale and synchronous control of the output channel voltages are reliably provided.

All amplifier channels have been calibrated individually together with their corresponding Fastino DAC channels to reliably provide the required voltage despite their different offset and voltage precision step values. With this calibration, each channel voltage can be set with an accuracy of few mV, which is comparable to the precision of the 6 mV steps generated by the 16 bit nature of the Fastino. The calibration procedure is described in the Appendix.

In addition to the electronics controlling the trap system and providing inter-system triggers, two additional Sinara crates have been successfully commissioned to run the laser system and provide synchronisation between the two involved lasers despite their difference in frequency (see Sect. 3). In order to be able to do this, the 1 T Kasli, in addition to controlling the respective trap system, is used as the master core for two satellite Kasli devices, both of which control a digital I/O card with BNC output for triggering the sequences needed for laser operation.[2] Furthermore, the new control electronics have been successfully integrated as part of the AEḡIS positron system to provide triggers for the positron preparation sequence and synchronize it to the rest of the experiment. Further extensions of the control infrastructure, e.g. dedicated Sinara crates for the positron system and to future-proof the use of the Rotating Wall technique for plasma compression, are ongoing.

---

[2]The BNC digital I/O units work in the same way as the MCX units except for comprising only eight channels instead of 16.

The Sinara hardware is a central component in the new AEḡIS control system, which drives all integral parts of the experiment. The software will be presented in the following section.

There are two relevant additional electronics components, which have been integrated in the new control system setup and are fully steerable programmatically. The first is a pulser device which provides ns-synchronised pulses of variable length to the electrodes, with tunable amplitude in the voltage range provided by the Sinara Fastino plus amplifier channels. The trigger signals for this pulser are given by the Sinara digital I/O units, while the amplitude is determined by internal DAC units. The inclusion of this functionality is vital for the efficient and fast transport of particles between the different trapping regions inside the experiment. The second component is a waveform synthesizer with multiple channels, which can be used to add phase-shifted sinusoidal signals of up to 5 V in a frequency range of 0 to 30 MHz to the so-called sectorised electrodes. These electrodes are separated into four sectors around their centre, i.e. around the central axis of the trap. By applying the sinusoidal signals with a phase shift between the four sectors, it is possible to employ the Rotating Wall (RW) technique for a manipulation of the radial dimension of the contained particle plasma. This component is also currently operated by the new control system.
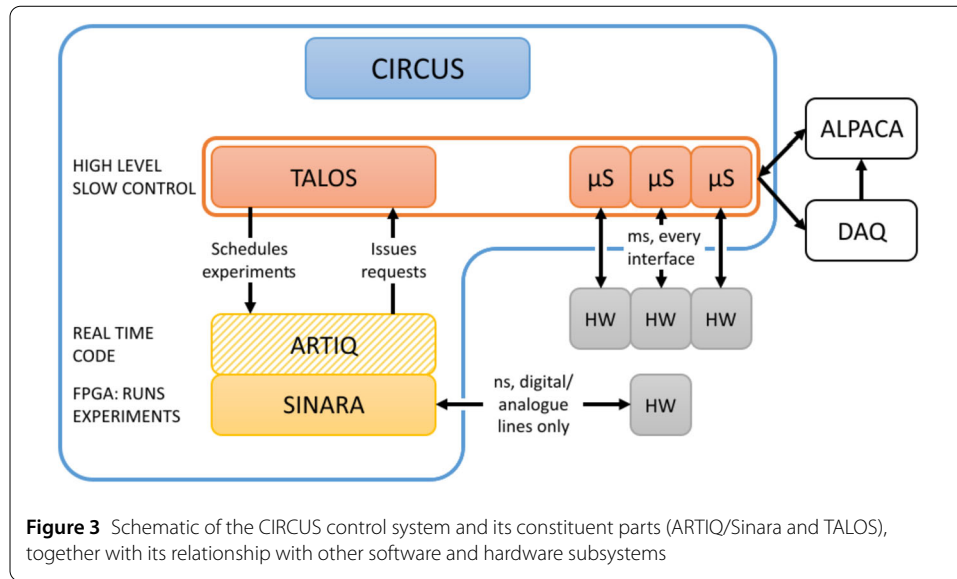
### 2.3 The control system software

While the CIRCUS heavily relies on the Sinara hardware to perform its operations, its core part is the software infrastructure. As introduced in Sect. 1, it consists of two parts: ARTIQ and TALOS (both presented in greater details in the two following sections). ARTIQ is the high-level programming language for scripting the ns-precise routines to be executed by Kasli, that we empowered with libraries to streamline the coding of experimental routines and to integrate its operations with TALOS. In principle, the Sinara/ARTIQ structure could be integrated in different overall control system structures as well. In contrast, TALOS is the framework that constitute the slow-control: it both provides the interface between the operators and the apparatus, and its flexibility makes it compatible with any range of hardware and control software units independent of their precise characteristics.

It is in the interplay of this ns-precise hardware control on the one hand and its full integration and automation of the surrounding experiment on the other hand that the presented control system, CIRCUS, manifests its strength in such a way that it can be applied to any experiment with similar requirements.

This interplay is evident especially when it comes to executing a sequence of measurements. In fact, the schedule of scripts (with parameters) is defined using the apposite TALOS interface, and when the schedule is launched, it is TALOS that assesses if the conditions for running the experiment are met. If positive, it passes the command to Sinara, which executes the script, and TALOS remains available to forward calls from the used ARTIQ/Python script to any part of the experimental apparatus. When the script terminates, the command passes back to TALOS, which, based on the outcome of the script, decides what action is to be taken – most of the time, running the same or the subsequent script in the schedule.

In Fig. 3 the schematic of the CIRCUS control system is given, outlining the relationship of its constituent parts and their functionality, together with the connection with the other software and hardware components of AEḡIS.

**Figure 3** Schematic of the CIRCUS control system and its constituent parts (ARTIQ/Sinara and TALOS), together with its relationship with other software and hardware subsystems

The CIRCUS control system is available open-source in a *git* repository (doi: 10.5281/zenodo.10371799)

In the following, both ARTIQ and TALOS are explained in greater detail.

*ARTIQ*    As explained in Sect. 2.2, the Sinara hardware relies upon the ARTIQ (Advanced Real-Time Infrastructure for Quantum physics) [37] language for a straightforward, reliable software control. ARTIQ is a Python-based, high-level programming language which supplies specialised pre-generated functions for communicating with the hardware. The resulting control routines are formed by clear and short run scripts, preventing long familiarisation phases of semi-experienced programmers and allowing for quick adaptations during data taking.

ARTIQ is designed to script experiments with nanosecond resolution and microsecond latency. To meet the requirements of real-time programming, ARTIQ code consists of two parts which can interact with one another: the first one, composed of regular Python commands, is executed on the host, while the *ARTIQ kernel* is executed on the CPU of the core device. This CPU can directly access a part of the "gateware"[3] responsible for specialised programmable I/O timing logic. A timeline, constituted by all programmed input and output events, keeps the synchronisation of the experimental routines: output events with a given timestamp are executed in a first-in-first-out mode when matching an internal, high-resolution clock, and input events are recorded with a stamp for the current clock value.

The ARTIQ environment includes a dedicated function to observe a given I/O TTL channel and register rising or falling edge events for a specified duration. A sequence of actions can then be performed within a deterministically programmed time window after receiving a trigger, one example of this being another ARTIQ function designed to set a specified voltage on a given Fastino channel. In order to control multiple different trap electrodes in a synchronous way, the use of the provided Direct Memory Access (DMA)

---

[3]By gateware is meant the code specifying the configuration of the digital logic gates of an FPGA.

```
from artiq.experiment import *                          import sys
from artiq.coredevice.kasli_i2c import port_mapping     sys.path.insert(1, 'C:\kasli-code\Libraries')
                                                        from AEgIS_imports import *
class HVAamp_Trigger(EnvExperiment):                    from AegisExperiment import _AegisExpOfficial

    def build(self):                                    class HVAamp_Trigger(_AegisExpOfficial):
        self.setattr_device("core")
        self.setattr_device("fastino0")                     def build(self):
        self.setattr_device("ttl0")                             self.Build()
        self.setattr_device("ttl_hvamp0_sw1")
        self.setattr_device("ttl_hvamp0_sw2")               def run(self):
        self.setattr_device("ttl_hvamp0_sw3")                   self.Init()
        self.setattr_device("dio_mcx_dir_switch")               self.SetVoltagesAtTrigger("Trigger", 10*us,
        self.setattr_device("i2c_switch0")                          "Channel1", 20.0, "Channel2", 20.0,
        self.dio_mmcx_i2c_port = port_mapping["EEM0"]               "Channel3", 20.0)

    @kernel
    def set_dio_outputs(self):
        self.i2c_switch0.set(self.dio_mmcx_i2c_port)
        self.dio_mcx_dir_switch.set(0b00000001)
        self.core.break_realtime()
        self.ttl0.input()
        self.core.break_realtime()

    @kernel
    def SignalAtTrigger(self):
        t_gate = self.ttl0.gate_rising(120*s)
        t_trig = self.ttl0.timestamp_mu(t_gate)
        at_mu(t_trig)
        delay(10*us)
        self.fastino0.update(1<<3|1<<2|1<<1)

    @kernel
    def SetVoltages(self):
        self.fastino0.set_dac(1, 1.0)
        self.core.break_realtime()
        self.fastino0.set_dac(2, 1.0)
        self.core.break_realtime()
        self.fastino0.set_dac(3, 1.0)
        self.core.break_realtime()
        self.SignalAtTrigger()

    @kernel
    def run(self):
        self.core.reset()
        self.fastino0.init()
        self.core.break_realtime()
        self.fastino0.set_hold(1<<3|1<<2|1<<1)
        self.SetVoltages()
```

**Figure 4** Left: Experimental routine to set a specified output voltage on three amplifier channels of the Sinara hardware system after an incoming trigger pulse, programmed in the ARTIQ environment. Right: The same experimental routine as on the left, reduced to a few lines of code when implementing library-based programming

is essential, as it allows to pre-define RTIO sequences in the Kaslis' SDRAM, which can then be run directly by the FPGA core.

ARTIQ allows for a library-based approach to programming run routines of an experiment. To simplify and standardise the procedure for creating run scripts, an experiment parent class has been developed. All routines inherit from this main class, which contains both the code for initialisation and configuration of the hardware, and function libraries for interacting with the hardware and trigger signals, whose constituents can be called from the scripts defining the different experimental routines. The effect of the outlined library-based approach can be observed in Fig. 4, which shows a very simple experimental routine. In both cases, the resulting sequence is the same: the system waits for an incoming trigger signal on one of the digital I/O lines and subsequently produces a voltage ramp to 1 V on three of the Fastino channels (which is amplified to 20 V by the corresponding amplifier units). The application of the calibration constants for the amplifier boards described in the Appendix is included in the function used in the routine on the right. The functionality to set up and initialise the used hardware, which is part of the first two function definitions on the left, is included in the standardised *Build* and *Init* functions on the right. All other functions defined explicitly in the script on the left are included in the

**Figure 5** Schematic of the ARTIQ/Python library structure of CIRCUS, as used in AEḡIS. Each library defines a class, which all the experimental scripts of AEḡIS inherit from. Most of the functions defined in the top-level libraries (*TCP*, *Build & Init*, *Utility* and *Analysis* libraries) are generic and could be utilised by other experiments as well
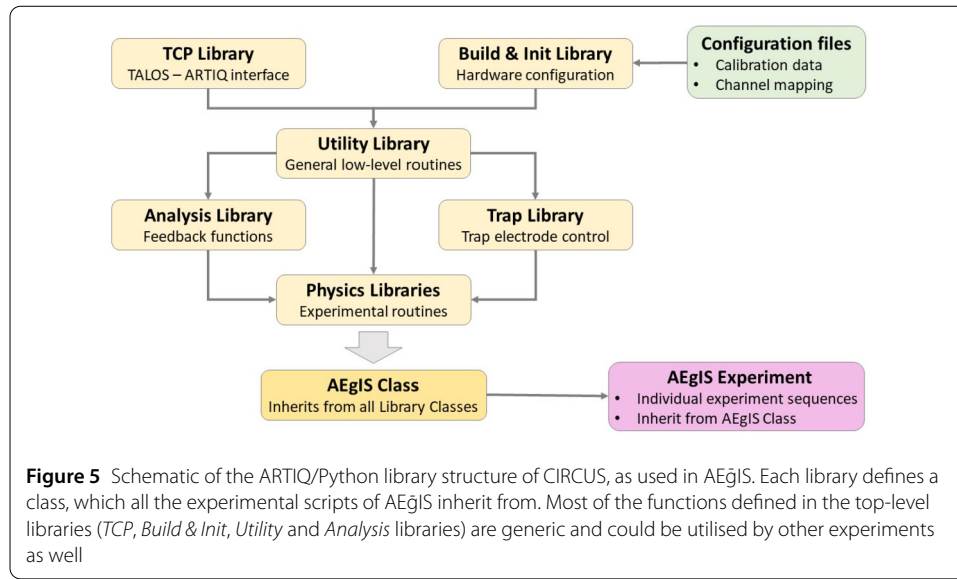
library structure and available without re-definition to all experiment scripts. This means that only one additional function call is needed in the actual run routine shown on the right side to achieve the same result as the code on the left.

The use of the AEḡIS library system reduces the ARTIQ script to a few lines of code when importing the parent classes and yields an immediate, simple overview of the routine. This effect is rendered more and more distinct the more complex (and closer to realistic run sequences) the experimental routines become.

In particular, a Python library, called the *TCP Library*, has been created to organise the interface with the TALOS part of the control system infrastructure, containing the functions that ensure the communication between them. The TALOS system underwent an in-depth test during the antiproton run, exhibiting reliable interaction with the Sinara/AR-TIQ setup.

Figure 5 shows the library structure developed in ARTIQ/Python code that is used as the basis of the hardware communication of the presented control system. Each shown library is formed by a class, which the AEgIS Class, i.e. the parent class of the experimental scripts, inherits from. As shown in the schematic, the higher-level libraries use functions of the base classes. The actual run routines are then sub-classes of the AEgIS Class and have all library methods available. Of course, several of the functions, particularly in the lower, experimental libraries, are specific to the AEḡIS experiment and would need to be replaced by corresponding functionalities in other environments. On the other hand, the base functions in the *TCP Library*, used to interface with TALOS, as well as the standard routines to configure and initialise the used hardware (with adapted configurations) and those general functions related to timing synchronisation, information logging, and data retrieval in the *Utility Library* and *Analysis Library* are re-usable as general functionalities of CIRCUS.

*TALOS*   TALOS (Total Automation of LabVIEW™ Operations for Science) is a control system framework[4] that unifies all the computers of an experiment into a coherent, coordinated, distributed system, and it increases the reliability and stability of the running apparatus by means of a (distributed) watchdog structure, with the ultimate goal to safely leave it running unsupervised for extended periods of time.

It is founded on two concepts: the "everything is a μService[5]" approach, and the distributed architecture. To satisfy both requirements, it was decided to base TALOS on the Actor Model [48], which is an information theory model specifically designed for the implementation of large distributed system architectures. The theory is based on the concept of *actors*, single entities that can react to a message arriving from another actor by executing a local action, sending further messages to other actors, changing its internal status, creating additional actors, or a combination of the above.

The first concept, "everything is a μService", consists of the division of the code into independent, autonomous parts, the μServices, each with a defined scope and task. Each μService runs separately from the other μServices, in a completely asynchronous way, communicating among each other via a built-in messaging system. This design choice makes the system both easily extendable and debuggable in a straightforward way, while also minimising system downtime: in fact, every μService can be separately tested before being deployed, and any problem can be readily isolated and solved.

The second concept, the distributed architecture, manifests in multiple instances of the same actor, called *Guardian*, taking the role of root actor, one on every computer. This Guardian has the function of monitoring both the status of all μServices running locally, all implemented as independent actors, and the status of the other active Guardians in the network. At the same time, the Guardians provide a common infrastructure to share messages and data between various μServices and among different computers. This new paradigm has a twofold result: it strengthens the reliability, the safety, and the stability of the system through a distributed watchdog system (in fact, no computer or program can become unresponsive without it being noticed), and it unifies all the computers into a single, distributed entity. The latter is what facilitates the full automation of the experimental procedures, as high-level decisions often depend on parameters generated by multiple computers.

The choice to base this new framework on LabVIEW™ (by NI[6]) was dictated mainly from the fact that an implementation of the Actor Model is present in LabVIEW™, called *NI Actor Framework*, which provides a readily available foundation block. Moreover, in AEḡIS, as in many other experiments, some fundamental hardware components are from NI, and therefore natively interfaced in LabVIEW™, simplifying μServices coding.

Some μServices developed with TALOS are of general use, independent of the AEḡIS experiment: CIRCUS comes with them integrated, so to be readily utilised by other experiments. Aside the μServices managing the communications with the FPGA (more below) and parts of TALOS internal mechanics itself, some good examples are: the *Error Manager*, which serves as a the single concentrated point for all the errors of the distributed

---

[4]We refer to TALOS as a *framework* because it does not only come with the functionalities described in this section, but it also creates a specific way of coding, in the form of guidelines to write μServices..

[5]Read *MicroService.*

[6]Formerly *National Instruments.*

system; the *Scheduler*, used as an interface for the user to define sequences of experimental scripts, each with specific parameters; the *Monkey*, which executes the scripts in the schedule and takes the high-level decisions at the core of the automation of the control system (such as retrying a script if it did not run correctly, or modifying the parameters based on the feedback from the analysis system); and the *Tamer*, used to coordinate the parallel execution of multiple *Monkeys*, in case multiple Sinara/Kasli crates need to be managed simultaneously. Furthermore, a standardised Graphical User Interface (GUI) is provided to the user, which is shown in Fig. 6.

As stated before, for the seamless functioning of the CIRCUS control system, a critical part of TALOS is the interface with Kasli. In fact, naturally, Kasli is managed by a user via a command-line interface from a terminal, and communication with external hardware is foreseen to happen only via its digital lines. In more complex experiments like AEḡIS, though, Kasli needs to communicate a huge variety of messages towards multiple different systems, in order to keep all the hardware operations synchronous, and this is impossible to be realised via physical digital lines, the more so because the messages often carry a non-trivial data structure. TALOS, in this respect, provides an interface to the FPGA to extend its capabilities: thanks to a direct TCP (Transmission Control Protocol) [49] communication between Kasli and two dedicated μServices, the FPGA can send (and receive) string messages to (and from) all μServices. This enables Kasli to have full "slow"[7] control over all the hardware and software interfaced with TALOS, which would be impossible by leveraging only Kasli native capabilities.

In addition, the usual terminal communication with Kasli is also integrated with TALOS via a specific μService, called *Kasli Wrapper*. It provides a low-level interface to communicate with it in a native manner, useful in case the TCP connection is not available (before the instantiation of the latter, or in case of errors).

This solution, coupled with a few digital lines controlled by the FPGA, enables the correct synchronisation of complex operations (e.g. setting the potential of a specific electrode to a specific voltage, configuring and starting the acquisition of a spectrometer) with a precision in the order of ns.

As mentioned before, TALOS could be easily modified in order to integrate a different real-time system. In fact, the terminal communication with a different FPGA can be simply assimilated by creating a child of the *Kasli Wrapper* μService, and coding it to redirect the messages between TALOS and the new FPGA. Similarly, to leverage the power of the TCP connection, the base functions present in the ARTIQ *TCP Library* have to be reimplemented in the language used by the new real-time system, or readily used if Python is supported.

The TALOS framework itself is the subject of a dedicated publication [50].

Moreover, also TALOS is freely available open-source in a *git* repository (doi: 10.5281/zenodo.10371404).

## 2.4  Data acquisition

Every experiment has the need to save and store the data collected during the measurements. For this purpose, the AEḡIS experiment operates an integrated run and monitoring

---

[7]The messages run over the network, so the speed of communication is inherently on the order of the ms.
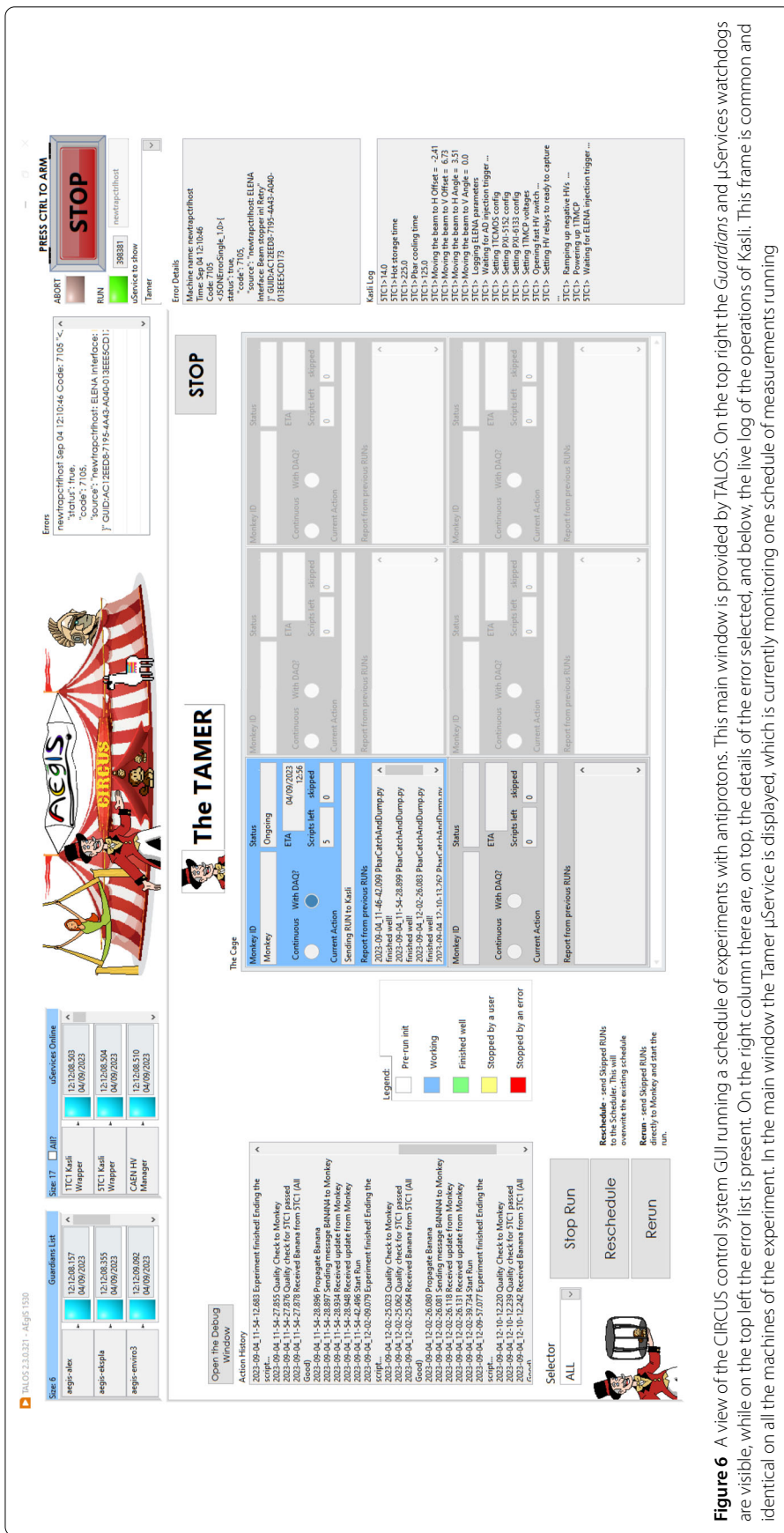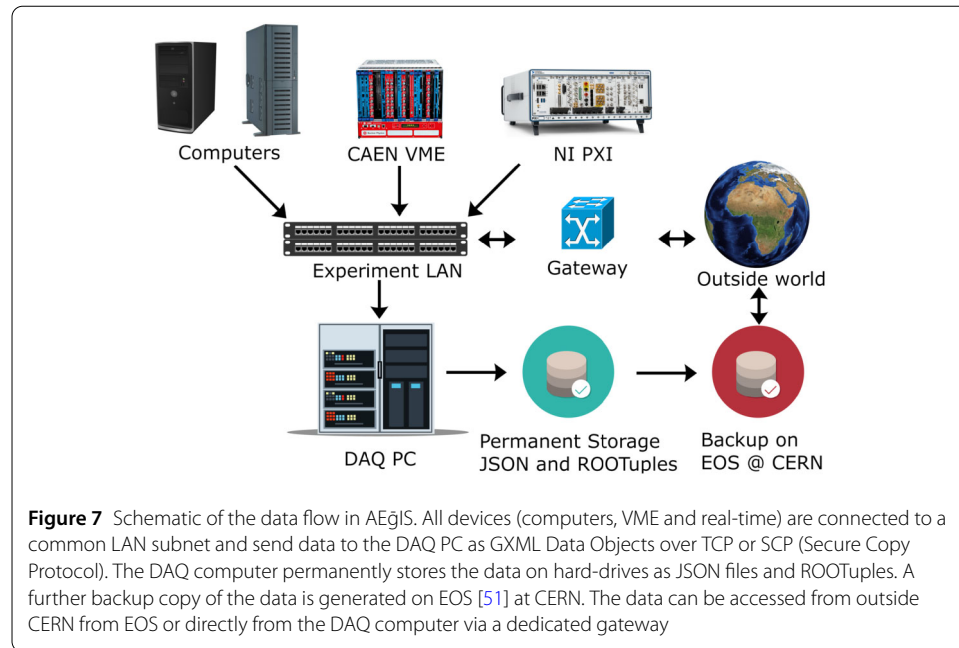
**Figure 6** A view of the CIRCUS control system GUI running a schedule of experiments with antiprotons. This main window is provided by TALOS. On the top right the *Guardians* and µServices watchdogs are visible, while on the top left the error list is present. On the right column there are, on top, the details of the error selected, and below, the live log of the operations of Kasli. This frame is common and identical on all the machines of the experiment. In the main window the Tamer µService is displayed, which is currently monitoring one schedule of measurements running

**Table 2** Structure of the AEḡIS data atom, representing *all* DAQ data objects

| Name | Alphanumeric String containing a (possibly hierarchical) unique name for the data atom. The format of the data associated with a given name should not be changed. |
|---|---|
| Timestamp | Instant when the data was acquired, in three formats:<br>1) character string, parsable by `strptime(3)`;<br>2) `struct timespec` containing time since the UNIX epoch;<br>3) 64-bit unsigned integer with RF clock count, if applicable. |
| Data | Instance of a scalar, vector, or structured (cluster) data type, compatible with LabVIEW™ types, and their conversion into either JSON-formatted files or ROOT `TTrees`. |



**Figure 7** Schematic of the data flow in AEḡIS. All devices (computers, VME and real-time) are connected to a common LAN subnet and send data to the DAQ PC as GXML Data Objects over TCP or SCP (Secure Copy Protocol). The DAQ computer permanently stores the data on hard-drives as JSON files and ROOTuples. A further backup copy of the data is generated on EOS [51] at CERN. The data can be accessed from outside CERN from EOS or directly from the DAQ computer via a dedicated gateway

data acquisition (DAQ) and logging system. Data atoms,[8] all cast in the standard format described in Table 2, are generated at various locations in the experiment, transferred over the local-area network (LAN), saved to local storage, then saved to long-term disk and tape storage systems at CERN. Data sources and sinks, along with the data transfer paths over the LAN, are identified in Fig. 7. This system is designed for the vital parameter monitoring needs for experiment commissioning and the long-term data logging for experimental runs, and has been running for over a decade.

The data are saved in JSON-formatted files, which provide a compact, clearly structured standard for efficient generation and transfer and are compatible with the GXML reference library (for serialisation) of the LabVIEW™ architecture used in many experiments.

For online access of monitoring data, CERN's ROOT data format system is currently still used preferentially thanks to its high data compression functionalities.

A side-by-side comparison of text representations of the general-purpose AEḡIS data atom in the GXML and JSON formats is shown in Fig. 8.

The presented DAQ system was built and adapted according to the specific needs of the AEḡIS experiment and is explained here for completeness. Other data acquisition systems, based on different hardware and software setups, can of course be easily integrated

---

[8]The term *data atom* refers to one unit of the smallest data container used.

```
<GXML_Root>                                    [ { "test_cluster": {
    <Name type='String'>                           "Timestamp": {
        test_cluster                                   "clock": 7856432,
    </Name>                                            "str":
    <Timestamp mems='4'>                                   {\scriptsize "16:17:18.020212 10/20/2021"},
        <str type='String'>                            "tv_nsec": 20212223,
            16:18:09.220036 09/20/2016                 "tv_sec": 1634739438
        </str>                                     },
        <tv_sec type='U64'>1474381089              "Type": "",
        </tv_sec>                                  "double_val": {
        <tv_nsec type='U32'>220036174                  "Type": "DBL",
        </tv_nsec>                                     "__value": 1.2345
        <Clock type='U64'>7856432</Clock>          },
    </Timestamp>                                   "float_array": {
    <Data mems='3'>                                    "MemberDims": "[3]",
        <double_val type='DBL'>                        "Type": "Array",
            1.2344999999999999307                      "v": [
        </double_val>                                      1.100000023841858,
        <int_val type='I32'>12345</int_val>                2.200000047683716,
        <float_array dim='[3]' type='SGL'>                 3.299999952316284
            <v>1.1</v><v>2.2</v><v>3.3</v>             ]
        </float_array>                             },
    </Data>                                        "int_val": {
</GXML_Root>                                           "Type": "I32",
                                                       "__value": 12345
                                               } } } ]
```

**Figure 8** Left: Example of GXML serialisation of an AEḡIS data atom containing a cluster of two numeric scalar values and one numeric array. Right: The corresponding JSON equivalent representation

in the overall control system structure analogously. Provided that the data acquisition system supports an interface with the commands *Start*, *Stop* and *Send data*, its integration in CIRCUS would simply consist of creating a child of the *DAQ Manager* µService, and implementing inside it the proper interface with these commands. After that, TALOS and all the other µServices will immediately use the new data acquisition system for data saving, without any further change in the code.
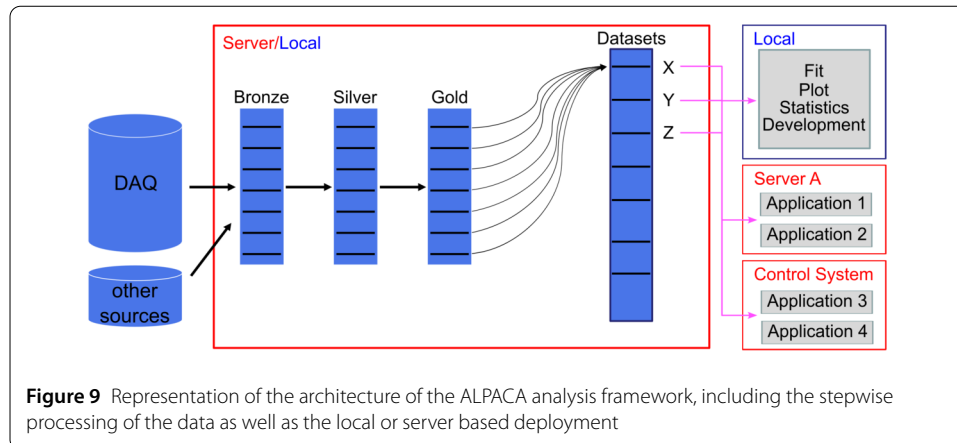
## 2.5  Integrated analysis pipelines

Analogously to the data acquisition system, every experiment also has the need for a series of algorithms to analyse the obtained data. Often, part of the data analysis is used to tune and improve the subsequent data acquisition: the capability of a control system to perform this task in autonomy is of great advantage to the scientists.

All Python Analysis Code of AEḡIS (ALPACA) is a Python data analysis framework written specifically for the AEḡIS experiment's different physics tracks. It leverages the functionality of the *NumPy* [52], *SciPy* [53] and *Plotly* [54] libraries to collect and transform the raw data acquired by all the detectors into observables, which can then be utilised by the scientists to perform dedicated studies. Figure 9 depicts the framework's linear architecture, where pipelines transform the data into different processing states.

First, all raw sources of an experiment's data, stored on different servers and in different formats (e.g. ROOT, json, png, txt, etc., and either originally plain or zipped) are concatenated into a bronze state as a Python dictionary. Raw sources include the data of each detector triggered, the settings of the detectors and the environmental data (for example, temperature and vacuum readings) during the experiment. At this stage, the originally stored files are just saved in a Python native format but no data manipulation is applied.

From the bronze to the silver state, the data is restructured depending on how each detector stores the acquired data according to its own configurations. For example, the json files for the acquired voltage readout of the MCP detector[9] always contain, as the

---

[9]A Micro-Channel Plate used to detect particles at the far end of the AEḡIS experiment. The electrons generated are converted into light by a phosphor screen and imaged with a camera. The voltage profile of the MCP itself is also acquired.

**Figure 9** Representation of the architecture of the ALPACA analysis framework, including the stepwise processing of the data as well as the local or server based deployment

first and second entries, the start time of the acquisition and the time increment, while the remaining entries hold the actual voltage readings after each time increment. In the bronze → silver pipeline, these data are parsed such that the start time, the time increment and the voltage readout become variables accessible on their own. Moreover, a three-layer nested data structure is established with the detector on the top, the acquisition number in the middle and the acquired data and run-specific configurations of the detectors at the deepest level.

Subsequently, the silver → gold pipeline computes and appends observables for each detector and acquisition, still preserving all the original data in the gold state as well. For example, in this step, the image taken from the MCP camera is first normalised for the set gain of the MCP, then the background is evaluated and subtracted, before sum, mean and standard deviation of all pixels are calculated and made available as three diffent observables.

In the last step, user-specified datasets of observables over many experiments are concatenated and made available for the user's personal analysis as well as for applications. Additionally, a dedicated package for the generation of statistics fits and plots as well as for the training, evaluation, and use of machine learning models using the generated datasets has been developed.

Thanks to the single end-point for querying datasets from ALPACA as well as the independence of the pipelines from each other, ALPACA is easily scalable in the number of applications as well as in the data sources and processing pipelines. Special emphasis is put on the scalability and reusability of the source code, which allows the seamless integration of new detectors installed at the AEḡIS apparatus as well as new analysis pipelines. Different applications utilising ALPACA's end-point for datasets beyond simple user's analyses have been envisioned for the future, enabling especially the introduction of automated feedback loops via the main control system to autonomously take decisions and promptly adjust the experimental settings for the subsequent experiment. Such feedback loops can be used for optimisation problems and event triggering, thereby increasing the overall progress speed of AEḡIS by integrating the ALPACA framework directly into CIRCUS as well.

Table 3 includes samples of the current runtime performance on a set of 177 experiments, which produced an average of 21.4 MB of raw data.

**Table 3** Runtime performance of the analyses framework using the experimental data from a parameter scan during the antiproton beam time 2022. These times are characteristics of the AEḡIS system

| Number of experiments | Loading times, in seconds, from: | | | |
|---|---|---|---|---|
| | Raw | Bronze | Gold | Datasets |
| 1 | 3.96 | 1.42 | 0.22 | 0.009 |
| 10 | 36.20 | 15.86 | 2.33 | 0.015 |
| 100 | 380.27 | 176.76 | 24.47 | 0.06 |
| 177 | 658.81 | 310.85 | 46.04 | 0.09 |

A significant speed-up in development and analyses is achieved by reloading the data from the different processing states. Loading the data from "Raw" takes exceptionally long due to the necessary download from the AEḡIS servers, while the locally stored datasets are available almost instantaneously. Processing the data of a single experiment usually takes few seconds, which is feasible for feedback loops with the control system.

In the framework of the presented control system, ALPACA is a powerful tool to aid automation and enable self-optimisation, and it is used as the main analysis framework in AEḡIS. In principle, its design serves as a foundation and its use can be adapted to different experiments as needed. However, different software architectures that fulfil this purpose can also be used in its place. In particular, the capability of CIRCUS to autonomously modify the experiment parameters based on the feedback loop from the data taken (an example of which is given in the following section) relies on a simple interface with the analysis framework. It consists of two shell commands: one for retrieving the last measured value of a specified observable, and another one for receiving new parameters to use, given a list of parameters used and results obtained. Any analysis framework capable of producing such a simple interface would be straightforwardly integrable in CIRCUS.

## 2.6  First automation with feedback loop: timing stabilisation of a laser pulse

The combination of the new control system and the new framework for data taking, storing, and pre-processing yields another desirable feature: decision-making based on a feedback loop. Complex systems typically depend on a multitude of parameters, of which not all are directly controllable.

A good study case is the stabilisation of the pulse timing of one of the AEḡIS lasers. In fact, the AEḡIS laser system for positronium excitation to the $n = 2$ state displays a strong correlation between ambient humidity and the resulting generation instant of the light pulse. The humidity in the environment, on the other hand, is coupled to the temperature, which in turn affects the output laser energy. Since the current "climate control" system can either stabilise the humidity or the temperature, the other needs to be allowed to run freely. The nanosecond-precise control system opens up the opportunity to tune the timing of the laser pulse by means of triggering a Pockels cell at the right moment, whereas the energy of the laser cannot be adjusted that easily. Thus, the temperature (and consequentially the energy of the laser pulse) is chosen to be controlled by the climate system, while the humidity is left to run freely. In turn, the time drift caused by the humidity variation is compensated by the control system via a feedback loop, which is detailed below.

A few seconds before the actual positronium production instant, a test laser pulse is produced by triggering the Pockels cell and the data acquisition chain. The generation instant of this pulse, depending on the environmental conditions, may vary with respect

**Figure 10** A feedback loop uses the uncorrected laser pulse timings (red squares) to calculate the deviation from the user setting (solid black line) over the course of an hour, and corrects the timing of the subsequent desired laser pulse that is used for the actual experiment (blue circles). Independent of short-term to long-term drifts or even sudden jumps, the resulting timing is always close to the desired value

to the moment the Pockels cell is triggered, e.g. because humidity drifts over time. The acquired spectrum of a photo diode is immediately stored by the DAQ system and analysed by a dedicated function in the experimental script, which extracts the arrival time of the test laser pulse. It is then compared to a user-defined value and a correction term is calculated. Imminent to positron implantation into the converter target, the Pockels cell is triggered again for the actually used pulse, applying the correction term obtained from the test pulse to account for the temporal offset. As a result, the synchronisation is now sufficiently precise to guarantee an overlap of the laser pulse and the positronium cloud, independently of the origin of the drift. This can be seen in Fig. 10, where the timings of the test laser pulses (red squares) and the desired laser pulses (blue circles) are plotted for a series of experimental trials executed over the course of one hour (with some interruptions). The user-defined value is given as the horizontal line. The statistical errors on the determination of the timings are of the order of a few hundred picoseconds and thus not visible in the plot.

This active feedback loop, exemplified for the timing of a laser pulse, is versatile and can be applied to any parameter of any part of the system, given that there is enough time to obtain the test data and analyse it before the real experiment occurs. With this step, the control system becomes self-governed and self-stabilising, obtaining the ability to tune parameters autonomously for an optimal result.

## 3  Results and discussion

Throughout the data taking of the 2021 antiproton beam time, three computers and two Sinara crates were used to perform the experiments. The computers were executing the CIRCUS control system and running in total 17 μServices, and they operated continuously during the whole period of beam time. Although the system was de facto undergoing its first field test campaign, it exhibited a very good stability, with an up-time close to 100 % of all the foreseen beam time. Moreover, albeit not yet complete, the new control system already proved capable of operating the AEḡIS experimental apparatus and routines in a completely unsupervised mode: in fact, it ran in an unmanned way throughout all

nights of the data taking. In addition, the automation was advanced further to perform parametric scans within multi-dimensional phase-space: again, the system displayed the ability of running up to (in total) ~1000 data points over four different parameters, autonomously pausing and resuming the measurements when detecting manageable exceptions, e.g. when there was an interruption in the beam delivery from ELENA.
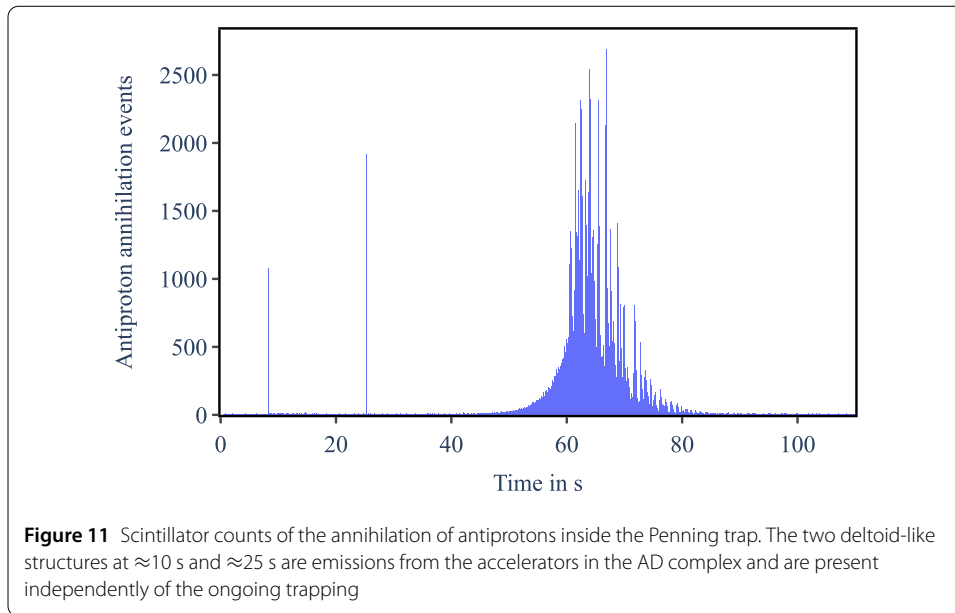
In 2022, the control system was further upgraded and refined, rendering it more stable, with a better error management and handling of external events (i.e. retrying a run if a μServices could not contact the DAQ, or if the beam of antiprotons from ELENA was empty). A total of six PCs were running more than 100 μServices (some of them were multiple instances of the unique 42 μServices coded). Apart from down time due to technical development on the experiment (or the decelerator complex), the system took data continuously. The interface with the Sinara electronics has been refined to allow for the option of using multiple, independently running units simultaneously. This feature will become critical once antihydrogen is routinely produced in large numbers. Ultimately, the integration of the analysis framework has enabled the system to autonomously derive certain values of some parameters of the experiment, based on a feedback-loop-driven machine-learning optimiser. This has completely changed the operation modality: from long scans and offline analyses to find the best working settings, to programming the machine to actively, and continuously, find them in an autonomous way.

The triggers from AD/ELENA were reliably registered by the digital I/O units of the Sinara crates and propagated through the control system to all involved hardware. The working principle was the following: upon reception of the "ELENA Injection" trigger (which arrives approximatively 30 s before the antiprotons actually reach the experiment) by Kasli, all the hardware systems are initialized and prepared to respond to a trigger signal, which is then given from a digital line of Sinara upon reception of either the "Bunch arriving – 20 $\mu$s" (for slower hardware such as cameras) or the "Bunch arriving"[10] (for fast hardware such as high voltage electrode gates) signals.

Thanks to the features of the CIRCUS control system as well as other recent improvements to the experiment, antiproton capture in the trap was efficiently performed: the synchronisation capabilities provided by Sinara, coupled with the fast iteration regime facilitated by ARTIQ, enabled a fully parameter-optimised capture of the energy-degraded portion of the antiprotons in less than 10 days after the first beam was acquired.[11] To monitor the capture efficiency, three different scintillating fibres, each connected to a photomultiplier tube (PMT), were used: by operating the PMT in the non-saturation regime, the quantity of antiprotons was estimated by the amplitude of the detectors' signals. The difference in the signals between measurements without raising the electrode gates ("passthrough mode") and with the electrodes raised at the correct time ("capture mode") confirms the capture of a significant amount of antiprotons available from ELENA (preliminary estimates point towards a record trapping efficiency around 70% [55]). As shown in Fig. 11, the annihilation signals in the surrounding scintillators indicate trapping of antiprotons for up to 50 s, a lifetime in agreement with the initially very poor vacuum level ($\approx 10^{-8}$ mbar at the time of this measurement) and the absence of electron cooling.

---

[10]The time difference between the arrival of this signal and the effective arrival of the particles is settable by the experiment, and it is typically in the order of hundreds of ns..

[11]For comparison, a similar optimised results was achieved with the previous system in more than 3 months..

**Figure 11** Scintillator counts of the annihilation of antiprotons inside the Penning trap. The two deltoid-like structures at ≈10 s and ≈25 s are emissions from the accelerators in the AD complex and are present independently of the ongoing trapping

The characteristic bell shape of the annihilation events is given by the fact that initially, the antiprotons are trapped at several kV energy, but the cross-section of annihilation is effectively greater than zero only for energies in the order of tenths of eV. Therefore, at the beginning, no annihilations take place because the antiprotons are losing energy by elastic collision with the residual gas. When their energy is low enough, they start to annihilate, which here happens from around 45 s. Since the population of low-energy antiprotons increases in time, the annihilation count rises, reaching a peak when the depletion of the antiprotons in the trap starts to be significant. From there, the curve decreases, terminating with an exponential decay with the characteristic lifetime of the cold antiprotons in the trap.

In parallel, and unrelated to the experiments performed with antiprotons, the experiment employs two laser systems, the so-called "EKSPLA" (205 nm and 1064 nm), which is a Nd:YAG pump-based system for antihydrogen formation, and an alexandrite-based system (in the following referred to as "Alex", 243 nm), used in experiments with positronium. These two setups are operated independently from each other, and they are spatially separated by more than 5 m, but during measurements, it is essential to keep them synchronised. This has been achieved by taking advantage of the master-satellite operation mode of Kasli devices. Two configurations have been tested: the continuous, standalone operation, and the on-demand operation. In the first scenario, the EKSPLA pulses at a frequency of 10 Hz are synchronised with the 4 Hz Alex pulses by a couple of Sinara crates, kept in master-satellite configuration through an optic fibre connection. On the master, an idle script continuously runs without the need of a computer and simultaneously re-triggers both lasers every 30 s, so as to temporally realign them and to eliminate any accumulated drift. In on-demand operation mode, by contrast, the lasers and the Sinara crates are kept idle, and the user can, at will, run a script which synchronously activates the pumping of the two lasers and subsequent simultaneous triggering.

## 4  Conclusions

The AEḡIS collaboration has implemented CIRCUS, a novel, high level and very general system for controlling complex physics experiments based on the Sinara/ARTIQ open hardware/software ecosystem and the TALOS software infrastructure.

The first in-depth stress tests of the new control system during the regular antiproton run time at CERN's Antiproton Decelerator have successfully validated its usability for ns-precise synchronisation of the involved procedures and its continuous reliable operation. It demonstrates sound reproducibility of the experiments.

Consequently, the control system will be extended further to include additional parts of the experiment and enable an autonomous execution of the more complex activities foreseen for future beam times. Additionally, the interface between TALOS and ARTIQ will be improved with a more advanced library structure and the possibility of operating multiple Sinara units simultaneously, but with different modes of synchronisation among them (i.e. some running synchronously, and others asynchronously). A higher level of on-line data analysis integration is being implemented, since the new optimisation-driven approach is significantly improving the operation modality, reducing the beam time required and enabling manoeuvres previously unfeasible.

By providing such automation of the entire run operation, the CIRCUS control system will continue to optimise the uptime and quality of data taken during the upcoming measurement campaigns of AEḡIS, including complex experiments such as the formation of antihydrogen atoms and the study of their quantum level distributions, as well as the exploration of antiprotonic atoms production.

On a broader scale, CIRCUS represents a novel kind of approach to managing experimental routines (and setups in general) with a focus on autonomy, which can be employed for a variety of different applications. In particular, experiments relying on precise synchronisation and coordination of subsystems handling individual tasks from different fields, reliable operation over many months, and flexible adaptations of the setup, such as those focused on atomic and quantum physics studies, can benefit from the introduction of this control system. The self-optimisation capabilities further render the system minimally sensible to external changes and very stable in its operation.

Another experiment, PsICO (Positronium Inertial and Correlation Observation), has started implementing the CIRCUS to operate its apparatus: its main goal is to study the three-body entanglement properties of the three photons produced by the decay of ortho-Positronium, relating it to the initial spin state [56, 57].

Both the hardware and software of the presented control system are available open-source to be adapted as needed for use in individual experiments, which is easily enabled by the modular and standardised library-based approach of the system's design.

### Appendix: Calibration of the voltage amplifiers

The calibration of the amplifier channels providing the electrode voltages to the AEḡIS traps is based on a scan from the minimum to the maximum of the range of possible Fastino channel voltages ($-10$ V to $+10$ V, which corresponds to $-200$ V to $+200$ V on the amplifier outputs). As a compromise between statistical precision and timing efficiency, a step size of 327 machine units of the Fastino (approximately 0.1 V depending on the exact configuration of each individual channel) is chosen for the scanning measurements.
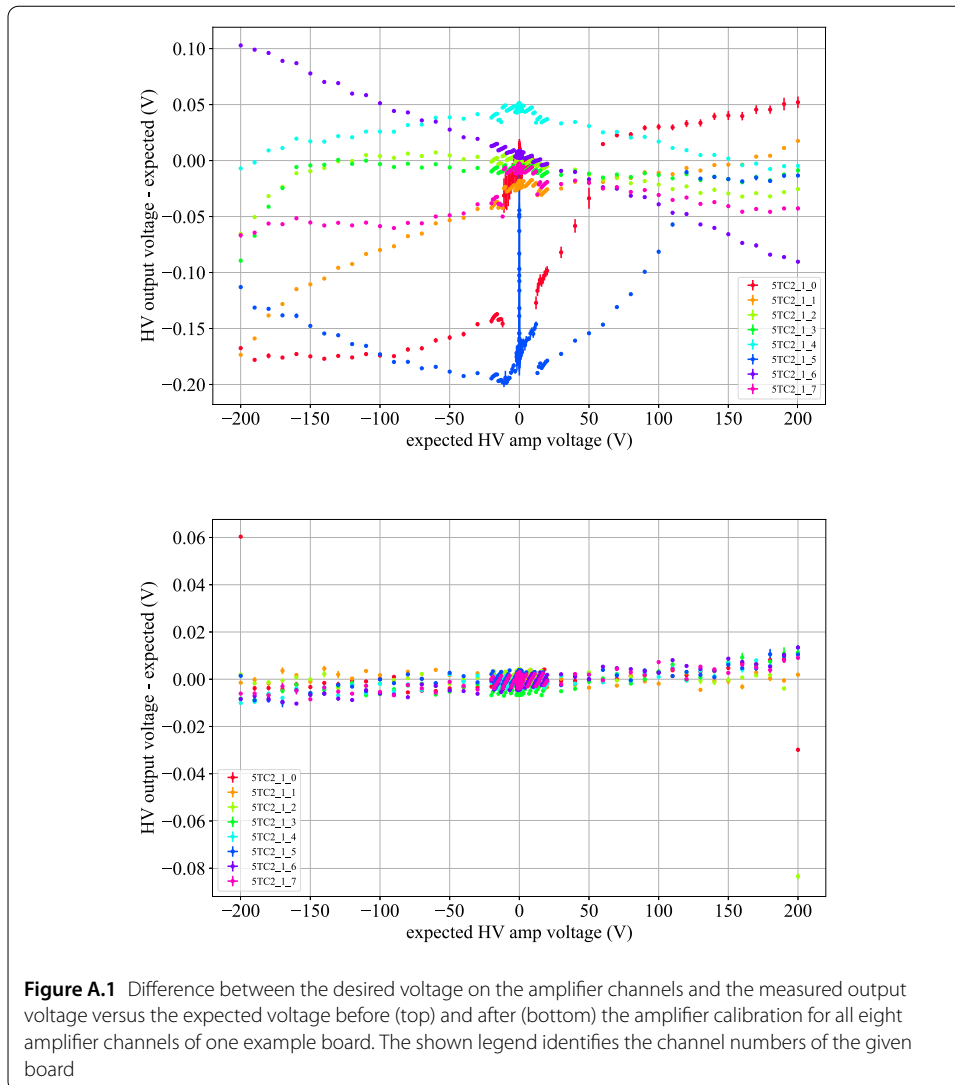
**Figure A.1** Difference between the desired voltage on the amplifier channels and the measured output voltage versus the expected voltage before (top) and after (bottom) the amplifier calibration for all eight amplifier channels of one example board. The shown legend identifies the channel numbers of the given board

At every voltage step, five measurement iterations are performed for every channel to be calibrated, where the actually produced voltage on the amplifier output is registered by a multimeter and read to a calibration file in JSON format.

The software calibration routine for all channels is then done at the same time: the data is fitted with a linear function from lowest to highest setting and the slope and offset are saved as calibration parameters for every channel individually. These parameters are imported into the corresponding ARTIQ library and directly applied as correction factors when setting a voltage on one of the trap electrodes from software.

A verification measurement for each channel is executed after waiting for an arbitrary amount of time, thus excluding a systematic influence from environmental conditions. For these measurements, a different scan through the range of Fastino voltages is performed, directly applying the calibration correction. The produced voltages are read out in the same way and the data is analysed to verify the minimisation of the differences between desired and produced voltage by the calibration for all channels.

Figure A.1 shows the result of this verification measurement before and after calibration for one of the amplifier boards, taken as example.

The absolute voltage accuracy reached after calibration is significantly improved, reaching the mV level on all channels, rendering it comparable to the 16-bit, i.e. 6 mV, precision of the Fastino settings. The precise reachable minimum and maximum voltage depends on the individual internal configuration of the Fastino channel and causes larger deviations in either positive or negative direction when pushing to the very boundaries. However, the reachable value is in no case further away than 0.1 V from the extremes of $\pm 200$ V, which suffices for the purposes of AEḡIS, as voltages beyond $\pm 190$ V are never required for the application of the trap potentials. The clustering of data points at low absolute voltage values stems from the procedure of adapting the step size to the scanned voltage range for the verification measurements; the internal step structure is a consequence of the 16-bit precision of the voltage settings. The large fluctuations and resulting error bars for low voltage settings of some channels were caused by an intrinsic condition of the hardware, which has since been fixed by the mounting of additional capacitors in the amplifier circuits.

### Availability of data and materials
Data and materials are available under the conditions of the AEgIS collaboration Data Management Plan.

## Declarations

### Competing interests
The authors declare no competing interests.

### Author contributions
MV and SH have implemented, commissioned, and maintained the main pillars of the control system: MV has developed the TALOS LabVIEW framework and the slow-control elements for the integration of all experimental subsystems; SH has built up the Sinara electronics system and the ARTIQ library structure for the fast control and synchronisation of the experiment. MV and SH have also been part of the core team running the experiment during the antiproton beam times. MV and SH are the major contributors in writing the manuscript. RC has devised the requirements of the control system and guided its implementation with his experience. RC also been part of the core team running the experiment during the antiproton beam times and lead the corresponding data analysis. JZ has designed and implemented core components of the TALOS infrastructure for a reliable operation of the experiment and contributed to the data taking during the antiproton beam times. GK, GK, and DN have introduced the collaboration to the ARTIQ/Sinara portfolio and developed the high-voltage amplifier units. TR has built up the ALPACA framework used for direct data analysis and for the integration of the self-optimisation capabilities of the system and contributed significantly to the data analysis. BR has contributed to the ARTIQ library structure and the data taking with regard to the positronium and laser routines and the feedback loop used for the parameter optimisation. FP has developed and maintained the AEḡISdata acquisition system. All authors revised and approved the final manuscript.

### Author details
[1] Physics Department, CERN, 1211 Geneva 23, Switzerland. [2] TIFPA/INFN Trento, via Sommarive 14, 38123 Povo Trento, Italy. [3] Department of Physics, University of Trento, via Sommarive 14, 38123 Povo, Trento, Italy. [4] Institute for Experimental Physics, Universität Hamburg, 22607 Hamburg, Germany. [5] Warsaw University of Technology, Faculty of Physics, ul. Koszykowa 75, 00-662, Warsaw, Poland. [6] Felix Bloch Institute for Solid State Physics, Universität Leipzig, 04103 Leipzig, Germany. [7] University of Liverpool, Liverpool, UK. [8] The Cockcroft Institute, Daresbury, UK. [9] INFN Milano, via Celoria 16, 20133 Milano, Italy. [10] University of Latvia, Department of Physics, Raina boulevard 19, LV-1586, Riga, Latvia. [11] Institute of Experimental and Applied Physics, Czech Technical University in Prague, Husova 240/5, 110 00, Prague, Prague 1, Czech

Republic. [12]Department of Physics, University of Oslo, Sem Sælandsvei 24, 0371 Oslo, Norway. [13]Department of Physics "Aldo Pontremoli", University of Milano, via Celoria 16, 20133 Milano, Italy. [14]Institute of Physics, Faculty of Physics, Astronomy, and Informatics, Nicolaus Copernicus University in Toruń, Grudziadzka 5, 87-100 Toruń, Poland. [15]Department of Aerospace Science and Technology, Politecnico di Milano, via La Masa 34, 20156 Milano, Italy. [16]Department of Physics, NTNU, Norwegian University of Science and Technology, Trondheim, Norway. [17]Czech Technical University, Prague, Brehová 7, 11519, Prague, Prague 1, Czech Republic. [18]Raman Research Institute, C. V. Raman Avenue, Sadashivanagar, Bangalore 560080, India. [19]Institute of Physics, Polish Academy of Sciences, Aleja Lotnikow 32/46, PL-02668 Warsaw, Poland. [20]INFN Pavia, via Bassi 6, 27100 Pavia, Italy. [21]Department of Civil, Environmental, Architectural Engineering and Mathematics, University of Brescia, via Branze 43, 25123 Brescia, Italy.

## References

1. Drobychev G, Nedelec P, Sillou D, Gribakin G, Walters H, Ferrari G, Prevedelli M, Tino GM, Doser M, Canali C, Carraro C, Lagomarsino V, Manuzio G, Testera G, Zavatarelli S, Amoretti M, Kellerbauer AG, Meier J, Warring U, Oberthaler MK, Boscolo I, Castelli F, Cialdi S, Formaro L, Gervasini A, Giammarchi G, Vairo A, Consolati G, Dupasquier A, Quasso F, Stroke HH, Belov AS, Gninenko SN, Matveev VA, Byakov VM, Stepanov SV, Zvezhinskij DS, De Combarieu M, Forget P, Pari P, Cabaret L, Comparat D, Bonomi G, Rotondi A, Djourelov N, Jacquey M, Büchner M, Trénec G, Vigué J, Brusa RS, Mariazzi S, Hogan S, Merkt F, Badertscher A, Crivelli P, Gendotti U, Rubbia A. Proposal for the AEgIS experiment at the CERN antiproton decelerator (antimatter experiment: gravity, interferometry, spectroscopy). Technical report SPSC-P-334. CERN-SPSC-2007-017. CERN, Geneva; 2007. http://cds.cern.ch/record/1037532.
2. Hangst JS, Bowe P. ALPHA proposal. Technical report. CERN, Geneva; 2005. https://cds.cern.ch/record/814351.
3. Ulmer S, Yamazaki Y, Blaum K, Quint W, Walz J. Direct high-precision measurement of the g-factor of a single antiproton stored in a cryogenic penning trap. Technical report. CERN, Geneva; 2012. https://cds.cern.ch/record/1455847.
4. Azuma T, Bakos JS, Bluhme H. Atomic spectroscopy and collisions using slow antiprotons. Technical report. CERN, Geneva; 1997. https://cds.cern.ch/record/622250.
5. Chardin G, Grandemange P, Lunney D, Manea V, Badertscher A, Crivelli P, Curioni A, Marchionni A, Rossi B, Rubbia A, Nesvizhevsky V, Hervieux P-A, Manfredi G, Comini P, Debu P, Dupré P, Liszkay L, Mansoulié B, Pérez P, Rey J-M, Ruiz N, Sacquin Y, Voronin A, Biraben F, Cladé P, Douillet A, Gérardin A, Guellati S, Hilico L, Indelicato P, Lambrecht A, Guérout R, Karr J-P, Nez F, Reynaud S, Tran V-Q, Mohri A, Yamazaki Y, Charlton M, Eriksson S, Madsen N, Werf D-P, Kuroda N, Torii H, Nagashima Y. Proposal to measure the gravitational behaviour of antihydrogen at rest. Technical report. CERN, Geneva; 2011. https://cds.cern.ch/record/1386684.
6. Aumann T, Bartmann W, Bouvard A, Boine-Frankenheim O, Broche A, Butin F, Calvet D, Carbonell J, Chiggiato P, De Gersem H, De Oliveira R, Dobers T, Ehm F, Ferreira Somoza J, Fischer J, Fraser M, Friedrich E, Grenard J-L, Hupin G, Johnston K, Kubota Y, Gomez-Ramos M, Indelicato P, Lazauskas R, Malbrunot-Ettenauer S, Marsic N, Müller W, Naimi S, Nakatsuka N, Necca R, Neyens G, Obertelli A, Ono Y, Pasinelli S, Paul N, Pollacco EC, Rossi D, Scheit H, Seki R, Schmidt A, Schweikhard L, Sels S, Siesling E, Uesaka T, Wada M, Wienholtz F, Wycech S, Zacarias S. PUMA: antiprotons and radioactive nuclei. Technical report; 2019. https://cds.cern.ch/record/2691045.
7. Gabrielse G, Fei X, Helmerson K, Rolston SL, Tjoelker R, Trainor TA, Kalinowsky H, Haas J, Kells W. First capture of antiprotons in a Penning trap: a kiloelectronvolt source. Phys Rev Lett. 1986;57:2504–7. https://doi.org/10.1103/PhysRevLett.57.2504.
8. Knoop M, Madsen N, Thompson RC. 8. Physics with trapped charged particles: lectures from the Les Houches winter school. 1st ed. London: Imperial College Press; 2014. p. 219–38.
9. Gabrielse G, Fei X, Orozco LA, Tjoelker RL, Haas J, Kalinowsky H, Trainor TA, Kells W. Cooling and slowing of trapped antiprotons below 100 mev. Phys Rev Lett. 1989;63:1360–3. https://doi.org/10.1103/PhysRevLett.63.1360.
10. Aghion S, et al (AEgIS collaboration). Compression of a mixed antiproton and electron non-neutral plasma to high densities. Eur Phys J D. 2018;72(4):76. https://doi.org/10.1140/epjd/e2018-80617-x.
11. Danielson JR, Dubin DHE, Greaves RG, Surko CM. Plasma and trap-based techniques for science with positrons. Rev Mod Phys. 2015;87:247–306.
12. Amoretti M, et al. Production and detection of cold antihydrogen atoms. Nature. 2002;419(6906):456–9.
13. Gabrielse G, et al. Background-free observation of cold antihydrogen with field-ionization analysis of its states. Phys Rev Lett 2002;89:213401.
14. Andersen GB, et al. Confinement of antihydrogen for 1000 seconds. Nat Phys. 2011;7:558–64.
15. Ahmadi M, et al. Observation of the 1 s-2 s transition in trapped antihydrogen. Nature. 2017;541:506–10. https://doi.org/10.1038/nature21040.
16. Kuroda N, et al. A source of antihydrogen for in-flight hyperfine spectroscopy. Nat Commun. 2014;5:3089. https://doi.org/10.1038/ncomms4089.
17. Doser M, et al (AEgIS collaboration). Aegis at elena: outlook for physics with a pulsed cold antihydrogen beam. Philos Trans R Soc Lond A. 2018;376:20170274.
18. Aghion S, et al (AEgIS collaboration). A moiré deflectometer for antimatter. Nat Commun. 2014;5:4538.
19. Charlton M. Antihydrogen production in collisions of antiprotons with excited states of positronium. Phys Lett A. 1990;143(3):143–6. https://doi.org/10.1016/0375-9601(90)90665-B.
20. Krasnický D, Caravita R, Canali C, Testera G. Cross-section for Rydberg antihydrogen production via charge exchange between Rydberg positronium and antiprotons in magnetic field. Phys Rev A. 2016;94:022714.
21. Testera G, et al (AEgIS collaboration). The aegis experiment. Hyperfine Interact. 2015;233:1–31320.
22. Storey J, et al. Particle tracking at 4K: the Fast Annihilation Cryogenic Tracking (FACT) detector for the AEgIS antimatter gravity experiment. Nucl Instrum Methods Phys Res, Sect A. 2013;732:437–41.
23. Amsler C, Antonello M, Belov A, Bonomi G, Brusa RS, Caccia M, Camper A, Caravita R, Castelli F, Comparat D, Consolati G, Demetrio A, Di Noto L, Doser M, Ekman PA, Fanì M, Ferragut R, Gerber S, Giammarchi M, Gligorova A, Guatieri F, Hackstock P, Haider D, Haider S, Hinterberger A, Kellerbauer A, Khalidova O, Krasnický D, Lagomarsino V, Malbrunot C, Mariazzi S, Matveev V, Müller SR, Nebbia G, Nedelec P, Nowak L, Oberthaler M, Oswald E, Pagano D, Penasa L, Petracek

V, Prelz F, Prevedelli M, Rienaecker B, Robert J, Røhne OM, Rotondi A, Sandaker H, Santoro R, Storey J, Testera G, Tietje IC, Toso V, Wolz T, Wuethrich J, Yzombard P, Zimmer C, Zurlo N. A cryogenic tracking detector for antihydrogen detection in the AEgIS experiment. Nucl Instrum Methods Phys Res, Sect A, Accel Spectrom Detect Assoc Equip. 2020;960:163637. https://doi.org/10.1016/j.nima.2020.163637. Accessed 2023-09-19.

24. Zurlo N, et al. Calibration and equalisation of plastic scintillator detectors for antiproton annihilation identification over positron/positronium background. Acta Phys Pol B. 2020;51:213–23. https://doi.org/10.5506/APhysPolB.51.213.

25. Mariazzi S, et al. High-yield thermalized positronium at room temperature emitted by morphologically tuned nanochanneled silicon targets. J Phys B. 2021;54(8):085004. https://doi.org/10.1088/1361-6455/abf6b6.

26. Mariazzi S, Bettotti P, Brusa RS. Positronium cooling and emission in vacuum from nanochannels at cryogenic temperature. Phys Rev Lett. 2010;104(24):243401. https://doi.org/10.1103/PhysRevLett.104.243401. Accessed 2023-09-19.

27. Mariazzi S, Bettotti P, Larcheri S, Toniutti L, Brusa RS. High positronium yield and emission into the vacuum from oxidized tunable nanochannels in silicon. Phys Rev B. 2010;81(23):235418. https://doi.org/10.1103/PhysRevB.81.235418. Accessed 2023-09-19.

28. Amsler C, et al (AEgIS collaboration). Pulsed production of antihydrogen. Commun Phys. 2021;4(19). https://doi.org/10.1038/s42005-020-00494-z.

29. Aghion S, et al (AEgIS collaboration). Laser excitation of the n = 3 level of positronium for antihydrogen production. Phys Rev A. 2016;94:012507.

30. Perego E, Pomponio M, Detti A, Duca L, Sias C, Calosso CE. A scalable hardware and software control apparatus for experiments with hybrid quantum systems. Rev Sci Instrum. 2018;89(11):113116. https://doi.org/10.1063/1.5049120. Accessed 2023-07-25.

31. Starkey PT, Billington CJ, Johnstone SP, Jasperse M, Helmerson K, Turner LD, Anderson RP. A scripted control system for autonomous hardware-timed experiments. Rev Sci Instrum. 2013;84(8):085111. https://doi.org/10.1063/1.4817213. Accessed 2023-07-25.

32. Agraz J, Grunfeld A, Li D, Cunningham K, Willey C, Pozos R, Wagner S. LabVIEW-based control software for para-hydrogen induced polarization instrumentation. Rev Sci Instrum. 2014;85(4):044705. https://doi.org/10.1063/1.4870797. Accessed 2023-07-25.

33. Trenkwalder A, Zaccanti M, Poli N. A flexible system-on-a-chip control hardware for atomic, molecular, and optical physics experiments. Rev Sci Instrum. 2021;92:105103. Publisher: AIP Publishing. https://doi.org/10.1063/5.0058986.

34. Keshet A, Ketterle W. A distributed GUI-based computer control system for atomic physics experiments. 2012. https://doi.org/10.1063/1.4773536.

35. Bartmann W, Belochitskii P, Breuker H, Butin F, Carli C, Eriksson T, Maury S, Kersevan R, Pasinelli S, Tranquille G, Vanbavinckhove G. Progress in ELENA design. In: Proceeding of IPAC 2013. Shanghai (China), 2013. p. 2651–3.

36. Kasprowicz G, Kulik P, Gaska M, Przywozki T, Pozniak K, Jarosinski J, Britton JW, Harty T, Balance C, Zhang W, Nadlinger D, Slichter D, Allcock D, Bourdeauducq S, Jördens R, Pozniak K. ARTIQ and Sinara: open software and hardware stacks for quantum physics. In: Optical society of America. 2020. https://doi.org/10.1364/QUANTUM.2020.QTu8B.14. http://www.osapublishing.org/abstract.cfm?URI=QUANTUM-2020-QTu8B.14.

37. Bourdeauducq S, Jördens R, Zotov P, Britton J, Slichter D, Leibrandt D, Allcock D, Hankin A, Kermarrec F, Sionneau Y, Srinivas R, Tan TR, Bohnet J. ARTIQ 1.0. Zenodo. 2016. https://doi.org/10.5281/zenodo.51303.

38. Bitter R, Mohiuddin T, Nawrocki M. Labview: advanced programming techniques. 2nd ed. 2017. https://doi.org/10.1201/9780849333255.

39. Chou C-W, Hume DB, Koelemeij JCJ, Wineland DJ, Rosenband T. Frequency comparison of two high-accuracy Al+ optical clocks. Phys Rev Lett. 2010;104(7):070802. Accessed 2023-12-01.

40. Gonzalez FM, Fries EM, Cude-Woods C, Bailey T, Blatnik M, Broussard LJ, Callahan NB, Choi JH, Clayton SM, Currie SA, Dawid M, Dees EB, Filippone BW, Fox W, Geltenbort P, George E, Hayen L, Hickerson KP, Hoffbauer MA, Hoffman K, Holley AT, Ito TM, Komives A, Liu C-Y, Makela M, Morris CL, Musedinovic R, O'Shaughnessy C, Pattie J, Ramsey J, Salvat DJ, Saunders A, Sharapov EI, Slutsky S, Su V, Sun X, Swank C, Tang Z, Uhrich W, Vanderwerp J, Walstrom P, Wang Z, Wei W, Young AR. Improved neutron lifetime measurement with UCN$\tau$. Phys Rev Lett. 2021;127(16):162501. Accessed 2023-12-01.

41. Hinkley N, Sherman JA, Phillips NB, Schioppo M, Lemke ND, Beloy K, Pizzocaro M, Oates CW, Ludlow AD. An atomic clock with $10^{-18}$ instability. Science. 2013;341(6151):1215–8. Accessed 2023-12-01.

42. Millen J, Stickler BA. Quantum experiments with microscale particles. Contemp Phys. 2020;61(3):155–68. https://doi.org/10.1080/00107514.2020.1854497. Accessed 2023-12-01.

43. Thomas RA, Parniak M, Østfeldt C, Møller CB, Bærentsen C, Tsaturyan Y, Schliesser A, Appel J, Zeuthen E, Polzik ES. Entanglement between distant macroscopic mechanical and spin systems. Nat Phys. 2021;17(2):228–33. https://doi.org/10.1038/s41567-020-1031-5. Accessed 2023-12-01.

44. Beck BR, Fajans J, Malmberg JH. Measurement of collisional anisotropic temperature relaxation in a strongly magnetized pure electron plasma. Phys Rev Lett. 1992;68:317–20. https://doi.org/10.1103/PhysRevLett.68.317.

45. Andresen GB, Ashkezari MD, Baquero-Ruiz M, Bertsche W, Bowe PD, Butler E, Cesar CL, Chapman S, Charlton M, Fajans J, Friesen T, Fujiwara MC, Gill DR, Hangst JS, Hardy WN, Hayano RS, Hayden ME, Humphries A, Hydomako R, Jonsell S, Kurchaninov L, Lambo R, Madsen N, Menary S, Nolan P, Olchanski K, Olin A, Povilus A, Pusa P, Robicheaux F, Sarid E, Silveira DM, So C, Storey JW, Thompson RI, Werf DP, Wilding D, Wurtele JS, Yamazaki Y. Evaporative cooling of antiprotons to cryogenic temperatures. Phys Rev Lett. 2010;105:013003. https://doi.org/10.1103/PhysRevLett.105.013003.

46. Gabrielse G, Kolthammer WS, McConnell R, Richerme P, Kalra R, Novitski E, Grzonka D, Oelert W, Sefzick T, Zielinski M, Fitzakerley D, George MC, Hessels EA, Storry CH, Weel M, Müllers A, Walz J. Adiabatic cooling of antiprotons. Phys Rev Lett. 2011;106:073002. https://doi.org/10.1103/PhysRevLett.106.073002.

47. Antonello M, et al (AEgIS collaboration). Rydberg-positronium velocity and self-ionization studies in a 1t magnetic field and cryogenic environment. Phys Rev A. 2019;102:013101. https://doi.org/10.1103/PhysRevA.102.013101.

48. Hewitt C, Bishop P, Steiger R. A universal modular ACTOR formalism for artificial intelligence. In: Proc. International joint conference on artificial intelligence. 1973. p. 235–45.

49. Cerf V, Kahn R. A protocol for packet network intercommunication. IEEE Trans Commun. 1974;22(5):637–48. https://doi.org/10.1109/TCOM.1974.1092259.

50. Volponi M, Zielinski J, Rauschendorfer T, Huck S, Caravita R, Auzins M, Bergmann B, Burian P, Brusa RS, Camper A, Castelli F, Cerchiari G, Ciurylo R, Consolati G, Doser M, Eliaszuk K, Giszczak A, Glöggler LT, Graczykowski L, Grosbart M, Guatieri F, Gusakova N, Gustafsson F, Haider S, Janik MA, Januszek T, Kasprowicz G, Khatri G, Klosowski L, Kornakov G, Krumins V, Lappo L, Linek A, Malamant J, Mariazzi S, Nowak L, Nowicka D, Oswald E, Penasa L, Petracek V, Piwinski M, Pospisil S, Povolo L, Prelz F, Rangwala SA, Rawat BS, Rienäcker B, Rodin V, Røhne OM, Sandaker H, Smolyanskiy P, Sowinski T, Tefelski D, Welsch CP, Wolz T, Zawada M, Zurlo N. Talos: a framework for autonomous control systems for complex experiments. Submitted. 2023.
51. Peters AJ, Janyst L. Exabyte scale storage at cern. J Phys Conf Ser. 2011;331(5):052015. https://doi.org/10.1088/1742-6596/331/5/052015.
52. Harris CR, Millman KJ, Walt SJ, et al. Array programming with numpy. Nature. 2020;585:357–62.
53. Virtanen P, Gommers R, Oliphant TE, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. Nat Methods. 2020;3(17):261–72.
54. Inc., PT. Collaborative data science. https://plot.ly.
55. Caravita R, et al. Progress report on the AEgIS experiment (2023). CERN SPSC report; 2023.
56. Bass SD, Mariazzi S, Moskal P, Stępień E. Colloquium: positronium physics and biomedical applications. Rev Mod Phys 2023;95(2):021002. https://doi.org/10.1103/RevModPhys.95.021002. Accessed 2023-12-06.
57. Hiesmayr BC, Moskal P. Genuine multipartite entanglement in the 3-photon decay of positronium. Sci Rep. 2017;7(1):15349. https://doi.org/10.1038/s41598-017-15356-y. Accessed 2023-12-06.

**Publisher's Note**