STATUS OF THE CONTROL SYSTEM FOR THE DELTA SYNCHROTRON LIGHT SOURCE

D. Schirmer*, A. Althaus, A. Erpelding
Center for Synchrotron Radiation (DELTA), TU Dortmund University, Dortmund, Germany

Abstract

The 1.5 GeV electron storage ring DELTA, operated by the University of Dortmund in Germany, celebrated its 30th anniversary last fall. Over the past three decades, the control system has undergone many different IT infrastructure development cycles. It was commissioned between 1994 and 1998, utilizing a series of command-line-based in-house applications that operated on individual, low-performance networked HP workstations and VME-based real-time CPUs, initially without the support of graphical user interfaces (GUIs). These GUIs were gradually implemented later with the introduction of the EPICS software package (1999-2001). Based on a combination of EPICS and a newly installed Linux PC-based client/server architecture, a variety of software tools and hardware extensions were introduced in the following years. Today, the DELTA control system utilizes an open-source virtual environment with a server management platform that integrates kernel-based virtual machines (KVM), software-defined storage and network functions on a single platform. In addition, web-based user interfaces simplify the configuration of the integrated disaster recovery tool and enhance the management of high availability and redundancy within the server cluster. Furthermore, machine learning algorithms have been incorporated into the control and optimization of the storage ring. This report gives a historical review, summarizes the most important developments in recent years and provides an outlook on future projects.

INTRODUCTION

The DELTA synchrotron light source, which is operated by the TU Dortmund University in Germany, consists of a linear pre-accelerator (70 MeV), a full energy booster synchrotron (70 MeV to 1.5 GeV) and a storage ring with 115.2 m circumference (see Fig. 1). The maximum stored beam current amounts to 140 mA (nominal multibunch operation) with a lifetime of about 60 hours (at 100 mA). The facility is operated 24 hours a day, 5 days a week, where 2000 hours/year were used for synchrotron radiation (SR) studies and 1000 hours/year for maintenance work and accelerator physics research. To save electricity costs, accelerator operation has been cut by 50% since 2023 [1–3].

The control system of the storage ring facility is based on the open-source framework EPICS (Experimental Physics and Industrial Control System [4]) and comprises a typical three level client/server architecture. A field bus level, a real-time control level, and the process control level. Over the past 30 years, the systems have been continuously adapted to the latest hardware and software developments.

* detlev.schirmer@tu-dortmund.de

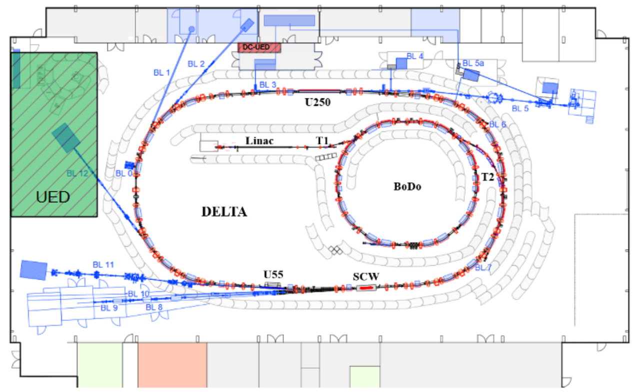

Figure 1: DELTA floor plan: The green area shows where the upcoming UED project will be located. The red area depicts the site of the currently installed 100-keV DC-UED demonstration project. The SR beamline infrastructure is shown in blue [1, 3].

HISTORICAL REVIEW

From 1999 to 2001, the control system underwent a step-by-step transformation, changing from a non-standard, in-house developed system to one based on the EPICS framework [5]. At that time, it consisted mainly of a few HP-UX-based workstations and about two dozen VMEbus single-board computers (Force/MicroSys CPU types) running the real-time operating system (OS) VxWorks [6]. They were primarily connected to accelerator input/output (I/O) devices via control area network (CAN) nodes (ESD digital/analog I/O modules [7]) or a general purpose interface bus (GPIB). Initially, all computers used a shared 10 Mbit/s 'Thick Ethernet' network consisting of a 10 mm thick, rigid, yellow coaxial cable (10Base5).

The HP-UX workstations were replaced successively by standard Linux-based PCs running the EPICS client software, along with a migration to VME-PowerPC CPUs operating as EPICS servers. As a result, DELTA became one of the first storage rings to operate a synchrotron radiation facility based on Linux PCs in conjunction with the EPICS framework.

Since October 2001, all ramped magnet power supplies of the booster synchrotron (dipole, 6 quadrupoles, 2 sextupoles, steerers), RF power, and all booster diagnostics (beam position measurement, tune control, beam loss measurement, power supply currents, etc.) were handled by VME-based digital signal processing (DSP) boards developed at DELTA ('DeltaDSP'). These DSP cards were designed with generic EPICS driver/device support [8]. This setup has not been modified and operates flawlessly up to now.

In the years that followed, the system was kept up to date through a series of enhancements and new developments (see Fig. 2).

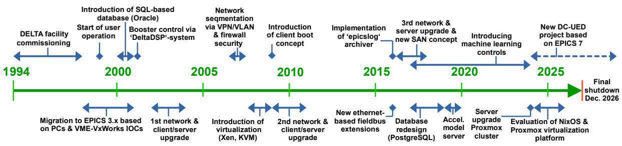

Figure 2: Major historical milestones.

Major Developments in the Past

Network Upgrades: In order to meet increasing bandwidth and security requirements, DELTA's computer network has undergone a series of network technology upgrades. From the original bus-type coaxial cabling (10Base5) to the introduction of star-shaped ethernet over twisted pair (10Base-T/100Base-TX) to today's Gbit/s backbone fiber optic technology (1000Base-SX/LX). In addition, the early implementation of highly configurable firewalls [9] and the introduction of network segmentation via virtual private networks (VPNs) and virtual local area networks (VLANs) significantly improved stability and performance while protecting against cyberattacks [10].

Virtualization: Adding more dedicated computers to the client-server system over time increased administrative overhead and created inefficient processor loads on the servers. Server consolidation was achieved by introducing virtualization techniques, beginning with Xen's paravirtualization [11] in 2009. This was later replaced by kernel-based virtualisation (KVM) [12], enabling multiple physical machines to be migrated to a single host. These changes supported legacy systems, simplified backup execution, and made it easier to create flexible testing environments [13].

Client Boot Concept: A centralized client boot concept was implemented to solve maintenance problems caused by the growing number of client PCs with different Linux kernels and configurations. All new clients were equipped with PXE-enabled network cards, which enabled them to boot their individual operating systems from a central server. This boot server provided customized but consistent Linux kernels and root file systems. Client-specific settings (e. g., monitor configurations) and log files were stored on the server, and common-use files were accessed via shared NFS directories. This centralized management, which is still in place today, leads to a more efficient distribution of processor load and simplifies client management, thereby improving consistency in the overall control system [13].

Database Redesign: To enable high-performance accelerator fault analysis and for machine learning studies, large amounts of data must be processed effectively. For this purpose, DELTA's SQL database (DB) was fundamentally redesigned in 2017 [10]. All critical machine data is continuously stored in EPICS records. A logger daemon (Linux systemd service) collects the data and archives it in the open-source object relational database management system PostgreSQL [14]. The TimescaleDB plug-in [15, 16] facilitates the storage of time-series data. This PostgreSQL extension increases analytical query performance while reducing data size. The improvement of query processing in large tables is achieved by utilizing PostgreSQL's fundamental table partitioning functionality. DB data can be read via the in-house command-line interface (CLI) 'epicslog', and an associated Python library. Both options are based on standard SQL commands [10].

Fieldbus Extension: The DELTA accelerator devices are controlled in many places via ESD-CAN nodes [7]. These are connected to VME-CAN cards, which are driven by VME-VxWorks/EPICS servers. This extensive VME middle layer has been replaced in newer installations (e. g., RF water cooling control, wiggler SR outlet vacuum chamber monitoring, magnet power supply interlock systems, etc.) by dedicated, less complex TCP/IP-based networks using a DIN rail-mountable, modular, compact, and expandable WAGO I/O-System [17]. It offers a wide range of multichannel digital and analog input/output modules for various signal types, as well as intelligent stepper motor controllers. In addition, it is fieldbus-independent and supports numerous communication protocols (e. g., Profibus, CANopen, EtherCAT, Modbus) and industrial ethernet standards [10].

Accelerator Modeling: Early accelerator design studies were carried out through offline simulation programs such as OPTICS [18], MAD [19] and ELEGANT [20]. Since 2001, Matlab [21] channel access libraries labCA [22] and mca [23] have enabled direct read/write access to EPICS PVs. This feature has been used extensively within the Matlab-based accelerator simulation toolbox [24, 25] and for prototype scripting [9]. In 2019, a Python/EPICS-based model server was integrated into the DELTA control system, enabling model calculations and storage ring simulations involving live machine data [26]. The Python toolkit 'Ocelot' [27] was employed for accelerator physics computations. The implementation is based on a modular and flexible client-server architecture according to the remote presentation model software design concept. This allows for parallel studies of storage ring models with various configurations. For example, online and offline simulations can be compared in real-time. Additionally, magnet settings obtained from simulation results can be transferred to the real machine, and vice versa. EPICS PVs manage the data exchange and are available transparently throughout the entire control system for further processing.

RECENT DEVELOPMENTS

The DELTA EPICS-based control system underwent continuous maintenance, including hardware replacements, software updates, network improvements, and the integration of new device controls (e. g., for the new 7 T superconducting wiggler [3] and the SPEED project [28]). A major upgrade involved replacing outdated server infrastructure and implementing a NixOS environment [29], a Linux distribution built on the Nix package manager [30].

Renewal of the Server Infrastructure

The majority of deprecated servers (out-of-warranty and maintenance) were replaced by two new 19-inch rack servers equipped with EPYC 7443P CPUs (24C/48T), purchased and installed in 2024. These servers also replace the legacy storage area network (SAN) system [10] and are configured as a redundant Proxmox cluster for high availability. The Proxmox virtual environment (PVE [31]) is an open-source virtualization platform based on the KVM hypervisor [12] and Debian Linux [32], enables features like high availability clustering, web-based management, and support for various storage backends. It integrates virtualization, containers, storage, and networking into one platform with a user-friendly interface. The cluster hosts virtual machines for general-purpose tasks and DELTA-specific software services like 'epicslog' queries, DB management, monitoring tools, and documentation platforms (see Fig. 3).

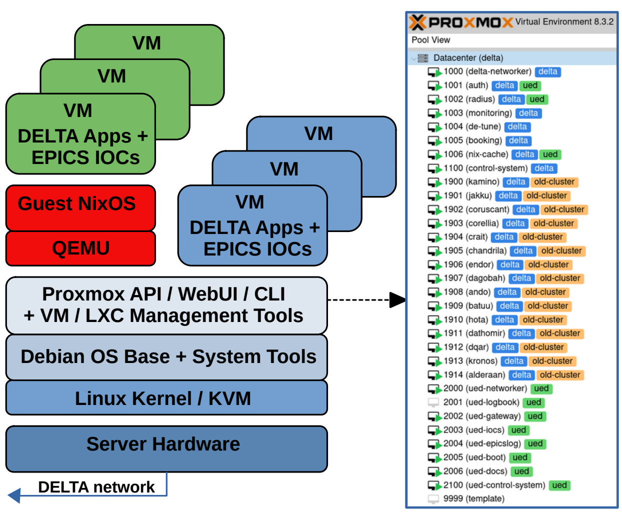

Figure 3: Proxmox virtual environment (PVE) schematic layout (left) and an example Proxmox UI showing typically running VMs (right).

An additional computer serves as both a Proxmox backup server (PBS [33]) and a third node in the cluster for failover capabilities. It complements the PVE by offering incremental, deduplicated VM backups integrated directly with PVE. Both systems utilize the ZFS file system [34], which combines volume management with advanced data protection features such as redundancy, snapshots, and replication to safeguard against physical errors and data corruption. The upgrade has significantly improved performance metrics such as DB access times, enhanced reliability, reduced maintenance efforts, and eliminated downtime across DELTA's IT infrastructure.

Evaluation of NixOS

Despite the use of IT automation tools such as Ansible [35], DELTA's Debian-based OS relies on manual, ad hoc managed configurations, which are error-prone due to the order of operations or unintended side effects during configuration changes. Therefore, NixOS has been evaluated as a replacement OS to simplify administrative efforts, improve reliability, and increase reproducibility.

NixOS uses the Nix domain-specific language (DSL) [29] to define complete system configurations, enabling reproducible setups stored under version control. Its repository includes over 100,000 pre-built binaries, reducing manual software compilation while allowing custom configurations. Dedicated modules were developed for EPICS packages and DELTA-specific Python libraries to ensure seamless integration into workflows. Currently several virtual machines are already running on NixOS (see Fig. 3), with plans to migrate others from Debian-based distributions. The standalone Nix package manager [30] also supports compatibility across other Linux systems without compromising reliability.

Machine Learning Applications

The DELTA facility provides an excellent testing environment for the development and proof of concept studies of novel, innovative machine learning (ML)-based control and optimization methods due to its high availability of beam time for accelerator physics research. Potential use cases span a wide range of applications [36]. Successful implementations to date include feed forward neural network (FFNN)-based orbit correction methods, which serve as an alternative to singular value decomposition (SVD)-based techniques (see Fig. 4) [37], as well as NN-based feedback systems designed to control the betatron tune and chromaticity values of the storage ring [38, 39]. Furthermore, the electron transfer rate from the booster synchrotron to the storage ring (injection efficiency) has been optimized automatically using ML-based algorithms [40]. Additionally, convolutional multilayer neural networks (CNNs) have been applied in the analysis of CHG-based radiation spectra [41].

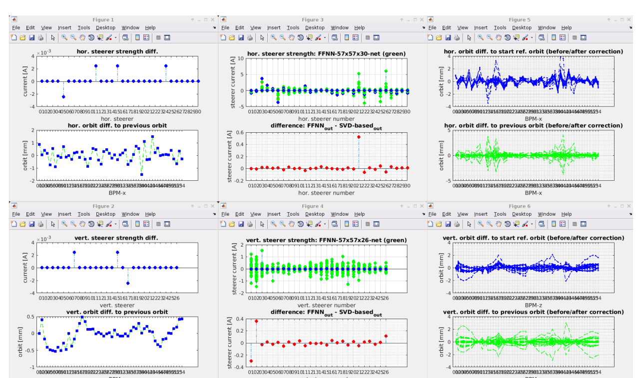

Figure 4: Sample plots during ML-based (FFNN-57x57x30/26) orbit correction measurements for the horizontal (top row) and vertical (bottom row) orbit plane.

NEW PROJECTS

A new facility dedicated to ultrafast electron diffraction (UED) in the MeV range is currently under development and aims to enable direct visualization of structural dynamics at femtosecond resolution (see Fig. 1, green area). In the final stage, this cutting-edge system will feature two 5-MeV electron sources powered by advanced photocathode guns designed specifically for high temporal and spatial precision experiments across various fields such as material science and molecular biology. Commissioning of the facility is planned for 2027.

Parallel to the development work on the MeV-UED project, a demo setup utilizing a 100 keV DC electron gun has entered an operational testing phase (see Fig. 1, red area). It was originally built at the University of Duisburg/Essen (Germany) and was transferred to DELTA in 2024, where it is currently being upgraded [42]. This prototype is used for first electron diffraction experiments and serves as an testbed for the planned MeV-UED setup.

The foundation of both UED projects' control systems relies on EPICS 7 (successor to EPICS 3), a robust toolkit widely used within accelerator-based research infrastructures worldwide due to its scalability and modular design capabilities [43].

DC-UED Demo Project

Operating System: Most of the DC-UED control software runs in VMs, utilizing two PVE hypervisors and a PBS installed on DELTA's newly established standard server hardware (see previous chapter: Renewal of the Server Infrastructure). Other DC-UED VMs use NixOS to host a variety of EPICS 7 IOCs (see Fig. 3 and previous chapter: Evaluation of NixOS).

EPICS 7: The EPICS control system is used for the DC-UED experiment to provide a user-friendly interface and enable remote operation. EPICS 7, with its 'pvAccess' protocol [44], builds on DELTA's established software while introducing enhanced capabilities. The system consists of several EPICS input/output controllers (IOCs) that utilize community libraries (e. g., asyn, streamdevice, areaDetector, Modbus, etc.) for device control, such as handling cameras, temperature sensors, power supplies, and stepper motors.

The novel 'pvAccess' protocol is primarily used for communication due to its improved support for features like long strings and NDArray-based images. High-level automation is achieved through Python scripts using the 'p4p' library [45] for tasks like image analysis and more sophisticated machine control. All required EPICS libraries and dependencies are managed via the Nix package manager [30], ensuring stable and reproducible development environments supported by the 'deltapkgs' git repository.

Device Control: As a diffraction experiment, the main data comes from the Andor iXon detector camera [46]. It is managed via the areaDetector library [47], which provides plugins for image processing and exports images using the 'pvAccess' protocol in a standard format (normative type NTNDArray). A Python-based 'pvAccess' server is under development for real-time image analysis, with results seamlessly integrated into the EPICS control system.

The Wago I/O-System 750 [17] handles basic I/O control, utilizing Modbus modules for digital and analog I/O, temperature channels, and stepper motor controllers that manage settings for mirrors, sample holders, wave plates, and delay stages. Historical EPICS values are archived using the 'epicslog' software toolkit with TimescaleDB [15, 16] for time-series data storage and SQL-based querying (see previous chapter: Database Redesign).

USB devices like phase shifters and attenuators are controlled through a NixOS-based IOC server using custom asyn drivers.

User Interface: The DC-UED experiment's main user interface (UI) operates on a network-booted NixOS system, that ensures immutability and reproducibility while providing persistent storage for settings (e. g. monitor setup) and detector images. It provides all necessary programs and libraries to run control UI panels and operator written Python scripts. The system is defined entirely in text files within a git repository, enabling version control and easy revocation if needed. Its immutability ensures local changes don't affect configuration, while persistent storage is provided via network-mounted directories for settings, global data, and detector image storage.

Control panels use the 'deltadm' display manager, which was developed at DELTA and rewritten in C++ with pvxs [48] for EPICS-7 integration. The panels are built using the Qt framework [49] and the Qwt widget library, as well as the Qt Designer WYSIWYG-editor for ease of creation by operators (see Fig. 5). Features include basic widgets, subpanels, macros for templates, plugins for panel providers, and specialized widgets like image tools. Future plans include specialized widgets and deeper integration of 'epicslog' for data visualization.

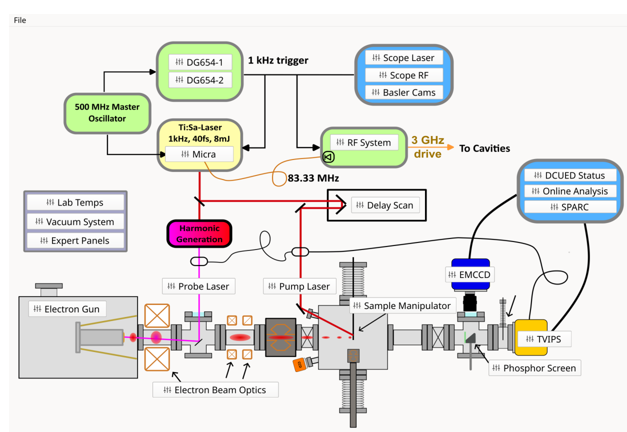

Figure 5: GUI of the control panel for the DC-UED demo experiment (courtesy of UED work group).

IT Concept for the planned MeV-UED Project

The data acquisition system (DAQ) for the proposed UED beamline is planned to be built around the JUNGFRAU 1M detector module, which will operate at a repetition rate of 1 kHz [50]. The detector is expected to generate 1-megapixel images with a bit depth of 16 bits, resulting in an image size of 2 MiB. At this repetition rate, it will produce a data rate of 2000 MiB/s, posing significant challenges for both IT infrastructure and fast live data analysis.

To address these demands, state-of-the-art computing hardware and software are planned to be employed. The system will integrate field-programmable gate array (FPGA) boards, advanced NVMe storage drives, and machine learning methods. A dedicated detector server is intended to be equipped with an HPC-FPGA compute card, a graphics processing unit (GPU), and high-speed NVMe SSD storage. The FPGA card will utilize PSI-developed JUNFRAUJOCH firmware [50] to ingest the data stream from the detector module and pre-process it for the server CPU. Additionally, it will perform real-time spot-finding on portions of image data to provide immediate feedback to operators.

Efficient data compression and reduction are also anticipated as critical tasks to minimize storage requirements. Therefore, the server CPU will collect images from the FPGA card and store them in a standardized compressed format, widely regarded as the 'gold standard' for macromolecular crystallography diffraction data, on local SSDs. Portions of this data may also be sent to GPUs for further live analysis using advanced techniques such as machine learning.

For comprehensive offline analysis, all measurements are planned to be archived for at least six months. Even after applying compression algorithms, this high-volume dataset is expected to require several petabytes of usable storage capacity. To achieve this within limited server rack space constraints, modern high-capacity servers equipped with at least 60 drives each are required.

Managing distributed datasets across multiple servers while ensuring redundancy against hardware failures will present additional challenges. To address these effectively while maintaining full control over stored information, DELTA plans to employ scalable clustered storage systems based on the open-source 'Ceph' software platform [51]. This solution is designed to enable seamless scalability for future requirements.

As with previous projects like DC-UED, the control system responsible for monitoring sensors and controlling devices is planned to be hosted on KVM-based VMs and in Linux containers [52]. Resource management will be handled by the Proxmox open-source virtualization platform, which is expected to ensure superior compartmentalization while facilitating software maintenance workflows. Furthermore, this setup aims to provide an efficient backup and restore mechanism essential for operational reliability.

Finally, enterprise-grade network switches offering bandwidths up to 25 Gbit/s per port will enable seamless communication between servers. This planned infrastructure is expected to support live analysis capabilities for measurement data while enabling rapid real-time feedback control. Additionally, the high-bandwidth network will facilitate efficient data transfer between storage systems and computational nodes, ensuring scalability and reliability for future requirements of the UED beamline project (see Fig. 6).

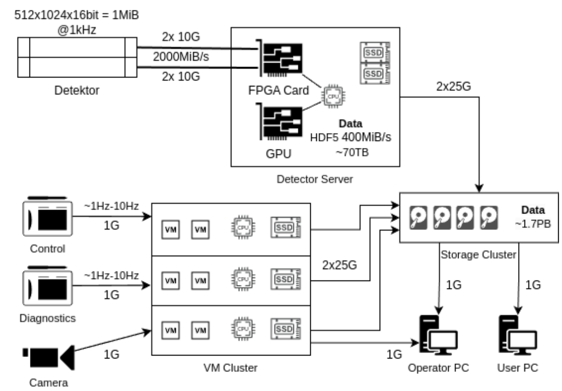

Figure 6: Planned data acquisition system for the proposed MeV-UED project.

ACKNOWLEDGEMENT

We would like to thank all technicians, engineers, scientists, funding institutions, local authorities, cooperating universities, and research institutes worldwide for their commitment and continuous support, which have made DELTA a success over the past 30 years.

REFERENCES

[1] DELTA Homepage, `https://delta.tu-dortmund.de/`.

[2] M. Tolan, T. Weis, C. Westphal, and K. Wille, "DELTA: Synchrotron light in nordrhein-westfalen", in *Synchrotron Radiat. News*, vol. 16, no. 2, pp. 9–11, Mar. 2003. doi:10.1080/08940880308603005

[3] D. Schirmer *et al.*, "Status of the DELTA synchrotron light source", presented at IPAC'25, Taipei, Taiwan, Jun. 2025, paper MOPM072, unpublished.

[4] EPICS, `https://epics-controls.org/`.

[5] D. Schirmer, E. Kasel, B. Keil, and D. Zimoch, "The Evolution of the DELTA Control System", in *Proc. ICALEPCS'01*, San Jose, CA, USA, Nov. 2001, pp. 292–294.

[6] VxWorks, real-time OS, `https://www.windriver.com/products/vxworks/`.

[7] ESD, I/O modules, `https://esd.eu/en/`.

[8] B. Keil, "A Unified Distributed DSP-Based Beam Diagnostics and Global Feedback System for Ramped Electron Storage Rings: Development, Construction and Applications", dissertation, TU Dortmund, Germany, 2003.

[9] D. Schirmer, P. Hartmann, O. Kopitetzki, G. Schünemann, P. Towalski, "Status of the DELTA Control System", in *Proc. ICALEPCS'07*, Oak Ridge, TN, USA, Oct. 2007, pp. 356–358.

[10] D. Schirmer, A. Althaus, P. Hartmann, and D. Rohde, "Control System Projects at the Electron Storage Ring DELTA", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 1361–1365. doi:10.18429/JACoW-ICALEPCS2017-THPHA013

[11] Xen paravirtualization, https://xenproject.org/.

[12] Kernel-based Virtual Machine (KVM), https://linux-kvm.org/.

[13] D. Schirmer, O. Kopitetzki, and P. Towalski, "First Experience with the Introduction of Virtualization Techniques into the DELTA Control System", in *Proc. ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper THP038, pp. 741–743.

[14] PostgreSQL, open-source relational database, https://www.postgresql.org/.

[15] TimescaleDB plug-in, PostgreSQL extension (open-source), https://github.com/timescale/timescaledb/.

[16] Timescale, Postgres extension for time-series (TigerData), https://www.tigerdata.com/time-series/.

[17] Wago I/O-System 750/753, https://www.wago.com/de-en/automation-technology/discover-io-systems/750.

[18] K. Wille, "Manual: OPTICS Version 4.00", Dortmund University, Germany, December 2004, unpublished.

[19] H. Grote, F. C. Iselin, "The MAD Program (Methodical Accelerator Design)", Version 8.19, CERN, Geneva, Switzerland, 1996.

[20] M. Borland, "elegant: A Flexible SDDS-Compliant Code for Accelerator Simulation", in *Proc. ICAP'2000*, Darmstadt, Germany, Sep. 2000. doi:10.2172/761286

[21] MATLAB/SIMULINK, Release 2017b, The MathWorks Inc., Natick, Massachusetts, United States.

[22] labCA library, Matlab/Scilab channel access (EPICS) interface, https://www.slac.stanford.edu/grp/ssrl/spear/epics/extensions/labca/manual/node30.html.

[23] mca library: Matlab channel access (EPICS) interface, https://github.com/epics-modules/mca/.

[24] A. Terebilo, "Accelerator Modeling with Matlab Accelerator Toolbox", in *Proc. PAC'01*, Chicago, USA, 2001, pp. 3203–3205. doi:10.1109/PAC.2001.988056

[25] atcollab, Accelerator Toolbox (AT) framework, https://github.com/atcollab/at/.

[26] D. Schirmer and A. Althaus, "Integration of a Model Server into the Control System of the Synchrotron Light Source DELTA", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1427. doi:10.18429/JACoW-ICALEPCS2019-WEPHA137

[27] Ocelot, Python-based toolkit for accelerator physics simulations, https://github.com/ocelot-collab/ocelot/.

[28] S. Khan *et al.*, "SPEED: Worldwide first EEHG implementation at a storage ring", in *Proc. IPAC'23*, Venice, Italy, May 2023, pp. 1057–1060. doi:10.18429/JACoW-IPAC2023-MOPM032

[29] NixOS, https://nixos.org/.

[30] nixpkgs, Nix packages collection, https://github.com/NixOS/nixpkgs/.

[31] Proxmox Virtual Environment (PVE), https://proxmox.com/en/products/proxmox-virtual-environment/overview/.

[32] Debian Linux OS, https://www.debian.org/.

[33] Proxmox Backup Server (PBS), https://www.proxmox.com/en/products/proxmox-backup-server/overview/.

[34] OpenZFS, https://openzfs.org/wiki/Main_Page/.

[35] Ansible: Open source IT automation engine, https://www.redhat.com/en/ansible-collaborative/.

[36] A. Edelen *et al.*, "Opportunities in Machine Learning for Particle Accelerators", arXiv:1811.03172 [physics.acc-ph], 2018. doi:10.48550/arXiv.1811.03172

[37] D. Schirmer, "A machine learning approach to electron orbit control at the 1.5 GeV synchrotron light source DELTA", *J. Phys.: Conf. Ser.*, vol. 2420, no. 1, p. 012069, Jan. 2023. doi:10.1088/1742-6596/2420/1/012069

[38] D. Schirmer, "Machine Learning Applied to Automated Tunes Control at the 1.5 GeV Synchrotron Light Source DELTA", in *Proc. IPAC'21*, Campinas, Brazil, May 2021, pp. 3379–3382. doi:10.18429/JACoW-IPAC2021-WEPAB303

[39] D. Schirmer *et al.*, "Machine Learning Methods for Chromaticity Control at the 1.5 GeV Synchrotron Light Source DELTA", in *Proc. IPAC'22*, Bangkok, Thailand, Jun. 2022, pp. 1141–1144. doi:10.18429/JACoW-IPAC2022-TUPOPT059

[40] D. Schirmer, A. Althaus, S. Hüser, S. Khan, and T. Schüngel, "Machine learning-based optimization of storage ring injection efficiency", *J. Phys.: Conf. Ser.* vol. 2687, no. 6, p. 062033, Jan. 2024. doi:10.1088/1742-6596/2687/6/062033

[41] A. R. Krishnan *et al.*, "Spectro-temporal properties of coherently emitted ultrashort radiation pulses at DELTA", in *Proc. of FEL'22*, Trieste, Italy, Aug. 2022, pp. 317–321. doi:10.18429/JACoW-FEL2022-TUP71

[42] L. Bölte, "Upgrading and Recommissioning a 100 keV Ultrafast Electron Diffraction Instrument at DELTA", Master's thesis, TU Dortmund University, Germany, 2025.

[43] EPICS 7, https://epics-controls.org/resources-and-support/base/epics-7/.

[44] pvAccess library, https://docs.epics-controls.org/en/latest/pv-access/protocol.html.

[45] p4p library, Python bindings for pvAccess network client/server, https://github.com/epics-base/p4p/.

[46] Andor iXon detector camera, https://andor.oxinst.com/products/ixon-emccd-camera-series/ixon-ultra-888/.

[47] areaDetector library, https://github.com/areadetector/.

[48] pvxs, PVA protocol client/server library and utilities, https://epics-base.github.io/pvxs/.

[49] Qt, GUI framework, https://www.qt.io/product/framework/.

[50] F. Leonarski *et al.*, "Jungfraujoch: hardware-accelerated data-acquisition system for kilohertz pixel-array X-ray detectors", *J. Synchrotron Radiat.*, vol. 30, no. 1, pp. 227–234, Jan. 2023. doi:10.1107/s1600577522010268

[51] Ceph, Distributed storage system, https://ceph.io/en/.

[52] LXC, Linux container, https://linuxcontainers.org/.