High Power Laser Science and Engineering, (2023), Vol. 11, e44, 7 pages.
doi:10.1017/hpl.2023.17

HIGH POWER LASER
SCIENCE AND ENGINEERING

RESEARCH ARTICLE

Tango Controls and data pipeline for petawatt laser experiments

Nils Weiße, Leonard Doyle, Johannes Gebhard, Felix Balling, Florian Schweiger, Florian Haberstroh, Laura D. Geulig, Jinpu Lin, Faran Irshad, Jannik Esslinger, Sonja Gerlach, Max Gilljohann, Vignesh Vaidyanathan, Dennis Siebert, Andreas Münzer, Gregor Schilling, Jörg Schreiber, Peter G. Thirolf, Stefan Karsch, and Andreas Döpp

Ludwig-Maximilians-Universität München, Garching, Germany

(Received 1 December 2022; accepted 16 February 2023)

Abstract
The Centre for Advanced Laser Applications in Garching, Germany, is home to the ATLAS-3000 multi-petawatt laser, dedicated to research on laser particle acceleration and its applications. A control system based on Tango Controls is implemented for both the laser and four experimental areas. The device server approach features high modularity, which, in addition to the hardware control, enables a quick extension of the system and allows for automated data acquisition of the laser parameters and experimental data for each laser shot. In this paper we present an overview of our implementation of the control system, as well as our advances in terms of experimental operation, online supervision and data processing. We also give an outlook on advanced experimental supervision and online data evaluation – where the data can be processed in a pipeline – which is being developed on the basis of this infrastructure.

Keywords: data processing; high-power laser experiments; laser–plasma acceleration; online diagnostics

1. Introduction

Petawatt laser facilities[1] enable the study of a plethora of phenomena, ranging from fundamental physics[2,3] and over laser-driven radiation sources[4–8] to applications with high societal relevance, such as medical imaging[9,10] and fusion energy[11]. However, the wide range of applications also results in a high variability of experimental configurations, and frequently shifting experimental requirements demand continuous modifications to the used technical infrastructure. The integration of new devices, such as motors, cameras or special instruments, into an experimental setup can be a time-consuming task for experimental physicists. Reliable control and supervision of all implemented devices is the first step in conducting any successful experiment. Thus, having a highly dependable and customizable server infrastructure that enables data acquisition and control of the entire experiment can be of great benefit for daily experimental work. However, this server infrastructure should still be easy to maintain and to work with because it must be used by many different employees and collaborating scientists from other facilities with specific needs and varying levels of programming experience. If these requirements are met, the server infrastructure then allows for even more advanced steps, such as online diagnostics, that enable the automation of the experiment and the implementation of safety features. In addition, a common server infrastructure standardizes the acquired data and streamlines the evaluation process into a data pipeline. The ongoing development of the data pipeline is planned to further enable advanced evaluation and control methods enabled by machine learning[12].

In this paper we are going to review the implementation of a control system that satisfies the requirements outlined above at the Centre for Advanced Laser Applications (CALA), its petawatt laser and experimental chambers[13,14]. The paper is structured as follows. Section 2 outlines the experimental infrastructure at CALA and the need for a common server infrastructure. An overview of the implemented control system, Tango Controls[15], and its basic features in terms of supervision and experimental operation is given

Correspondence to: Nils Weiße, Ludwig-Maximilians-Universität München, Am Coulombwall 1, 85748 Garching, Germany. Email: nils.weisse@physik.uni-muenchen.de

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press

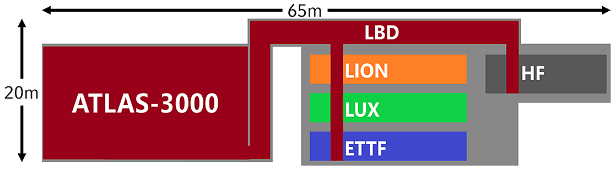


Figure 1. Sketch of the experimental infrastructure at CALA: the ATLAS-3000 laser provides the experimental chambers LION, LUX, ETTF and HF with multi-petawatt laser pulses via the LBD.

in Section 3. We present our own software development in these fields and show how it improved daily operation and supervision in Section 4. The developed data acquisition and processing pipeline, which made data handling faster and more streamlined, is shown in Section 5. An outlook on future software development is given in Section 6.

2. ATLAS-3000 laser and experimental infrastructure

The CALA in Garching, Germany, hosts the ATLAS-3000 laser. With up to 60 J pulse energy after compression to 20 fs pulse length and 1 Hz repetition rate, the laser offers an exclusive combination of multi-petawatt peak power and repetition rate. A custom-built laser beam delivery (LBD) supplies four different experimental areas with high-peak-power laser pulses. Each of these experimental areas is dedicated to different flavours of laser applications, namely research on electron acceleration (Electron and Thomson test facility, ETTF), laser-ion acceleration (LION), high fields (HFs), and novel X-ray sources (LUX). The experimental infrastructure is sketched in Figure 1. Each of the individual research areas requires different specific instruments to successfully conduct an experiment. However, there are also devices commonly used in all experimental chambers, such as cameras, energy meters, spectrometers and different motors. Furthermore, the data from the ATLAS-3000 laser are relevant for all groups. Thus, there is interest at CALA in having a server infrastructure that enables access to the data required by each individual group. A common laser infrastructure allows for easy integration of new instruments and provides researchers with data in a standardized format.

3. Tango Controls at the Centre for Advanced Laser Applications

We have adopted Tango Controls[15] as the main server infrastructure for supervisory control and data acquisition in recent years. As open-source software, it can be easily implemented and adapted. Tango Controls establishes an abstraction of the communication protocol that allows the use of different operating systems (Windows, Unix, MacOS) and the implementation of devices by applying different programming languages (e.g., Python, Java, MATLAB, C++).

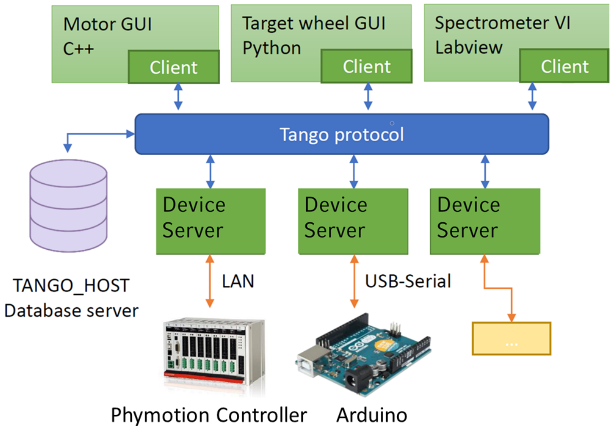


Figure 2. Sketch of the Tango server infrastructure. Device servers allow communication with physical devices.

In this way, a wide variety of instruments can be included in the control system. At CALA mainly the Python version of Tango, PyTango, is used. The general structure of PyTango, as well as its inherent capacities in server creation, data acquisition and instrument supervision, will be presented in this section.

An overview of a typical Tango server infrastructure is sketched in Figure 2. Each physical device, such as a motor control panel or microcontroller, is an instance of a device server. Communication between the device server and the physical device can be achieved in multiple ways, for example, by means of a local area network (LAN) or Universal Serial Bus (USB)–serial connection. The device server is most commonly written in C++ or Python. In this example, all device servers are in control of a physical device. However, device servers could also be used to only evaluate data, as will be described in the following sections. Instances of the device servers can be run on different physical computers. This includes, but is not limited to, any Windows or Linux computer as well as ARM systems, such as the Raspberry Pi. Each instance is registered on a common database server, which is the central point for establishing new connections between two devices. Communication between devices is carried out as a peer-to-peer (p2p) connection using the Tango protocol.

A client such as a graphical user interface (GUI) running on a computer can access all devices via such a p2p connection. In its current implementation, the server infrastructure is highly decentralized. The main laser system, as well as each of its four experimental areas, has independently operating database servers. This is done to reduce the risk of interfering with group-specific configurations and requirements.

Setting up and operating the different components of a working PyTango server infrastructure is relatively easy to learn through the use of software provided in the Tango package. Programming a device server is supported by the

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press

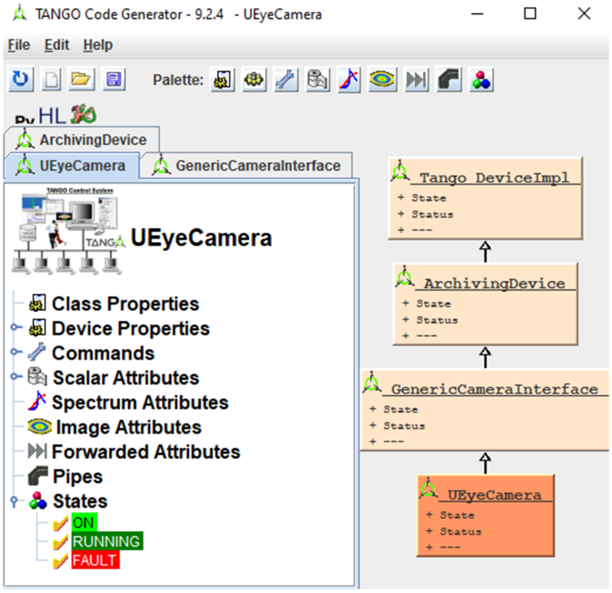
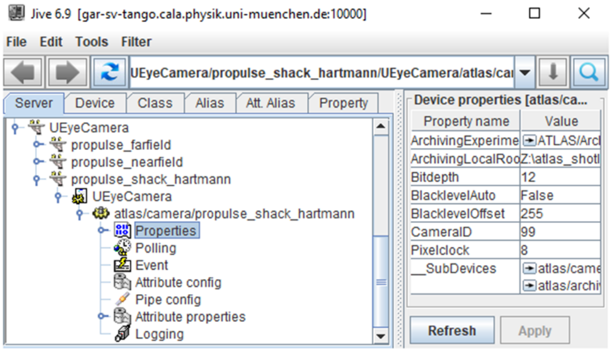


Figure 3. Screenshot showing the GUI of the Tango code generator called Pogo. Pogo allows the definition of properties, commands, attributes and states for a device server.

Tango code generator called Pogo. Pogo can be used to generate a Python class that contains dummy functions for a device server. Figure 3 shows the GUI of Pogo. In this example, a Python device server called 'UEyeCamera' is created to control Ethernet cameras (IDS 'UEye'). In order to control physical devices, such as cameras, commands, device properties, attributes and states are used:

- **commands** are used to execute functions, for example, to turn a camera on or off;
- **device properties** are constant values for one device, for example, a static Internet Protocol (IP) address corresponding to a camera;
- **scalar attributes** are single values that can be read and written, for example, the exposure time of a camera;
- **spectrum attributes** describe a 1D array data type that can be read and written, for example, the 1D measurement data of a spectrometer;
- **image attributes** describe a readable and writable 2D array, for example, a camera image;
- **states** are used to provide users or other programs with information about the device server, for example, whether it is currently running correctly or a fault has occurred.

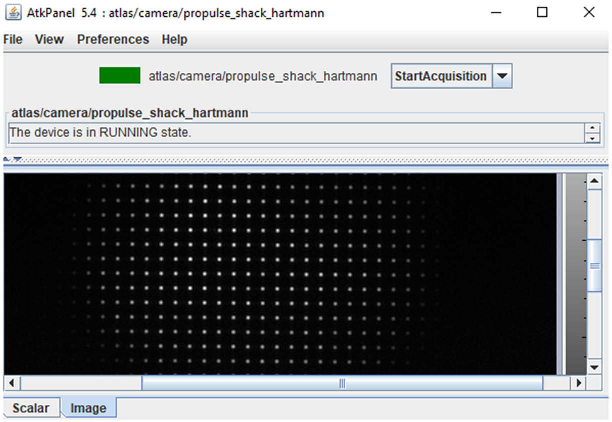
Furthermore, Pogo allows a device server to inherit attributes and commands from a parent class. In the example shown in Figure 3, the device server with the name UEyeCamera inherits a class called GenericCameraInterface that contains many attributes, functions and commands



Device properties [atlas/ca...

Property name	Value
ArchivingExperime	ATLAS/Arc
ArchivingLocalRoo	Z:\atlas_shot
Bitdepth	12
BlacklevelAuto	False
BlacklevelOffset	255
CameraID	99
Pixelclock	8
__SubDevices	atlas/came atlas/archi

Figure 4. Screenshot of the Jive GUI showing a list of camera device servers and their device properties.


AtkPanel 5.4 : atlas/camera/propulse_shack_hartmann
File View Preferences Help
atlas/camera/propulse_shack_hartmann
StartAcquisition
atlas/camera/propulse_shack_hartmann
Scalar
Image

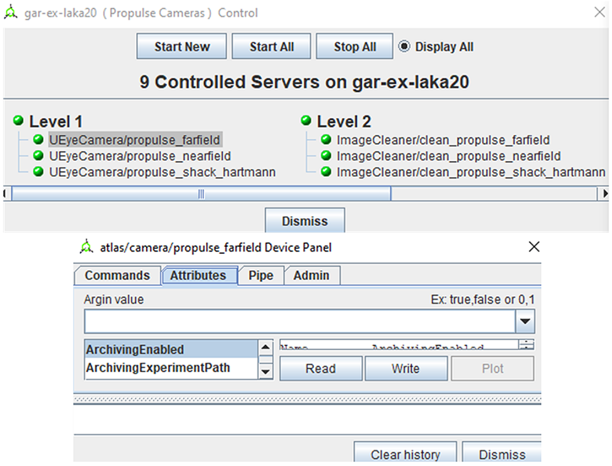
The device is in RUNNING state.

Figure 5. Display of the image attribute of a Shack-Hartmann camera in Jive.

applicable to all types of cameras, for example, setting the exposure time or acquiring images in a loop. This increases the readability of different camera classes between research groups. The UEyeCamera class, like all devices in our infrastructure, inherits from the ArchivingDevice class. Due to this, all devices share common attributes defined in the ArchivingDevice class that we use to distribute and store data at the time of a laser shot, as explained in Section 5.

Each device server has to be registered in the database server to be able to create device server instances. This and all device server instances can be managed using the tool Jive. Figure 4 shows parts of the GUI of Jive. The GUI gives the operator some basic information about the registered device servers, such as their state or when the device server was last online. Jive also allows the values of the device properties to be defined or changed (as shown in Figure 4) and the execution of all the commands defined for the device server. Furthermore, all attributes of any server (e.g., a camera image) can be viewed, as shown in Figure 5. The advantages of implementing physical devices, such as cameras, into a server infrastructure become immediately clear when looking at Jive. One key advantage of making a hardware device available in Tango is that it can afterwards

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press



gar-ex-laka20 (Propulse Cameras) Control

9 Controlled Servers on gar-ex-laka20

Level 1
- UEyeCamera/propulse_farfield
- UEyeCamera/propulse_nearfield
- UEyeCamera/propulse_shack_hartmann

Level 2
- ImageCleaner/clean_propulse_farfield
- ImageCleaner/clean_propulse_nearfield
- ImageCleaner/clean_propulse_shack_hartmann

atlas/camera/propulse_farfield Device Panel

Argin value Ex: true,false or 0,1

ArchivingEnabled
ArchivingExperimentPath

Name ArchivingEnabled

Figure 6. Display of the Astor user interface.

be accessed by many clients simultaneously. Where in the past operators often had to, for example, close a camera in one program to re-open it in another, the image can now be accessed at the same time by Jive, a custom GUI and the evaluation pipelines presented in Section 5.

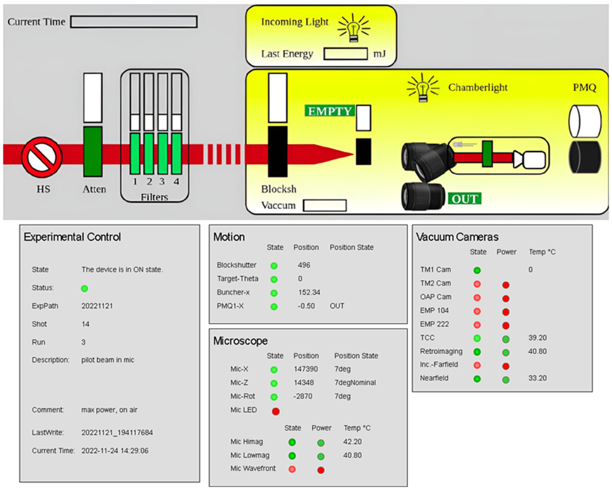
For the general management of Tango device servers, daily operation is further eased by a program called Astor that shows the running state for each device server. When configured correctly, Astor can control or restart device servers running on a remote PC. Figure 6 shows the user interface of Astor. In this case, three device servers for Ethernet cameras (‘UEyeCamera’) are running on a computer, as well as device servers used for the image and data processing described in Section 5.

Even though Jive and Astor provide some basic functionality for the operation and supervision of a device server, custom-made code and GUIs are still required in the case of CALA, as we will discuss in the next section.

4. Operation and supervision at the Centre for Advanced Laser Applications

Conducting daily experiments on, for example, particle acceleration with a petawatt laser, requires considerable experience and care. The success of an experiment can only be ensured when all important parameters can be safely and reliably controlled and supervised by the operator. By implementing all important devices into a common server infrastructure, this was made easier to achieve. At the time of writing, there are more than 50 motors, 20 cameras, multiple spectrometers and energy meters and many more special instruments integrated as device servers in the PyTango server infrastructure at CALA.

In addition, custom GUIs ease the operation and supervision of such a complex technical infrastructure by allowing the operator to focus on the most important components of the system. A notable example of a GUI that does this is the CALA overview dashboard that is operated in the laser control room, depicted in Figure 7. A dedicated user interface retrieves live data from multiple Tango devices and displays them accurately, improving the operator’s decision-making. The current approach to implementing the overview dashboard consists of a combination of Taranta[16] and Plotly Dash[17]. Taranta (formerly known as WebJive) is a Tango native open-source web interface providing the so-called widgets to create dashboards interactively per drag and drop. Those widgets connect to Tango devices to monitor the system by displaying and plotting attributes or controlling the system by executing commands and setting attributes. Furthermore, since Taranta is open source, we can adjust it to our needs and create individual widgets optimized for our purposes. Plotly Dash is a Python package that includes nearly all HyperText Markup Language (HTML) components and allows for more complex processes, such as displaying custom graphics. Furthermore, it is still able to connect to Tango with some workarounds. This dashboard pictographically summarizes the current state of the ATLAS-3000 laser and the experimental chamber, here LION, in the control room. It includes beamline information, such as the attenuation state and filter settings of the laser beam. Also, important parameters such as incoming energy of the laser pulses and vacuum pressures are displayed.



Experimental Control

State	The device is in ON state.
Status:	●
ExpPath	20221121
Shot	14
Run	3
Description:	pilot beam in mic
Comment:	max power, on air
LastWrite:	20221121_194117684
Current Time:	2022-11-24 14:29:06

Motion

	State	Position	Position State
Blockshutter	●	496	
Target-Theta	●	0	
Buncher-x	●	152.34	
PMQ1-X	●	-0.50	OUT

Microscope

	State	Position	Position State
Mic-X	●	147390	7deg
Mic-Z	●	14348	7degNominal
Mic-Rot	●	-2870	7deg
Mic LED	●		

	State	Power	Temp °C
Mic Himag	●	●	42.20
Mic Lowmag	●	●	40.80
Mic Wavefront	●	●	

Vacuum Cameras

	State	Power	Temp °C
TM1 Cam	●		0
TM2 Cam	●	●	
OAP Cam	●	●	
EMP 104	●	●	
EMP 222	●	●	
TCC	●	●	39.20
Retroimaging	●	●	40.80
Inc.-Farfield	●	●	
Nearfield	●	●	33.20

Figure 7. CALA overview dashboard for the supervision of laser parameters and the LION experimental chamber.

Furthermore, operating experiments can be made safer for the technical infrastructure if all instruments are included in a common server infrastructure. While human safety is ensured by a programmable logic controller (PLC) that monitors external laser and radiation safety, experimental equipment can still be damaged. For example, accidental exposure of cameras to full-power laser beams or even attenuated but focused beams can easily result in pixel damage. Even more expensive equipment, such as optical gratings or mirrors, can be destroyed if the wrong settings for the laser’s

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press

spectral content are selected by the operator. Such events can potentially shut down the experimental operation for months. Implementing automated safety features becomes easier when all important devices report their status in a common control system. In this way, the risk of a camera being destroyed by an unattenuated laser pulse was made highly unlikely when Tango Controls is in use. Every time the operator requests an unattenuated laser shot, the program checks if all cameras have been moved out of the laser's path. Since both the attenuation of the laser and the position of the motorized stage on which the camera is placed are attributes accessible via Tango, this became very easy to program. Currently, more advanced safety systems are being developed on the basis of the Tango server infrastructure. An important example of this is a safety device consisting of a spectrometer and a shutter that checks if the laser spectrum is still broad enough for amplification and compression after modification with an ultrafast pulse-shaping system. Only then is the laser allowed to pass through the amplification chain, preventing the destruction of optical components by a too narrow spectrum.

Finally, automation of repetitive steps during an experiment becomes more easily possible with this infrastructure. Repetitive tasks, such as taking 100 images in a row for a focus analysis while simultaneously scanning one parameter, are difficult to perform by hand without making simple mistakes. The implementation of analysis features in the GUI massively increases the reliability of the results and enables the experimentalist to focus on the experiment rather than infrastructure.

5. Data acquisition and processing pipeline at the Centre for Advanced Laser Applications

Given the 1 Hz repetition rate of the ATLAS-3000 laser, automated data acquisition is necessary to handle data from various diagnostic devices, such as energy meters, spectrometers and cameras. The multitude of devices requires a structured data acquisition to keep track of the data and synchronize it so that each measurement corresponds to a specific laser shot. As mentioned in Section 3, all device servers inherit attributes and commands from an archiving device server and subscribe to an experimental control server. By inheriting common attributes and commands, all data can be standardized in terms of file and date format. By subscribing to the experimental control server, a device server can receive information such as the current laser shot number, time and location of the storing folder from the experimental control server that sets these parameters.

Data acquisition and synchronization within the pulsed laser system are based on a command structure. The flow of commands for the CALA trigger system is sketched in Figure 8. The operator can set the laser and target parameters using a GUI. Once the parameters are set, a shot can be requested, which triggers the experimental control server to store and update meta-data such as the current shot number. The request for a shot is then forwarded to the laser device server that stores further meta-data and asks the laser control PLC for a shot. All device servers that are subscribed to the experimental control server will activate their trigger system, so that the next time a hardware trigger is received, the measurements are stored. If the shot is not blocked by external safety interlocks, an electrical trigger is sent 200 ms before the laser pulse arrives at the target.

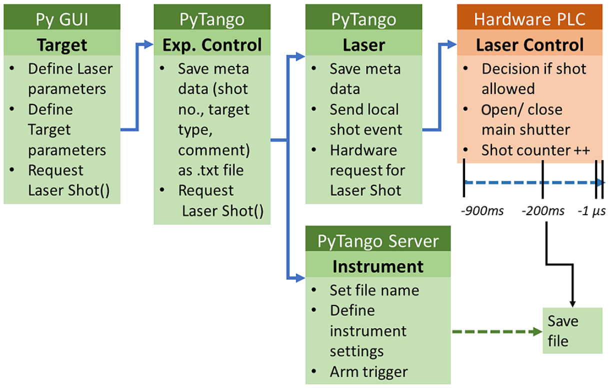


Figure 8. Flow of commands for the CALA trigger system.

The integration of physical devices into the Tango server infrastructure allows for a faster online evaluation of important physical parameters by channelling the acquired data into a data processing pipeline. Every time a device server acquires new data, it automatically emits a 'data ready' event. Other device servers that are capable of analysing data can subscribe to this event, creating a pipeline. One such example is an image processing pipeline, where a camera image is received by an ImageCleaner device. This device will subtract a stored dark image and apply some cropping and filtering if desired. The output image is again accessible as a Tango attribute and an ImageCalculator device can receive each new cleaned output image to calculate the image centre of the mass, peak and integral. When applied, for instance, to a far-field monitor camera, this allows for an online evaluation of the beam pointing. When storing only the calculated quantities, this also allows a significant reduction in storage requirements for certain cameras.

In addition, improved diagnostics and data acquisition can also provide insight into the system's operation. As an example, the temperature at essential parts of the laser, for example, the temperature of optical tables, crystals, cooling water and the ambient air temperature of the ATLAS-3000 laser, is measured using multiple temperature sensors, which are integrated into the Tango system. Because of this, data from these diagnostics can easily be combined with various laser output parameters to measure correlations between temperature and laser performance. As an example, Figure 9 shows a plot of the air temperature against the corresponding

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press

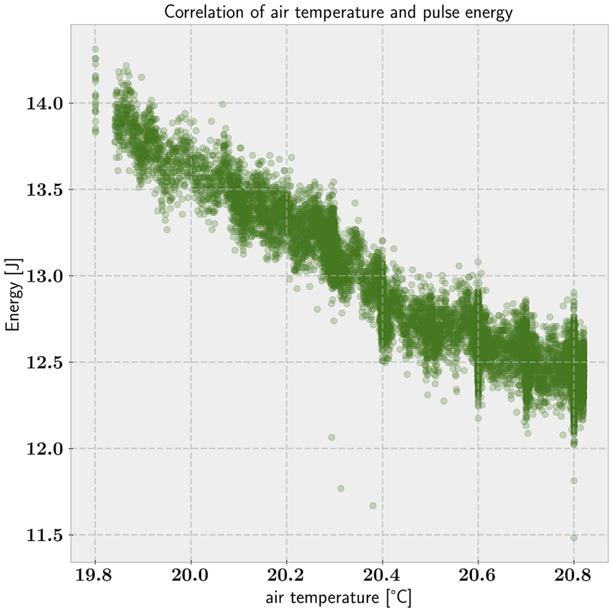


Figure 9. Correlation of laser pulse energy with air temperature measured at the amplification crystal.

laser pulse energy, measured at the last amplification crystal. The plot shows a clear (anti-)correlation between air temperature and laser energy.

These developments lay the foundation for more advanced machine learning-based data processing and control techniques[12] by providing stored data, as well as live measurements in a standardized format. For instance, researchers at CALA have been actively working on developing control systems based on Bayesian optimization[18] and have used object detection networks to actively monitor various features or patterns in diagnostics (e.g., optical damage or few-cycle images of laser-driven plasma waves)[19]. As an example, Figure 10 shows objects detected by an object detection network[20], which was fine-tuned to detect and distinguish features that regularly occur in few-cycle shadowgraphy, such as plasma waves or diffraction patterns from dust on optics or hydrodynamic shocks. Importantly, this fine-tuning only requires a small number (~ 50) of manually labelled training images. The detected objects can be used to estimate physical quantities, such as the plasma wavelength and plasma density. Although the object detector so far has only been applied to offline data, the fast inference speed of the algorithm will make an integration into Tango possible. It would allow live analysis during an experiment of thousands of laser shots where the objects can have different positions on the diagnostics.

As another example, Bayesian optimization has recently been applied at the ETTF facility to optimize and automate the generation of electron beams. The Tango system facilitates such an optimization by providing online information of the laser and experiment systems, which is then used by

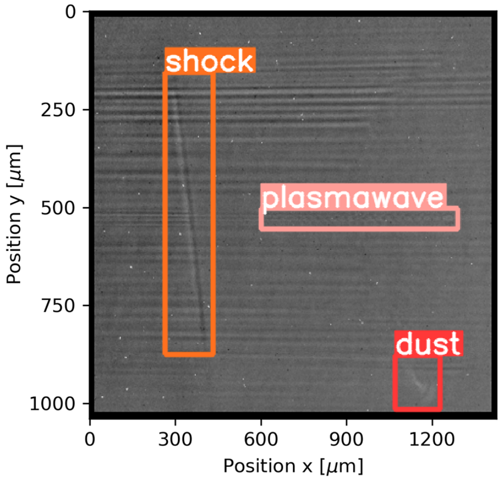


Figure 10. Machine learning application in few-cycle probing in a hybrid[21] laser–plasma accelerator. The objects to be detected in the shadowgram are the plasma wave, the shock and the diffraction patterns from dust particles.

Bayesian optimization to generate complex models of the experiment. Based on these models, the optimizer is able to send commands to the Tango system to move different motors to new positions and gas regulators to new gas pressure set points, resulting in better electron beams.

6. Conclusions and outlook

In this paper, we have described our server infrastructure based on Tango Controls and its Python implementation, PyTango. We give an overview of Tango Controls and its capabilities in terms of experimental supervision and control. Our own development in improving the experimental supervision and control, as well as our data acquisition and data processing pipeline based on Tango Controls, was further presented. We view the integration of most instruments in a common server infrastructure as well as the described development as an important step towards improved operation and evaluation procedures at CALA.

At the time of writing, multiple goals have been defined to optimize the experimental procedures at CALA. Firstly, Tango Controls as a server framework can enable automation and supervision of alignment steps at the ATLAS-3000 laser at CALA. This includes optimization of motor positions, for example, changing the position of alignment mirrors, by feedback loops with additional diagnostics. Furthermore, safety aspects with respect to the implemented instruments can be improved, for example, by devices such as the ultrafast pulse-shaping system safety device mentioned in Section 4. Based on the processing pipeline described in Section 5 and the dashboard described in Section 4, online analysis tools that allow for quick visualizations of measurements and recorded data are being developed. This could ease daily experimental operations, for example, plotting the stage

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press

position versus the mean count rate on a camera to find the optimal position for a particular experiment. Finally, the developments described in this paper are prerequisites for the ongoing implementation of advanced methods based on machine learning to understand, control and analyse petawatt laser experiments.

Acknowledgements

We thank the Federal Republic of Germany and the Free State of Bavaria for funding the CALA infrastructure (15171 E 0002) and its operation. This work was supported by the Independent Junior Research Group "Characterization and control of high-intensity laser pulses for particle acceleration", DFG Project No. 453619281. N.W. was supported via the IMPULSE project by the European Union Framework Program for Research and Innovation Horizon 2020 under grant agreement No. 871161. This work was supported by the Bundesministerium für Bildung und Forschung (BMBF) within project 01IS17048. J.G. acknowledges support from the German Academic scholarship foundation.

References

1. C. N. Danson, C. Haefner, J. Bromage, T. Butcher, J.-C. F. Chanteloup, E. A. Chowdhury, A. Galvanauskas, L. A. Gizzi, J. Hein, D. I. Hillier, N. W. Hopps, Y. Kato, E. A. Khazanov, R. Kodama, G. Korn, R. Li, Y. Li, J. Limpert, J. Ma, C. H. Nam, D. Neely, D. Papadopoulos, R. R. Penman, L. Qian, J. J. Rocca, A. A. Shaykin, C. W. Siders, C. Spindloe, S. Szatmári, R. M. G. M. Trines, J. Zhu, P. Zhu, and J. D. Zuegel, High Power Laser Sci. Eng. **7**, e54 (2019).
2. A. Pukhov, Rep. Prog. Phys. **66**, 47 (2003).
3. M. Marklund and P. K. Shukla, Rev. Mod. Phys. **78**, 591 (2006).
4. E. Esarey, C. B. Schroeder, and W. P. Leemans, Rev. Mod. Phys. **81**, 1229 (2009).
5. V. Malka, Phys. Plasmas **19**, 055501 (2012).
6. A. Macchi, M. Borghesi, and M. Passoni, Rev. Mod. Phys. **85**, 751 (2013).
7. J. Schreiber, P. R. Bolton, and K. Parodi, Rev. Sci. Instrum. **87**, 071101 (2016).
8. F. Albert and A. G. R. Thomas, Plasma Phys. Control. Fusion **58**, 103001 (2016).
9. J. M. Cole, D. R. Symes, N. C. Lopes, J. C. Wood, K. Poder, S Alatabi, S. W. Botchway, P. S. Foster, S. Gratton, S. Johnson, C. Kamperidis, O. Kononenko, M. De Lazzari, C. A. J. Palmer, D. Rusby, J. Sanderson, M. Sandholzer, G. Sarri, Z. Szoke-Kovacs, L. Teboul, J. M. Thompson, J. R. Warwick, H. Westerberg, M. A. Hill, D. P. Norris, S. P. D. Mangles, and Z. Najmudin, Proc. Natl. Acad. Sci. U.S.A. **115**, 6335 (2018).
10. A. Döpp, L. Hehn, J. Götzfried, J. Wenz, M. Gilljohann, H. Ding, S. Schindler, F. Pfeiffer, and S. Karsch, Optica **5**, 199 (2018).
11. A. B. Zylstra, O. A. Hurricane, D. A. Callahan, A. L. Kritcher, J. E. Ralph, H. F. Robey, J. S. Ross, C. V. Young, K. L. Baker, D. T. Casey, T. Döppner, L. Divol, M. Hohenberger, S. Le Pape, A. Pak, P. K. Patel, R. Tommasini, S. J. Ali, P. A. Amendt, L. J. Atherton, B. Bachmann, D. Bailey, L. R. Benedetti, L. B. Hopkins, R. Betti, S. D. Bhandarkar, J. Biener, R. M. Bionta, N. W. Birge, E. J. Bond, D. K. Bradley, T. Braun, T. M. Briggs, M. W. Bruhn, P. M. Celliers, B. Chang, T. Chapman, H. Chen, C. Choate, A. R. Christopherson, D. S. Clark, J. W. Crippen, E. L. Dewald, T. R. Dittrich, M. J. Edwards, W. A. Farmer, J. E. Field, D. Fittinghoff, J. Frenje, J. Gaffney, M. Gatu Johnson, S. H. Glenzer, G. P. Grim, S. Haan, K. D. Hahn, G. N. Hall, B. A. Hammel, J. Harte, E. Hartouni, J. E. Heebner, V. J. Hernandez, H. Herrmann, M. C. Herrmann, D. E. Hinkel, D. D. Ho, J. P. Holder, W. W. Hsing, H. Huang, K. D. Humbird, N. Izumi, L. C. Jarrott, J. Jeet, O. Jones, G. D. Kerbel, S. M. Kerr, S. F. Khan, J. Kilkenny, Y. Kim, H. G. Kleinrath, V. G. Kleinrath, C. Kong, J. M. Koning, J. J. Kroll, M. K. G. Kruse, B. Kustowski, O. L. Landen, S. Langer, D. Larson, N. C. Lemos, J. D. Lindl, T. Ma, M. J. MacDonald, B. J. MacGowan, A. J. Mackinnon, S. A. MacLaren, A. G. MacPhee, M. M. Marinak, D. A. Mariscal, E. V. Marley, L. Masse, K. Meaney, N. B. Meezan, P. A. Michel, M. Millot, J. L. Milovich, J. D. Moody, A. S. Moore, J. W. Morton, T. Murphy, K. Newman, J.-M. G. Di Nicola, A. Nikroo, R. Nora, M. V. Patel, L. J. Pelz, J. L. Peterson, Y. Ping, B. B. Pollock, M. Ratledge, N. G. Rice, H. Rinderknecht, M. Rosen, M. S. Rubery, J. D. Salmonson, J. Sater, S. Schiaffino, D. J. Schlossberg, M. B. Schneider, C. R. Schroeder, H. A. Scott, S. M. Sepke, K. Sequoia, M. W. Sherlock, S. Shin, V. A. Smalyuk, B. K. Spears, P. T. Springer, M. Stadermann, S. Stoupin, D. J. Strozzi, L. J. Suter, C. A. Thomas, R. P. J. Town, E. R. Tubman, C. Trosseille, P. L. Volegov, C. R. Weber, K. Widmann, C. Wild, C. H. Wilde, B. M. Van Wonterghem, D. T. Woods, B. N. Woodworth, M. Yamaguchi, S. T. Yang, and G. B. Zimmerman, Nature **601**, 542 (2022).
12. A. Doepp, C. Eberle, S. Howard, F. Irshad, J. Lin, and M. Streeter, arXiv:2212.00026 (2022).
13. F. S. Englbrecht, A. Döpp, J. Hartmann, F. H. Lindner, M. L. Groß, H.-F. Wirth, P. G. Thirolf, S. Karsch, J. Schreiber, K. Parodi, and G. Dedes, J. Radiol. Protection **40**, 1048 (2020).
14. J. Hartmann, T. F. Rösch, F. Balling, M. Berndl, L. Doyle, L. Flaig, S. Gerlach, L. Tischendorf, and J. Schreiber, Proc. SPIE **11779**, 117790N (2021).
15. https://www.tango-controls.org/.
16. https://gitlab.com/tango-controls/web/taranta.
17. https://dash.plotly.com/.
18. F. Irshad, S. Karsch, and A. Döpp, Phys. Rev. Res. **5**, 013063 (2023).
19. J. Lin, F. Haberstroh, S. Karsch, and A. Döpp, High Power Laser Sci. Eng. **11**, e7 (2023).
20. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), p. 779.
21. F. M. Foerster, A. Döpp, F. Haberstroh, K. v. Grafenstein, D. Campbell, Y.-Y. Chang, S. Corde, J. P. Couperus Cabadağ, A. Debus, M. F. Gilljohann, A. F. Habib, T. Heinemann, B. Hidding, A. Irman, F. Irshad, A. Knetsch, O. Kononenko, A. Martinez de la Ossa, A. Nutter, R. Pausch, G. Schilling, A. Schletter, S. Schöbel, U. Schramm, E. Travac, P. Ufer, and S. Karsch, Phys. Rev. X **12**, 041016 (2022).

https://doi.org/10.1017/hpl.2023.17 Published online by Cambridge University Press