20th Int. Conf. Accel. Large Exp. Phys. Control Syst. (ICALEPCS'25), Chicago, IL, USA JACoW Publishing
ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEPD076

GEOFF DEVELOPMENTS IN 2025

P. Madysa*, S. Appel, L. Dingeldein, J. Fitzek, GSI, Darmstadt, Germany
V. Kain, M. Schenk, CERN, Geneva, Switzerland

Abstract

The complexity of the CERN and GSI/FAIR accelerator facilities requires a high degree of automation to maximize beam time and performance for physics experiments. Geoff, the Generic Optimization Framework & Frontend, is an open-source tool developed within the EURO-LABS project by CERN and GSI to streamline access to classical and AI-based optimization methods. It provides standardized interfaces for optimization problems and utility functions to speed up implementation. Plugins are independent packages with their own dependencies, allowing scaling from simple prototypes to complex state machines that communicate with devices in different timing domains. This contribution presents Geoff's design, features, and current applications.

At GSI, multi-objective Bayesian optimization was applied to SIS18 multi-turn injection, building a Pareto front from experimental data. At CERN, Geoff and ML/AI contributed to a record ion beam intensity for the LHC in 2024 through LEIR and SPS optimization. In addition, Geoff underwent major updates in 2024 and 2025, aligning it with the latest developments in Python-based numerical and machine-learning software.

GEOFF AT GSI/FAIR

FAIR—the Facility for Antiproton and Ion Research—is going to be an international center of heavy-ion accelerators that will drive the forefront of heavy-ion and antimatter research [1]. The complexity of FAIR requires a high level of automation for future operation [2]. One part of this automation effort is to provide a framework that allows both machine experts and operators to solve concrete optimization problems, and to make these solutions reusable in an operational context. We call this project the "Generic Optimization Framework and Frontend", or *Geoff* for short [3].

Geoff is based on the Python programming language, which is widely used in scientific research, has a vibrant ecosystem of machine learning (ML) algorithms, and is perceived as very beginner-friendly. Both language and framework have proven themselves flexible enough to be quickly adapted to new problems.

Geoff is used extensively at CERN—from linacs to SPS and ISOLDE—and in the process of being deployed at GSI/FAIR [4], It is usually embedded into a GUI application, but can also be used in command-line scripting. *Geoff* standardizes interfaces for optimization tasks [5] and provides adapters for various third-party packages, for example: *SciPy, Stable Baselines 3, Scikit-Optimize*. Optimization problems, implemented as plugin packages, can scale to arbitrary complexity and depend on any Python package. They can use any controls systems and communicate with external simulation tools, as long as they have Python bindings (see Fig. 1).

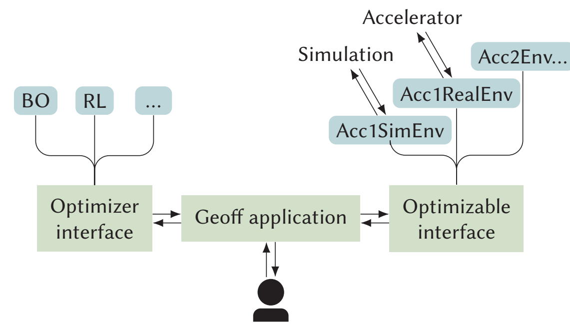

Figure 1: Model of *Geoff* and its components. The optimizer interface includes BOBYQA, Bayesian optimization (BO), and reinforcement learning (RL).

Figure 2 shows an optimization run that used *Geoff* for beam steering in the TK transfer channel at GSI. To shorten the edit-check-run cycle between code changes and test runs, we used *Geoff* in a command-line script and ran it inside a terminal window. Custom figures can be shown and updated continuously to monitor the algorithm's progress.

Figure 2: Screenshot of a *Geoff* optimization run from the terminal. The terminal window with logging messages is in the background, live graphs for monitoring are in the foreground.

GUI APPLICATION

Part of the Geoff project is a reference GUI application originally developed for the CERN accelerator complex. The application lists available optimization problems (which are imported dynamically as plugins); allows the user to configure them; and runs them such that their monitoring figures are embedded into the application window.

* p.madysa@gsi.de

20th Int. Conf. Accel. Large Exp. Phys. Control Syst. (ICALEPCS'25), Chicago, IL, USA JACoW Publishing
ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEPD076

Work at GSI/FAIR has been ongoing to port this GUI application to a non-CERN system. A first prototype can be seen in Fig. 3. CERN-specific elements have been removed and the *context select* (black box on the left) has been adjusted to GSI-specific properties of LSA [6].

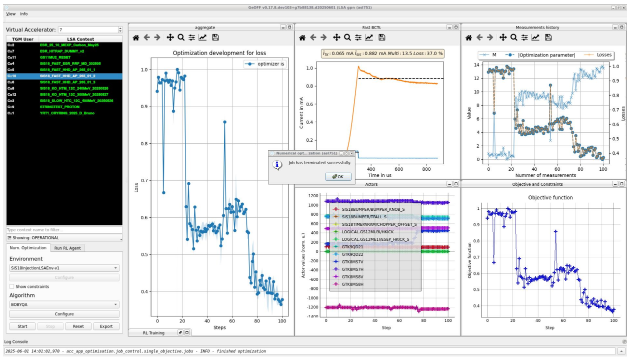

Figure 3: Screenshot of a *Geoff* optimization run in a GUI application at GSI/FAIR. The live graphs are embedded into the application window and logging messages are shown in a status line at the bottom, which can be expanded into a console view of the full log.

The next steps are to extend the *Reset parameters* feature to allow any previously evaluated point; to further modularize the package to make more parts reusable in other laboratories; and to deploy the application in a stable and maintainable manner within the larger accelerator controls system of GSI/FAIR.

REMOTE GEOFF

One of the obstacles we encountered at GSI/FAIR in 2025 was a mismatch in computational requirements between the optimization problem and the optimization algorithm.

While the problem was computationally cheap to evaluate, it had to be run within the protected accelerator network to access all devices of interest. Terminals in this network are usually shared machines with limited computational resources.

The optimization algorithm under investigation at the time[1], on the other hand, was computationally expensive, but had no requirements in terms of network access.

The project *Remote Geoff* addresses this mismatch. Its architecture is shown in Fig. 4. It consists of a client and a server that communicate via TCP.

Inside the accelerator network, a TCP server listens for incoming connections. Each connection is treated as a distinct user session. During each session, a request–reply scheme is employed such that each interaction is initiated by the client sending a message and finished as soon as it receives a reply. Interactions do not overlap in time.

On request by the client and after authentication via a shared secret, the server instantiates one of the optimization

[1] A multi-fidelity Bayesian optimization algorithm, see the results section.

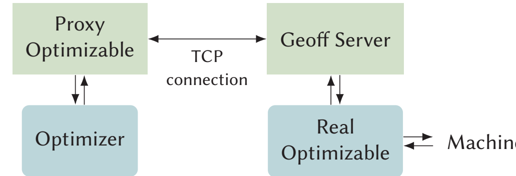

Figure 4: Model of *Remote Geoff* and its components. The left-hand side is fully local to the client and could include a host application between optimizer and optimizable, like in Fig. 1.

problem classes that are available to it. Towards the optimization problem, it acts like a host application. Any further requests from the client are deserialized and turned into a method call on the problem object. Its return value is then serialized again and send back to the client as a reply.

On the client side, a class `RemoteSingleOptimizable` takes in a server to connect to, an authentication token, and the name of an optimization problem to request. The instantiated object is a *proxy* that implements Geoff's regular `SingleOptimizable` interface, but forwards all method calls to the server.

The serialization protocol is based on the standard-library function `ast.literal_eval` and only permits a small number of built-in types to be transferred. In all other cases, both sides of Remote Geoff must expect the more complex type and convert it to and from a representation with the allowed types as necessary. For example, where NumPy arrays are expected, they are converted to and from lists of numbers on both sides of the connection.

An example of how complex data can be transferred by Remote Geoff is the task of rendering live graphs, visualized in Fig. 5. In a local setup, this is done by calling the method `render()` and using the Pyplot API of the Matplotlib library. This API operates on global state and eventually presents the plotted results to the user.

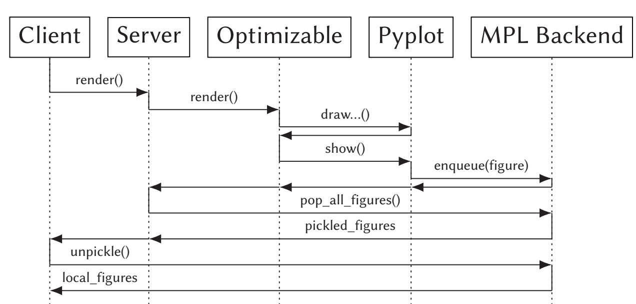

Figure 5: Sequence diagram of the client requesting the server to render a live graph. The actors *Server*, *Optimizable*, and *Pyplot* all run in the same process. The actor *MPL Backend* represents a Matplotlib backend on both the server and the client side. They have been treated as one for space reasons.

In a remote setup, plotting occurs on the server side, but the results are sometimes supposed to be shown on the client

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEPD076

side. To achieve this, the server installs a custom Matplotlib backend to take over management of figure objects. For each figure, the `show()` method must eventually be invoked to make it visible. These calls are forwarded to the backend, and Remote Geoff uses this opportunity to remember which figures are to be sent to the client.

When the `render()` call eventually returns, the server queries the custom backend directly for a list of all registered figures. These figures are serialized using a custom instance of `pickle.Pickler` from the standard library. The pickler serializes Matplotlib figures in such a way that during unpickling, special callback functions of Remote Geoff are called instead of the regular reconstruction functions. The figures thus pickled are sent as a `bytes` primitive to the client.

On the client side the remote request `render()` is eventually replied with the bytes of a pickled object. Unpickling them calls into Remote Geoff functions that track on the client side which figure has been shown to the user before and which has been sent for the first time. In the case of a new figure, it is unpickled as usual. If the figure however has already been shown to the user on the client side, the existing figure object is instead updated with the new state. Unpickling the bytes returns the existing figure object in this case.

RESULTS HIGHLIGHTS

In 2024 and 2025, Geoff has been used extensively in accelerator operations at CERN [7] and in several machine development studies at GSI/FAIR. For a broader overview of its impact, see [3].

At CERN's LINAC3 linear accelerator, Geoff is used to optimize and continuously tune various parameters of the ion source to provide a high and stable output current for ion beam operations [8]. The problem is a mix of slow and fast dynamics and the nested-loop structure used to solve it would be difficult to implement without Geoff's flexibility.

At the SPS, Geoff was used in the compensation of main-power converter noise from the electrical grid and allowed the continuous upgrade of algorithms over multiple years; the latest one (based on Bayesian optimization) improves ion beam transmission to the LHC by 15–20 % [9], as Fig. 6 shows. It further facilitated the operational deployment of simulation-trained RL agents for beam trajectory correction in the transfer lines from SPS to the North Experimental Area [10].

At GSI, Geoff has been used to study the automation of various beam automation tasks [11]. In particular, the Fragment Separator (FRS) serves as a useful study case: both because up to a third of the time available to experiments is currently being used for manual adjustment of its settings, and because the plans for FAIR include a new *Super-FRS* with four times more magnets and a corresponding increase in set-up complexity.

The beam focusing in the main dispersive focal plane of the FRS is a concrete set-up task where Geoff has been

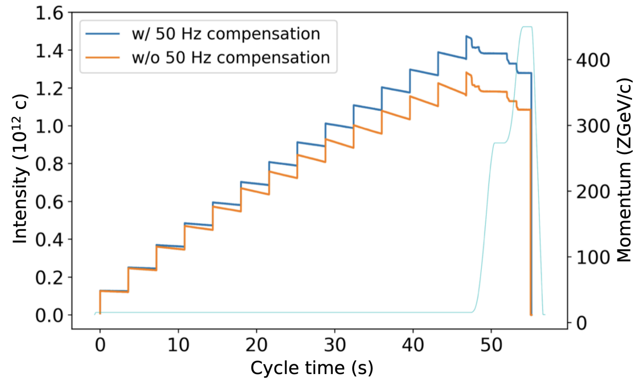

Figure 6: Intensity of the beam injected from the SPS into the LHC with and without Bayesian optimization of main-power converter noise [9].

used. A time projection chamber (TPC) reconstructs the trajectories of all individual particles. These trajectories are classified into several beam spots (depending on the particles' charge state) via the DBSCAN algorithm implemented in Scikit-Learn. A fitting algorithm then estimates the Twiss parameters for each beam spot. The actual optimization task is then to adjust the quadrupole settings of the FRS in such a way that the Twiss parameters correspond to an upright elliptical beam spot.

Using Geoff for this task was advantageous because of the flexibility that it affords to plugin authors. Because each plugin is a full Python package, it may depend on third-party packages like Scikit-Learn. The independence of any particular controls system further helped to cross the separation between the accelerator network at GSI/FAIR (which contains the quadrupoles to be manipulated) and the experimental network (which contains the TPC used for evaluation). The ability to organize code into multiple modules meanwhile kept the code maintainable.

In addition to automation tasks like the above, Geoff has also been used to study a number of advanced optimization techniques [12]. These include Multi-Objective Bayesian optimization of the SIS18 multi-turn injection; multi-fidelity Bayesian optimization, which combines information from real data and a fast surrogate model; and data-driven model predictive control (MPC), which combines model-based RL with various techniques of control theory.

In all these cases, Geoff provided a uniform API between the optimization problem and the algorithm in question. Although all these algorithms were written by external collaborators, it was trivial in all cases to write adapters between their algorithms and the GSI-internal code that drove the accelerator settings to the machine.

CONCLUSION

The Geoff project continues to benefit the participating laboratories in 2025. Progress on making the GUI application available to a wider audience is ongoing and has reached the milestone of a working prototype.

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEPD076

New packages continue to unlock new use cases. As an example, *Remote Geoff* allows using computationally expensive optimization algorithms by separating the machine that runs the algorithm from the machine that runs the optimization problem.

CERN continues to use Geoff in an operational context and has seen ion beam intensity improvements of 15–20 % by solving one particular optimization problem.

At GSI, efforts to automate operational tasks and to deploy Geoff in an operational context continue. The FRS provides here an excellent opportunity to study and improve automation solutions that will become critical in the operations of Super-FRS. Collaborating research institutions have the opportunity to test their algorithms in a challenging context and Geoff provides to them a uniform interface to adapt their algorithms to.

ACKNOWLEDGMENTS

The EURO-LABS project has received funding from the European Union's *Horizon Europe* Research and Innovation programme under Grant Agreement no. 101057511.

REFERENCES

[1] H. Gutbrod *et al.*, "FAIR baseline technical report, vol. 1: Executive Summary", GSI, Darmstadt, Germany, Rep. GSI-2013-04785, 2006. `https://repository.gsi.de/record/54062`.

[2] S. Reimann, M. Sapinski, P. Schütt, and M. Vossberg, "Building an Operation Team for FAIR nearly from Scratch", in *Proc. WAO'16*, Shanghai, China, Sep. 2016. `https://www.researchgate.net/publication/316324398`

[3] P. Madysa, S. Appel, V. Kain, and M. Schenk, "Geoff: The Generic Optimization Framework & Frontend for Particle Accelerator Controls", *SoftwareX*, vol. 32, p. 102335, Sep. 2025. `doi:10.1016/j.softx.2025.102335`

[4] R. Roussel *et al.*, "Bayesian optimization algorithms for accelerator physics", *Phys. Rev. Accel. Beams*, vol. 27, no. 8, p. 084801, Aug. 2024. `doi:10.1103/PhysRevAccelBeams.27.084801`

[5] COI — Common Optimization Interfaces, `https://cernml-coi.docs.cern.ch/`.

[6] J.-P. A. Hucka, "Control System for the Next Generation Inflight Separator Super-FRS applied to New Isotope Search with the FRS", Ph.D. dissertation, Technische Universität Darmstadt, Darmstadt, Germany, Jan. 2023. `doi:10.26083/tuprints-00023299`

[7] G. Trad *et al.*, "Results and plans for integration of automation and optimization in operation", presented at the CERN Joint Accelerator Performance Workshop (JAPW'24), Montreux, Switzerland, Dec. 2024, unpublished. `https://indico.cern.ch/event/1439972/contributions/6159155/`.

[8] V. Kain *et al.*, "Continuous data-driven control of the GTS-LHC ion source at CERN", in *Proc. ECRIS'24*, Darmstadt, Germany, Sep. 2024, pp. 56–59. `doi:10.18429/JACoW-ECRIS2024-MOP11`

[9] V. Kain *et al.*, "Eliminating mains noise with Machine Learning", presented at MaLAPA'25, Geneva, Switzerland, Apr. 2025, unpublished. `https://indico.cern.ch/event/1382428/contributions/6283299/`.

[10] A. Menor de Onate, N. Bruchon, V. Kain, and M. Schenk, "Autonomous trajectory steering of DC beams at CERN's transfer lines using reinforcement learning", presented at RL4AA'25, Hamburg, Germany, Apr. 2025, unpublished. `https://indico.kit.edu/event/4216/contributions/19231/`.

[11] P. Madysa, "Automation & optimization with classical optimizer towards RL at GSI with Geoff", presented at the 13th MT ARD ST3 Meeting 2025, Zeuthen, Germany, Jun. 2025, unpublished. `https://indico.desy.de/event/40831/contributions/187911/`.

[12] R. Assmann *et al.*, "The FAIR and GSI Accelerator Facility: Automation of key beam manipulations with AI methods", submitted for publication.