ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF044

AUTOMATIC TUNING OF PETRA, ITS INJECTOR COMPLEX, AND PROSPECTS OF AUTONOMOUS OPERATION OF PETRA IV

I. Agapov *, H. Ehrlichmann, J. Keil, G. K. Sahoo, R. Wanzenberg
Deutsches Elektronen Synchrotron, Notkestrasse 85, 22607, Hamburg, Germany

Abstract

We present the progress in tuning automation of the PETRA injection complex. The OCELOT optimizer has been ported to the PETRA control system and proof-of-principle tests of transmission efficiency optimization done. We further argue that the next steps in tuning and automation are impossible without rethinking the architecture of the high level control system. A possible approach to the new system is then sketched.

INTRODUCTION

Automatic empirical tuning of accelerators has attracted significant attention in the recent years. It has been successfully used in both linac-driven FELs [1–3] and storage ring facilities [4]. The reasons this approach became popular include the increased attention to facility performance, proliferation of high-level programming languages and software libraries making developing of such tools easy, and aversion to menial work. The challenges of empirical tuning methods are their speed and robustness. In virtually all applications the advantage brought by such tuning tools has been limited to saving the operator time. Manual tuning could almost always lead to a better result if the operator is skilled enough and is willing to dedicate sufficient time. In what follows we discuss next steps in the tuning and automation project, focusing on following issues:

First, we report on the progress with automatic tuning at PETRA injectors. The OCELOT generic optimizer developed for the European XFEL has been ported to the PETRA environment, and tuning various subsystems of PETRA and its injector complex could now be done with it. Some details are given in what follows.

Then, in contrast to XFELs, the optimization of parameters such as the injection efficiency or dynamic aperture have an indirect influence on the source performance, at least in the case of PETRA. Empirical tuning methods do work, but in most cases have little direct impact on the key facility performance characteristics such as the brightness, the photon beam stability, or the mean time between failures [1]. Moreover, DESY II – the PETRA III booster – can deliver more charge than is required for the top-up operation of PETRA. A reduced transfer efficiency from the gun through the whole injector chain is undesirable, but has only delayed consequences (e.g. through accumulation of deposited radiation dose).

* ilya.agapov@desy.de

[1] One exception is the optimization of the residual orbit distortion during the top-up injection [5].

The biggest challenge in operation of the PETRA III storage ring is not the tuning but its stability and availability. Reliable hardware systems are absolutely essential to minimize failures, however both software malfunctioning and human errors contribute significantly to the failure statistics. These factors can be addressed by a next generation high level control system that would make human error less likely. In the last section we try to rethink the approach to automation from that perspective. A conceptual sketch of the proposed next generation high level control system is presented.

EMPIRICAL OPTIMIZATION OF INJECTORS

The OCELOT optimizer [3] has been adapted to the *tine* [6] control system. An arbitrary number of control channels can be used for actuators, and a numerical expression involving an arbitrary number of control channels (up to 5 if the non-expert GUI version) for the objective function. Readout delays and the selection of the algorithm can be done through the GUI. A screenshot of the GUI during the matching of the tune at PETRA with the main quadrupole circuits is shown in Figure 1. The main objective of adapting the optimizer was to use it for transfer efficiency optimization in the injector chain. The injector chain consists of an electron source (GUN), a linac (LINAC2), an accumulator ring (PIA), transfer line to the booster synchrotron (L-WEG), the booster synchrotron (DESY II), and the transfer line to the PETRA ring (E-WEG). The actuators used for transfer optimization mostly consist of orbit correctors and the RF modulator phases (see Table 1). In all test cases the optimization worked robustly and a transfer efficieny of about 80 % could be achieved, similar to what an operator typically gets albeit in a shorter time. The optimizer allows creating configuration files with pre-defined actuator, objective function and optimization algorithm settings. Such configuration with most effective actuators was created for the transfer optimization, which could now be used by operators. The system setup is available only on the Apple Mac test workstation, which is also used as a testbed for evaluating the possibility of moving the control system away from Windows. A considerable complication is related to non-uniform naming convention and non-standardized data structures within the PETRA control system; some logic related to dealing with different device classes had to be incorporated into the code. That would have not been necessary if the naming convention and data type standards were adhered to. We hope that in the future this situation can be improved.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF044

Figure 1: OCELOT optimizer screenshot, example of tune matching at PETRA.

Table 1: Summary of Control Parameters for Transfer Optimization

Subsystem	Parameter
Gun, LINAC2	RF modulator phases, attenuation, orbit
PIA	orbit
L-WEG	orbit
DESY II	orbit
EWEG, transfer	kicker timing, orbit, kicker voltages

TOWARDS AUTONOMOUS OPERATION

The development of the OCELOT optimization tools started after observing that operators at FELs such as FLASH or LCLS perform an extremely large amount of relatively straightforward tuning tasks (many hundreds of hours per year). Considerable amount of tuning time could be cut by introducing appropriate software. At synchrotrons the situation turned out to be rather different. These machines run with availability of 98% and higher, and as soon as the machine is set up (which typically happens once a week during the maintenance day at PETRA) very little intervention is usually required. Already now the load on operators is rather small, and with increased automation it will be even further reduced. Soon the operator will hardly need to actually control the machine. The reduced load on operators has a surprising negative impact on the availability since the operators are faced with the need to perfom certain procedures so seldom that they lack the routine to perform them properly. One could argue that the next natural step is to achieve autonomous operation of the machine and release the operator from her duties entirely [2]. However, if autonomous operation is what we wish, then the present high level control system architecture is fundamentally inappropriate: it is designed with the view of a human operator being in charge and taking care of the interaction between software tools. We need to design a new system where a software module assumes central control role from the start. All other applications should be able to be orchestrated by this centralized control entity. In the following we sketch an approach to the *high level* control system which we believe will make it possible to achieve this goal. Note that the high level control is mostly independent of the low level control approach and could be built on top of any system such as *epics* [7], *doocs* [8], *tango* [9], *tine* [6], etc.

[2] Note that this is probably within reach for storage ring light sources only; for more complex machines such as the SRF linacs or the LHC this possibility migh be much further removed.

Definition of Autonomous Operation

The operation of a synchrotron facility is interrupted by technical faults or maintenance. By autonomous operation we mean a completely automatic restart of the machine and switching into the user operation mode after any shutdown such as hardware installation or exchange. The steps include: powering the machine on, magnet cycling, filling with beam, orbit correction, dispersion correction, on some occasions – optics correction. A similar functionality is already present in the control system in the form of a so-called sequencer. The important difference is, however, that, as its name suggests, the sequencer is able to perform only linear sequences of actions, while in relity the space of possible decisions always branches out. A more appropriate structure for the central control unit would be some sort of *decision graph*.

Operation

Operation of a synchrotron is a non-trivial enterprise and considerable experience has been accumulated at DESY and elsewhere in this area. It is natural to map this process on software components without much modification. Synchrotron operation is performed by an operator who typically has a sort of "mandate" to do some things but is not allowed to do other things. E.g. at PETRA an operator would be allowed to correct the orbit but not the optics, a machine physicist is required for the latter. The whole process is orchestrated by a run coordinator who decides on the timing and gives a "mandate" to certain people to start certain procedures. This could be easily mapped onto software, with people replaced by *controllers*, each performing certain tasks, and the central controller arbitrating the whole procedure.

High-level Controller Network

A controller can act as a central control unit for the whole facility or for a subsystem. Following the divide and conquer approach, we could divide functionality in various domains and assign controllers to each of them. The structure need not be hierarchical, i.e. a single controller can accept requests from different other controllers. Some domains within the high level control system are presented in Table 2.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF044

Table 2: Examples of Possible High Level Controllers

Controller	Responsibility
Central	Analogy for the run coordinator. Major task is machine restart
Slow orbit	orbit feedback and correction functionality
Fast orbit	fast orbit feedback
Optics	optics correction, dispersion correction
Beam	Runs top-up
MPS	machine protection provides sensor and actuator limits

Application GUI Ideology

It is expected that all the functionality should be able to run in the background, with the possibility of connecting to separate processes for visualization and control. No substantial control logic should be incorporated into any of the UI tools. Establishing this technology could be a major challenge to the controls group.

No Optics Server, Optics Calculations are Available to all High-level Applications

The speed of modern computers and the complexity of light source optics is such that most of the matching required in operation can be done instantaneously. Moreover, the optics or orbit matching is normally single-threaded, and the HPC processors are typically not faster than workstation processors wrt. single-core performance. We thus believe that having an optics server for light sources only complicates the system and should be abandoned. A standard configuration directory containing current optics in conjunction with linear optics calculation functionality in the control system software should be enough. MML [10] or OCELOT [11] follow this approach, which makes it possible to create high level control application either in the MATLAB or PYTHON environment.

Case Study Orbit Correction

Consider the case of orbit correction, and consider the possibility of having multiple orbit correction client instances: at the central accelerator control center and at the experimental hutches. Suppose one client requests orbit correction, e.g. readjustment of the orbit in an undulator. The flow of events is like this: the GUI orbit correction tools connects to the controller and requests orbit correction. The controller uses the optics module to calculate the projected correction and reports to the GUI if it is possible. The controller then requests a *mandate* to perform this operation from the central controller. The central controller checks that the particular UI instance is allowed to perform the action (i.e. the scope of possible orbit correction should be different for accelerator controls or experimental control applications). It contacts the machine protection controller and also potentially makes use of the optics modules to perform such a check. If the action is authorized, the orbit contoller receives the mandate and performs necessary correction. The whole workflow is sketched in Figure 2. The central controller need not be a bottleneck: a long-term "wildcard" mandate can be given to certain types of controllers. Also note that some sort of authorization mechanism has to be added.

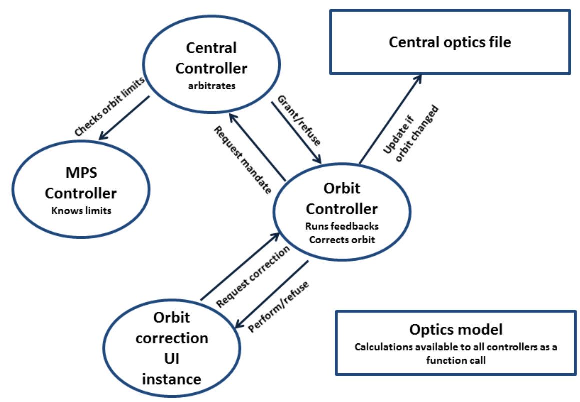

Figure 2: Sketch of interaction between controllers, case study of orbit correction.

CONCLUSION AND OUTLOOK

The OCELOT tools have been adapted for the operation of PETRA and its injectors and several optimization configurations were set up. During this process it was however noted that not the tuning but the refactoring of the high level control system is the more pressing issue to improve the level of automation. Given that synchrotrons are rather stable and reliable machines, their autonomous operation looks within reach. A path for refactoring the PETRA high control tools based on a controller network to achive this goal has been sketched.

REFERENCES

[1] I. Agapov, G. Geloni, I. Zagorodnov, Statistical optimization of FEL performance, in Proceedings of IPAC15, Richmond, USA (2015)

[2] I. Agapov, G. Geloni, S. Tomin, I. Zagorodnov, Automatic Tuning of Free Electron Lasers, DESY17-054, arXiv:1704.02335 [physics.acc-ph] (2017)

[3] I. Agapov, W. Decking, G. Geloni, M. Scholz, S. Tomin, I. Zagorodnov, On-line Optimization of European XFEL with OCELOT, in Proceedings of ICALEPCS 17, Barcelona, Spain (2018)

[4] X. Huang, Beam-Based Optimization of Storage Ring Nonlinear Beam Dynamics, in Proceedings of IPAC17, Copenhagen, Denmark (2017)

[5] J. Keil et al., Optimization of the Injection Kicker Bump Leakage at PETRA III, in Proceedings of IPAC18, Vancouver, Canada (2018)

[6] http://winweb.desy.de/mcs/tine/

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-WEPAF044

[7] https://epics.anl.gov/

[8] http://tesla.desy.de/doocs/doocs.html

[9] http://www.tango-controls.org/

[10] G. Portmann, J. Corbett and A. Terebilo, An accelerator control midle layer using MATLAB, in Proceedings of PAC 2005, Knoxville, Tennessee, USA (2005)

[11] I. Agapov, G. Geloni, S. Tomin, I. Zagorodnov, OCELOT: A software framework for synchrotron light source and FEL studies, Nuclear Instruments and Methods in Physics Research Section A, 768 (2014) pp 151-156