AUTOMATIZED OPTIMIZATION OF BEAM LINES USING EVOLUTIONARY ALGORITHMS

S. Appel*, S. Reimann†, V. Chetvertkova, W. Geithner, F. Herfurth, U. Krause, M. Sapinski, P. Schütt
GSI, Darmstadt, Germany
D. Österle, KIT, Karlsruhe, Germany

Abstract

Due to the massive parallel operation modes at GSI accelerators, a lot of accelerator setup and re-adjustment has to be made by operators during a beam time. This is typically done manually using potentiometers and is very time-consuming. With the FAIR project the complexity of the accelerator facility increases further and for efficiency reasons it is recommended to establish a high level of automation for future operation. Modern Accelerator Control Systems allow a fast access to both, accelerator settings and beam diagnostics data. This provides the opportunity to implement algorithms for automated adjustment of e.g. magnet settings to maximize transmission and optimize required beam parameters. The fast-switching magnets in GSI-beamlines are an optimal basis for an automatic exploration of the parameter-space. The optimization of the parameters for the SIS18 multi-turn-injection using a genetic algorithm has already been simulated [1]. The first results of our automatized online parameter optimization at the CRYRING@ESR injector are presented here.

INTRODUCTION

FAIR – the Facility for Antiproton and Ion Research – will constitute an international center of heavy ion accelerators that will drive forefront heavy ion and antimatter research. The goal of the FAIR facility is to provide antiproton and ion beams of unprecedented intensities as well as qualities. As a special feature, the facility will provide a broad range of high-intensity ion, antiproton and rare-isotope beams parallel to multiple experiments.

The High Energy Beam Transport System of FAIR, with a total length of more than 2350 meters, forms a complex system connecting seven accelerator- and storage rings, the experiment caves, beam dumps, stripping stations, the antiproton target and the Super Fragment Separator. The variety of beams to be transported is considerable, ranging from slow extracted beams with long spills of up to 100 s to short intense bunches with lengths of a few nanoseconds and momentum spreads of up to ±1%. The range of beam intensity covers more than six orders of magnitude [2]. The complexity of the FAIR facility demands a high level of automation for future operation, because otherwise the anticipated manpower requirements for operators would be excessive, as shown in [3]. Modern accelerator control systems allow a fast access to both, accelerator settings and beam diagnostics data. This provides the opportunity to implement

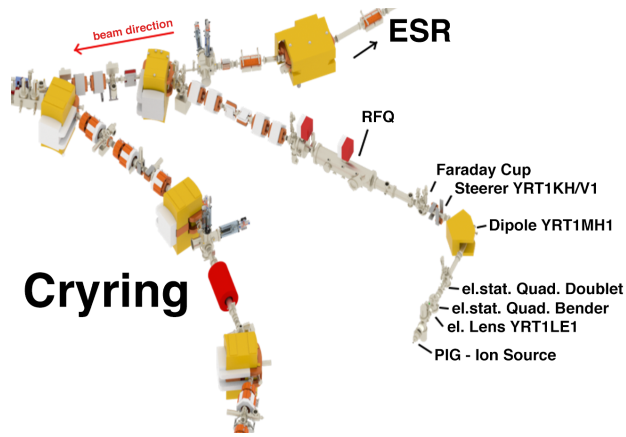

Figure 1: The settings of the steerers and electrostatic quadrupoles between ion source and Farady Cup after the dipol of CRYRING@ESR injector at GSI have been automatized optimized with evolutionary algorithm to maximize the beam transmission.

algorithms for an automated adjustment. An automatized machine based optimization using genetic algorithms for a storage ring has been already successfully demonstrated experimentally [4].

In the frame of the Swedish in-kind contribution to the FAIR project the storage ring CRYRING@ESR is planned to be used for experiments with low-energy ions and antiprotons. The ring is already installed in the existing GSI target hall and commissioning has started in 2015 [5, 6]. Since CRYRING@ESR has its own local injector it can be used stand-alone for testing novel technical developments like automatized configuration of beam line devices. Figure 1 shows the part of the CRYRING@ESR injector (from ion source to Faraday Cup), which has been used for testing automatized online genetic algorithm optimization. A semi-automatized optimization has been already preformed at the CRYRING in Sweden [7].

GENETIC ALGORITHMS

Genetic algorithms (GA) are inspired by natural evolution. GA search for solutions using techniques such as *selection*, *mutation* and *crossover*. By employing a wide range of different algorithms, GA are very flexible and can be adapted to a large range of different problems.

In GA terminology, a solution vector is called an *individual* and represents a set of variables; one variable is a *gene*. A group of individuals form a *population*, the following child populations are counted in *generations*. The first popula-

* s.appel@gsi.de
† s.reimann@gsi.de

ISBN 978-3-95450-182-3

tion is created randomly. The crossover operator exchanges variables between two individuals - the parents - to discover with their offspring promising areas in the solution space (*exploration*). For the optimization within a promising area, the mutation operator changes randomly the characteristics of individuals on the gene level (*exploitation*). Reproduction of individuals for the next generation involves selection. The *fitness* of an individual reflects how well an individual is adapted to the optimization problem and determines the probability of its survival for the next generation. The fitness is evaluated by an objective function, by a simulation code or by a real running system. During the single-objective optimization the most promising individuals are chosen to create the next generation. By allowing individuals with poor fitness to take part in the creation process the population is prevented to be dominated by a single individual. The most popular techniques are proportional selection, ranking and tournament selection [8–10].

A careful choice of the algorithm and operator is necessary to get the best performance of GA algorithms. The optimal choice of the offspring production probability through crossover or mutation is important for a proper balance between exploration and exploitation. The mutation operators are mostly used to explore, which is preferred at beginning of the search process. On the other hand, at the end of search process more exploitation through the crossover operators is needed to ensure convergence of the population. According to these facts, an incorrect production probability can lead to local optimum convergence.

To overcome the GA optimization slowness in the simulation (many different parameter settings need to be evaluated), parallel computing techniques are used. For an automatic optimization of real machine this advantage is not available. On the other hand an automatized configuration would instantly adapt to errors sensitive to the adjustment parameters as well as other technical influences. An automatic optimization of the transmission is possible if a pre-conditioned initial generation, a smaller number of individuals and generations can be used to shorten the optimization time. Very important for this kind of an automatic optimization is a fast reaction of the beam line devices as well as a fast and accurate beam diagnostic. For a short optimization within a few minutes a single cycle of setting beam line parameters and reading out detector values should not be longer than two seconds.

SIMULATION

The CRYRING@ESR injector model has been implemented in the particle tracking code pyORBIT - the python implementation of ORBIT (Objective Ring Beam Injection and Tracking) [11]. For the GA optimization the Distributed Evolutionary Algorithms in Python (DEAP) [12] together with pyORBIT has been used. DEAP includes evolution strategies, multi-objective optimization, and allows the development of new genetic algorithms. DEAP decouples the GA operators like crossover from the evolutionary algorithms, which allows for example to easily exchange the selection operator and leave the remaining algorithm unchanged.

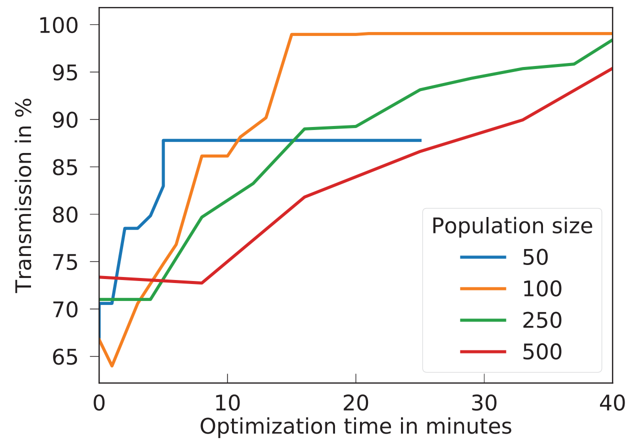

Figure 2: Simulated transmission evolution with the $(\mu + \lambda)$-algorithm for different population sizes for a beam line with four quadrupoles. It has been assumed a single beam line cycle is two seconds long.

Simulations indicated the $(\mu + \lambda)$-evolutionary algorithm from the DEAP python package would be the perfect candidate for a rapid automatic optimization. In the $(\mu + \lambda)$-algorithms as first step the individual fitness of μ-individuals are evaluated. μ is the population size and λ the offspring's size. Secondly, the evolutionary loop begins by producing $\lambda < \mu$-offspring's from the population through crossover and mutation. The offspring's are then evaluated and the next generations population is selected from both the offspring's and the current population. Finally, when a given number of generations has been evaluated, the algorithm returns the final population including the best solution [12]. In simulation the $(\mu + \lambda)$-algorithm could sufficient optimize the transmission of a beam line with four quadrupoles after 10 generations using a small population of 100 individuals and offspring size of 50, presented in Figure 2. Assuming a single beam line cycle is two seconds long, an automatic optimization would last 15 minutes.

As a result of the promising GA optimization simulation outcome of a beam line and multi-turn injection presented in [1] the Parameter Evolution Project (PEP) has been launched for automatized online parameter optimization in beam lines.

EXPERIMENT

Currently most of the GSI facility is undergoing heavy construction work or large up-grade measure and is therefore not available for beam time until 2018. An exception is the CRYRING@ESR and its local injector beam line. Since the CRYRING@ESR has just recently been installed at GSI, the infrastructure follows the guidelines of a modern accelerator control system and allows a fast access to both, accelerator settings and beam diagnostics data. GSI has selected the CERN Front-End Software Architecture (FESA) to operate accelerator devices and LSA-Database for the new control system. A lightweight python interface to FESA for the development of novel ideas, fast and easy, is available. The slow response of the CRYRING@ESR injector electrostatic quadrupoles of a few seconds is a disadvantage for testing evolutionary algorithm optimization and resulted in a cycle time longer than five seconds. An enhancement of electrostatic devices response is maybe possible in the future.

The aim of the optimization was to maximize the beam transmission through the beam line. During the genetic algorithm optimization the parameters on which the beam transmission depends were altered in consideration of the limiting technical and physical conditions. The algorithm allows independent variation of the steerer strengths and electrostatic quadrupoles voltages, in total nine different parameters. The 90°-Dipole shown in Figure 1 has been excluded from the genetic algorithm optimization due to is low transmission influence. The beam current has been measured at the current transformer behind the ion source and the Faraday cup after the dipole. Unfortunately, due to lack of time the current transformer has not been calibrated. Still without calibration the measurements from the current transformer could be used as reference in the transmission optimization process. Because of the slow response of the electrostatic devices, a holding period of five seconds before the readout of the beam diagnostic devices has been included. The result of the first successful evolutionary algorithm's optimization performed at GSI is presented in Figures 3 and 4. The population evolution has been limited to five generation in order not to exceed an optimization time of 30 minutes. Fortunately, during the optimization beam current fluctuation from the CRYRING@ESR source has been low. Even in the first generation a similar transmission as with a manual optimization could be reached, since the parameter space of the first generation has been limited to ±10% of the known optimal settings. As the next population is selected from both the offspring's and current population the number of fitter individuals grows with generations. Nevertheless, the generated and evaluated offspring covers a large parameter space indicated through different beam currents.

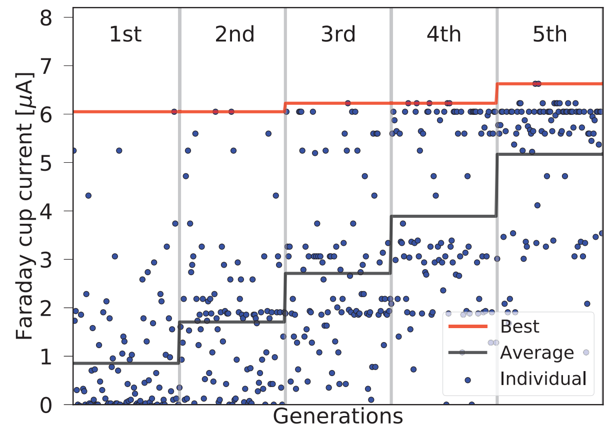

Figure 3: Evolution of the population fitness represented through beam current along generations. As the next population is selected from both the offspring's and current population the number of fitter individuals grows with generations.

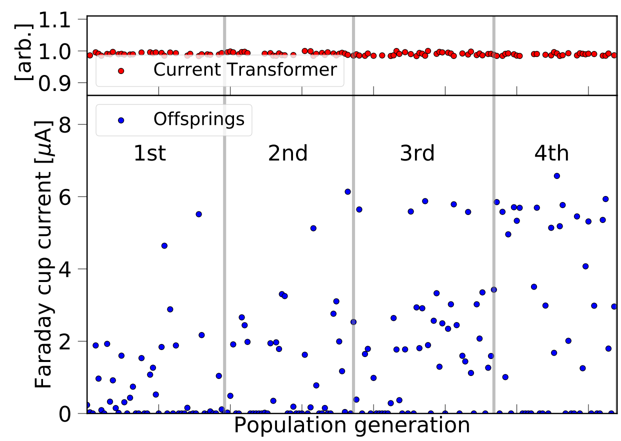

Figure 4: The offspring's fitness represented through beam current along generations. Only the best offspring's replace parents in the next population. The measurements from the uncalibrated current transformer are in arbitrary units, still show the low ion source fluctuations.

CONCLUSION AND OUTLOOK

As a result of the promising simulation outcome of optimizing the multi-turn injection as well as beam lines, the PEP Project has been launched. The first automatic PEP version at the CRYRING@ESR injector has been implemented and tested. A good transmission could be reached in half an hour of optimization time. Still, the PEP Project is at its beginning and many improvements as well as detailed studies have to be made. The influence of population, generation size, crossover and mutation should be studied as well as other genetic algorithms, particle swarm algorithms or machine learning algorithms should be tried. Before the parameter space can be expanded, some trigger must be included like 'measurement failed and has to be repeated' and 'Set value of devices have been reached'. Crucial for the transmission is an optimization of beam size as well as position and must therefore be included. For the GSI beam time in 2018 it is planned to test PEP at the transfer channel to SIS18 for optimizing the injection.

REFERENCES

[1] S. Appel, O. Boine-Frankenheim, and F. Petrov, "Injection optimization in a heavy-ion synchrotron using genetic algorithms", *Nucl. Instrum. Methods* A, 852 (2017) pp. 73–79.

[2] S. Ratschow, F. Hagenbuck, and P. J. Spiller, "The High Energy Beam Transport System for FAIR", in *Proc. EPAC'08*, paper THPP104, pp.3608–3610 (2008).

[3] S. Reimann, M. Sapinski, P. Schütt, and M. Vossberg, "Building an Operation Team for FAIR Nearly from Scratch", presentation at WAO'10 (2016).

ISBN 978-3-95450-182-3

[4] K. Tian, J. Safranek, and Y. Yan, "Machine based optimization using genetic algorithms in a storage ring", *Phys. Rev. ST Accel. Beams* 17, 2 (2014).

[5] W. Geithner, Z. Andelkovic, D. Beck, H. Bräuning, A. Bräuning-Demian, H. Danared, C. Dimopoulou, M. Engström, S. Fedotova, O. Gorda, F. Herfurth, R. Hess, A. Källberg, C. Kleffner, N. Kotovskiy, I. Kraus, M. Lestinsky, S. Litvinov, F. Nolden, A. Reiter, T. Sieber, M. Steck, and G. Vorobyev, "Status and outlook of the CRYRING@ESR project", *Hyperfine Interact.* 238, (2017).

[6] O. Gorda, A. Braeuning-Demian, H. Danared, A. Dolinskii, M. Engstroem, F. Herfurth, A. Kaellberg, C.-M. Kleffner, M. Lestinsky, A. Simonsson, J. Sjoeholm, and G. Vorobjev, "Ion-optical design of CRYRING@ESR", *Phys. Scr. T* 166, 14043 (2015).

[7] A. Källberg and A. Simonsson, "Beam Steering with Image Processing in the CRYRING Injection Beamline", in *Proc. of DIPAC'99*, pp. 118–119 (1999).

[8] A. Konak, D.W. Coit, and A.E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial", *Reliab. Eng. Syst. Saf.* 91 (9) (2006) 992–1007, `doi:10.1016/j.ress.2005.11.018`

[9] A. Hofler, B.Terzić, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis, "Innovative applications of genetic algorithms to problems in accelerator physics", *Phys. Rev. ST Accel. Beams* 16 (1) (2013) 010101, `doi:10.1103/PhysRevSTAB.16.010101`

[10] X. Pang and L. Rybarcyk, "Multi-objective particle swarm and genetic algorithm for the optimization of the LANSCE linac operation", *Nucl. Instrum. Methods* A, 741 (2013), pp. 124–129.

[11] A. Shishlo, S. Cousineau, J. Holmes, and T. Gorlov, "The Particle Accelerator Simulation Code PyORBIT", *Procedia Comput. Sci.* 51 (2015) 1272–1281, `doi:10.1016/j.procs.2015.05.312`, `https://github.com/PyORBIT-Collaboration`

[12] F. Fortin and D. Rainville, "DEAP: Evolutionary algorithms made easy", *J. Mach. Learn. Res.* 13 (2012) 2171–2175, `https://github.com/DEAP/deap`