ELECTRON TRANSPORT LINE OPTIMIZATION USING NEURAL NETWORKS AND GENETIC ALGORITHMS

D. Schirmer, P. Hartmann, T. Büning, D. Müller
Center for Synchrotron Radiation (DELTA),
University of Dortmund,
Maria-Goeppert-Mayer Str. 2,
44221 Dortmund, Germany

Abstract

Methods of computational intelligence (CI) were investigated to support the optimization of the electron transfer efficiency from the booster synchrotron BoDo to the electron storage ring DELTA. Neural networks (NNs) and genetic algorithms (GAs) were analyzed alternatively. At first both types of methods were tested on the basis of a theoretical model of the transport line. After the training various algorithms were used to improve the magnet settings of the real transport line elements with respect to the electron transfer efficiency. The results of different strategies are compared and prospects as well as limitations of CI-methods to the application of typical optimization problems in accelerator operation are discussed.

INTRODUCTION

The Dortmund 1.5 GeV electron storage ring facility DELTA serves universities and industry as a source of synchrotron radiation. It consists basically of three accelerator components. A linear accelerator (LINAC), where the electrons are pre accelerated to an energy of 60 MeV, a computer-ramped booster synchrotron (BoDo) for the acceleration up to the maximum final energy of approx. 1.5 GeV and the main storage ring (Delta) in order to store electrons for many hours providing synchrotron radiation for miscellaneous beamlines [1]. For each transition there exists an electron transport line between these accelerator components (T1: LINAC to booster; T2: booster to storage ring). The object of this paper is a software based injection efficiency optimization of the transport line T2 between the booster synchrotron BoDo and the main storage ring Delta.

LAYOUT OF THE TRANSFER LINE

The transport line T2 consists of the BoDo extraction septum (BOEXSEPT; pulse length 88μs), two pulsed dipoles (B1, B2; 100ms), four dc-quadrupoles (Q1-Q4), four horizontal steerers (HK0-HK3) as well as two vertical steerers (VK1-VK2) and an injection septum (DESEPT; 83μs). The online beam position measurement in both planes is performed by three BPMs (BPM1-BPM3) [3]. Figure 1 depicts the layout of the transfer line.

The energy ramping cycle of the synchrotron booster is software controlled and requires 6.5 seconds from 60 MeV up to 1.5GeV. In average a charge corresponding to approximately $I_{BoDo} \approx$ 4mA is ramped each cycle and completely extracted by a septum magnet into the transfer channel. Assuming an injection efficiency of 40% one needs about 170 ramp cycles ($\hat{=}$ 18 minutes) to fill the storage ring up to the maximum current of I_{Delta} =120mA.

Both, Genetic Algorithms (GAs) as well as Neural Networks (NNs) need a large number of real time data for the production of individuals or training the neural nets (generation of input/output pairs). 170 ramp cycles correspond to 170 input/output net training pairs or individuals of a population, which are not sufficient for an CI-method optimization. To overcome this problem a software optics/orbit-model (T2-simulator) of the T2 has been established for fast data acquisition.

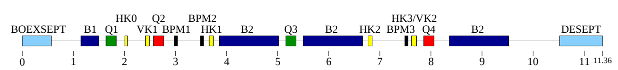

Figure 1: Schematical view of the transport line T2.

BASIC IDEA

The goal is an automated online optimization of the injection efficiency during the injection phase. Unfortunately this value depends on a huge number of parameters including essentially [2]:

The starting conditions at the extraction point of the booster BoDo (beam position and angle). The settings of all deflecting magnets along the T2 as well as the quadrupole strengths and thus the overlap of the phase space ellipses at the injection point. The orbit of the storage ring especially the seclusion of the Delta kicker bump in combination with an add on DC-bump in the injection region. The tune of the storage ring and the timing of all pulsed T2-magnets are also sensitive parameters.

Since not all parameters could be measured and controlled by the CI-algorithms, we limited, for a first test, the number of controlled devices only to the deflecting elements and kept all other above-named parameters, as far as possible, constant. As an input we used only the T2 BPM orbit data and as the figure of merit the injection efficiency measurement.

Measurement of the Beam Position and the Transfer Efficiency

The injection efficiency $G = \Delta Q_{Delta}/Q_{BoDo}$ is defined by the ration of charge transfer Q_{BoDo} from the booster BoDo to the charge change ΔQ_{Delta} in the storage ring Delta. With $Q = I \cdot T$ and under consideration of the different revolution times (T=384ns resp. 168ns for Delta and BoDo) one gets:

$$\mathrm{G} = \frac{384}{168} \cdot \frac{I_{\texttt{Delta},\,\text{after injection}} - I_{\texttt{Delta},\,\text{before injection}}}{I_{\texttt{BoDo},\,\text{before extraction}}}. \tag{1}$$

The error of the measurement is a few percent depending mainly on the amount of transfered charge and the beam lifetime of Delta [2].

The amount of transfered charge per injection cycle is in the order of $0.1 - 1.0$nC with a pulse duration of 10 to 40ns. The resulting peak current is not sufficient for a self triggering of the single shot BPM electronics. Therefor it was necessary to establish an external forced trigger hooked on the BoDo extraction kicker timing. The overall statistical beam position error at each BPM results in less than 300μm taking into account the BPM calibration data, the fluctuations of the BoDo current and the jitter of the trigger timing system [4],[3].

Controlling and Modelling the T2 Devices

All devices at DELTA are controlled by the control system software EPICS [5]. The Accelerator Toolbox (AT) [6] is a collection of tools to model particle accelerators and beam transport lines in the MatLab [7] workbench environment. The 'mca' packages [8] provide an interface to the EPICS channel access (CA) client library which can be integrated with MatLab applications and toolboxes. Additionally, MatLab provides a extensively bibliography of GA/NN-algorithms. Thus, MatLab unifies all essential ingredients in one workbench: read/write control of all devices; T2 modelling; GA/NN-algorithm test bed; visualization; file handling and programming.

APPLIED CI-METHODS

Two different categories of CI-techniques have been investigated [4]. At first GA-algorithms with various rules of selections, mutations, recombinations as well as population sizes have been studied. This includes also a T2-model versions, including 'virtual' correction coils inside the quadrupole magnets, simulating a steering effect of misaligned magnets.

The second approach are neural networks. Classical Feed-Forward Neural Networks (FFNN) as well as Radial Basis Function (RBF) networks with different training schemes (batch training, Delta rule, resilient backpropagation) have been studied.

Optimization Using Genetic Algorithms (GAs)

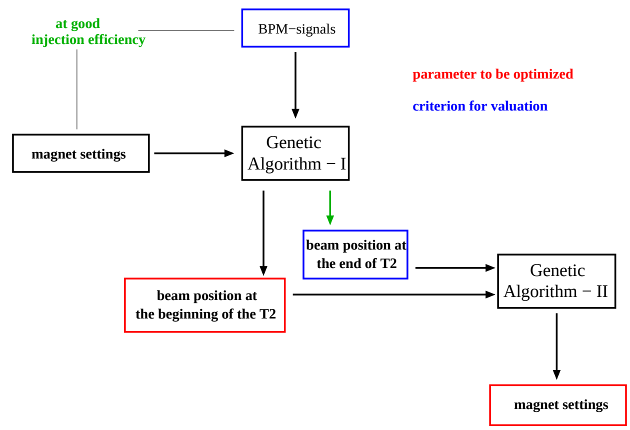

Figure 2: Optimization procedure.

The GA optimization proceeds in two major steps: Matching the T2-model and optimizing the magnet settings (see Figure 2). Initial step: Loading the T2 input lattice file to simulate the transfer line optics/orbit. Second step: Reading the real machine data at good injection efficiency (more than 30%) representing a snapshot of all BPM data and the corresponding T2 magnet setup (setting of all dipoles, quads, steerer and septa). Third step: Starting optimizing by launching GA-I. This step simulates the beam position at all BPMs by mutating the start orbit vector $\vec{X}_{start} = (x_0, x\prime_0, y_0, y\prime_0)$. The deviations between the real BPM data and simulated BPM data are a measure of the quality (so called fitness) for each individuum of the GA (here the start vector).

$$fitness_{GA-I} = \sum_{i=1}^{3} (\mathrm{BPM}_{i,real} - \mathrm{BPM}_{i,sim})^2 . \tag{2}$$

As a result one gets the optimal start orbit vector which fits the real BPM data and the appropriate computed end vector under good transfer conditions. Assuming that a good transfer efficiency is primarily defined by the beam position and slope at the end of the T2 ($\hat{=}$ Delta injection point) one defines this optimized end vector as the figure of merit for the next optimization step. Whensoever the transfer efficiency is poor (e.g. few percent), the subsequent GA-II mutates the magnet settings of the T2 (in that case the individuals of the GA-II). Now, the minimization of deviations between simulated and the previously optimized end vector is the criterion for evaluation (fitness function)

$$fitness_{GA-II} = \sum_{i=1}^{4} \left(\vec{X}_{end,opt}(i) - \vec{X}_{end,sim}(i)\right)^2 . \tag{3}$$

Several types of rules for miscellaneous GAs have been tested (e.g. intermediate crossover, uniform mutation, with/without recombination/crossover, gaussian mutation etc.)[4]. On the basis of this procedure a recovery of the original beam position was successful, but the restoration of the 'real' injection efficiency was not always reproducible. One result is shown in Figure 3 as an example.

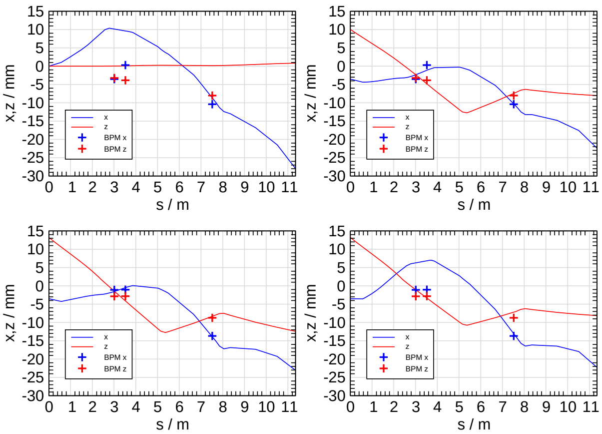

Figure 3: Orbit along the T2 (model). Up: before/after 'GA-I-matching' the initial values of the start vector. Down: before/after 'GA-II-optimization' of the magnet settings. Crosses indicate real orbit data.

Optimization Using Neural Networks (NNs)

The neural networks were also trained using the T2 simulator. A snapshot of the real T2 at good transfer efficiency defines a reference orbit and a corresponding reference magnet setting. Based on this reference, arbitrary offsets on the T2-simulator magnets generates new simulated orbit data. Hence, the input for the net is a 6-dim vector of beam positions at the BPMs and the output is a 10-dim vector of magnet corrections (only deflecting devices). Several hundred of input/target pairs were generated to train the neural networks.

Mainly two network topologies have been investigated. A three layer Feed-Forward NN (FFNN) and a two layer Radial Basis Function (RBF) model. For each network different kinds of learning algorithms, transfer (bias) functions (gaussian, hyperbolic tangent, linear) and number of neurons per layer have been studied. For more details see [4], [7]. Figure 4 compares T2 simulator examples for a beam correction at the BPM positions applying RBF and Feed-Forward neural networks (e.g. FFNN with Resilient Backpropagation (RPROP) training).

It turned out, that within the model a orbit correction was possible with almost all nets. In general RBF neural networks produce better results and could be trained faster than FFNNs [4]. By contrast, applying the simulator trained NNs to the real T2 failed, so that the results of the simulation were non-transferable. To some extend, the NNs proposed magnet settings which were not in the scope of their physical limits. The assumption is, that the real T2 is not covered by the simulated produced training data pairs, which could also be interpreted as an hint for an insufficient T2 model.

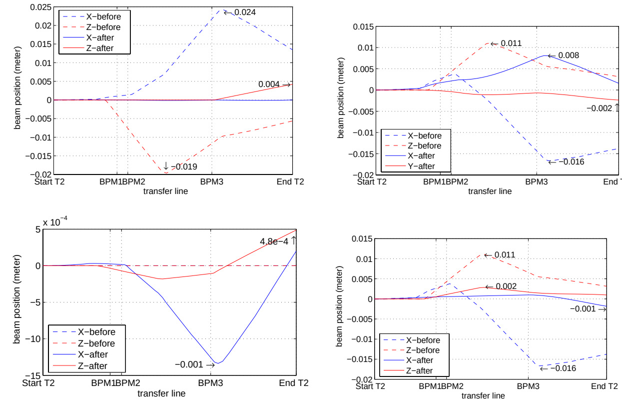

Figure 4: T2 orbit correction using different kinds of neural networks (RBF, FFNN).

CONCLUSION AND PROSPECTS

The investigated GAs and NNs fulfilled the requirements only in parts, the reproduction of the beam position was possible but the reproduction of the 'real' injection efficiency could not be achieved at all times.

Because both optimization methods, GAs as well as NNs, require a large number of input data, it was necessary to establish a transfer simulator for fast data generation. Especially in case of the Delta transfer channel this simulator demands a theoretical model which represents the problem as accurately as possible to reflect the real behavior of the machine sufficiently.

However, unknown misalignments and field errors of the T2 magnets, inducing a steering effect of the quadrupoles, seemed not to be considered satisfactory. A timing jitter of all pulsed magnets and beam current dependent multibunch instabilities appeared to be responsible for a statistical spread of the transfer efficiency. All these effects make it rather difficult to simulate the injection efficiency, the core value of both CI-methods.

An alternative approach will be the postprocessing of real time data accumulated over a long period of time. NNs trained with these data should describe the real machine much better.

REFERENCES

[1] D. Schirmer, et. al., *Status of the Synchrotron Light Source DELTA*, EPAC 2004, Luzern, Switzerland.

[2] E. Zimoch, Ph.D. Thesis, University of Dortmund, 2003.

[3] P. Hartmann, et. al., *A Compact Single Shot Beam Position Monitor for DELTA*, EPAC 2002, Paris, France.

[4] T. Büning, D. Müller, Diploma Thesis, University of Dortmund, Germany, 2005.

[5] *EPICS Home Page*, `http://www.aps.anl.gov/epics/`.

[6] A. Terebilo, *Accelerator Toolbox for MATLAB*, SLAC-PUB-8732, 2001.

[7] The MathWorks, *MATLAB*, `www.mathworks.com`

[8] A. Terebilo, *Channel Access Client Toolbox for MATLAB*, 8th ICALEPCS San Jose, 2001.