ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA138

ORBIT CORRECTION WITH MACHINE LEARNING TECHNIQUES AT THE SYNCHROTRON LIGHT SOURCE DELTA

D. Schirmer*
Center for Synchrotron Radiation (DELTA), TU Dortmund University, Germany

Abstract

In the last years, artificial intelligence (AI) has experienced a renaissance in many fields. AI-based concepts are nature-inspired and can also be used in the field of accelerator controls. At DELTA, various studies on this subject were conducted in the past. Among other possible applications, the use of neural networks for automated correction of the electron beam position (orbit control) is of interest. Machine learning (ML) simulations with a DELTA storage ring model were already successful. Recently, conventional Feed-Forward Neural Networks (FFNN) were trained on measured orbits to apply local and global beam position corrections to the 1.5–GeV storage ring DELTA. First experimental results are presented and compared with other orbit control methods.

INTRODUCTION

DELTA is a 1.5–GeV electron storage ring facility operated by the TU Dortmund University supplying radiation ranging from THz to the hard x-ray regime [1, 2]. The transverse position of the electron beam is measured at 54 capacitive multiplexed pick-up monitors (BPMs) installed along the 115 m long vacuum chamber of the storage ring [3]. For beam position correction, 30 horizontal (HC) and 26 vertical dipole correctors (VC) are available. A slow orbit feedback system (SOFB), ranging from 0.2 Hz to 1 Hz and based on a singular value decomposition (SVD) algorithm of measured orbit response matrices (ORM) is in operation since 2005 [4].

Because of increasing software issues and for control system maintenance reasons, a revised software version was required. Therefore, a cone-programming-based global correction approach has been developed and is in a testing phase now [5, 6]. An alternative concept applies machine learning techniques as an heuristic method, inspired by the pioneering work done at NSLS/BNL [7, 8].

ORBIT CORRECTION AT DELTA

All steerer magnets are additional coils on quadrupole yokes which can be ramped to a current of max. ± 10 A according to beam kicks of max. ±3 mrad at 1.5 GeV. They are controlled via 12-bit digital-to-analog converters (DACs) integrated on CAN to serial bus converter modules [9]. The DACs allow current changes with a granularity of 2.4 mA which corresponds also to the minimum read-back resolution [10].

The analog BPM signals are read out via a mix of I-Tech Libera [11] and Bergoz MX [12] electronics. The MX-BPMs provide the measured beam position as an analog voltage that is digitized by 12-bit analog-to-digital converters (ADCs) [13] and fed over a CAN-bus into the EPICS control system [14]. A 10 Hz low-pass filter reduces sampling noise while maintaining sufficient bandwidth for the SOFB. The measuring accuracy is approx. ±5 µm mainly limited by the resolution of the ADCs. At beam currents above 2 mA the 'slow acquisition' data from Libera BPMs is of roughly the same quality as the data from the MX-BPMs. For a more detailed description see [3, 15].

The 'zero orbit', i. e., the orbit with all correctors switched off, deviates from the design orbit of the storage ring due to alignment and field errors of magnets as well as unavoidable non-linear magnetic fringe fields. At DELTA, this orbit is normally not usable for standard synchrotron user operation, since it implies large beam amplitudes and angles and thus causes vacuum chamber heatings, reduces injection efficiency and does not optimally illuminate the synchrotron radiation beamlines. Through empirically adjusted orbit shifts and optionally by putting weights at dedicated BPM positions, which increase the impact of orbit deviation at important ring positions (e. g., beamline source points or injection septum), a new so-called reference orbit is defined. Nevertheless, due to temperature drifts, malfunctions and miss-calibrations of BPMs, failure of steerer magnets as well as variations in BPM weight and offset values, the currently measured orbit still deviates slightly from the desired reference orbit. An orbit correction (OC) algorithm keeps these deviations as small as possible. The quality of an OC algorithm can be expressed by the Euclidean norm $E_{x,z} = \sqrt{\sum_{i=1}^{N} (\Delta_{x,z})_i^2}$. It considers the orbit errors in both planes ($\Delta_{x,z}$) between the currently measured orbit and the reference orbit at all 54 BPM positions i=1..54. For a more detailed description see [4].

MACHINE LEARNING DESIGN STEPS

The development of an ML-based OC application passes six major steps (see Fig. 1). The first steps are data acquisition (DAQ) and cleaning. Afterwards, the neuron network topology is defined and optimized. Finally, after multiple training sessions with continuous performance tests, the OC application has to be tested in real machine operation. A number of special programming environments are available for implementing ML applications. First tests were carried out within the frameworks TensorFlow [16] and Keras [17]. For investigations conducted in this work, the computing environment MatLab (Matrix Laboratory) and corresponding

[*] detlev.schirmer@tu-dortmund.de

machine learning toolboxes were used [18]. The essential development steps are explained in the following sections.

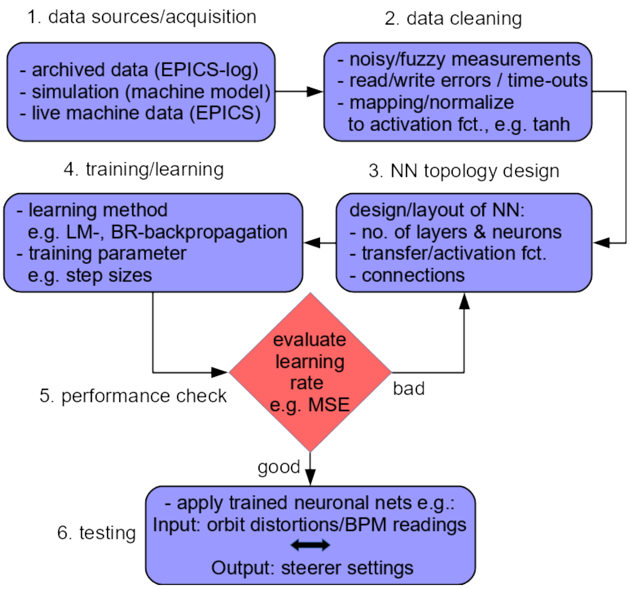
1. data sources/acquisition
- archived data (EPICS-log)
- simulation (machine model)
- live machine data (EPICS)
2. data cleaning
- noisy/fuzzy measurements
- read/write errors / time-outs
- mapping/normalize
to activation fct., e.g. tanh
3. NN topology design
design/layout of NN:
- no. of layers & neurons
- transfer/activation fct.
- connections
4. training/learning
- learning method
e.g. LM-, BR-backpropagation
- training parameter
e.g. step sizes
5. performance check
evaluate
learning
rate
e.g. MSE
bad
good
6. testing
- apply trained neuronal nets e.g.:
Input: orbit distortions/BPM readings
Output: steerer settings

Figure 1: Development stages for an ML-based OC.

Data Mining and Data Cleaning

In order to perform supervised neural network (NN) training (see Fig. 2a), a large number of data pairs, i. e., BPM readings (NN inputs) correlated with steerer settings (NN targets), must be available. These data sets can either be generated by accelerator simulations, obtained from logged archive data (e. g., EPICS-log database) or be measured. ML-based OC simulations were already successful, but only partially reflect real storage ring behavior [19]. Since adequate experimental data, i. e., large number of randomly disturbed orbits, from EPICS-log archive was not available, a special data mining program was implemented. This program varies randomly all steerer strengths within an interval of typically ±200 to 300 mA. The interval limits are a compromise between risk of beam loss and minimizing relative measurement errors due to the limited steerer strength resolution of 2.4 mA. After each perturbation, the differences in orbit response and in steerer strengths are recorded. Erroneous readings (e. g., read/write time-outs, noisy data) are directly filtered out. On average, approx. 15-20 seconds are required per single measurement cycle. This way, several hundred random steerer-BPM data combinations (corresponding to ML input-target data pairs) were recorded in several measuring sessions.

Measurement of standard orbit response matrices for various positive and negative single steerer kicks were carried out similary. All measurements of both sources were merged to a common data pool of approx. 1500 data pairs to obtain at least ten times as much training data as neurons in the neural network (see below).

As an example Fig. 3 depicts the random orbit error ($E_{x,z}$) distribution (ML input data) and the corresponding random steerer strengths distribution (ML target data). For fixed dicing limits of ±0.2 A (horizontal steerer), ±0.3 A (vertical steerer) the center of the orbit error distribution is in the range of 4 mm (horizontal) and 2 mm (vertical), whereby larger orbit errors are overrepresented [1].

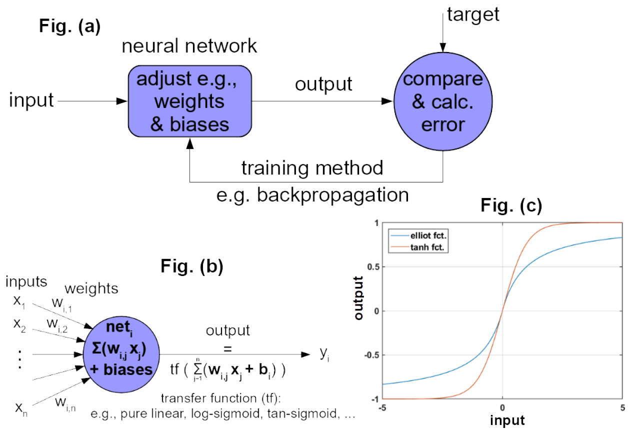

Figure 2: Illustrations of supervised learning (a), neuron model (b) and neuron activation function (c).

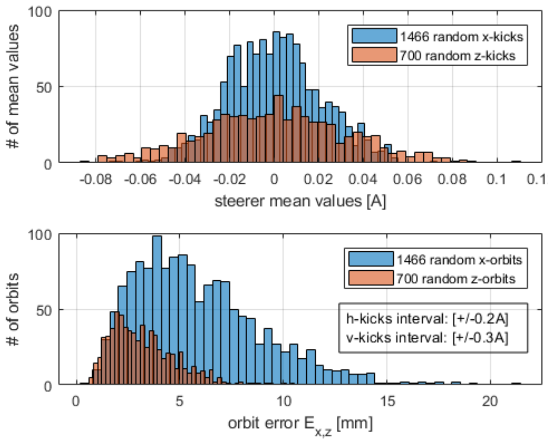

Figure 3: Measured ML training target (top) and input (bottom) data for the horizontal (blue) and vertical (red) plane.

Definition of the Neural Network Topology

At first a basic NN structure adapted to the orbit correction problem was defined. It consists of 3 neuron layers with 54 input layer neurons (corresponding to the number of BPMs) and 30/26 output layer neurons (corresponding to the number of horizontal/vertical steerers). The number of hidden neurons was determined empirically by trial and error. It turned out, that a too large number of hidden neurons (much more than input neurons) leads to ‘overfitting’ [18] and too few hidden neurons (much less than output neurons) downgrades the learning performance. Using the same number

[1] Reducing and randomizing the interval limits would shift the $E_{x,z}$-distributions to smaller values.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA138

of input neurons as hidden neurons was a good compromise in many cases.

Since each corrector strength change usually effects the beam position at all BPMs in the storage ring, a classical fully-connected feed-forward neural network (FFNN) was utilized for the neuron network connection topology. This basic network was extended by further neurons. One neuron represents the radio frequency (RF) of the storage ring cavity. An RF change varies the electron energy and thus generates a dispersion trajectory with orbit shifts at all BPMs. Additional three neurons consider the actual status of the three insertion devices (SAW, U250, U55) installed at the DELTA storage ring. At DELTA, large beam amplitudes at sextupole magnets integrated into the quadrupole magnets generate also strong orbit kicks which can be of the same order as steerer kicks. Up to now, this effect is not considered in the ML-based OC algorithm but could be taken into account by additional 'sextupole neurons'. Figure 4 illustrates the neural network topology utilized for the horizontal orbit plane only.

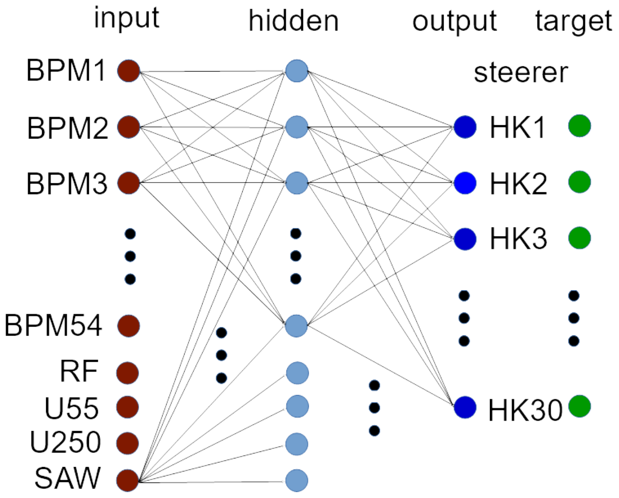

Figure 4: Feed Forward Neural Network (FFNN) topology used for ML-based horizontal orbit correction.

Supervised Learning

Before actual FFNN supervised learning can be performed (see Fig. 2a), all training input data must be normalized to the value range ±1 of an neuron transfer function (so-called activation) (see Fig. 2b, 2c). From a multitude of possible functions, the hyperbolic tangent function $\tanh(x)$ has proven to be most suitable. It is point-symmetric and continuously differentiable, a necessary condition for many back-propagation training methods and also shows non-linear behavior close to the interval limits ±1. An alternative function is the Elliot sigmoid function $f(x) = x/(1 + |x|)$ (see Fig. 2c), whose expression does not include a call to any higher order functions[2] (e. g. $\exp(x)$) [18]. It was used for first performance studies for a more efficient training on a grafical processor unit[3] (GPU).

The network training performance is usually rated by the mean squared error (MSE). It is calculated by the actual neural network output o_i and the desired target t_i values $\frac{1}{N}\sum_{i=1}^{N} w_i(o_i - t_i)^2$ whereby, as an option, each squared error can be weighted individually (w_i).

ML software frameworks provide a large pool of training methods and functions [16–18]. They were compared regarding their training performance. Best results were reached with back-propagation methods using the Levenberg-Marquardt (LM) algorithm with or without Baysian Regularization (BR) [21, 22] as well as Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm which updates network weight and bias values according to the BFGS quasi-Newton method [23]. Pure gradient decent learning turned out to be slow and often got stuck in local MSE minima.

All network trainings were also performed with two different numbers of hidden layer neurons, equal to the number of input or output neurons, respectively. Moreover, for each training method a variety of so-called training metaparameters to tweak the underlying numerical algorithms (e. g., steps sizes, max./min.-limits for gradients or validation failures, etc.), were checked too. The training data pool (sum of all related measurements) was divided into three sets with a fixed splitting ratio:

1. Data for pure training, i. e., adjusting weights and bias values to minimize the MSE (80%).
2. Validation data is used to prevent learning by pure memorization, i. e., avoid the overfitting problem (10%).
3. Testing data is used to measure how accurately the network was trained, i. e., test with "unseen" data (10%).

As an example, Fig. 5 depicts the FFNN supervised learning performance (MSE) trained with approx. 1500 measured machine data records (orbit/steerer-data sets). In this case, best validation performance of $1.4 \cdot 10^{-4}$ mm was reached after 27 LM-BR-training iterations (so-called epochs) applying all training records at once (so-called full batch training).

Application Results on Real Machine Operation

The best trained NNs, with MSE values less than $3 \cdot 10^{-4}$ mm, were tested for OC during real machine operation of the DELTA storage ring. For this purpose, the current orbit was defined as an arbitrary reference which could also be any other predefined orbit. Random kicks were set to all or individual selected steerers and the orbit error $E_{x,z}$ of the orbit response was determined (iteration 0). The resulting disturbed orbit was fed into the previously trained NNs, which were now able to determine the orbit-related steerer strengths. These values were inverted and transferred back to the steerers, giving a new closed orbit with a significantly smaller MSE (first iteration). Repeating this procedere itera-

[2] Partly a necessary condition for GPU-based learning algorithms.

[3] NVIDIA GeForceGTX1060, CUDA Vers. 6.1 [20].

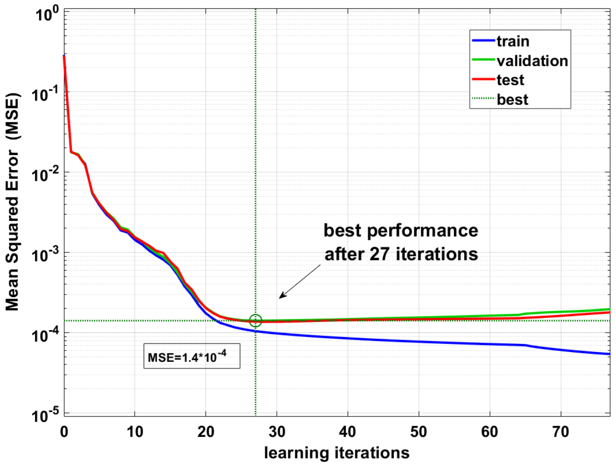

Figure 5: FFNN supervised learning performance trend (LM-BR) for the full batch of approx. 1500 measured orbit/steerer data sets. Best validation performance (MSE) of $1.4 \cdot 10^{-4}$ mm was reached after 27 LM-BR-training iterations (so-called epochs) of all records.

tively reduced the MSE in average nearly by two orders of magnitude (see Figs. 6, 7).

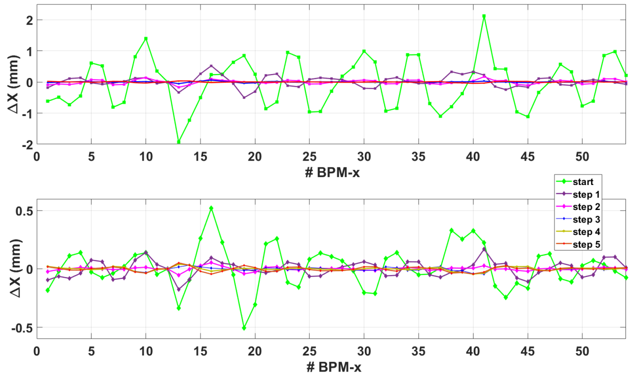

Figure 6: Iterative application of the pretrained FFNN referred to the previously corrected orbit, starting from a randomly disturbed orbit (start). After 3 successive correction steps, an error of < 200 µm was achieved.

The performance of ML-based OC was compared to two other computing methods, singular value decomposition (SVD) [4] and qpcone-based (convex optimization) [5, 6], respectively. Figure 7 depicts exemplarily the horizontal orbit error E_x depending on correction iterations for randomly disturbed orbits (mean values and standard deviation). All tested OC algorithms converge to comparable results after multiple successive correction steps. For ML-based OC, the absolute closed orbit error was reduced to less than 200 µm after three iterations. This error can be further reduced by increasing the number of training data sets in the small $E_{x,z}$-value range (see Fig. 3). The minimum residual error must be related to the limited BPM readout resolution of $E_{x,z} = \sqrt{54 \cdot (5\mu\text{m})^2} \approx 40\,\mu\text{m}$.

Since sufficiently well trained NNs for both planes were available, two more OC tests were carried out. (i) Stepwise power down of a strongly unbalanced orbit bump induced by steerer-independent DC-coils (starting at an amplitude of 10 mm). (ii) Variation in strength of an unmatched insertion device (undulator U250). In both cases strong *x*/*z*-orbit disturbances were provoked. The ML-based OC was able to compensate these distortions in both planes without beam losses, even though the related data were not teached during ML.

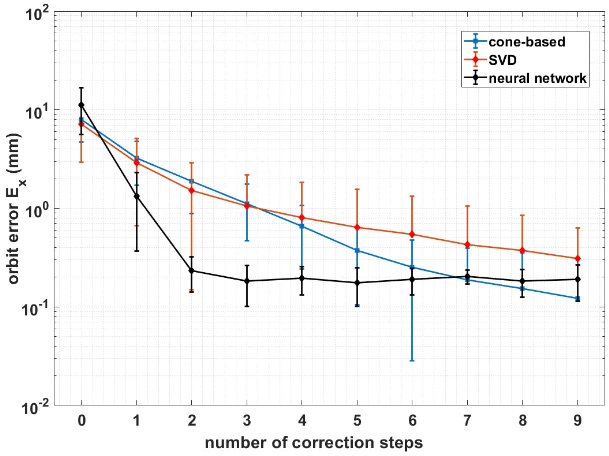

Figure 7: Performance comparison of different OC implementations.

SUMMARY

Although the ML-based OC is only a prototype application so far, it has been shown that ML techniques are an alternative approach for automated orbit correction of the DELTA storage ring. ML incorporates the real storage ring as it was set up at the time of training data acquisition. Therefore, all machine imperfections like non-linearities (e.g. sextupoles) or other-higher order multipoles due to magnetic fringe fields effects are intrinsically considered. Even alignment errors and their impact on the closed orbit can be taken into account. Due to the heuristic approach, they show once trained fast correction convergence and high numerical stability and robustness, since only simple matrix multiplications have to be applied. Moreover, although the potential has not yet been fully exploited, the residual orbit error is competitive to SVD-based methods and close to qpcone-programming-based approaches.

OUTLOOK

Despite ML-based OC is competitive, there is still potential for optimizations.

Up to now, BPM weights are not taken into account but can be considered by an additional neuron network layer or by introducing weighted bias learning.

In order to keep the neural network small and therefore the training time acceptably short, horizontal and vertical

OC were treated independently so far. To study *x/z*-coupling effects, the fusion of both *x/z*-NNs is already in preparation.

Large orbit amplitudes in strong sextupole magnets generate distortions similar to steerer strength changes. This could be taken into account by including additional 'sextupole neurons' into the neural network structure. Other NN topology expansions like the integration of already installed additional weak but fast steerer magnets for minor orbit corrections will also be investigated in the near future.

During routine machine operation, the standard OC applies always small orbit amplitude variations which are close to the desired reference orbit. These data could be used for adaptive (on the fly) ML techniques like the so-called 'sliding windows method' [24]. However, continuous background training via 'on the fly' DAQ requires high computing performance. Here, GPU-based computing could be much more efficient. First tests showed learning speed gains of about two orders of magnitude. The disadvantage of time-consuming NN retraining, due to significant storage ring setup changes can also be mastered by future introduction of adaptive learning techniques.

Last but not least, the longterm stability and robustness of the ML-based OC must further be proven in standard machine operation.

ACKNOWLEDGEMENT

I would like to thank all colleagues of the DELTA team for many helpful discussions and suggestions as well as for providing sufficient data mining time during accelerator shifts. Furthermore, I thank all colleagues who inspired me to do this work.

REFERENCES

[1] T. Weis *et al.*, "Status of the 1.5 GeV Synchrotron Light Source DELTA and Related Accelerator Physics Activities", in *Proc. 20th Russian Particle Accelerator Conf. on Charged Particle Accelerators (RuPAC'06)*, Novosibirsk, Russia, Sep. 2006, paper MONO02, pp. 138-140.

[2] S. Khan *et al.*, "Generation of Ultrashort and Coherent Synchrotron Radiation Pulses at DELTA", *Synchrotron Radiation News*, vol. 26, pp. 25–29, 2013. doi1:0.1080/08940886.2013.791213

[3] A. Jankowiak, "Strahldiagnose und Closed-Orbit-Charakterisierung mit HF-Strahllagemonitoren am Beispiel der Synchrotronstrahlungsquelle DELTA", dissertation, Dortmund University, 1999.

[4] M. Grewe, "SVD-basierte Orbitkorrektur am Speicherring Delta", dissertation, TU Dortmund University, Germany, 2005.

[5] S. Kötter, B. Riemann, and T. Weis, "Status of the Development of a BE-Model-Based Program for Orbit Correction at the Electron Storage Ring DELTA", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, pp. 673–675. doi:10.18429/JACoW-IPAC2017-MOPIK065

[6] S. Kötter, A. Glaßl, B. D. Isbarn, D. Rohde, M. Sommer, and T. Weis, "Evaluation of an Interior Point Method Specialized in Solving Constrained Convex Optimization Problems for Orbit Correction at the Electron Storage Ring at DELTA", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 3507–3510. doi:10.18429/JACoW-IPAC2018-THPAK114

[7] E. Bozoki and A Friedman, "Neural Networks and Orbit Control in Accelerators", in *Proc. 4th European Particle Accelerator Conf. (EPAC'94)*, London, UK , Jun.-Jul. 1994, pp. 1589–1591.

[8] E. Bozoki and A Friedman, "Neural network technique for orbit correction in accelerators/storage rings", in *Orbit Correction and Analysis in Circular Accelerators Workshop*, Upton, NY, USA, Dec. 1993, Brookhaven National Lab., NY, USA, Rep. BNL-61253.

[9] esd-electronics system design gmbh, "can-ccom4 Hardware-Handbuch", CAN-CCOM4 Hardware Rev. 2.0, Hannover, Germanny, 1994.

[10] D. Zimoch, "Implementierung eines Orbitkorrektursystems an der Synchrotronstrahlungsquelle DELTA", dissertation, TU Dortmund University, Germany, 2002.

[11] Instrumentation Technologies, http://www.i-tech.si.

[12] J. Bergoz, "Multiplexed Beam Position Monitor User's Manual", Bergoz Instrumentation, Rev. 1.5.4.

[13] esd-electronics system design gmbh, "can-cai812 Hardware-Handbuch", CAN-CAI812 Hardware Rev. 2.0, Hannover, Germany, 1993.

[14] EPICS, https://epics.anl.gov/index.php.

[15] P. Hartmann, J. Fuersch, D. Schirmer, T. Weis, and K. Wille, "Experience with Libera Beam Position Monitors at DELTA", in *Proc. 8th European Workshop on Beam Diagnostics and Instrumentation for Particle Accelerators (DIPAC'07)*, Venice, Italy, May 2007, paper TUPB21, pp. 111–113.

[16] TensorFlow, https://www.tensorflow.org/

[17] Keras, https://keras.io/

[18] MATLAB/SIMULINK and Neural Network, Fuzzy Logic, Genetic Algorithm Toolboxes, Release 2017b, The MathWorks, Inc., Natick, Massachusetts, United States.

[19] D. Schirmer, "Intelligent Controls for the Electron Storage Ring DELTA", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 4855–4858. doi:10.18429/JACoW-IPAC2018-THPML085

[20] NVIDEA, https://www.nvidia.com/de-de/

[21] David J. C. MacKay, "Bayesian Interpolation", in *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[22] F. Dan Foresee and Martin T. Hagan, "Gauss-Newton Approximation to Bayesian Learning", in *Proc. Int. Conf. on Neural Networks (ICNN'97)*, Jun. 1997, vol. 3, pp. 1930–1935. doi:10.1109/ICNN.1997.614194

[23] Philip E. Gill, Walter Murray and Margret H. Wright, "Practical Optimization", Academic Press Inc. (London) Limited, 1981, ISBN: 0.12.283950.1.

[24] F. M. Dias, A. Antunes, J. Vieira, and A. Mota, "A sliding window solution for the on-line implementation of the Levenberg–Marquardt algorithm", in *Engineering Applications of Artificial Intelligence*, vol. 19, no. 1, pp. 1–7, Feb. 2006. doi:10.1016/j.engappai.2005.03.005