

MACHINE LEARNING TOOLS IMPROVE BESSY II OPERATION

L. Vera Ramírez*, T. Birke, G. Hartmann, R. Müller, M. Ries, A. Schällicke, P. Schnizer
 Helmholtz-Zentrum Berlin, Germany

Abstract

At the Helmholtz-Zentrum Berlin (HZB), user facility BESSY II Machine Learning (ML) technologies aim at advanced analysis, automation, explainability and performance improvements for accelerator and beamline operation. The development of these tools is intertwined with improvements of the prediction part of the digital twin instances at BESSY II [1] and the integration into the Bluesky Suite [2, 3]. On the accelerator side, several use cases have recently been identified, pipelines designed and models tested. Previous studies applied Deep Reinforcement Learning (RL) to booster current and injection efficiency. RL now tackles a more demanding scenario: the mitigation of harmonic orbit perturbations induced by external civil noise sources. This paper presents methodology, design and simulation phases as well as challenges and first results. Further ML use cases under study are, among others, anomaly detection prototypes with anomaly scores for individual features.

MOTIVATION

The complexity of a large-scale facility such as the light source BESSY II in Berlin-Adlershof represents a perfect benchmark for the development, implementation and testing of Machine Learning (ML) tools due to the enormous set of use cases that can be identified - some of which were already presented in [4]. An important factor in order to prioritise these applications is the added value that might be gained through ML, which is enormous in the two cases presented in this paper.

We will first focus on a very challenging application: the mitigation of harmonic orbit perturbations. In this case ML tools aim to improve existing correction systems in the frequency domain (beyond the possibilities of current analytical methods), seeking an increase of the electron beam stability - a critical factor in order to achieve light radiation with high quality brilliance and brightness over time. Besides we will also further introduce original developments towards an anomaly detection system with feature anomaly assignment. This automatic system might extend existing preprogrammed alert system in BESSY's control room and provide additional support to human operators.

MITIGATION OF HARMONIC ORBIT PERTURBATIONS

At the light source BESSY II, the stability of the orbit in the storage ring (within the transverse beam dimensions $100 \times 20 \mu\text{m}$) is currently pursued with a system called Fast Orbit Feedback (FOFB, [5]) running at 150 Hz. FOFB correction is based on the linear approximation $\Delta \mathbf{x} \approx S \Delta \mathbf{c}$ with

\mathbf{x} relative beam position, \mathbf{c} corrector magnets strength and S the so-called *response matrix* (calculated or measured at the accelerator). Hence, for $\mathbf{x}_{t+1} = \mathbf{0}$ we can apply recursively $\mathbf{c}_{t+1} := \mathbf{c}_t - \alpha S^{-1} \mathbf{x}_t$ with S^{-1} Moore-Penrose pseudoinverse of S and α positive constant (at BESSY $\alpha = 0.8$).

FOFB manages to correct orbit perturbations due e.g. to imprecisions of the magnet positioning in a very efficient way. But beyond that, there are several external elements such as civil noise, main power at 50 Hz and some imperfectly isolated magnetic sources (e.g. booster power supply at 10 Hz) that also produce additional inherent perturbations (see Fig. 1). The correction induced by the FOFB system (Fig. 2) is able to mitigate perturbations at lower frequencies (less than ca. 15 Hz) but beyond that point the FOFB system is not that effective and even induces further perturbations (especially in the region 20-40 Hz).

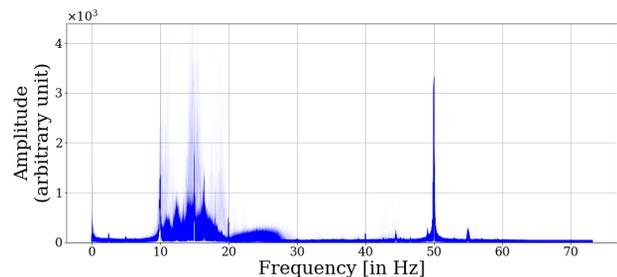


Figure 1: Horizontal beam motion spectra between injections without FOFB (cumulated along 22/04/20, BESSY Archiver data).

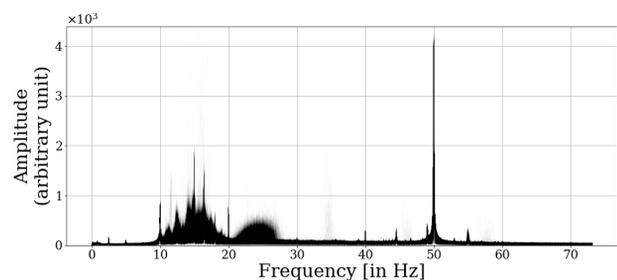


Figure 2: Horizontal beam motion spectra between injections with FOFB (cumulated along 13/05/20, BESSY Archiver data).

A first approach to face this problem was proposed, implemented and tested with simulations in [6]: an explicit correction of the 10 Hz perturbation with a *inverse wave* and an improvement of the PID correction coefficients (proportional-integral-derivative response) in the FOFB algorithm. In this work, we explore the application of ML techniques (in particular Reinforcement Learning, RL) in order to extend the analytical approach with an agent that learns the dynamics

* luis.vera_ramirez@helmholtz-berlin.de

of the system through observation and interaction directly at the machine.

Chronology

- **Early 2020:** first RL tests with simulations (code from [6] and OCELOT [7]). These feasibility tests showed that a RL agent acting in time domain with a proper configuration should be able to also reduce perturbations in frequency domain.
- **July 2020:** infrastructure set-up during machine commissioning and first plausibility tests of the Bluesky-based framework ([2]) up to 20 Hz ([3]).
- **September 2020:** direct zmq-communication with the mBox (fast orbit correction infrastructure), communication up to 100 Hz. First learning results.
- **Since February 2021:** zmq-communication improved, stable RL-interaction loop at 75 Hz, further attempts at 150 Hz.

Baselines

In all the experiments presented in this section, we will compare the performance of our models with three different baseline correction systems:

- *Static steerers:* steerers set back to the initial strength, precalculated with the slow correction system ([8]). The performance of this baseline varies due to machine drifting.
- *FOFB:* standard FOFB system for orbit correction at BESSY II ([5]). For comparison we took a 5-minute period at the end of the corresponding learning tests.
- *FOFB (zmq):* reimplementaion of the response-matrix correction (conditioning 0.04) synchronized with the zmq loop.

Model-Free Approach

The first approach that was tested in BESSY’s control room was a direct translation of the simulations carried out during 2020: the orbit correction is completely undertaken by a RL agent learning and interacting in time domain in a model-free way - i.e. no explicit modelling of the environment is learnt, only the *optimisation dynamics*. According to the simulations, this agent should also be able to stabilise the orbit in frequency domain.

Following our positive experiences in [4] the chosen algorithm was Deep Deterministic Policy Gradient (DDPG, [9]). DDPG is a well-known deep-RL algorithm for continuous environments based on the update of the so-called action-value function $Q : S \times A \rightarrow \mathbb{R}$ with deterministic target policy $\mu : S \rightarrow A$, where S denotes the state space and A the action space. In the DDPG agent proposed in [9], both Q -function and policy are approximated with neural networks - a so-called *deep actor-critic* architecture. The detailed hyperparameter configuration used in our tests can be found in the Appendix.

Regarding the environment settings, we defined the **state** as the last observation of 6 beam position monitors (BPMs) along the ring measured in mm (Δ ’s w.r.t. the reference

orbit). On the other side, we define the **action** as the strength of 6 horizontal steerers along the ring modified up to ± 20 mA (Δ ’s w.r.t. the initial strength). The picked BPMs and steerers are those presenting higher absolute values in the response matrix - i.e. those presenting higher linear correlation. Finally, among the many different definitions tested for the **reward** function, we are presenting the most minimalistic and the most ambitious cases so far.

Reward defined for a single BPM: In our first tests, conceived as a proof-of-concept, the reward was defined through a exponential transformation of the deviation of the next reading from BPMZ5D7R:

$$2e^{-\beta |x_{t+1}|} - 1$$

with β positive constant (for this experiment $\beta = 20$). This BPM was picked because it is used as reference for the beam motion supervision at BESSY II. During this experiment the machine was set in decay mode starting at ca. 20 mA and the zmq-communication was carried out at 75 Hz.

Figure 3 shows the evolution of the beam deviation (in this case measured at a single BPM) during the whole experiment. It alternates exploration periods with baselines and exploitation periods, and the convergence can be easily perceived. Figure 4 carries out the comparison of the last baseline and exploitation period of the experiment with a subsequent FOFB running in comparable machine conditions. One can see that the agent achieves better results in time domain (for this single BPM) than any baseline.

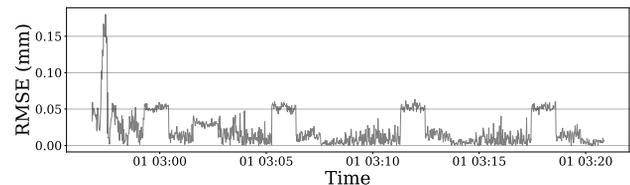


Figure 3: Beam deviation (RMSE) during the whole learning process (test with BPMZ5D7R, 1/3/21, Archiver data).

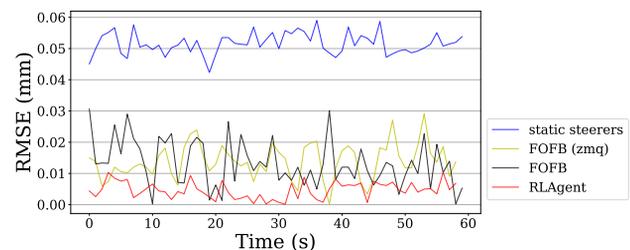


Figure 4: Comparison of the beam deviation (RMSE) after learning (test with BPMZ5D7R, 1/3/21, Archiver data).

If we switch to frequency domain with the reference beam motion measurement at BESSY II (Figs. 5 and 6), we can see that the RL agent carries out the correction of the beam position in a *smooth* way: beyond 20 Hz it improves the stability of standard FOFB and, although it does not manage to correct the inherent harmonic perturbations, it does not add any

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

extra perturbations either, as both FOFB implementations do.

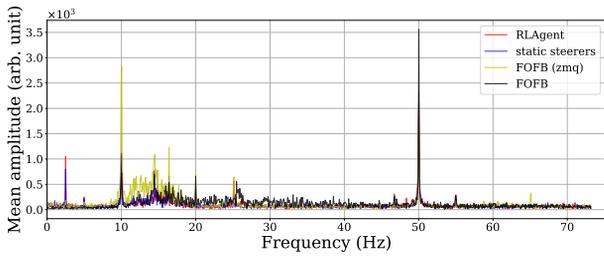


Figure 5: Mean horizontal beam motion spectra comparison (test with BPMZ5D7R, 1/3/21, Archiver data).

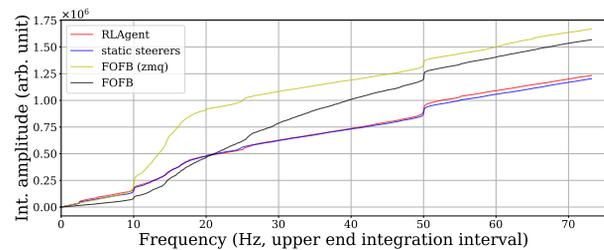


Figure 6: Integrated mean horizontal beam motion spectra comparison (test with BPMZ5D7R, 1/3/21, Archiver data).

Nevertheless, if we carry out an additional frequency analysis through the spectra of the beam motion RMSE with data gathered during the experiment (Figs. 7 and 8), we can observe a reduction of the harmonic perturbations (in norm). In this case, *FOFB* baseline can not be plotted since the data was obtained through the zmq-loop in real time and no comparable data for *FOFB* can be hence obtained.

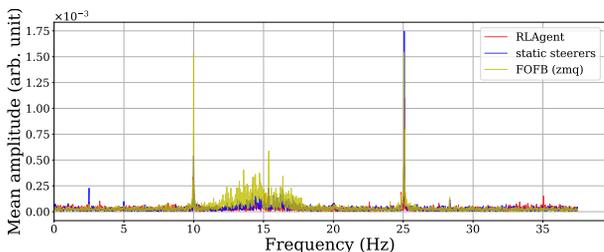


Figure 7: RMSE spectra comparison (test with BPMZ5D7R, 1/3/21, on-the-fly data).

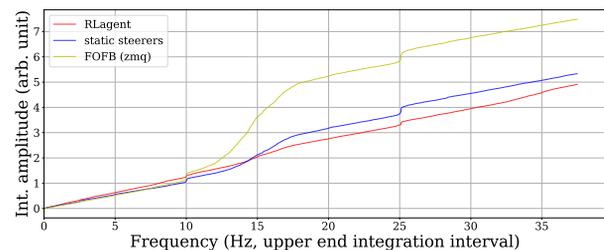


Figure 8: Integrated RMSE spectra comparison (test with BPMZ5D7R, 1/3/21, on-the-fly data).

Reward defined for all BPMs: Now we redefine the reward as an exponential transformation of the RMSE of the next BPM observation for all active BPMs ($m = 102$):

$$2e^{-\beta \text{RMSE}[\mathbf{x}_{t+1}]} - 1 = 2e^{-\beta} \sqrt{\frac{\sum_{i=1}^m (x_{t+1}^i)^2}{m}} - 1$$

with β positive constant (here also $\beta = 20$). This is indeed a much more challenging case for the RL agent (recall that it is only observing 6 BPMs and acting on 6 steerers). In this case the machine was also set in decay mode at ca. 20 mA but the zmq-loop was carried out at 150 Hz, bringing more correction power but also some instabilities in the communication.

Figure 9 shows the evolution of the beam deviation (RMSE for all BPMs) during the whole experiment. In this period the machine suffered from drifting, which made the learning process even more difficult, affecting the baselines as well; Nevertheless, convergence can be also perceived. In the comparison showed in Fig. 10 we can see that this time the agent improves the beam deviation but does not manage to beat the FOFBs in time domain - probably the 6 steerers might not be enough for the global correction.

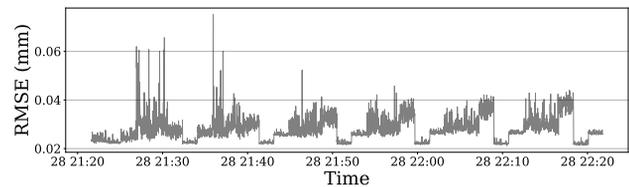


Figure 9: Beam deviation (RMSE) during the whole learning process (test with all BPMs, 1/3/21, BESSY Archiver data).

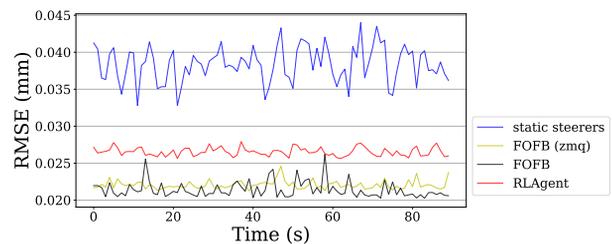


Figure 10: Comparison of the beam deviation (RMSE) after learning (test with all BPMs, 1/3/21, BESSY Archiver data).

On the other side, this experiment's outcome is more positive in frequency domain (Figs. 11 and 12): in the region between ca. 18 Hz and 50 Hz it improves the stability of all baselines, including the *static steerers* - meaning that the inherent perturbations get also slightly mitigated.

The additional frequency analysis through the spectra of the beam motion RMSE (Figs. 13 and 14) shows an even stronger reduction of the harmonic perturbations (in norm).

A natural question might be if the reason for stability in frequency domain is a *low* activity of the steerers when they are controlled by the RL agent. Figure 15 shows that it does not seem to be the case - the 6 steerers manipulated by the

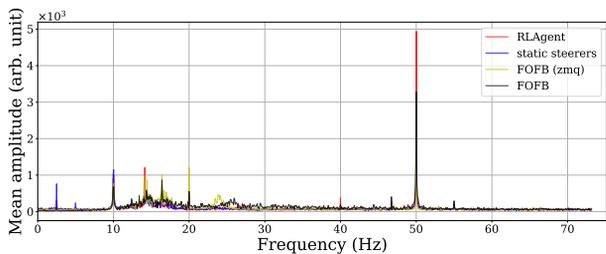


Figure 11: Mean horizontal beam motion spectra comparison (test with all BPMs, 1/3/21, BESSY Archiver data).

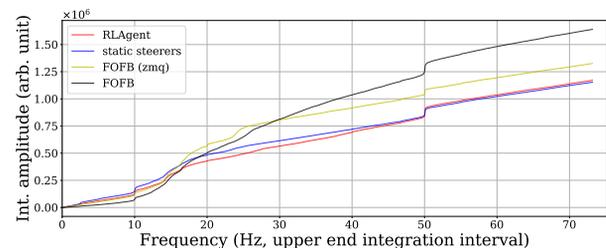


Figure 12: Integrated mean horizontal beam motion spectra comparison (test with all BPMs, 1/3/21, BESSY Archiver data).

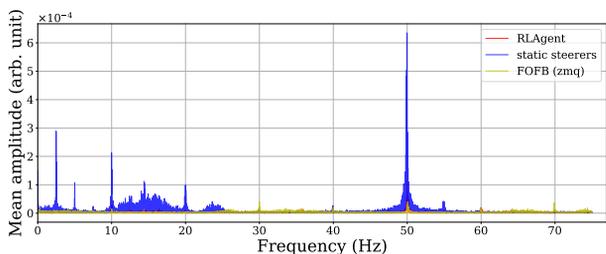


Figure 13: RMSE spectra compasion (test with all BPMs, 1/3/21, on-the-fly data).

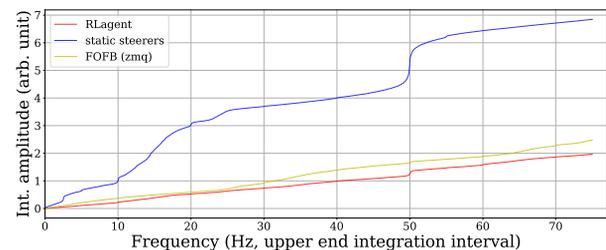


Figure 14: Integrated RMSE spectra compasion (test with all BPMs, 1/3/21, on-the-fly data).

agent present active motion patterns, which remind of the beam motion perturbations they are supposed to counteract.

Towards a Model-Based Approach

Previous experiments with a model-free RL agent were able to carry out sensible orbit corrections in frequency domain. Nevertheless, in time domain traditional algorithms still have a major advantage when the target involves the correction of all BPMs: the need to include more horizontal

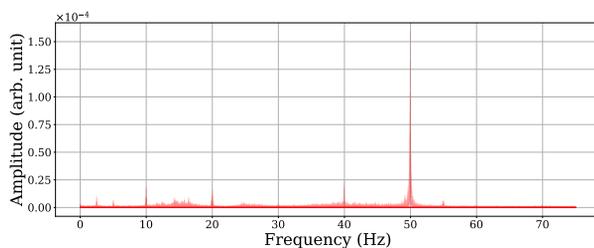


Figure 15: Steerers motion spectra with RL agent (test with all BPMs, 1/3/21, on-the-fly data).

steerers implies an increase of the action space dimensionality and complicates the convergence of the RL agent.

In this sense, an optimal set up might be a *combination* between global response-matrix-based correction "smoothed" with RL for the frequency domain. Our first attempts in this direction (RL agent action added to or introduced into the global response-matrix correction) were not successful - probably because the dynamics of the environment got more complicated to learn with the standard model-free RL loop.

As a consequence, we have started exploring some model-based approaches. The idea is to *encapsulate* the dynamic system induced by the harmonic perturbation patterns within a surrogate model that can be used for combination with the response-matrix-based correction but also for policy training in a model-based RL context. We seek hence a surrogate model in the form

$$F : \mathbb{R}^{n \times m} \times \mathbb{R}^l \rightarrow \mathbb{R}^m$$

$$(\mathbf{x}_{t-n}, \dots, \mathbf{x}_t, \mathbf{c}_{t+1}) \mapsto \mathbf{x}_{t+1}$$

where n is the window size, m the number of BPMs and l the number of steerers.

In our first tests of this approach at the machine, the role of surrogate model was undertaken by a neural network whose architecture is sketched in Fig. 16. This network was directly fed with real data obtained via zmq in real-time interaction with the machine: random steerer strengths were set sporadically and the BPM response to these modifications was tracked. But since the steerer modifications were carried out slowly (see Appendix), the (windowed) BPM data also included the inherent harmonic perturbations.

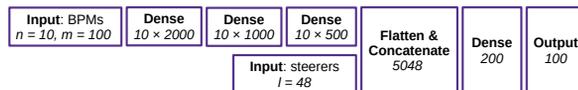


Figure 16: Surrogate model architecture schema.

Table 1 shows that system dynamics were accurately learnt by the model. In this table, model prediction error on unseen test data is compared with several baselines: average of the test data, previous observations in the test data ($\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \mathbf{x}_{t-3}$ and \mathbf{x}_{t-4}) and prediction obtained by the application of the response matrix: $\mathbf{x}_{t+1} \approx \mathbf{x}_t + S(\mathbf{c}_{t+1} - \mathbf{c}_t)$.

Table 1: Surrogate Model and Baseline Prediction Errors

Error	Test Set Avg.	Previous BPM Data					Resp. Matrix:	Model:
		\mathbf{x}_t	\mathbf{x}_{t-1}	\mathbf{x}_{t-2}	\mathbf{x}_{t-3}	\mathbf{x}_{t-4}	$\mathbf{x}_t + S(\mathbf{c}_{t+1} - \mathbf{c}_t)$	$F(\mathbf{x}_{t-9}, \dots, \mathbf{x}_t, \mathbf{c}_{t+1})$
RMSE	0.0154	0.0274	0.0178	0.0236	0.0204	0.0229	0.0098	0.0036
R^2	0	-2.172	-0.3455	-1.3492	-0.769	-1.2223	0.5896	0.9441

The impressive performance of the surrogate model encouraged us to embed it into an algorithm (sketched as Algorithm 1) where its prediction accuracy is used in order to recursively improve the correction given by the response matrix pseudo-inverse. In each iteration, the implicit linear prediction calculated with the response matrix is replaced by the model prediction, which takes into account the inherent orbit perturbations that are not captured by the measured response matrix.

Data: S response matrix, S^{-1} response matrix pseudoinverse, \mathbf{x}_t current BPMs, \mathbf{c}_t current steerer strengths (corrections), F surrogate model

$$\mathbf{c}_{t+1}^0 := \mathbf{c}_t - \alpha S^{-1} \mathbf{x}_t$$

repeat

$$\begin{aligned} \mathbf{x}_{t+1}^k &:= \mathbf{x}_t + S(\mathbf{c}_{t+1}^k - \mathbf{c}_t) \\ \tilde{\mathbf{x}}_{t+1}^k &:= F(\mathbf{x}_{t-n}, \dots, \mathbf{x}_t, \mathbf{c}_{t+1}^k) \\ \mathbf{c}_{t+1}^{k+1} &:= \mathbf{c}_t + \alpha S^{-1} [-(\tilde{\mathbf{x}}_{t+1}^k - \mathbf{x}_{t+1}^k) - \mathbf{x}_t] \end{aligned}$$

until convergence or k too large

Algorithm 1: Combination of surrogate model and response-matrix-based correction.

A test run at the machine of Algorithm 1 (300 mA in decay mode, 150 Hz, $k = 1$) can be visualised in Figs. 17, 18 and 19. Notice that the algorithm is directly constructed over the baseline *FOFB* (*zmq*), and the results are coherent with its theoretical construction: the performance in time domain is quite similar to *FOFB* (*zmq*)'s, whereas it is much more stable in frequency domain.

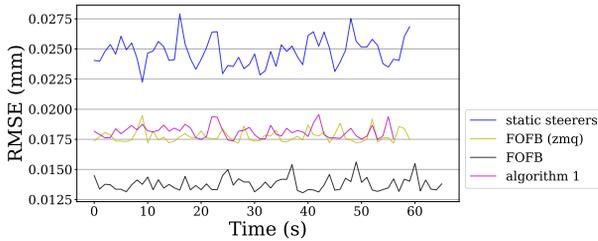


Figure 17: Comparison of the beam deviation (RMSE) after learning (test with surrogate model, 21/6/21, BESSY Archiver data).

Interpretability: A major advantage of the model-based approach we are seeking is the possibility of a more direct interpretability analysis. As an example, we want to briefly analyse the behaviour of the surrogate model F

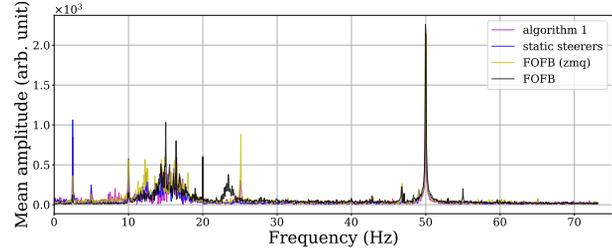


Figure 18: Mean horizontal beam motion spectra comparison (test with surrogate model, 21/6/21, BESSY Archiver data).

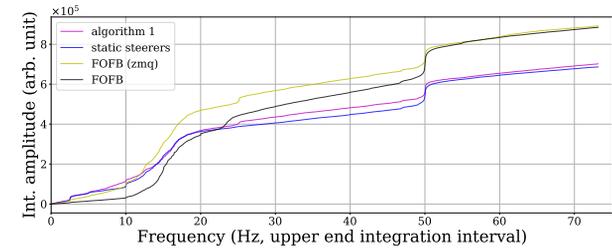


Figure 19: Integrated mean horizontal beam motion spectra comparison (test with surrogate model, 21/6/21, BESSY Archiver data).

trained in the previous section. Recall that F is a neural network with ReLU and linear activation functions; it means that F is differentiable almost everywhere and with the help of Tensorflow's automatic differentiation we can approximate its *jacobian* w.r.t. the steerer strengths:

$$J^c := \begin{pmatrix} \frac{\partial F^1}{\partial c_{t+1}^1} & \dots & \frac{\partial F^m}{\partial c_{t+1}^1} \\ \vdots & \ddots & \vdots \\ \frac{\partial F^1}{\partial c_{t+1}^l} & \dots & \frac{\partial F^m}{\partial c_{t+1}^l} \end{pmatrix}$$

An averaged evaluation of matrices J^c with test data (approximating $\mathbb{E}_{(\mathbf{x}_{t-n}, \dots, \mathbf{x}_t, \mathbf{c}_{t+1})} [J^c |_{(\mathbf{x}_{t-n}, \dots, \mathbf{x}_t, \mathbf{c}_{t+1})}]$) can be visualised in Fig. 20. Notice that, as expected, the matrix structure is almost identical to the measured response matrix (Fig. 21).

Furthermore, a visualisation of the standard deviation of the evaluated matrices J^c (Fig. 22) gives us additional insights about which steerer-BPM interactions are captured by the surrogate model F in a more *non-linear* way: those entries presenting higher variance in the set of evaluated J^c 's.

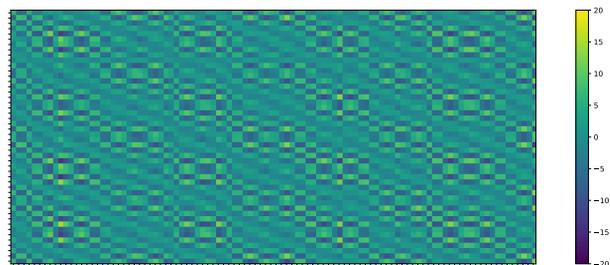


Figure 20: Average of matrices J^c evaluated at 4000 test points.

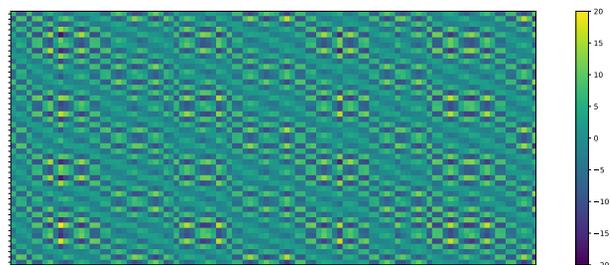


Figure 21: Horizontal response matrix measured at BESSY II (February 2021).

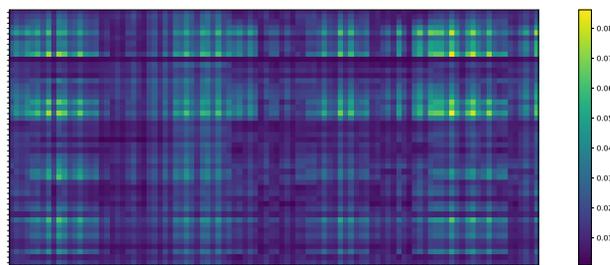


Figure 22: Standard deviation of matrices J^c evaluated at 4000 test points.

ANOMALY DETECTION WITH FEATURE ASSIGNATION

The use of anomaly detection algorithms for accelerator parameters is another field in current development at BESSY II. In particular we have started experimenting with Isolation Forests [10], whose simple and elegant foundations (outliers as points with *short path lengths*) have allowed us to develop additional anomaly scores that can be individually assigned to the input features. Our feature anomaly scores are conceived online and complement the information given by the global anomaly score: They should evaluate *how anomalous* the features of the current observation are. The idea is based on the philosophy of feature importance for fully randomized trees in [11] and similar to LFI in [12]. The results are comparable to SHAP [13] but with much better time performance for large data sets.

Given a point x , its node path in i -th tree T_i will be denoted as $\mathcal{P}_i(x) = \{N_0, N_1, \dots, N_l\}$. Notice that N_0 will be the root node and N_l a leaf. For each node N in T_i let $s(N)$ denote the number of samples assigned to this node after training.

We can hence define a *splitting path* of a given point x in T_i that tracks the reduction of training samples along the path

$$\mathcal{S}_i(x) := \left\{ \left(1 - \frac{s(N_{j+1})}{s(N_j)} \right)^p \right\}_{N_j, N_{j+1} \in \mathcal{P}_i(x), j=1, \dots, l}$$

with $p > 0$ *sparsity* coefficient. If we denote as $F(N_j)$ the feature used for splitting at node N_j with $j = 0, \dots, l - 1$, we can assign to it the j -th quantity within $\mathcal{S}_i(x)$. Therefore, given a point x and a feature F , we can define their *anomaly score* at tree T_i as

$$a_i(F, x) := \sum_j \{s_j \in \mathcal{S}_i(x) \mid F(N_j) = F\}$$

and extend it naturally to all trees in the forest:

$$a(F, x) := \frac{1}{k} \sum_{i=1}^k a_i(F, x)$$

Figure 23 shows an anomaly detection application prototype including these feature anomaly scores and conceived as a first proof-of-concept. It corresponds to an isolation forest trained with ca. 125k data points along three weeks from 22 BESSY's top-up variables. Once the model has been trained, the application starts evaluating data points read in real-time and detects a decay of the global anomaly score, mainly assigned to a single variable (booster current per bunch). This value was indeed anomalous according to BESSY's predefined alert intervals; The algorithm had been able to recognize it without any previous information about these intervals.

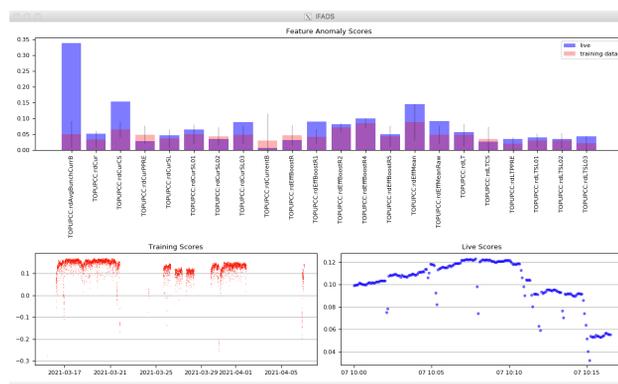


Figure 23: Screenshot of the live application for anomaly detection at BESSY II: An anomaly score decay (lower right plot) is assigned in real time to the booster current per bunch (first bar in the upper plot).

CONCLUSION

The central use case presented in this paper (mitigation of harmonic orbit perturbations) was firstly faced with a model-free deep RL agent working with 6 BPMs and horizontal steerers that, as expected, improved stability in frequency domain but does not overcome traditional global methods in time domain. First steps towards model-based global optimization with all 100 BPMs and 48 horizontal steerers keeping

the time domain correction quality and improving also stability in frequency domain were introduced as well. These first model-based approaches are based on very accurate surrogate models encapsulating the inherent perturbations of the orbit. Finally, we have introduced further original ideas and a proof-of-concept for anomaly detection with feature assignation, whose first prototypes are ready for integration at the machine.

APPENDIX

- **Frameworks:** [14–16]
- **DDPG - Hyperparameters:**
 - **Actor:** feedforward network with three hidden layers (50-20-10 neurons), ReLU as activation function (output with \tanh)
 - **Critic:** feedforward network with four hidden layers (50-(50+action)-20-10) neurons, ReLU as activation function, Adam as optimizer
 - **Learning:** $\gamma = 0.99$, target model update rate = 0.01, batch size = 32
 - **Exploration:** Ornstein-Uhlenbeck process ($\sigma = 0.05$, $\theta = 0.1$, no annealing), memory buffer with 20000 steps, short episodes (50 steps) followed by steersers random restart.
- **Surrogate model correction - Hyperparameters:**
 - **Model:** architecture plotted in Fig. 16, ReLU as activation function, linear output, Adam as optimizer
 - **Learning:** 16000 points for training, 4000 for test (gathered at 150 Hz), validation split 0.05, batch size = 32, 50 epochs.
 - **Steerer randomisation:** steerer intensities sampled from $\mathcal{N}(0, 5)$ (in mA) and actually set with probability $p = 0.005$.

REFERENCES

- [1] P. Schnizer *et al.*, “Online Model Developments for BESSY II and MLS,” in *Proc. IPAC’21*, Campinas, SP, Brazil, 2021, paper WEPAB317, pp. 3413–3416, doi: 10.18429/JACoW-IPAC2021-WEPAB317
- [2] D. Allan, T. Caswell, S. Campbell, and M. Rikitin, “Bluesky’s Ahead: A Multi-Facility Collaboration for an a la Carte Software Project for Data Acquisition and Management,” *Synchrotron Radiation News*, vol. 32, no. 3, pp. 19–22, 2019, doi: 10.1080/08940886.2019.1608121
- [3] W. Smith, S. Kazarski, R. Müller, P. Schnizer, S. Vadilonga, and L. V. Ramirez, “Status of Bluesky Deployment at BESSY II,” presented at ICALEPCS’21 in Shanghai, China, paper FRBR03, this conference.
- [4] L. V. Ramirez, G. Hartmann, T. Mertens, R. Müller, and J. Viehhaus, “Adding Machine Learning to the Analysis and Optimization Toolsets at the Light Source BESSY II,” in *Proc. ICALEPCS’19*, New York, NY, USA, 2020, pp. 754–760, doi: 10.18429/JACoW-ICALEPCS2019-TUCPL01
- [5] R. Mueller, R. Goergen, R. Lange, I. Mueller, and J. Rahn, “Introducing Fast Orbit Feedback at BESSY,” in *Proc. ICALEPCS’09*, Kobe, Japan, 2009, paper THP059, pp. 773–775, <https://jacow.org/icaleps2009/papers/thp059.pdf>
- [6] O. Churlaud, “Localization and correction of orbit perturbations in BESSY II storage ring,” M.S. thesis, TU Berlin, HZB, 2016.
- [7] I. Agapov, G. Geloni, S. Tomin, and I. Zagorodnov, “OCELOT: A software framework for synchrotron light source and FEL studies,” *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 768, pp. 151–156, 2014, (c) Elsevier B.V., doi: 10.1016/j.nima.2014.09.057
- [8] J. Feikes, K. Holldack, P. Kuske, and R. Mueller, “Orbit Stability at BESSY,” in *Proceedings of the 2005 Particle Accelerator Conference (PAC’05)*, 2005, pp. 2366–2368, doi: 10.1109/PAC.2005.1591112
- [9] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016, <http://arxiv.org/abs/1509.02971>
- [10] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17
- [11] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, “Understanding variable importances in forests of randomized trees,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, 2013, pp. 431–439, <http://dl.acm.org/citation.cfm?id=2999611.2999660>
- [12] M. Carletti, M. Terzi, and G. A. Susto, *Interpretable Anomaly Detection with DIFFI: Depth-based Feature Importance for the Isolation Forest*, 2020.
- [13] S. M. Lundberg *et al.*, “Explainable AI for Trees: From Local Explanations to Global Understanding,” *CoRR*, vol. abs/1905.04610, 2019, <http://arxiv.org/abs/1905.04610>
- [14] M. Abadi *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015, <https://www.tensorflow.org/>
- [15] F. Chollet, *Keras*, <https://github.com/fchollet/keras>, 2015.
- [16] M. Plappert, *Keras-rl*, <https://github.com/keras-rl/keras-rl>, 2016.