Adding Machine Learning to the Analysis and Optimization Toolsets at the Light Source BESSY II

Luis Vera Ramírez, Gregor Hartmann, Tom Mertens, Jens Viefhaus, Roland Müller

Helmholtz-Zentrum Berlin (HZB), Germany

17th Biennial International Conference on Accelerator and Large Experimental Physics Control Systems, 08/10/2019, New York
At the synchrotron radiation source BESSY II (Helmholtz-Zentrum Berlin, HZB) both the beamline and the machine groups have started working towards setting up the infrastructure to introduce modern analysis, optimization and automation in order to improve the performance and the experimental setups.

We present an initial study with Machine Learning tools (started at the end of 2018) including analysis and prediction models with real accelerator data as well as first prototypes and use cases for parameter tuning with (Deep) Reinforcement Learning agents.
The Helmholtz Association (the largest scientific organisation in Germany) has initiated the implementation of the **Data Management and Analysis** concept across its centers.

Further HZB collaborations:
- Accelerating Machine Learning for physics (AMaLea) (DESY, KIT, HZDR and HZB) - *more about this at talk TUCPL06*
- Advanced Computational Tools (ACT) (TU Darmstadt, HZB)
- Findable, Accessible, Inter-operable and Re-usable Data Infrastructure for material research (FAIRmat/FAIRDI) (IRIS Berlin, HU Berlin, Max Planck Institute and HZB)

Overview

Prediction models
  Beam lifetime prediction

RLControl - Parameter tuning with Deep RL
  Optimization of booster current
  Optimization of injection efficiency

Building the digital twin

Conclusion

References

Backup
Standard context for measurement prediction → **supervised learning**.

Further **unsupervised** tools for e.g. anomaly detection.

**Reinforcement learning** allows (model-free) parameter tuning.

Figure from [http://www.isaziconsulting.co.za/machinelearning.html](http://www.isaziconsulting.co.za/machinelearning.html)
Overview: Current state at BESSY II

- **Device**
  - Injector
  - Ring accelerator
  - Undulator
  - Beamline
  - Experiment
- **Data**
  - Archive, Diagnostic
  - Simulations
  - Online optimization
  - Diagnostic
  - Raytracing
  - Scans/online data
  - Demands
  - Simulations
  - Beamtimes
- **Methods**
  - SVR-RFF
  - DNN
  - Deep-RL-Control
  - RNN, LSTM
  - Autoencoder
  - CNN, MLP, GBoost
  - Dataloader
  - Tensor product
  - kNN, auto-diff.
  - Reasonable random generator
- **Agent**
  - Operator
  - Beamline scientist
  - (Random-) User
Overview

Prediction models
  Beam lifetime prediction

RLControl - Parameter tuning with Deep RL
  Optimization of booster current
  Optimization of injection efficiency

Building the digital twin

Conclusion

References

Backup
Goal: **time-series-based prediction** where the input given to the model consists only of simultaneous accelerator readbacks, i.e. **without previous target measurements**. This allows us to:

- Avoid an excessive reliance on the previous target measurements.
- Force the model to identify further correlations and patterns in the readbacks.
- Reuse the information and experience gained in a RL context.
**Ensemble methods:** Random Forests, Extremely Randomized Trees... ([Bre01], [GEW06]). For regression, **MSE** as loss $\rightarrow$ variance as impurity measure. **Self-explaining:** allows individual analysis of each variable’s behavior.

**Support Vector Regression** [Smo98] with **Random Fourier Features** ([RR08]). SVR extends traditional SVM (for classification) via Vapnik’s $\epsilon$-insensitive loss function ([Vap95]).

**Neural Networks** (e.g. see [Roj96]). Feed-forward NNs for regression (i.e. MSE as loss function).

Figs. from https://dsc-spidal.github.io/harp/docs/examples/rt/, [SS03], [Roj96].
Beam lifetime prediction: Experiments

- **Target:** instant approximation to **beam lifetime** defined via current decay rate \((k = 20): \frac{1}{\tau} = -\frac{\dot{I}_t}{I_t} \approx -\frac{1}{I_t} \frac{\sum_{i=0}^{k} (I_{t-i}-I_{t_0})(t-i-t_0)}{\sum_{i=0}^{k}(t-i-t_0)^2}\)

- **Input variables** (185 after preprocessing):
  - Gap and shift of **insertion devices** undulators (21 vars)
  - **Power supply currents** and offsets into **quadrupoles** (58 + 38 vars) and **sextupoles** (7 vars)
  - **Collisions** with rest gas particles, **vacuum pressures** (12 vars)
  - **Local beam loss fractions** (49 vars)

- Feature importance analysis with **RandomTrees:** even distribution, but **quadrupoles** (offsets) and **insertion devices** stand out

<table>
<thead>
<tr>
<th>Test set</th>
<th>Algorithm</th>
<th>RMSE</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Random 20%</td>
<td>ExtraTrees</td>
<td>0.201</td>
<td>0.099</td>
<td>0.091</td>
<td>0.068</td>
<td>0.757</td>
<td>0.794</td>
<td>0.885</td>
</tr>
<tr>
<td></td>
<td>SVR-RFF</td>
<td></td>
<td></td>
<td></td>
<td>0.077</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td></td>
<td></td>
<td></td>
<td>0.069</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last 20%</td>
<td>ExtraTrees</td>
<td>0.231</td>
<td>0.096</td>
<td>0.079</td>
<td>0.195</td>
<td>0.829</td>
<td>0.884</td>
<td>0.292</td>
</tr>
<tr>
<td></td>
<td>SVR-RFF</td>
<td></td>
<td></td>
<td></td>
<td>0.121</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td></td>
<td></td>
<td></td>
<td>0.125</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Beam lifetime prediction: SVR-RFF and DNN with chronological split
Table of Contents

Overview

Prediction models
  Beam lifetime prediction

RLControl - Parameter tuning with Deep RL
  Optimization of booster current
  Optimization of injection efficiency

Building the digital twin

Conclusion

References

Backup
Deep Deterministic Policy Gradient [LHP^{+}16]: Actor-critic Reinforcement Learning algorithm for continuous environments.

Off-policy data and the Bellman equation used to learn the $Q$-function.

$Q$-function used to learn the policy.

Approximated with NNs.

Figure: From [SB18]
After long interruptions of the machine operation, the booster current tends to be low - as for today, **manual parameter tuning** is required.

We seek an automatized, RL-based solution.

- **State variables:**
  - High (radio) frequency - master clock.
  - Voltage in LINAC.
  - Klystron current diagnostic measurements.

- **Action variable:** **time phase in LINAC.**

Observations show that this parameter does not affect the injection efficiency.

- **Reward:** (normalized) **booster current per bunch.**
RLControl - Parameter tuning with Deep RL: Case description

Agent

Environment

State

Reward

Action
RLControl - Parameter tuning with Deep RL: Case description

Agent

RLControl

Environment

BESSY II

State
klystron diagnostics
LINAC voltage
master clock

Reward
booster current
per bunch

Action
LINAC time phase
Short test (20/05/19). **Reward (booster current per bunch)** in **blue**, **action (LINAC time phase)** in **red** - remaining lines correspond to state variables. Pretraining with 30 days of historical data. **Exploration** with Parameter Space Noise ([PHD\textsuperscript{+}17]) appears **shaded** - period length of ca. 2 min chosen as experiment for automatic scheduling...
Optimization of booster current: Automatic scheduling

State:
- klystron diagnostics
- LINAC voltage
- master clock

Agent:
RLControl

Action:
LINAC time phase

Reward:
- booster current per bunch

Environment:
BESSY II
Optimization of booster current: Automatic scheduling

Agent \textit{RLControl}

Environment \textit{BESSY II}

State
\begin{itemize}
  \item klystron diagnostics
  \item LINAC voltage
  \item master clock
\end{itemize}

Reward
\begin{itemize}
  \item booster current per bunch
\end{itemize}

Action
\begin{itemize}
  \item LINAC time phase
\end{itemize}

Injection soon?
\begin{itemize}
  \item Yes: optimize
  \item No: explore
\end{itemize}

automatic scheduling
Exploration is scheduled in the meantime between injections to avoid disturbing user activity - optimization activated shortly before each injection.
Long test during user time with **automatic exploration schedule** (09/07/19). Reward (booster current per bunch) in blue, action (LINAC time phase) in red - remaining lines correspond to state variables. Pretraining with 30 days of historical data. Exploration with automatic schedule shaded - **first hour**. The agent optimizes (and learns) successfully during the next **8.5 hours of user operation**.
Injection efficiency is known to be affected by temperature - nowadays it also needs **manual tuning**. RL-based optimization is a work in progress.

- **State variables:**
  - Number of **bunches** generated by the LINAC (1, 3 or 5).
  - Injection angle **mismatch**, measured by the beam position in the transfer line.
  - **Current** measured during the booster acceleration phase.
  - Measured **loss rate after extraction** from the booster.

- **Action:** **deflection angle into the storage ring**, generated by the 2nd septum.
- **Reward:** **last injection efficiency** - fraction of current increase generated in the storage ring by the charge accelerated in the booster.
Short test (23/09/19). Pretraining with 23 days of historical data. **Reward (injection efficiency)** is plotted in *blue*, actions (septum deflection angle) is plotted in *red*. Exploration periods appear shaded. *Ad-hoc* modifications in the number of pulses (in black) and booster current (in purple) are carried out during the test - the agent manages to find and improve the optimal action regions.
Overview

Prediction models
  Beam lifetime prediction

RLControl - Parameter tuning with Deep RL
  Optimization of booster current
  Optimization of injection efficiency

Building the digital twin

Conclusion

References

Backup
The accelerator/source side: OCELOT surrogate models for RL ([AGTZ14]). Challenge: export a RL-agent trained on the virtual lattice to the real accelerator.

The beamline/x-ray side: general initiative to take advantage of ML methods also for the retrieval of scientific data from the measurements. Close coordination of activities developing toolsets for the accelerator and the beamlines (talk MOCPL02 - [M+19]).

E.g., work by Dr. Gregor Hartmann et al.: beamline raytracing inversion (photon footprint screens → beamline parameters) with neural networks.
Overview

Prediction models
  Beam lifetime prediction

RLControl - Parameter tuning with Deep RL
  Optimization of booster current
  Optimization of injection efficiency

Building the digital twin

Conclusion

References

Backup
Beam lifetime at BESSY II can be successfully predicted in a time-series fashion through supervised learning models trained only with 185 accelerator variables readbacks - i.e. excluding previous beam lifetime measurements.

The accelerator optimization framework RLControl is able to solve a first use case (booster current) at BESSY II through Deep Reinforcement Learning. It has also been tested even during user operation with the help of injection-based automatic scheduling.

Further effort has been put into the application of RLControl for more advanced use cases such as injection efficiency optimization, leading also to successful preliminary tests.
Conclusion: Next steps

- Measurement prediction framework:
  - Build models for further target variables such as purity
  - Classification approach
  - Surrogate models

- RLControl:
  - Further tests, investigation and use cases (injection efficiency with more state and action variables, orbit correction with OCELOT ([AGTZ14]) pretraining...)
  - Bluesky integration
  - User interfaces


