Modelization of an injector with Machine Learning

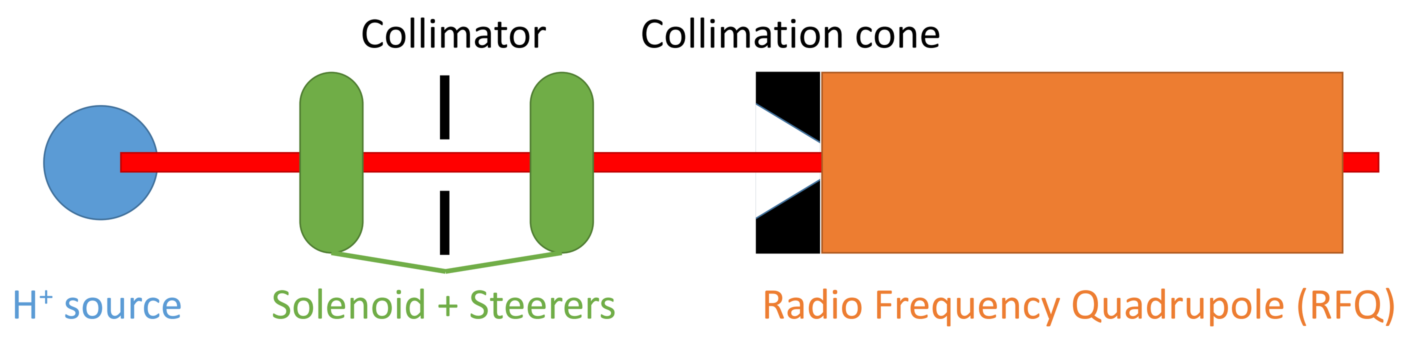
M. Debongnie*[1,2], T. Junquera[1], F. Bouly[2], D. Uriot[3], N. Chauvin[3] & A. Gatera[4]

[1] Accelerators and Cryogenic Systems, Orsay, France
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, LPSC-IN2P3, 38000 Grenoble, France
[3] CEA Saclay, Saclay, France
[4] SCK*CEN, Mol, Belgium

Abstract

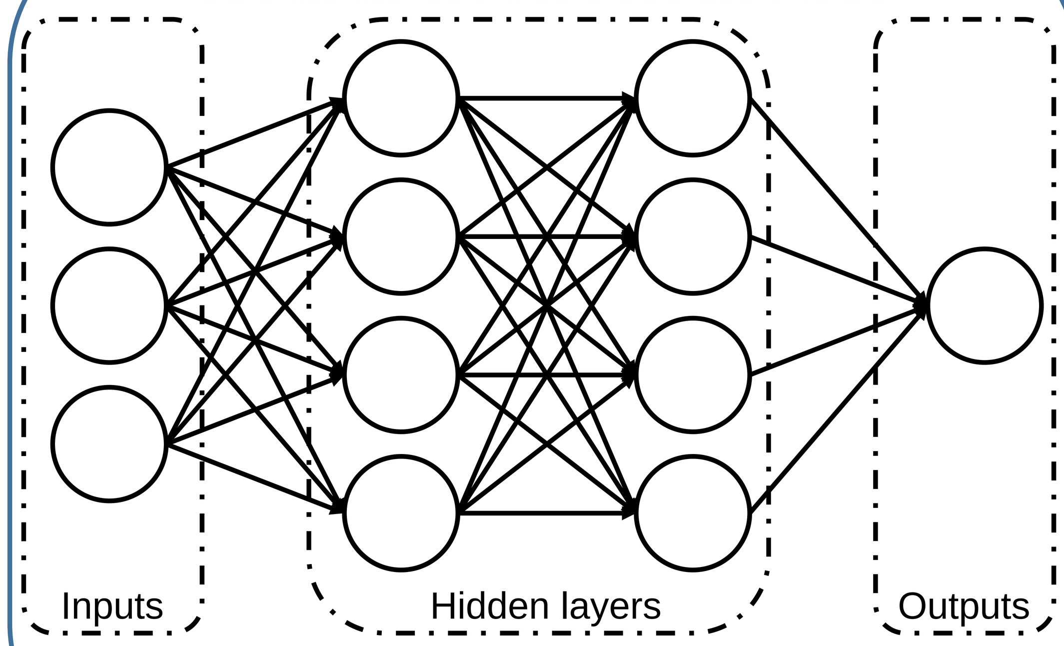
Modern particle accelerator projects, such as MYRRA [1], have very high stability and/or reliability requirements. To meet those, it is necessary to optimize or develop new methods for the control systems. One of the difficulties lies in the relatively long computation time of current beam dynamics simulations. In this context, the very low computation time of neural network is of great attraction. In this work, we show the method used to train neural networks to model the IPHI [2] and MYRRHA injectors. The resulting models are able to predict key quantities of the beam with good accuracy for a wide range of injector configurations and with a computation time down to $\sim 10\ \mu s$ per injector configuration.

The MYRRHA & IPHI injectors architecture



The MYRRHA and IPHI injectors have a similar design, first a beam of proton is extracted from an ion source (~8 mA, 30 keV and ~100 mA, 95 keV for MYRRHA and IPHI respectively). Then the beam is transported and shaped using two solenoids with steerers inside and a collimator to ensure a good injection through the collimation cone. In the case of IPHI, the beam is then accelerated up to 3 MeV with a Radio Frequency Quadrupole (RFQ). In the case of MYRRHA, the RFQ is currently in commissioning therefore the beam is dumped after the collimation cone.

Neural network model



- Activation function: Relu with He initialization
- TensorFlow [3] API v1.13 for python

Datasets generation

Both datasets were obtained experimentally following similar approaches: multiple scans were performed by varying the current applied first in the solenoids then in the steerers for different position of the collimator.
In the case of MYRRHA, the beam current was measured for ~ 20000 different configurations using a Faraday cup placed after the collimation cone.
In the case of IPHI, the beam current was measured for ~ 10000 different configurations using a DCCT before the collimation cone and an ACCT after the RFQ. The transmission of the RFQ is estimated as the current measured in the ACCT divided by the current measured in the DCCT.
In both cases, the datasets were randomly divided into three subset: training (~ 60 %), validation (~ 20 %) and test (~ 20 %) datasets.

MYRRHA		IPHI	
Inputs	Outputs	Inputs	Outputs
Current in solenoids (x2)	Beam current	Current in solenoids (x2)	Beam current
Current in steerers (x4)		Collimator position (x1)	RFQ transmission
Collimator position (x4)			
Pressure (x3)			

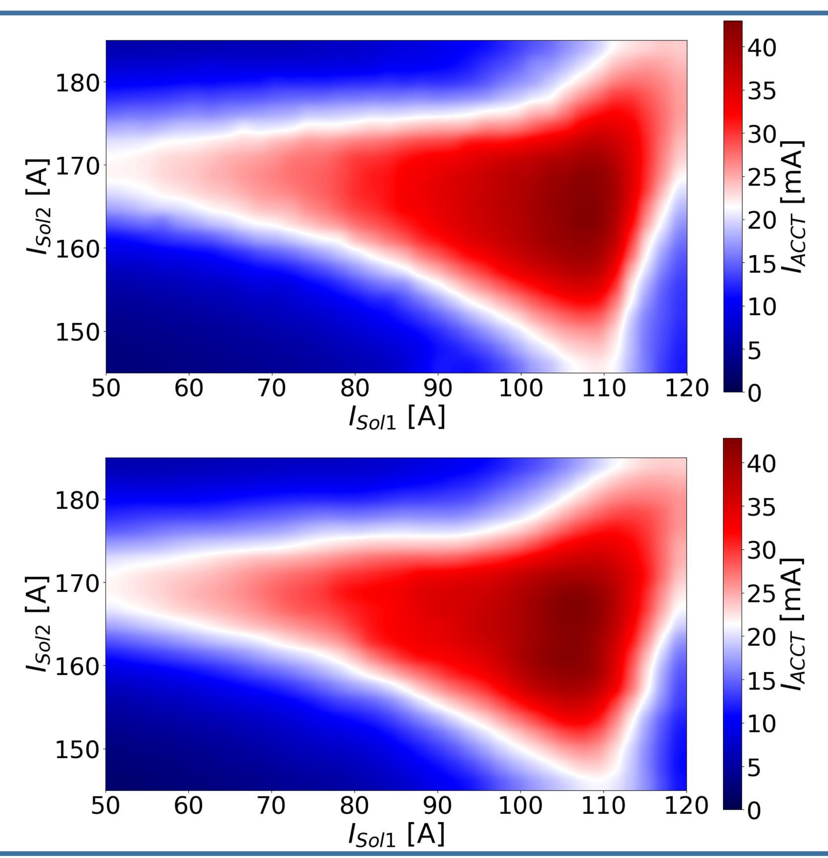
Training and evaluation

The models are trained following a standard supervised learning approach with a stochastic gradient descent optimizer. The cost function minimized by the optimizer is the mean squared error between the model predictions and the experimental data. The learning rate is initially set to 0.1 then progressively decreased down to 0.001 each time the cost function does not change for longer than 5 consecutive epochs. One epoch is defined as 1000 training steps with a batch size of 128. At the end of every epoch, the model is tested on the validation dataset to monitor the training progress. Every 10 epoch, the model is tested on the test dataset to monitor the training progress and interrupt the training in case of overfitting.

Results

The performance of trained models is estimated using the Root Mean Square Error (RMSE) between the models predictions and the test datasets. A summary of the RMSEs of the trained models on their respective datasets is given in the table on the right.
The figures next to the table give a visual comparison between an experimental scan on the beam current (top) and the corresponding scan as predicted by the trained model for IPHI (bottom).
The time required to compute a prediction using trained models for one set of inputs (= one configuration of the injector) is very short (a few μs).

	MYRRHA	IPHI	
Outputs	Beam current [mA]	Beam current [mA]	RFQ transmission [%]
RMSE on training dataset	0.46	0.66	1.25
RMSE on validation dataset	0.44	0.79	1.62
RMSE on test dataset	0.45	0.81	1.65
RMSE on whole dataset	0.46	0.72	1.42



Conclusions & perspectives

Throughout this work, multiple neural network models were trained successfully on the MYRRHA and IPHI injectors following a Supervised Learning approach. This shows that neural network models are suited to model key quantities governed by complex physics in a particle accelerator.
Overall, the trained models show good performances. The RMSE on the test datasets indicates that the neural network models are able to interpolate for injector configurations that it has not been trained for. Hence, it is possible to obtain a model over a wide range of configurations with only few actual measurements on the injectors.
With these models, it becomes possible to substantially speed up control strategies that involve optimization algorithms. In theory, any optimization algorithm that can be applied to a beam dynamics simulation can also be applied on trained models. One of the interesting application would be to use trained models for coarse optimization then use the results to bootstrap a finer optimization on slower beam dynamics simulations.
Another possibility is to use trained models to train neural networks controller using a Reinforced Learning approach. In this case, the neural network controller is trained to act in a similar way to a human operator by interacting with the trained model. This approach will be tested soon with the aim to develop a solution for the fast reconfiguration of the MYRRHA superconducting linac in the context of a fault tolerant strategy [4].

Acknowledgments

We would like to thank D. Uriot (CEA), N. Chauvin (CEA) and D. Vandeplassche (SCK*CEN) for providing experimental time on the IPHI and MYRRHA injector, respectively.

References

[1] https://myrrha.be/
[2] http://irfu.cea.fr/
[3] https://www.tensorflow.org/
[4] F. Bouly *et al.*, Proc. 27th Linear Accelerator Conf. (2014), MOPP103, pp 297-299

* Contact: mathieu.debongnie@lpsc.in2p3.fr