ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS79

MODELING AND OPTIMIZATION OF THE FACET-II INJECTOR WITH MACHINE LEARNING ALGORITHMS *

S. Chauhan†, A. Edelen, C. Emma, S. Gessner
SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

Abstract

Linear particle accelerators are elaborate machines that demand a thorough comprehension of their beam physics interactions to enhance performance. Traditionally, physics simulations model the physics interactions inside a machine but they are computationally intensive. A novel solution to the long runtimes of physics simulations is replacing the intensive computations with a machine learning model that predicts the results instead of simulating them. Simple neural networks take milliseconds to compute the results. The ability to make physics predictions in almost real time opens a world of online models that can predict diagnostics which typically are destructive to the beam when measured.

This research entailed the incorporation of an innovative simulation infrastructure for the SLAC FACET-II group, aimed at optimizing existing physics simulations through advanced algorithms. The new infrastructure saves the simulation data at each step in optimization and then improves the input parameters to achieve a more desired result. The data generated by the simulation was then used to create a machine learning model to predict the parameters generated in the simulation. The machine learning model was a simple feedforward neural network and showed success in accurately predicting parameters such as beam emittance and bunch length from varied inputs.

INTRODUCTION

Challenge and Motivation

Operating and optimizing a linear particle accelerator involves the calibration and control of hundreds of variables to ensure that the accelerator functions at its peak performance. Traditionally, these optimizations are carried out using computationally intensive physics simulations that model the behavior of particles as they move through the accelerator. While these simulations are invaluable for understanding the nuanced interactions within the system, they can be prohibitively expensive in terms of computational time and resources. Additionally, certain diagnostic procedures can be invasive or destructive to the particle beam, limiting real-time adjustments and optimization.

Given these challenges, there is a growing need for a more efficient and less invasive method to simulate and optimize linear accelerators. This necessity not only aims to reduce computational overhead but also opens up the possibility of real-time diagnostic and predictive modeling—something that was previously unattainable.

* Work supported by the U.S. Department of Energy, United States under Contract DE-AC02-76SF00515
† sanjeev.chauhan@duke.edu

Background: FACET-II Photo-Injector

FACET-II (Facility for Advanced Accelerator Experimental Tests II) is an experimental setup focused on high-energy electron beams. It aims to produce a high electron peak current with a small beam size and low emittance, essential for plasma wakefield acceleration (PWFA) experiments. The injector, a key component, is based on the Orion RF photocathode gun, similar to the LCLS (Linac Coherent Light Source) injector with minor changes. Figure 1 provides a schematic representation of the FACET-II setup, illustrating the injector, radio frequency cavities, and diagnostic area.

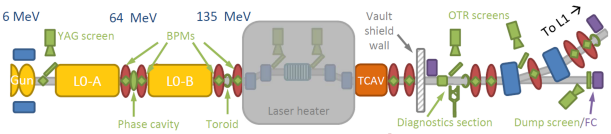

Figure 1: Diagram of the FACET-II photo injector.

Limitations of Traditional Physics Simulations

Traditional physics simulations are essential for understanding and predicting physical systems' behavior. Using mathematical principles, they provide a virtual lab for testing and optimizing designs.

The General Particle Tracer (GPT) [1] and Lucretia [2] simulation frameworks are advanced tools for modeling charged particle dynamics and high-performance electron beam transport systems. GPT excels in simulating accelerator components, including magnetic and electric fields, and beamline elements, especially space charge effects at lower energies, such as at the start of the FACET-II injector. Lucretia, initially developed for Linear Collider Low Emittance Transport studies, efficiently simulates bunch compressors, linear accelerators, final focus systems, and linac-driven free electron lasers, ignoring low-energy space charge effects. This work uses GPT until particles reach sufficient energy for accurate Lucretia simulation.

Despite their effectiveness, GPT and Lucretia have limitations. They use macro particles to reduce computational load, leading to approximations that may not fully capture real-world interactions.

These simulations are also computationally intensive, averaging 107.77 seconds per run, which prevents real-time synchronization with live experiments. As a result, they are primarily used for pre-experimental design and analysis rather than real-time diagnostics or adaptive control during beam-time.

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS79

This delay inhibits experimental iterations and on-the-fly adjustments, posing a significant challenge that requires further innovation to overcome.

SIMULATION INFRASTRUCTURE

To simulate the FACET-II photo injector, a computational pipeline using both GPT (General Particle Tracer) and Lucretia was developed. A MATLAB script, employing a MATLAB class file, orchestrated the simulation parameters for GPT.

In the first phase, GPT simulated particle motion up to an energy of 66 MeV (section L0-A in Figure 1), accounting for space charge effects relevant at lower energies, such as in the FACET-II injector gun. The particle tracking data from GPT was then saved and passed to Lucretia, which continued the simulation up to 135 MeV. Lucretia is more efficient at higher energies as it ignores the low-energy space charge effects considered by GPT.

The MATLAB script was controlled by a Python wrapper using the XOPT framework [3] to implement a CNSGA optimization algorithm. This algorithm adjusted input parameters to find optimal conditions for the FACET-II photo injector. The goal of the optimization was to minimize bunch length and mean emittance.

Additionally, the Python script converted the beam file data from each Lucretia simulation run into the openPMD format using an openPMD Python library [4], improving interoperability and ease of plotting beam statistics across different platforms.

This process is illustrated in Fig. 2. Python with XOPT runs a MATLAB-based physics simulation at each optimization step, and the beam state is saved in openPMD format.

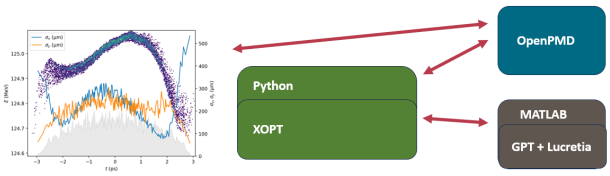

Figure 2: Diagram of developed simulation infrastructure.

Input and Output Parameters

Table 1 summarizes the key input and output parameters manipulated and tracked during the simulations with XOPT.

Simulation Results

The computational pipeline integrating XOPT with GPT and Lucretia was designed for flexibility, allowing various optimization strategies. One successful approach was the Controlled Non-dominated Sorting Genetic Algorithm (CNSGA), which effectively generated a Pareto front (Fig. 3).

In this case, the Pareto front helps identify optimal input parameters for minimizing both bunch length and mean beam emittance. Its emergence confirms the high-dimensional optimization landscape and CNSGA's capability to navigate it effectively.

Moreover, the flexibility of the XOPT framework allows the use of other optimization techniques as well. For instance, Bayesian optimization algorithms can also be employed within this computational infrastructure.

Thus, this simulation infrastructure proves to be not just robust but also versatile, capable of employing various optimization algorithms.

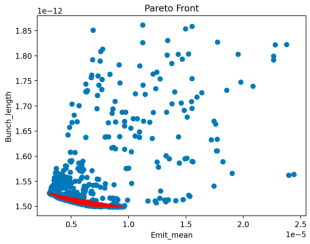

Figure 3: Recorded data showing a Pareto front.

MACHINE LEARNING INFRASTRUCTURE

Data Generation and Preprocessing

The machine learning model's training data was generated through physics simulations controlled by an optimization algorithm, enhancing data generation efficiency and relevance to ideal linear accelerator runs. This approach improved the model's ability to learn meaningful patterns.

Key features and targets were selected to represent critical accelerator variables, including solenoid variations, gun phases, bunch charges, laser pulse lengths, and quadrupole controls. These features were mapped to targets such as

Table 1: Simulation Input and Output Parameters

Parameter	Description
Input Parameters	
Quadrupole Controls	QUAD:IN10:361, 371, 425, 441, 511, 525:BCTRL
bunch_charge	Initial bunch charge
gun_phase	Photo-injector gun phase
laser_pulse_length	Laser pulse length
sol_var	Solenoid variable
Output Parameters	
Twiss Parameters	alpha_x, alpha_y, beta_x, beta_y, gamma_x, gamma_y
bunch_charge_final	Final bunch charge
bunch_length	Bunch length
emittance	emit_mean, emit_x, emit_y, norm_emit_x, norm_emit_y
energy	Final energy
Dispersion	eta_x, eta_y, etap_x, etap_y
Beam Size	sigx, sigy
Other Metrics	num_particles, xopt_error, xopt_runtime

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS79

normalized emittances, crucial for understanding beam dynamics.

Before model training, the dataset was standardized using Min-Max Scaling to ensure all variables operated on a comparable scale. The data was split into training, validation, and test sets, with an 80-20 split for training and testing, and a 75-25 split within the test set for validation.

To enhance training data size and variability, random noise was introduced to create a "noisy" duplicate set. This augmented dataset improved model robustness, enabling better generalization to unseen data.

Neural Network Architecture and Training

The machine learning model is a fully connected neural network, adapted from a US particle accelerators course [5]. It includes dropout layers with a 0.05 rate to reduce overfitting and improve generalization.

The network has multiple hidden layers with hyperbolic tangent (tanh) activation functions. The loss function is Mean Absolute Error (MAE), optimized using the Adam optimizer, with custom weighting to prioritize mean emittance error. This was necessary as the model performed better at predicting parameters like bunch length compared to mean emittance. A learning rate scheduler and early stopping based on validation loss were employed to enhance training efficiency and prevent overfitting.

Machine Learning Results

The model effectively identified patterns in the data, enabling accurate predictive modeling. There was no evidence of overfitting, suggesting it might be underpowered. Rapid learning was observed initially, but further training did not significantly improve performance, possibly due to data scarcity.

For evaluation, the model was tested on a subset not used during training. Its performance on mean beam emittance and bunch length predictions is shown in Figures 4 and 5, respectively.

A more focused evaluation was conducted using a new simulation dataset, varying only the laser pulse length. The model's predictive capabilities for mean emittance with this new data were slightly less accurate than during training but remained usable.

Additionally, feature importance was investigated using a gradient descent method, with results shown in Figure 6.

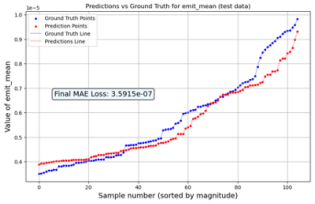

Figure 4: Ground truth vs predicted values of mean normalized beam emittance for validation dataset sorted by magnitude.

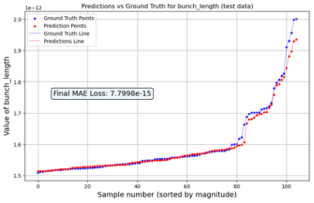

Figure 5: Ground truth vs predicted values of beam bunch length for validation dataset sorted by magnitude.

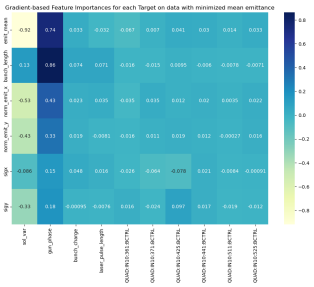

Figure 6: Gradient descent based feature correlation.

CONCLUSION

This work presents a multifaceted approach to simulating high-performance single-pass electron beam transport systems, combining conventional physics-based simulations like GPT and Lucretia with machine learning algorithms. Traditional tools provide essential insights but are limited by long computation times, hindering real-time applications.

To address this, a machine learning model was developed and trained on simulation-generated data. While effective in predicting key parameters like mean beam emittance and bunch length, the model showed signs of being underpowered, likely due to dataset constraints. Multiple scenarios were evaluated to confirm the model's generalizability and effectiveness.

Future improvements include deploying the model for virtual diagnostics. This work demonstrates that machine learning can complement traditional methods in particle accelerator simulations and proves model weights can reveal which features have a greater impact on output parameters, thus improving virtual diagnostics.

ACKNOWLEDGMENTS

The author would like to thank Auralee Edelen for the wonderful project guidance and advice on machine learning. Thank you to Claudio Emma for your amazing help setting up GPT and Lucretia. Furthermore, thank you to Spencer Gessner for putting together this project and overseeing the project. This work was supported by the U.S. Department of Energy, under the SULI program.

Thank you to Hillary Freemen for doing an amazing job at managing the SULI program.

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS79

REFERENCES

[1] M. J. de Loos and S. B. van der Geer, "General Particle Tracer: A 3D Code for Accelerator and Beam Line Design," in *Proc. EPAC'98*, Stockholm, Sweden, Jun. 1998, paper THP18F, pp. 1245–1247.

[2] P. Tenenbaum, "Lucretia: A Matlab-based Physics Toolbox for the Simulation of High-Performance Single-Pass Electron Beam Transport Systems," Available: https://www.slac.stanford.edu/accel/ilc/codes/Lucretia/.

[3] C. Mayes and R. Roussel, "Xopt: Flexible Optimization of Arbitrary Problems in Python," Available: https://github.com/ChristopherMayes/Xopt.

[4] C. Mayes, "openPMD: Tools for Analyzing and Viewing Particle Data in the openPMD Standard, Extension Beamphysics," Available: https://christophermayes.github.io/openPMD-beamphysics/.

[5] R. Lehe, A. Edelen, C. Mayes, R. Roussel, and A. Hanuka, "U.S Particle Accelerator School: Optimization and Machine Learning for Accelerators," Available: https://uspas.fnal.gov/programs/2022/onlinetamu/courses/machine-learning.shtml.