ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU1BCO04

LASER FOCAL POSITION CORRECTION USING FPGA-BASED ML MODELS

J. Einstein-Curtis*, S. J. Coleman, N. M. Cook, J. P. Edelen, RadiaSoft LLC, Boulder, CO, USA
S. Barber, C. Berger, J. van Tilborg, Lawerence Berkeley National Lab, Berkeley, CA, USA

Abstract

High repetition-rate, ultrafast laser systems play a critical role in a host of modern scientific and industrial applications. We present a diagnostic and correction scheme for controlling and determining laser focal position by utilizing fast wavefront sensor measurements from multiple positions to train a focal position predictor. This predictor and additional control algorithms have been integrated into a unified control interface and FPGA-based controller on beamlines at the Bella facility at LBNL. An optics section is adjusted online to provide the desired correction to the focal position on millisecond timescales by determining corrections for an actuator in a telescope section along the beamline. Our initial proof-of-principle demonstrations leveraged precompiled data and pre-trained networks operating ex-situ from the laser system. A framework for generating a low-level hardware description of ML-based correction algorithms on FPGA hardware was coupled directly to the beamline using the AMD Xilinx Vitis AI toolchain in conjunction with deployment scripts. Lastly, we consider the use of remote computing resources, such as the Sirepo scientific framework*, to actively update these correction schemes and deploy models to a production environment.

INTRODUCTION

Laser plasma accelerators (LPAs) rely upon accurate control of ultrafast lasers, typically Ti:Sapph and Nd:Yag amplifier systems [1]. The BELLA Center at Lawerence Berkeley National Laboratory (LBNL) features several ultra-short pulse, high-energy beamlines to develop LPAs. These accelerators require highly repeatable, stable interaction points to generate high-quality electron beams, which necessitates a collection of active and passive controls to mitigate environmental, mechanical, and component variations.

Recent work has primarily focused on enhancing transverse beam stability [2]. This paper describes a a strategy to address focal position stability, leveraging a machine learning (ML) enhanced wavefront diagnostic in tandem with a Field Programmable Gate Array (FPGA) controller to correct focal position at a kHz-scale rate. By building a model of wavefront at the interaction point, it is possible to use a non-perturbative measurement to calculate the focal position.

* joshec@radiasoft.net

Table 1: Optimal lens movement vs focal shift and beam size change. Focus shift is per mm lens translation. Beam size change is change per mm lens translation.

	Shift	Size Change
Transmissive Amp3-in	2 mm	x1.348
Transmissive Amp4-in	2 mm	x1.046
Reflective Amp4-out	1 mm	x1.002

FACILITY AND EQUIPMENT

The initial model was created for the BELLA HTU laser system, shown in Fig. 1. This beamline operates with 1 kHz seed pulses and a 1 Hz full-power pulse. A HASO FIRST Shack-Hartmann wavefront sensor was used as the ground-truth imaging device of the interaction and post-interaction region, with the pre-interaction region sensor a Thorlabs WFS20-7AR. A Xilinx Zynq ZCU104 FPGA evaluation kit was used for testing to provide flexibility during the prototype phase, including a variety of customizable I/O, well-supported manufacturer-provided software, and a variety of processing options in support of ML operations.

FOCAL POSITION INVESTIGATIONS

To determine the optimal lenses to move for a focal shift, we looked at the magnitude of the shift at final focus and the (unwanted) increase in beam size throughout the optical chain. Table 1 summarizes these parameters for three different lenses in the telescope.

From these simulations we determined that the reflective Amp4-out is not ideal as a motorized correction optic for focal location because it is more weakly responsive, shifting the focus by only 1 mm per mm translation. Moreover, the off-axis reflective geometry introduces beam centroid kicks, even in response to relatively mild beam size variations. Ultimately, we determined the Amp4-in telescope is the best choice.

To verify our model, we measured the focal location vs lens separation at high power. Our measurement used a comparable method of capturing leakage from the final steering mirror thus measuring raw focal location without the need for further calibration or renormalization. The inset of Fig. 2 provides details of the measured focal position and radius of curvature taken from the wavefront sensor.

When comparing measurements to the simulation, we note that the focus shift per mm stage motion depends on the nominal Amp4-in lens separation. For a perfectly collimated beam entering the $-f_1/+f_2$ telescope, and for a perfectly collimated beam leaving the telescope (lens separation is

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU1BC004

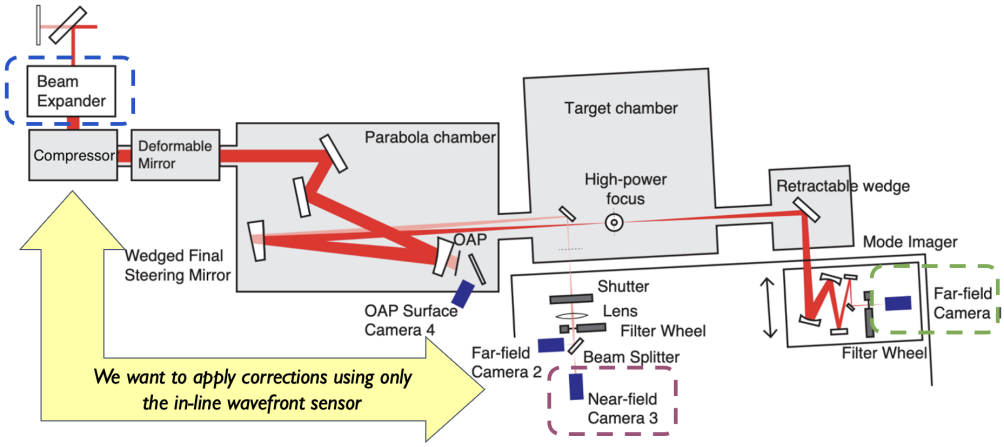

Figure 1: Diagram of HTU laser system at LBNL, highlighting the proposed correction scheme. Machine learning techniques are used to correlate a fast, non-perturbative sensor (2) with a high-quality, but perturbative wavefront sensor (1) which cannot be used for online correction. The resulting online diagnostic is used to deduce variations away from the desired focal position, which is then corrected for prior to the next shot by changes made to a transmissive lens beam expander (3).

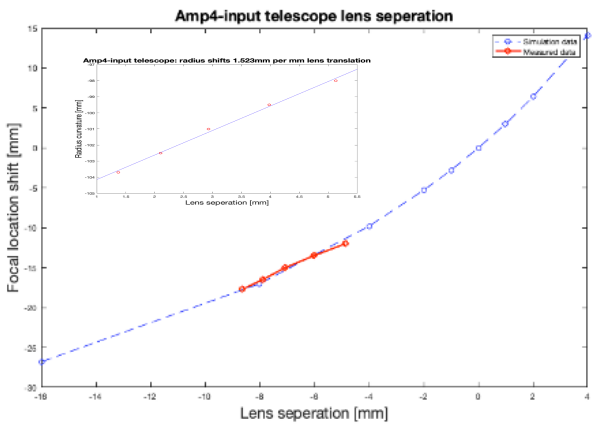

Figure 2: Focal location vs lens separation.

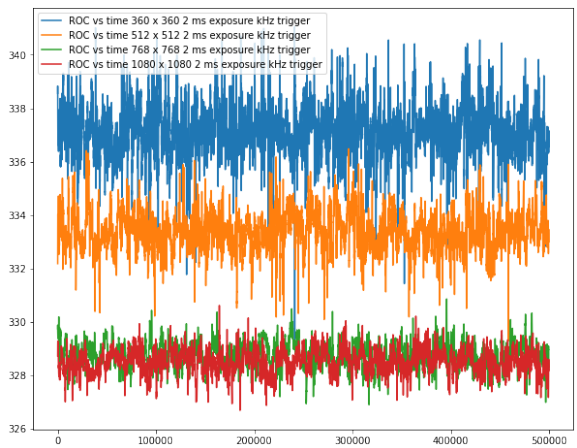

Figure 3: RoC calculated by Thorlabs Driver vs image resolution.

$f_2 - f_1$), the slope change is 2 mm focus shift per 1 mm change in lens separation.

However, for the situation where the lens separation is NOT equal to $f_2 - f_1$, for example because the input beam has a divergence or the output beam is not perfectly collimated, this slope will have a different value.

By overlapping the experimental data (red circles) with the simulation (blue circles), we find a good agreement for one very specific initial lens separation offset (circa −6 mm). The slope at this separation is 1.52 mm focal position shift for every 1 mm lens motion. Figure 2 confirms this result.

This validates the use of a telescoping optic configuration for making controlled adjustments to the laser focal position. This design was validated through simulation and experimental measurement.

Input data is also highly affected by the sampling method and instrument systematics. Figures 3 and 4 show the systematic effects of instrument settings on the calculated figure of merit. As camera image resolution is decreased, the calculated Radius of Curvature increases by several mm, while there is also an increase in overall 'noise' in the signal. These measurements were not taken with an equivalent ground truth image, and thus our assumption is that switching into a higher speed mode of operation will come with corresponding errors that need to be systematically identified or incorporated in a smart feedback mechanism.

MODEL DEVELOPMENT

Several datasets were collected to examine changes in focal position on a shot-by-shot basis. The intra- and inter-shot variation over time, as shown in Fig. 5, show millimeters of variation in the calculated radius of curvature, highlighting the need for correction schemes.

Examining the extrapolated focal positions from each dataset reveals significant discrepancies between the two

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU1BCO04

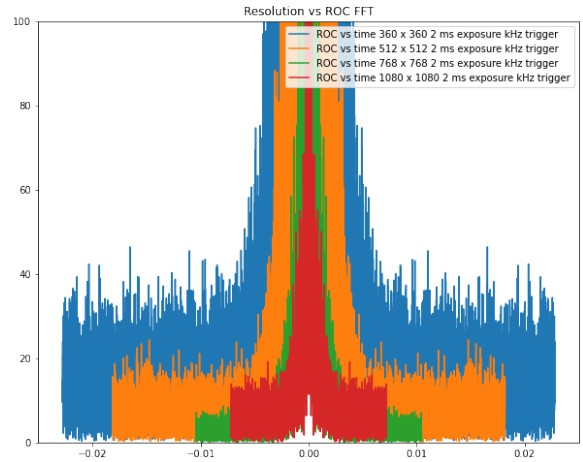

Figure 4: FFT of RoC calculated by Thorlabs driver vs image resolution.

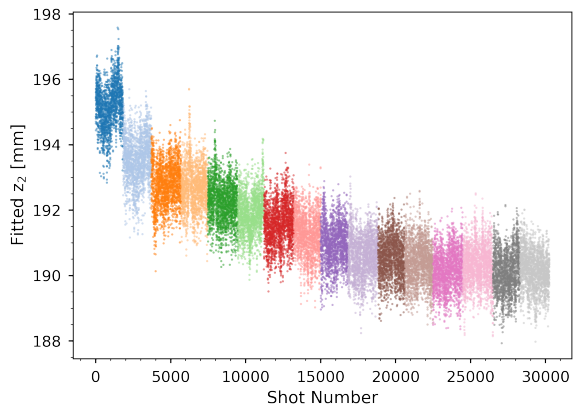

Figure 5: Representative dataset variation in fitted Z_2 calculation.

sensor measurements. Figure 6 shows a correlation plot between the two sensors, for which the raw correlation, as measured by the Pearson's coefficient between computed radius of curvature, is only 0.45.

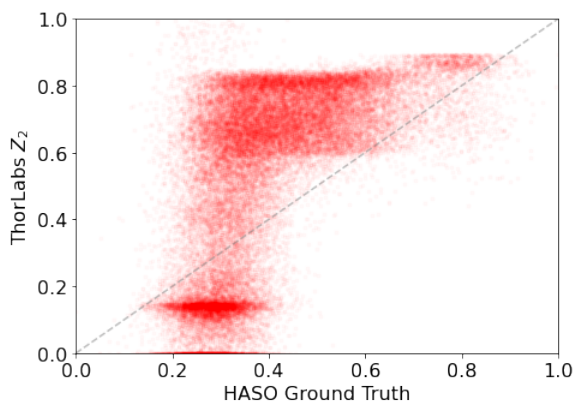

Figure 6: Raw HASO/ThorLabs correlation.

Due to the lack of correlation between the two sensors using raw pixel data, it became necessary to develop a pre-processing flow and system model to accurately capture systematic differences in the two measurements. We thus developed and trained a set of neural network models with the aim of improving the correlation between the two devices. The output for the trained network was a prediction of the radius of curvature, to be compared against the HASO WFS measurement.

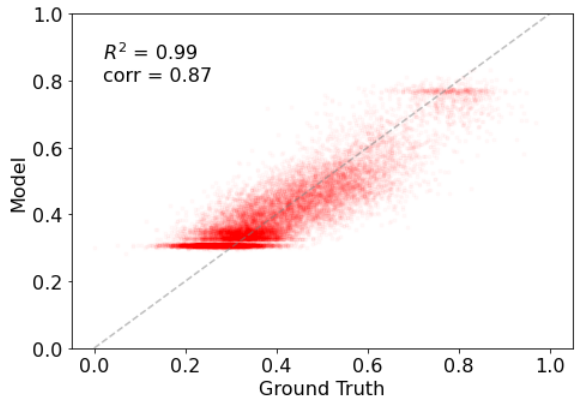

Figure 7: Feed-forward network accuracy.

Our initial efforts considered two different types of neural networks – convolutional neural nets (CNN) and more general feedforward neural nets (FFNN). Each network was trained using PyTorch, an open source library for developing machine learning models.

CNNs are designed to operate on images as inputs and are useful for computer vision applications. Our CNNs were trained using 12 × 12 pixel-by-pixel image data of the wavefront from the Thorlabs WFS. We found that the CNN produced only modest improvements in the correlation, to upwards of 0.63 from the initial value of 0.45. We thus transitioned to exploring more general FFNNs.

Our FFNN architecture featured 2-4 fully connected hidden layers, ReLU activation functions, and implemented a robust scaler on inputs and outputs. These features were chosen to be fast-executing and compatible with our FPGA deployment strategy. We concluded that the FFNN consistently produced better results than the CNN, but did exhibit tradeoffs between input space size and network complexity, with implications for performance at high repetition rate.

The best correlation was found by augmenting the pixel data with additional Zernike polynomial fitting terms; we explored several different strategies for generating the fit. Using the Thorlabs toolkit to produce a 5th order fit provided an additional 16 terms to include in the input space of the network, and improved dataset correlation to as high as 0.87, using only two hidden layers, as shown in Fig. 7.

Using external fitting libraries, such as the Mahotas library [3], permitted higher order fits, such as a 28-value, 6th-order fit. However, increasing the fit complexity showed diminishing returns, as correlations did not improve significantly, while speed of execution declined. Using a 6th-order fit does enable a network to be trained using only fitting data (28 inputs), and can result in comparable performance to

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU1BC004

Table 2: Correlation Values for Different Data

Dataset	Pearson Correlation
Zernike fits only	0.45
CNN - pixel data only	0.63
FFNN - pixel data only	0.82
FFNN - pixel data & Zernike fits	0.87

that of the full set of pixel values. Table 2 summarizes the correlation performance for each of our approaches.

IMPLEMENTATION

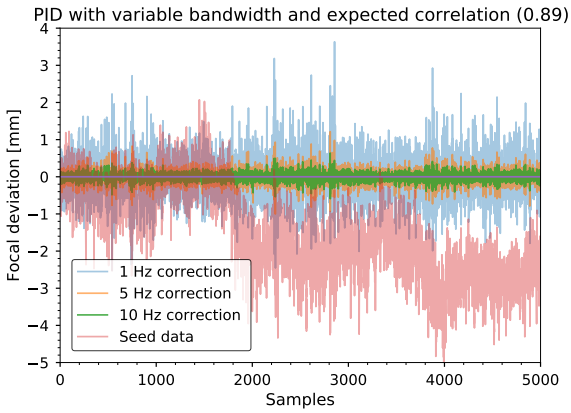

Figure 8: Testing of PID loop using simulated model at a number of data update rates.

Both machine learning and traditional control models were evaluated when looking at how to best correct errors. Figure 8 shows an estimated PID correction using a controlled designed to work on the 1 kHz seed pulse data (shown in red). A number of correction rates were simulated, along with a toy model of the focal position region, allowign for an estimation of the corrected signal and feedback performance.

While such a controller can operate on incoming data with no variations in a relatively straightforward manner, the BELLA laser system and interaction region still contest with a number of systematic concerns. A model-based control method allows integrating and learning these issues without prior knowledge, but can also increase the complexity of the controller. We are still in the process of evaluating which controller performance best manages running in the real experiment, along with instrument errors.

Figure 9 shows the importance of profiling.

A full correction implementation was prototyped using the FPGA system and was tested on the bench to meet-or-exceed the operational requirements of the HTU beamline. This implementation utilized the Xilinx Vitis AI toolkit in conjunction with the Xilinx Deep Learning Processor (DPU) to minimize the use of custom FPGA designs and software.

Due to driver limitations of the Thorlabs WFS20 sensor, in particular being limited to a Windows-based platform,

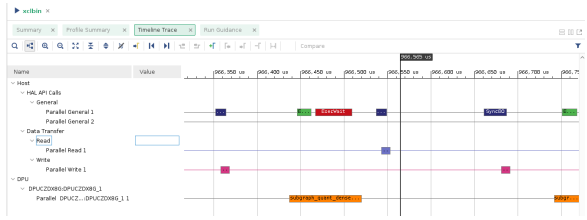

Figure 9: Trace from Vitis AI profiler showing inference being run on a single image. A single inference takes a bit less than 200 µs.

the sensor was unable to be directly connected to the processing platform. This necessitated the use of an alternative data communication channel. This channel was created in Python using ZeroMQ, and tested to transfer wavefront data to the processing platform at the limit of the sensor capture rate (about 0.9 kHz). Validation data from model development was used to test the model processing performance, and achieved a better than 5 kHz throughput, with well-understood bottlenecks and limitations. Vitis AI [4] profiling and trace tools were used to determine any issues with inference and data movement, as shown in Fig. 9. The DPU takes up a majority of this time, with much of the time spent handling overheaad outside of the Vitis AI code. Due to the minimal number of outputs, additional output data processing should not over-burden the system, enabling performance that meets the 1 kHz seed pulse rate on the HTU beamline.

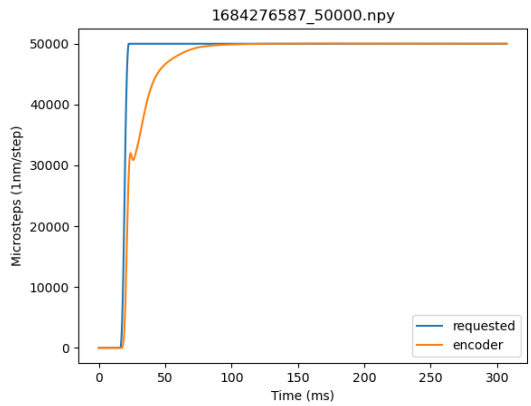

Figure 10: Response of motion stage to a request for a move of 50 µm.

We selected a Zaber Motion X-LDA-A linear stage to prototype the controller. Initial tests show a need to correct at the hundreds-of-µm level to suppress noise, but the bandwidth of the output motion stage is a limiting factor in how fast the controller can update the system due to a combination of an internal PID loop and relatively slow serial communication. In addition, the lack of a *park* feature means that the system is not easily locked into a set position while unpowered, necessitating both an online control scheme and the ability to transparently communication with

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TU1BCO04

the motion stage electronics for debug and high-level control. Figure 10 shows the trace of a requested 50 µm movement on the prototype motion stage with default stage settings, showing a settling time from scope start to acheived position with 100 nm of about 100 ms. Larger motion requires a longer settling time, and the motion stage can also take new requests while already moving.

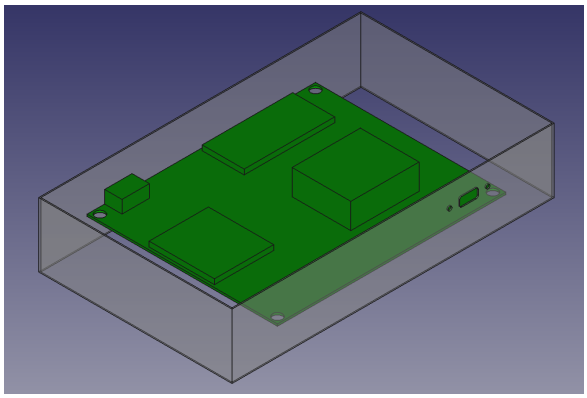

Figure 11: Feed-forward network accuracy.

The evaluation board is being installed in an off-the-shelf chassis to simplify rapid procurement. Figure 11 shows the ZCU104 in a Hammond Manufacturing chassis, allowing for neater packaging on the beamline.

CONCLUSION

We have demonstrated a model of the BELLA Center HTU beamline interaction region and developed a correction method for the focal position. This model, in conjunction with slow controllers, corrects for measured system variations in simulation. This method has been demonstrated in prototype hardware using simulated data and meets-or-exceeds the necessary performance requirements with room for expansion and increased model complexity as needed.

Limitations exist in that variations between seed and full-power pulses might require multiple models for proper correction. Plans exist to continue this work on additional beamlines to develop a flexible, plug-and-play framework for additional LPAs.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under Award Number DE-SC0021680.

REFERENCES

[1] E. Esarey, C. B. Schroeder, and W. P. Leemans, "Physics of laser-driven plasma-based electron accelerators," *Rev. Mod. Phys.*, vol. 81, no. 3, pp. 1229–1285, Aug. 2009. doi:10.1103/RevModPhys.81.1229

[2] F. Isono *et al.*, "Update on BELLA Center's Free-Electron Laser Driven by a Laser-Plasma Accelerator," in *Conference on Lasers and Electro-Optics*, San Jose, CA, USA, 2019, SF3I.1. doi:10.1364/CLEO_SI.2019.SF3I.1

[3] L. Coelho, "Mahotas: Open source software for scriptable computer vision," *Journal of Open Research Software*, vol. 1, no. 1, p. e3, 2013. doi:10.5334/jors.ac

[4] "Vitis AI," Xilinx. (Oct. 5, 2023), https://github.com/Xilinx/Vitis-AI