APPLICATIONS OF ARTIFICIAL INTELLIGENCE IN LASER ACCELERATOR CONTROL SYSTEM

F.N. Li*, Z. Guo†, M.X. Zang, Y.D. Xia, Q.Y. He, K. Chen, Q. Wang[1], C. Lin[1]
Peking University, Beijing, China
[1]also at Beijing Laser Acceleration Innovation Center, Beijing, China

Abstract

Ultra-intense laser-plasma interactions can produce TV/m acceleration gradients, making them promising for compact accelerators. Peking University is constructing a proton radiotherapy system prototype based on PW laser accelerators (CLAPA-II). This transient acceleration process becomes more challenging for stability control, which is critical for medical applications. This work demonstrates artificial intelligence's application in laser accelerator control systems.

Laser accelerator requires fast implementation of micro-precision alignment between the ultra-intense laser and the target. We proposed an automated positioning program using the YOLO algorithm. This real-time method employs the convolutional neural network, directly predicting object locations and class probabilities from input images. It enables precise, automatic solid target alignment in about a hundred milliseconds, reducing experimental preparation time. The YOLO algorithm is also integrated into the safety interlocking system for anti-tailing, allowing quick emergency response.

The intelligent control system also enables convenient, accurate beam tuning. We developed high-performance virtual accelerator software using OpenXAL and GPU-accelerated multi-particle beam transport simulations. The software allows real-time or custom parameter simulations and features control interfaces compatible with optimization algorithms. By designing tailored objective functions, the desired beam size and distribution can be achieved in a few iterations.

INTRODUCTION

Ultra-intense laser interaction with solid targets can produce acceleration gradients up to TV/m [1], which is considered a promising candidate for future compact accelerators. Pulsed proton sources from laser-plasma accelerators (LPA) offer significant application advantages in high-dose-rate tumor radiotherapy [2,3], extreme environmental material irradiation [4,5], and proton radiography [6,7] due to their short time pulse and high peak current intensity. Among various acceleration mechanisms, Target Normal Sheath Acceleration (TNSA) has been extensively studied both theoretically and experimentally [8–10]. This mechanism stands out for simplicity, robustness, and smooth beam profile compared to other mechanisms. Theoretical and experimental studies have shown that the cut-off energy of protons accelerated under the TNSA mechanism is sensitive to the laser intensity I_0 [11, 12]following an $E_{max} \sim I^{0.5}$ scaling low [13]. Laser acceleration experiments have already demonstrated high-quality particle beams with energies approaching nearly 100 MeV [14].

To optimize laser energy deposition on the surface of the targets, the main laser with a center wavelength of λ is focused on a spot with the size of several micrometers, denoted as r_0 , using an off-axis parabolic mirror (OAP). This configuration results in a Rayleigh length at the micrometer scale, as indicated by $L_R = \pi r_0^2/\lambda$. Consequently, precise alignment at the micrometer level between the laser spot and the target surface becomes imperative and optical methods capable of measuring distances are employed.

Traditional automatic alignment depends on multi-step algorithms [15, 16] that are time-consuming. The defocus distance is measured from the imaging system by designing a focus measure function. The focus measure function value will reach its extremum when the lens motor is approaching the optimal imaging position. Nowadays, artificial intelligence techniques for computer vision have been successfully used for rapidly processing imaging information [17, 18]. This new method ensures highly efficient target positioning. In the first session, we explored a deep learning method to realize rapid automated positioning. We successfully demonstrated that the YOLO [19,20] (You Only Look Once) object detection network enables fast and high-precision automatic positioning. Subsequently, we integrated this deep learning model into the laser accelerator control system. Additionally, YOLO algorithm has also been implanted into the safety interlocking system of CLAPA-II. It can display information on the operator interface and issue a sound alarm to help operators quickly take necessary actions in emergency situations.

In the second session, we delineated the developmental process of virtual accelerator software within laser accelerators, encompassing the development of GPU-accelerated multi-particle beam transport simulation algorithm, the development of software interfaces, and the incorporation of applications utilizing genetic optimization algorithms. The virtual accelerator software has accomplished the bi-directional conversion between physical quantities and control quantities within the control system. This advancement facilitates the more convenient and efficient application of physical and artificial intelligence algorithms to accelerators, laying a foundational framework for the realization of intelligent control in future laser accelerators.

* fnli@stu.pku.edu.cn
† zhen_guo0327@stu.pku.edu.cn

SESSION 1: REAL-TIME OBJECT DETECTION IN LPA

Methods

The experimental setup and positioning system of LPA are illustrated in Fig. 1. The key optical components of the positioning system include a high-magnification microscope lens and CCD camera with a specific depth of field, high-precision motors for controlling the six degrees of freedom in the orientation of OAP and the target holder, and a halogen lamp for controlling the illumination conditions. The four images in the bottom right corner of Fig. 1 display the target surface under high magnification. Typically, the CCD camera with the microscope lens is oriented at an angle relative to the target holder. Consequently, only a portion of the target surface can be imaged clearly. By identifying the clear area, the target position can be determined: a clear position on the right indicates that the target is located in front of the focal plane, a clear position on the left indicates that the target is positioned behind the focal plane, and a clear position in the center indicates that the target has been accurately aligned.

Once a clear area is identified within the field of view, the motors can be adjusted by scanning the distance until the clear position aligns with the center of the image. To achieve automated and fast target positioning, a feedback signal of the target is required to control the movement of the motors. In this study, we utilized a deep learning algorithm to detect the clear area and calculate the defocus distance.

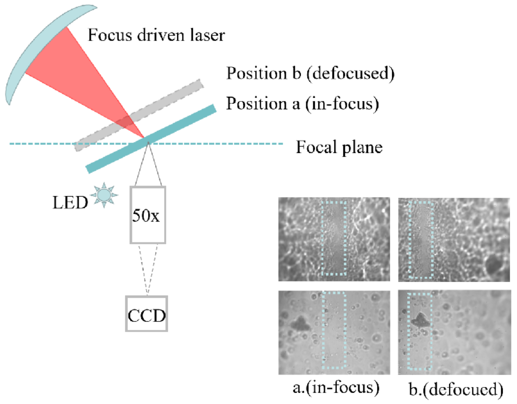

Figure 1: Laser-plasma Acceleration Experiment Setup Layout. The top row displays the imaging of the aluminum surface, while the bottom displays the imaging of the plastic surface. Within the field of view, the clear area can be observed in the middle (column a) at target position a (in-focus). Similarly, the clear area can be observed on the left (column b) at target position b (defocused).

A Faster-Automated Positioning Algorithms: YOLO

We proposed a faster-automated positioning algorithm using the YOLO deep learning model (as shown in Fig. 2). YOLO is a highly efficient convolutional neural network that directly performs object localization and classification on images.

The YOLO algorithm works by dividing the input image into a grid of cells and predicting bounding boxes and class probabilities for each cell. Each bounding box prediction consists of four coordinates (x, y, width, height) that define the location and size of the object. Additionally, there are class probabilities that indicate the likelihood of each object class being present within the bounding box. Another crucial aspect of YOLO's performance is the management of bounding box overlap. When the overlap between two bounding boxes exceeds this threshold, non-maximum suppression (NMS) is applied to retain only the most confident prediction. This prevents multiple overlapping boxes from being considered as distinct objects.

The compact size and faster inference speed of YOLO make it well-suited for real-time object detection. Once the relative positions of the clear and non-clear areas are identified, the Y-axis motor movement distance can be estimated through linear calculation.

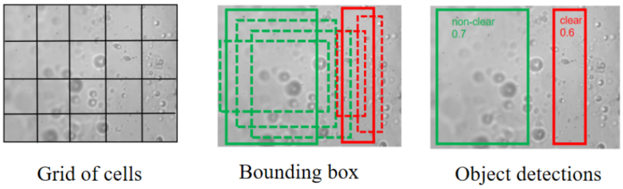

Figure 2: Object Detections on the Target Surface [19].

Experiments Details and Results

Data Collection and Annotations We acquire a series of target surface images at varying defocus distances automatically by controlling the motor's movement using the LabVIEW program. Initially, each target is manually positioned and the optimal spatial positions are stored in advance. Subsequently, we develop an auto-control program that scans the target both forward and backward with a 10um step size by controlling the y-axis of the motor. The camera captures and stores target images at defocus distances with each step size. Finally, we annotate the clear and non-clear positions on each image, saving the annotations as JSON files. These annotations are represented by rectangular box coordinates and object categories. About 230 plastic target images are annotated and randomly shuffled into training and validation sets in an 8:2 ratio.

Training the Network Training the model with the image data sets on a Tesla P100 GPU takes approximately 40 minutes. The model gradually converged within 400 epochs.

We also conducted a series of experiments to investigate how different training set sizes affect the training time and model convergence(Fig. 3).In experiment 1 and experiment 2, We randomly selected 50 and 100 images from the original training set as new training sets respectively. In experiment 4, we created a larger training set comprising 1,000 images annotated with location information. This training set includes 200 manually labeled images and 800 images labeled automatically by a neural network after a manual augmenting process. The augmentation of labels involves adding slight numerical noise to the coordinates and width of the bounding boxes, removing object annotations with low confidence, and occasionally duplicating annotations for images with the same defocus distances. Image augmentation techniques include random flipping and adjustments in brightness.

Liner Transformation of Lables To complete testing target positioning, it is necessary to convert the position labels of the detected objects into the movement distance on the y-axis motor through a linear transformation. The transformation method is divided into two ways. If a clear object is detected, we search for the annotation with the highest confidence value and obtain the corresponding label's horizontal coordinate x, to calculate the motor movement distance using the equation $L = k_1(0.5 - x)$. If only a non-clear position is detected, we search for the non-clear position annotation with the maximum confidence value and obtain the corresponding label's width w, and horizontal coordinate x. The motor movement distance should be $L = k_2 w \cdot sign(0.5 - x)$. Here,the linear transformation coefficients k_1 and k_2 are determined by the angle of the image plane and the target movement direction.

Test Results We present several metrics including MAE (Mean Absolute Error) and R2 score. MAE measures the average difference between the predicted and actual defocus distance. R2 score, also known as the coefficient of determination, assesses the goodness of fit of the regression line to the data points. The formulas for each metric are as follows:

$$MAE = 1/N \sum_{i=1}^{N} (|\hat{y}_i - y_i|) \tag{1}$$

$$R^2 = 1 - \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 / \sum_{i=1}^{N} (\bar{y}_i - y_i)^2 \tag{2}$$

Here, $|\hat{y}_i - y_i|$represents the error between the predicted values and the ground truth values. $\bar{y}_i$ represents the mean value of the ground truth. The influence of the training set size on the model's accuracy is illustrated in Fig. 3. It is observed enlarging the training dataset size augments the YOLO model's capacity for generalization. Upon reaching a training sample size of 1000 for plastic targets, the model exhibits similar performance levels when applied to both metal and plastic target test sets.

Inference

We selected the YOLO algorithm model trained in experiment 4 as the final deployment model due to its balanced performance in terms of inference speed and accuracy. For a single prediction from a 1600 × 1200 pixels image, it takes 20 ms on a desktop GTX-1080Ti GPU and 150 ms on a desktop CPU(2.60 GHz). Statistical analysis of the plastic target calibration result is illustrated in Figs. 4 and 5. Statistical tests indicate that the calibration errors follow a normal distribution. It can be considered that the average positioning error is less than 1/10 of the Rayleigh length range of the experiment and 95% of the targets' calibration errors(the range of $\mu \pm 2\sigma$) are within a 20-micrometer error.

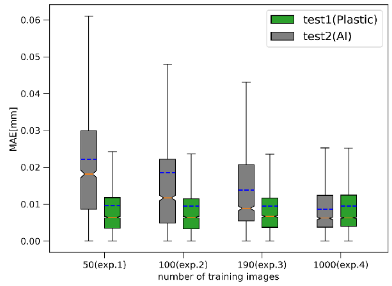

Figure 3: Mean Absolute Error (MAE) on test sets predicted by the YOLO model trained with different training set sizes.

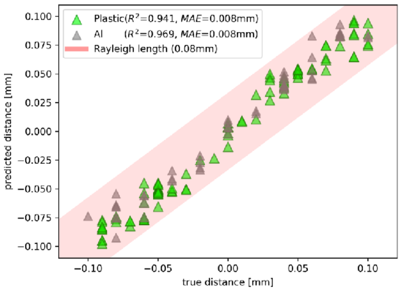

Figure 4: A linear regression analysis of the inference results by the YOLO model (exp.4) is depicted. The red area represents the 80-micron Raleigh length (corresponding to a 5-micron-focused spot). The green and gray dots respectively represent the testing results of plastic and aluminum images.

The interaction protocol between the YOLO model and the LabVIEW program is illustrated in Fig. 6. This architecture is implemented based on the FLASK framework, where the LabVIEW program controlling the target can issue calibration instructions. The server performs detection on the target images and outputs the motor movement distance via a linear transformation script. This methodology also alleviates the computational load on the local control computer.

Other Application

During the laser-plasma interaction process, other instantaneous radiation, including neutrons and gamma rays, can cause radiation injuries to the human body. Therefore, CLAPA-II requires a stable and reliable safety interlock system to shut down the laser source when humans enter the running experimental area. To address this, we have developed an anti-tailing subsystem for the safety interlock system. This subsystem utilizes the YOLO object detection algorithm to monitor the camera view and is deployed on edge mobile devices. When a human presence is detected, alarm signals are sent to the IOC (Input/Output Controller) of the EPICS (Experimental Physics and Industrial Control System) control system. This ensures immediate action is taken to prevent any potential harm.

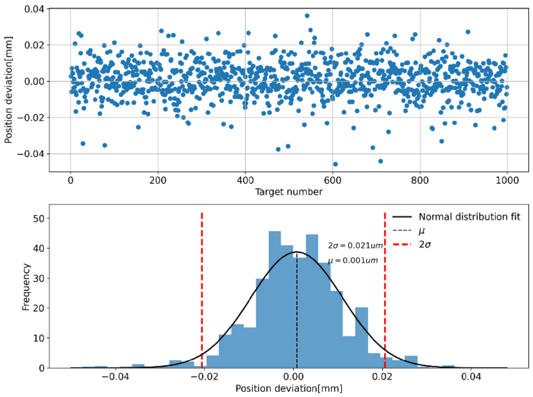

Figure 5: Position deviation of the plastic testing set (above, partial results shown) along with corresponding normal distribution fit (below).

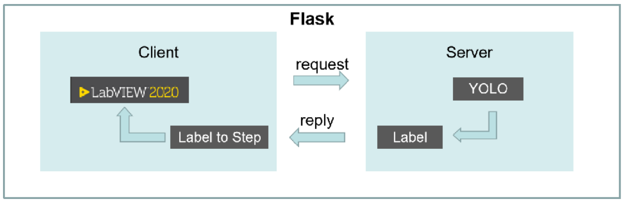

Figure 6: Architecture of the inference model and target control program. We deploy the model on the server, while the montor control program on the client (local computer). When a request command is sent to the server, the model predicts the calibration distance for the target, and the motor adjusts its position to achieve target positioning

Table 1: Comparison of Simulation Runtime Between the GPU Algorithm and TraceWin on the CLAPA-II Horizontal Beamline

Simulated Proton Count	**GPU Accelerated Algorithm** (s)	**TraceWin with Multi-threads** (s)	**Ratio**
10^3	0.106	2.194	20.7
10^4	0.116	2.677	23.1
10^5	0.2	4.302	21.5
10^6	0.95	23.614	24.9
10^7	7.684	199.468	25.9

SESSION 2: VIRTUAL ACCELERATOR SOFTWARE FOR BEAMLINE

The ongoing advancement in laser-plasma accelerators has instigated the development of multiple functional beam lines designed to transport pulsed particle beams characterized by considerable energy spread and divergence angle [21, 22]. These advancements necessitate the development of optimal design strategies and online feedback mechanisms for laser beam lines, prompting the inception of Virtual Accelerator (VA) technology [23]. Virtual Accelerators are crucial for real-time beam transport simulations, beam tuning, and communication with control systems, providing pivotal functionalities for the adjustment and operation of accelerators [24–29].

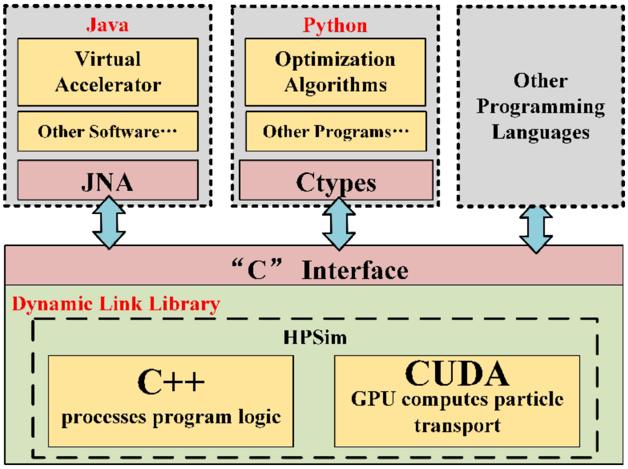

Figure 7: This figure shows the software architecture of the GPU-accelerated multi-particle beam transport simulation algorithm. Java programs call the algorithm through the "JNA" interface, and Python programs call the algorithm through the "ctypes" module.

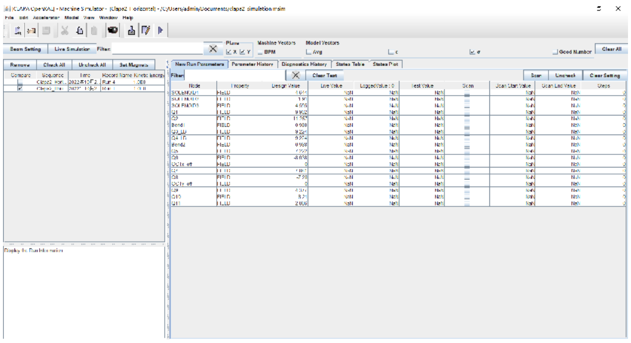

Figure 8: Main graphical user interface of virtual Accelerator software.

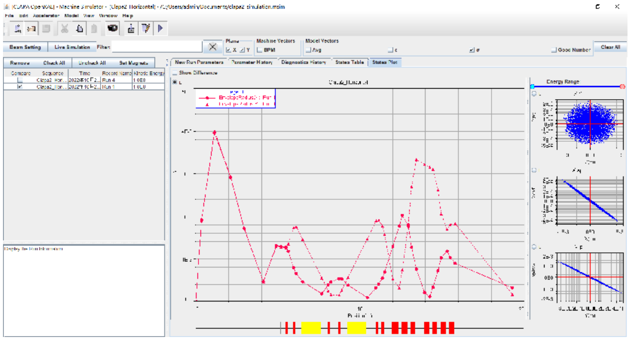

Figure 9: Virtual accelerator software simulation results visualization.

GPU Algorithm Development

We introduced a GPU-accelerated multi-particle beam transport simulation algorithm, derived from the open-source project "HPSim" [30, 31], aiming to bridge the accuracy-speed gap. This algorithm, utilizing Transfer Matrix-based methods, is 20 times more efficient than its con-

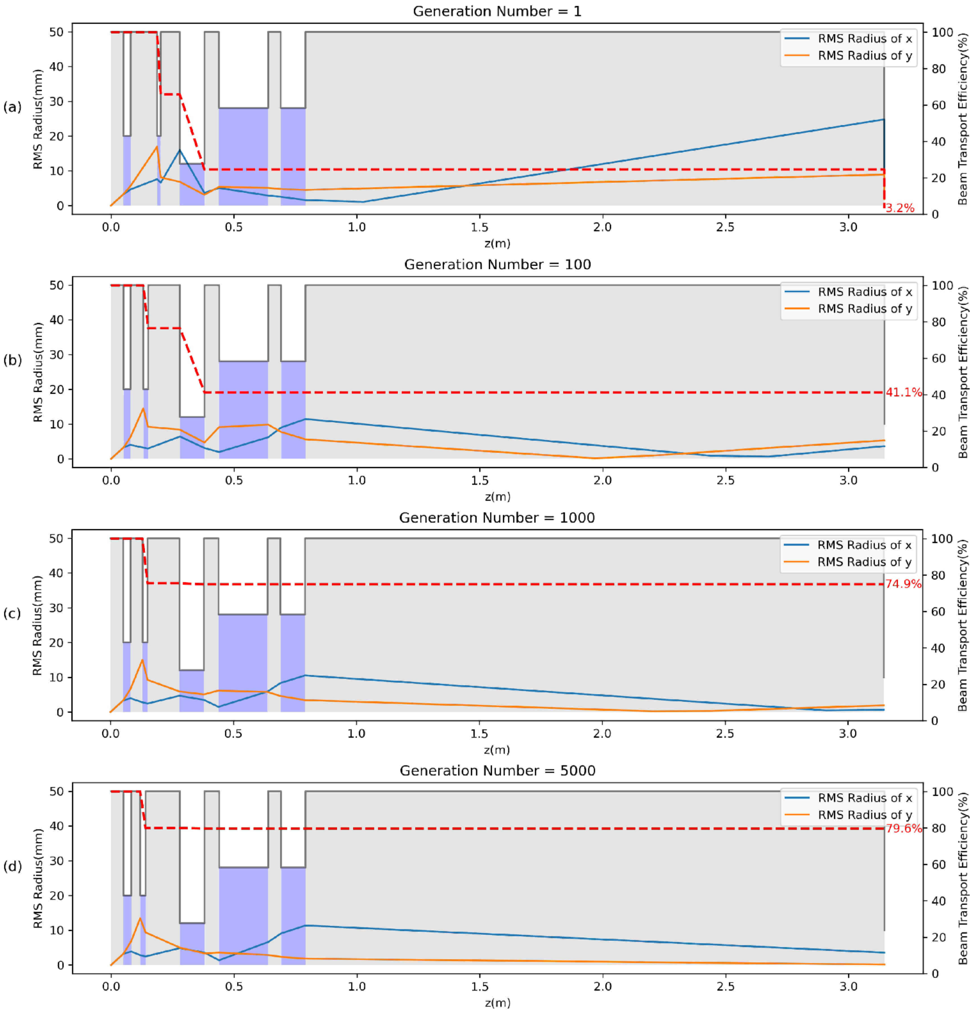

Figure 10: The optimal results in the 1st, 100th, 1000th, and 5000th generations of the genetic optimization algorithm. The solid line indicates the beam envelope change and the red dashed line indicates the beam transport efficiency.

temporaries like TraceWin [32], reducing simulation times to a few hundred milliseconds (as shown in Table 1). It is versatile, offering cross-programming and cross-language calls, and is adept at simulating solenoids and exponentially decreasing energy spectrum beams [33]. This algorithm is crucial for handling beams from laser-plasma accelerators due to their inherent misalignment and unique beam characteristics.

Figure 7 illustrates our 'C' interface, designed for the GPU-accelerated multi-particle beam transport simulation algorithm. Given the prevalent use of the Windows OS in the control room and among numerous researchers, we have compiled the simulation algorithm into a Dynamic Link Library (dll) on the Windows 10 platform. This approach ensures its accessibility to Java and Python programs, accommodating the diverse needs of users.

Development of Virtual Accelerator Software

The development of the Virtual Accelerator Software is pivotal for enhancing beamline diagnostics in laser accelerators. This software primarily serves to conduct simulations based on real-time or custom parameters of the accelerator, providing a virtual diagnostic framework. It efficiently computes variations in beam parameters along the beamline, furnishing essential references for beam tuning.

Integration and Development Framework Our integration of the GPU-accelerated multi-particle beam transport simulation algorithm into the OpenXAL framework [34] has resulted in the successful development of advanced virtual accelerator software. OpenXAL serves as an open-source environment designed for crafting applications, scripts, and services related to accelerator physics. Rooted in Java programming language, it supports the creation of feature-rich user interfaces and enables the implementation of intricate control algorithms.

User Interface and Control The software's main graphical user interface, as depicted in Fig. 8, permits users to switch between set values and real-time values from the EPICS control system [35] for simulations. It displays the magnetic field parameters' design values for each magnet and the real-time values retrieved from the EPICS control system. When conducting real-time simulations, the user can choose either design parameters or "Test Value" under the "Design" option or the real-time parameters under the "All Live" option.

Simulation Visualization The visualization of the beam simulation results is illustrated in Fig. 9. This feature encompasses a history section, beam parameters, and a section showing the transverse phase space distribution of the particles, reflecting the beam distribution and focusing in horizontal and vertical directions. This visualization also allows users to set the selected record to the EPICS control system during the experiment and filter out particles in a specific energy range using a slider.

Algorithm Implementation and Extension The Virtual Accelerator Software's structure allows for the incorporation of more complex control algorithms. These may involve the use of optimization algorithms to automate the adjustment of magnet parameters and achieve desired beam spot distribution, or a blend of real and simulated data to identify anomalies in beam diagnostic devices.

Beamline Magnets Optimization

The beamline magnet optimization process relied heavily on the enhanced speed of the GPU-accelerated algorithm, utilizing optimization techniques like the genetic algorithm [36] to identify ideal parameters. The efficiency of the algorithm enabled the completion of 1,000,000 beam transport simulations in approximately 2587.15 seconds, averaging 2.59 ms per simulation. The optimization process effectively augmented beam transport efficiency to 79.6%, minimizing beam loss rate and facilitating ideal beam spot distribution (as shown in Fig. 10). This optimization process is critical in designing proton collection systems that can handle high divergence beams resulting from laser acceleration.

CONCLUSION

In conclusion, this work has demonstrated the application of artificial intelligence in laser accelerator control systems. The utilization of the YOLO algorithm has enabled fast and automated positioning and the safety interlocking system, which have significantly reduced experimental preparation time and improved stability control. The developed virtual accelerator software has also facilitated accurate beam tuning by optimization algorithms. The desired beam size and distribution can be achieved in a few iterations

ACKNOWLEDGEMENTS

This work was supported by the National Grand Instrument Project (No.2019YFF01014400 and No. 2019YFF01014404) and also supported by High-performance Computing Platform of Peking University.

REFERENCES

[1] A. Maksimchuk *et al.*, "Forward Ion Acceleration in Thin Films Driven by a High-Intensity Laser",*Phys. Rev. Lett.*, vol. 84, no. 18, pp. 4108–4111, May 2000. doi:10.1103/PhysRevLett.84.4108

[2] M.C. Vozenin *et al.*, "Towards clinical translation of FLASH radiotherapy", *Nat. Rev. Clin. Oncol.*, vol. 19, no. 12, pp. 791–803, 2022. doi:10.1038/s41571-022-00697-z

[3] J. Han *et al.*, "Ultra-high dose rate FLASH irradiation induced radio-resistance of normal fibroblast cells can be enhanced by hypoxia and mitochondrial dysfunction resulting from loss of cytochrome C",*Front. Cell Dev. Biol.*, vol. 9, p. 672929, 2021. doi:10.3389/fcell.2021.672929

[4] N. Simos *et al.*, "Multi-MW accelerator target material properties under proton irradiation at Brookhaven National Laboratory linear isotope producer", *Phys. Rev. Accel. Beams*", vol. 21, no. 5, p. 053001, 2018. doi:10.1103/PhysRevAccelBeams.21.053001

[5] M. Barberio *et al.*, "Laser-accelerated particle beams for stress testing of materials",*Nat. Commun.*, vol. 9, no. 1, p. 372, 2018. doi:10.1038/s41467-017-02675-x

[6] D.Y. Li *et al.*, "Influence factors of resolution in laser accelerated proton radiography and image deblurring", *AIP Adv.*, vol. 11, no. 8, 2021. doi:10.1063/5.0039364

[7] A.Y. Faenov *et al.*, "Submicron ionography of nanostructures using a femtosecond-laser-driven-cluster-based source,"*Appl. Phys. Lett.*, vol. 95, no. 10, 2009. doi:10.1063/1.3210785

[8] R.A. Snavely *et al.*, "Intense high-energy proton beams from petawatt-laser irradiation of solids", *Phys. Rev. Lett.*, vol. 85, no. 14, p. 2945, 2000. doi:10.1103/PhysRevLett.85.2945

[9] E.L. Clark *et al.*, "Energetic heavy-ion and proton generation from ultraintense laser-plasma interactions with solids", *Phys. Rev. Lett.*, vol. 85, no. 8, p. 1654, 2000. doi:10.1103/PhysRevLett.85.1654

[10] M. Nishiuchi *et al.*, "Towards a novel laser-driven method of exotic nuclei extraction- acceleration for fundamental physics and technology", *Plasma Phys. Rep.*, vol. 42, pp. 327–337, 2016. doi:10.1134/S1063780X1604005X

[11] W.p. Wang *et al.*, "Focal spot effects on the generation of proton beams during target normal sheath acceleration", *Plasma Phys. Control. Fusion*, vol. 58, no. 2, p. 025010, 2016. doi:10.1088/0741-3335/58/2/025010

[12] N.p. Dover *et al.*, "Effect of small focus on electron heating and proton acceleration in ultrarelativistic laser-solid interactions", *Phys. Rev. Lett.*, vol. 124, no. 8, p. 084802, 2020. doi:10.1103/PhysRevLett.124.084802

[13] P. Mora, "Plasma expansion into a vacuum", *Phys. Rev. Lett.*, vol. 90, no. 18, p. 185002, 2003. doi:10.1103/PhysRevLett.90.185002

[14] Y.T. Hu *et al.*, "Above-100 mev proton beam generation from near-critical-density plasmas irradiated by moderate laguerre–gaussian laser pulses",*Plasma Phys. Control. Fusion*, vol. 64, no. 12, p. 125002, 2022. doi:10.1088/1361-6587/ac95c3

[15] L. Firestone *et al.*, "Comparison of autofocus methods for automated microscopy", *Cytom.: J. Int. Soc. Anal. Cytol.*, vol. 12, no. 3, pp. 195–206, 1991. doi:10.1002/cyto.990120302

[16] N. Kehtarnavaz and H.J. Oh, "Development and real-time implementation of a rule-based auto-focus algorithm", *RealT. Imaging*, vol. 9, no. 3, pp. 197–236, 2003. doi:10.1016/S1077-2014(03)00037-8

[17] H. Mir *et al.*, "An autofocus heuristic for digital cameras based on supervised machine learning", *J. Heuristics*, vol. 21, pp. 599–616, 2015. doi:10.1007/s10732-015-9291-4

[18] Z. Ren *et al.*, "Learningbased nonparametric autofocusing for digital holography", *Optica*, vol. 5, no. 4, pp. 337–344, 2018. doi:10.1364/OPTICA.5.000337

[19] K. Redmon, "You only look once: Unified, real-time object detection", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779-788. doi:10.1109/CVPR.2016.91

[20] YOLOv5 by Ultralytics, https://github.com/ultralytics/yolov5

[21] K. Wang *et al.*, "Achromatic beamline design for a laser-driven proton therapy accelerator"*Phys. Rev. Accel. Beams*, vol. 23, no. 11, p. 111302, 2020. doi:10.1103/PhysRevAccelBeams.23.111302

[22] J. G. Zhu *et al.*, "Experimental demonstration of a laser proton accelerator with accurate beam control through image-relaying transport", *Phys. Rev. Accel. Beams*, vol. 22, no. 6, p. 061302, 2019. doi:10.1103/PhysRevAccelBeams.22.061302

[23] A. Shishlo, p. Chu, J. Galambos, and T. A. Pelaia, "The EPICS Based Virtual Accelerator - Concept and Implementation", in *Proc. PAC'03*, Portland, OR, USA, May 2003, paper WPPE017, pp. 2366–2368.

[24] J. Galambos *et al.*, "XAL - The SNS Application Programming Infrastructure", in *Proc. EPAC'04*, Lucerne, Switzerland, Jul. 2004, paper THPLT168, pp. 2855–2857.

[25] R. Kammering *et al.*, "The Virtual European XFEL Accelerator", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 578–580. doi:10.18429/JACoW-ICALEPCS2015-TUD3O04

[26] H. Harada *et al.*, "Current Status of Virtual Accelerator at J-PARC 3 GeV Rapid Cycling Synchrotron", in *Proc. PAC'07*, Albuquerque, NM, USA, Jun. 2007, paper MOPAN028, pp. 215–217.

[27] D. Gu *et al.*, "Development of virtual accelerator environment for beam diagnostics", *High Power Laser Part. Beams*, vol. 26, p. 125101, 2014. doi:10.11884/HPLPB201426.125101

[28] Y. Li *et al.*, "Beam-based Alignment Simulation on Transport Line of CSNS", in *Proc. IPAC'15*, Richmond, VA, USA, May 2015, pp. 3818–3820. doi:10.18429/JACoW-IPAC2015-THPF051

[29] Y. Li *et al.*, "CSNS Linac Beam Commissioning Tools and Experience", in *Proc. LINAC'18*, Beijing, China, Sep. 2018, pp. 750–752. doi:10.18429/JACoW-LINAC2018-THPO032

[30] X.Y. Pang and Larry Rybarcyk, "Gpu accelerated online multi-particle beam dynamics simulator for ion linear particle accelerators", Comput. Phys. Commun., vol. 185, no. 3, pp. 744–753, 2014. doi:10.1016/j.cpc.2013.10.033

[31] Hpsim, https://www.osti.gov//servlets/purl/1333548

[32] D. Uriot and N. Pichoff, "Status of TraceWin Code", in *Proc. IPAC'15*, Richmond, VA, USA, May 2015, pp. 92–94. doi:10.18429/JACoW-IPAC2015-MOPWA008

[33] Y.X. Geng *et al.*, "Generating proton beams exceeding 10 MeV using high contrast 60 TW laser", *Chin. Phys. Lett.*, vol. 35, no. 9, p. 092901, 2018. doi:10.1088/0256-307X/35/9/092901

[34] OpenXAL, https://openxal.github.io

[35] EPICS, https://epics-controls.org

[36] D.E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*. Boston, Boston, MA, United States: Addison-Wesley Longman Publishing Co., Inc., 1989.