ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEXC2

ADVANCES IN PROTON LINAC ONLINE MODELING*

Xiaoying Pang†
Los Alamos National Laboratory, Los Alamos, NM, 87545, USA

Abstract

This paper will review current online modeling tools used for proton linacs and then focus on a new approach that marries multi-particle beam dynamics with modern GPU technology to provide pseudo real-time beam information in a control room setting. Benefits to be discussed will include fast turnaround, accurate beam quality prediction, cost efficiency, test bed for new control and operation scheme development and operator training.

INTRODUCTION

Accelerator online modeling codes are indispensable tools in proton linac control rooms. At their hearts are the online beam dynamics simulators. These simulators are developed to have direct access to real machine parameters and to be able to provide fast predictions of various beam quantities. Together with some optimization tools, they can guide accelerator operators to tune up and troubleshoot the linac systems efficiently. Existing online simulators for proton linacs are mostly developed based on the envelope model which represents a r.m.s. equivalent beam as an ellipse in phase space and uses a linear representation of the space charge force. While the envelope model enables fast response and good accuracy for beam matching and steering in most cases, it lacks the capability to simulate using real beam distributions and study a range of real-world scenarios where the beam is not a nicely formed ellipsoid or the beam intensity is high enough so that a more accurate representation of the space charge force is needed. For high power beam operations, there will be less room for operational error, therefore a more accurate multi-particle online simulator is needed to provide more realistic beam information and tuning guidance.

In the following sections, we will first review the existing online modeling tools and briefly discuss the need for a multi-particle simulator for high power beam operations. Then we will introduce the high performance simulator (HPSim) that we have been developing. Details about the design, performance and its applications will be presented.

EXISTING ONLINE MODELING TOOLS

The list of tools presented here is by no means exhaustive, but they can be considered representative of what have been used in proton linac control rooms worldwide.

TRACE2D/3D

TRACE2D and TRACE3D [1] are well-established envelope based simulation codes initially developed at LANL. TRACE2D represents a beam using phase space ellipses in the two transverse planes, while the TRACE3D has the additional capability of simulating the longitudinal beam dynamics by assuming a 3D ellipsoidal beam shape. In both cases, the beam is a r.m.s. equivalent representation and it is described as a sigma matrix whose elements are related to the Twiss or Courant-Snyder parameters. Since they only track beam through the beamline elements represented as R-matrices and only consider the linear part of the space charge force, TRACE2D/3D simulations can be very fast, which makes them perfect for accelerator online modeling. In fact, the TRACE2D is still used in the proton linac control room at the Los Alamos Neutron Science Center (LANSCE). The physics algorithms of TRACE2D/3D have been used in the development of other online modeling tools and they have been widely used to benchmark against other similar codes.

XAL Online Model

XAL is a Java development framework for accelerator applications [2]. It was first developed and tested at the Spallation Neutron Source (SNS) and has been widely adopted by several accelerator sites, not limited to proton accelerator facilities. The online modeling tools [3] of XAL can simulate not only linacs but also storage rings. It comes with a suite of tools [4] that can be very useful in accelerator tuning, i.e. PASTA [5] for warm linac RF setpoint tuning, SLACS for superconducting cavity [6] RF setpoint tuning and tools for orbit corrections, etc. Like TRACE3D, the XAL online model tracks a beam in the 6D phase space based on the envelope model and uses the linear space charge approximation.

ESS Linac Simulator

More recently, a new online beam dynamics simulator is being developed at the European Spallation Source (ESS) [7–9]. It is also an envelope model based simulator. In order to be able to simulate the 5MW beam correctly, better treatment of the RF transformation and nonlinear space charge effect have been implemented [8]. However, the nonlinear space charge effect is calculated by assuming the beam distribution is Gaussian within the 3D ellipsoid.

HIGH PERFORMANCE ONLINE MULTI-PARTICLE SIMULATOR

The common theme of the existing online simulators for proton linacs is that they are all based upon the envelope

* Work supported by U.S. DOE, NNSA under contract DE-AC52-06NA25396
† xpang@lanl.gov

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEXC2

model. While the envelope model can provide fast response and works sufficiently well for a lot of applications, i.e. beam matching, it lacks the capability to handle real beam distributions and to treat the nonlinear space charge force accurately. This diminishes somewhat the usefulness of the existing online modeling tools, especially for high power beam operations where it does not produce the final set-points. That is what we at LANSCE, and people at other proton linac facilities around the world have experienced where loss-based tuning is required following physics based tuning [10].

At LANSCE, the half-mile long 800-MeV linac provides both H^+ and H^- beams for several user programs. The tune-up procedures usually begin with direct low-power beam measurements and set up of the beam lines based on the predictions from an envelope model (TRACE2D). However, in the transition to high-power operations, due to the lack of non-interceptive beam diagnostics and a good online modeling tool for high power beam, machine setpoints are empirically adjusted by operators to achieve minimal beam loss along the linac. These adjustments are usually done in a very high dimensional parameter space, which can make the tuning process very inefficient, and the machine settings subjective and inconsistent. More non-interceptive diagnostics can always help, however the information provided by those diagnostics is still limited and the hardware can be expensive. The best way to eliminate or reduce the gap between the low power and high power operations would be to complement the non-interceptive diagnostics with a good online modeling tool that can provide accurate predictions and new insight into the high power beam performance.

Introduction to HPSim

To accurately model the nonideal nature of the real world high power proton beam, we need to use a multi-particle model, which can simulate real beam distributions and uses particle-in-cell (PIC) algorithms to calculate space charge effects. But these features can also make it too slow for online modeling. In fact, a lot of multi-particle beam dynamics simulation codes would need a computer cluster or even supercomputer to make them fast enough for online accelerator modeling. That is why those codes are usually reserved for offline accelerator design and simulation.

Our goal in making the high performance simulator (HPSim) is to strike a balance between accuracy and speed and to make an online modeling tool that is both accurate and fast enough to be used for high power beam operations.

Accurate physics algorithm from PARMILA The physics algorithms of the HPSim comes mostly from the well-established PARMILA code [11]. PARMILA has been used to design and simulate several ion linacs around the world and has been proven to be a good simulation tool for ion linacs for decades. It is multi-particle, and uses PIC algorithms to simulate the space charge effects, e.g. SCHEFF 2D and PICNIC 3D. Unlike a lot of time based multi-particle simulation tools, PARMILA is location based. It uses the transfer maps in beam transport regions and gives a good treatment of the RF transformations in the cavities. Comparisons between PARMILA and other simulation codes including the time-based code PARMELA have shown good agreement [12]. However, PARMILA runs much faster than the time-based codes.

The current PARMILA is not up to the standard for online modeling due to the following reasons: 1) Its algorithms are developed for designing machines from scratch. It is difficult to use PARMILA to simulate the off-design scenarios which happens most of time in real-world operations. 2) Real-world machine parameters can not be easily input into the simulation. 3) Although PARMILA runs faster than a lot of other multi-particle codes, it is still slow for online modeling which requires a fast turnaround on the orders of seconds.

We addressed all these issues with PARMILA when we designed the HPSim. 1) The interaction with the physics algorithms have been made more transparent in the following senses: first, the geometric design has been made static and stored in the model (and the database). Second, from the model one can easily view all the important RF parameters i.e. synchronous phase of the cavity, updated based on the real world RF setpoints. Third, we keep track of each macro particle's absolute phase instead of its relatives phase to the synchronous particle so that it is more convenient to track the phase shifts in off-design scenarios and also made the code more parallelizable. Also in the HPSim, the transit time factors are functions of the particle velocity in order to more accurately simulate the off-energy particles traversing the RF gaps. In order to simulate beams with large energy spread, we have modified the space charge algorithm to focus mainly on the particles in the RF buckets. These modifications enable us to simulate a wide range of operation scenarios more accurately, i.e. RF amplitude and phase changes, RF cavities without fields, and reaccelerating. 2) We use a database in HPSim to facilitate the direct access of the real-world machine parameters. 3) HPSim is accelerated utilizing GPU technology and it is fast enough for online modeling.

Moreover, instead of calculating the complete space charge field everytime it is applied, the space charge routine scales the previously calculated field when a ratio composed of beam energy and sizes has not changed beyond a user-defined value. This is similar to what is done in ASTRA [13] and it improves the performance without sacrificing measureable accuracy. Currently, HPSim can simulate beams through drift tube linac (DTL) and coupled cavity linac (CCL) structures and several standard beam transport elements. Other structures can be incorporated as needed.

Accelerated by GPU In order to speed up the simulation, we utilized the start-of-the-art high performance computing hardware, e.g. the graphics processing unit (GPU). The cost effectiveness and the impressive parallel computing performance of the GPU make it the world's most accessible and most popular supercomputing platform. With

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEXC2

the GPUs, we can very easily make a online modeling workstation that can out-perform a small CPU cluster and meanwhile is inexpensive and can be available for accelerator operations 24/7. We are able to achieve a significant performance boost and to get a start-to-end runtime ranging from subseconds to several seconds or longer depending on the size and scale of the simulations.

HPSim Code Design and Performance

Code Design The goal of the HPSim structure design is to ensure fast execution and enable ease of use, which has led us to write our number crunching simulation kernels in CUDA C and C++ in order to take advantage of the GPU performance, and then wrap these fast kernels with Python/C APIs so that the users need only to communicate with the simulator through easy-to-use Python scripts. The hierarchical structure of the code is shown in Figure 1. The combination of a low level compiled language, e.g. CUDA C and C++, and a high level scripting language, e.g. Python, really enables both fast simulations and quick prototyping. It also allows the users to take advantage of the rich numerical and visualization libraries that are available for Python without the need to reinvent the wheel.

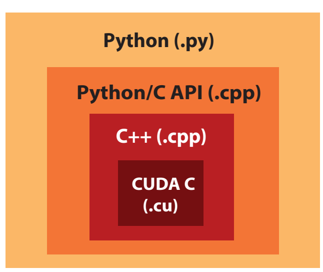

Figure 1: Hierachical structure of the HPSim code. Users only need to communicate directly with the outer Python layer.

Figure 2 shows the major components of the code structure. The simulator allocates and utilizes resources both on the GPU(device) and the CPU(host) sides. The components that are shaded in blue are stored on the CPU side, while the ones in yellow are generated and stored on the GPU during the simulation. The center box that is colored in green represents the beamline component that needs to be updated from the CPU side and is needed in the simulation on the GPU side. So it resides in the pinned memory, which is a preallocated region on the RAM on the host that can be efficiently accessed from both sides. The real machine parameters are taken from the EPICS control system and recorded in a flat file like, serverless SQLite database. This database contains not only the engineering units of the parameters but also the converted physics units and the conversion rules and the calibration factors required to transform machine setpoints to model values. The model values are used to generate the beamline and later used in the simulation on the GPU. More details about the code structures can be found in [14].

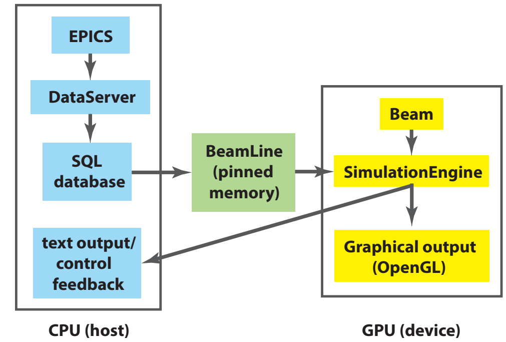

Figure 2: High level code structure and data flow indicated by the arrows.

GPU Performance Details about the GPU implementations of the beam transport and space charge algorithms have been reported in [14]. We compared the code performance using an Intel Xeon E5520 2.27GHz CPU and a NVIDIA GTX 580 GPU (Fermi architecture). The maximum speedup achieved for beam transported without space charge is around 160 and it is around 45 for the space charge routine alone.

A typical case we have simulated is the 800-meter-long LANSCE linac which consists of 2/3 of the 750-KeV low energy beam transport (LEBT), 100-MeV DTL, 100-MeV transport, and 800-MeV CCL. With a relatively newer NVIDIA Tesla K20 GPU, we were able to simulate 64K macro-particles through the LANSCE linac (> 5100 RF gaps + 400 quads + 6000 space charge kicks) in 5.5 seconds. For the cases where we only simulate up to the end of the LANSCE DTL, the front-to-end simulation only takes about 0.4 seconds.

Using newer generation GPUs, newer features of the GPUs and multiple GPUs, one can further improve the performance of the simulator.

Accuracy The HPSim has been calibrated using beam measurement and has been benchmarked against real ex-

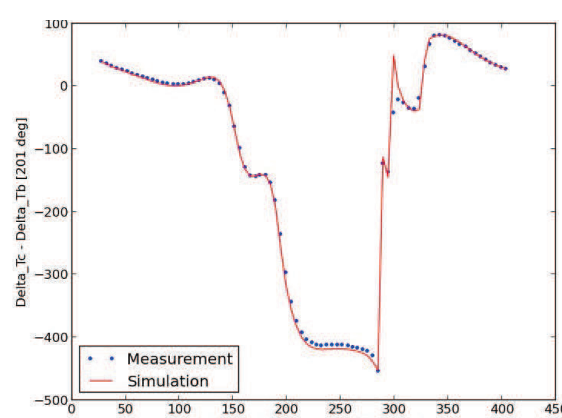

Figure 3: Comparison of HPSim result (red line) with measurement (blue dots) for the first CCL module phase scan, which utilizes beam phase measurement diagnostics.

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEXC2

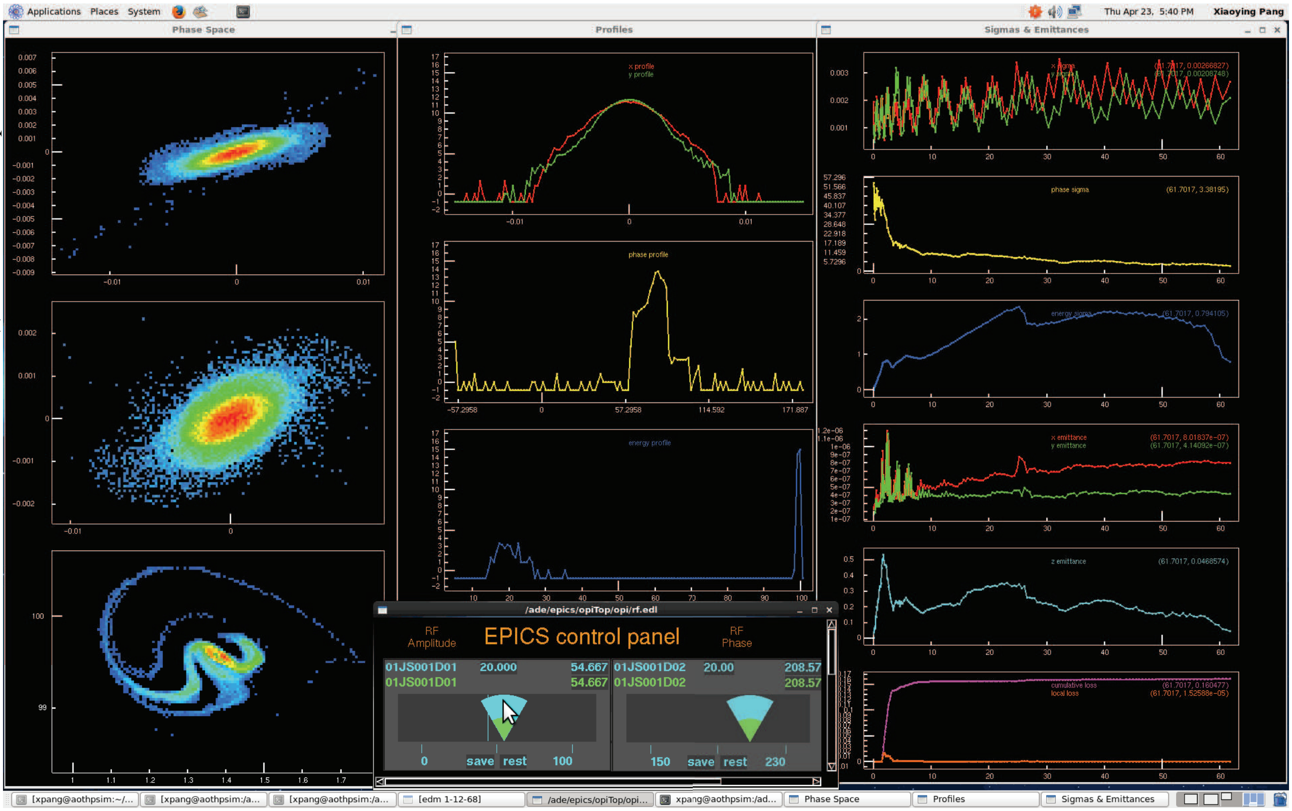

Figure 4: Screen shot of the online simulator graphical output showing phase space distributions, beam profiles, and emittance, beam r.m.s size and loss profiles for the LANSCE DTL.

periments. Figure 3 compares the simulated phase scan of the first module of the LANSCE CCL to the actual phase scan experiments. Good agreement between simulations and real measurement has been achieved.

Applications of HPSim

Guided Tuning and Troubleshooting The main goal of the HPSim development is to provide a virtual beam diagnostic to be used during accelerator operations to provide fast and accurate information about the beam evolution along the linac to aid in tune-up and troubleshooting. In these cases, the simulator can be configured to run continuously in the monitoring mode so that the operators can watch the beam characteristics change as they adjust the machine setpoints. Figure 4 is an example of the simulator 2D graphics output that can be updated continuously during the operations. During the start-up phase of linac operations, oftentimes, measurements do not meet expectations. The simulator can provide useful guidance to help diagnose and pinpoint problems with the operations. Details about the advantages of using HPSim are reported in [15].

What-if Studies and Operator Training The simulator can also be used offline for beam studies, i.e. finding the optimal location or the expected response of a new diagnostic, testing a new operational mode, adding a new beamline into the system. The advantage of the simulator compared to the traditional design tools is the use of the real machine parameter and its capability of simulating an off-design real world machine. The what-if studies can easily be extended to help train accelerator operators and physicists regarding the nuances of operating a high power linac and help them to develop better insight into the beam performance inside the linac. By utilizing a local copy of the EPICS system, i.e. soft IOC, an operator can manipulate machine settings and observe beam performance without impacting the acutal accelerator operations.

Multi-objective Optimization of Linac Setpoints We have used the simulator in an offline mode as the evaluator for the multi-objective particle swarm optimization (MOPSO) and the multi-objective genetic algorithm (MOGA) to find optimal operational settings in a high-dimensional parameter space [16]. Using these optimization techniques, we were able to produce machine settings that resulted in less beam loss, small beam emittances and less radiation damage to the linac than the design setting. From this study we also found that the MOPSO algorithm is far more efficient than the popular MOGA algorithm. PSO can even possibly be used as a online optimization algorithm [17].

Test Bed for Alternative Control Schemes A model independent automatic accelerator tuning method has been

ISBN: 978-3-95450-168-7 doi:10.18429/JACoW-IPAC2015-WEXC2

developed using the simulator as its test bed [18]. This real-time method can simultaneously tune several coupled components of an accelerator to achieve good beam quality. Since the acutal testing of the new scheme bears the potential risk of damaging the accelerator infrastructure and development time is expensive, HPSim allows one to first explore these new methods without the associated risk and cost. Using the simulator, the new control scheme has proven to be very efficient and robust to noise.

Autonomous Particle Accelerator Currently we are investigating the possibilities of combining HPSim and machine learning (ML) to create a more intelligent accelerator operating system. HPSim will play an indispensable role in training and testing the ML algorithms without risk damaging the accelerator infrastructure.

CONCLUSION

Existing online modeling tools for proton linacs are mostly based on the standard envelope model approach. Our newly developed GPU-accelerated multi-particle online simulator can be a very useful tool for high power accelerator operations. By combining multiparticle beam physics algorithms with GPU technology we are able to bring high-fidelity beam dynamics modeling capabilities to the control room environment in a cost effective way. The list of possible applications is by no means limited to the examples given above. Instead, the high-level Python interface enables the user to easily create and explore new applications. Further development of the simulator and its applications are ongoing.

ACKNOWLEDGMENT

Many thanks to my colleague Larry Rybarcyk who has been heavily involved in the development of this simulator since the beginning. I also want to thank my colleague Scott Baily for his help on setting up the EPICS system and Robert Garnett for his strong support for this work.

REFERENCES

[1] K. R. Crandall, D. P. Rusthoi, TRACE3D manual, LA-UR-97-886.

[2] J. Galambos et al., "XAL Application Programming Structure", PAC'05, Knoxville, Tennessee, 2005, ROPA001, p.79.

[3] C.K. Allen et al., "Novel Online Simulator for Applications Requiring a Model Reference", ICALEPCS2003, Gyeongju, Korea, 2003, WE116, p.315.

[4] J. Galambos et al., "Software Tools for Commissioning of the Spallation Neutron Source Linac", PAC07, Albuquerque, New Mexico, 2007, TUOCC01, p.883.

[5] J. Galambos et al., "PASTA – AN RF Phase Scan and Tuning Application", PAC'05, Knoxville, Tennessee, 2005, FPAT016, p.1491.

[6] J. Galambos et al., "Operational Experience of a Superconducting Linac Cavity Fault Recovery System", 5th Conf. Util. and Rel. of High Power Proton Linacs, Mol, Belgium, 2007.

[7] E. Laface et al., "The ESS Linac Simulator: a First Benchmark with Tracewin", IPAC'13, Shanghai, China, 2013, TUPWO046, p.1970.

[8] E. Laface et al., "Space Charge and Cavity Modeling for the ESS Linac Simulator", IPAC'13, Shanghai, China, 2013, WEPEA040, p.2588.

[9] I. List, E. Laface, "Benchmarking Beam Envelope Models for the European Spallation Source", IPAC'14, Dresden, Germany, 2014, TUPRO061, p.1174.

[10] J. Galambos et al, "Summary of the Commissioning, Operations and Performance Working Group of the HB-2010 Workshop", HB2010, Morschach, Switzerland, 2010, FROP04, p.703.

[11] H. Takeda, Parmila manual, Los Alamos Report, LA-UR-98-4478, Revised July, 2005.

[12] S. Nath et al, "Comparison of Linac Simulation Codes", PAC'01, Chicago, 2001, TOPB003, p.214.

[13] K. Floettmann, Astra manual.

[14] X. Pang, L. Rybarcyk., "GPU Accelerated Online Beam Dynamics Simulator for Linear Particle Accelerators" Comput. Phys. Commun. 185, 744, 2014.

[15] L. Rybarcyk et al, "Advantages to an Online Multiparticle Beam Dynamics Model for High-power Proton Linacs" Richmond, Virginia, 2015, MOPWI033, IPAC'15.

[16] X. Pang, L. Rybarcyk., "Multi-objective Particle Swarm and Genetic Algorithm for the Optimization of the LANSCE Linac Operation", Nucl. Instr. Meth. Phys. Res. A, 741, 124, 2014.

[17] X. Huang, J. Safranek, "Online Optimization of Storage Ring Nonlinear Beam Dynamics", http://arxiv.org/abs/1502.07799

[18] A. Scheinker et al, "Model Independent Particle Accelerator Tuning", Phys. Rev. ST Accel. Beams, 16, 102803, 2013.