RECENT PROGRESS AND PLANS FOR THE CODE elegant *

M. Borland †, V. Sajaev, H. Shang, R. Soliday, Y. Wang, A. Xiao, ANL, Argonne, IL 60439, USA
W. Guo, BNL, Upton, NY, 11973, USA

Abstract

`elegant` is an open-source accelerator code that has been in use and development for approximately two decades. In that time, it has evolved from a graduate student project with a narrow purpose to a general code for the design and modeling of linacs and storage rings. `elegant` continues to evolve, thanks in no small part to suggestions from users. `elegant` has seen extensive application to modeling of linacs, particularly for applications related to free-electron lasers and energy recovery linacs. Recent developments have emphasized both linac and storage-ring-related enhancements, along with parallelization. In this paper, we briefly review the features of `elegant` and its program suite. We then describe some of the recent progress made in the ongoing development of `elegant`. We also discuss several noteworthy applications and directions for future work.

INTRODUCTION

The program `elegant` [1] is now widely used in the accelerator community and is available as source code or in binary form for many operating systems. It started more than two decades ago as a graduate student project when the lead author concluded that it was easier to write a new code than to modify existing codes to include needed features. Since then, it has undergone almost continuous incremental improvement, with releases at approximately six-month intervals. The original structure and philosophy of the code are well suited to this process.

A basic `elegant` run has two inputs: a command input file and a lattice definition file. The command input file contains a series of namelist-like structures defining a series of commands to set up and execute a run. The lattice input file defines the lattice using a format that is very similar to that popularized by the program MAD [2].

One of the design goals of `elegant` was to make adding a new element no harder than writing code to implement the physics of the element. Toward this end, a set of data structures was defined that allows the developer to describe the properties and parameters of any new element, as well as the properties of those parameters. This element dictionary has made incremental improvement of the code relatively painless. (It is also used to automatically generate the manual pages for all elements.)

`elegant` attempts to implement as many features as possible using a lumped-element concept. For example, one may impart charge to a beam or change the Twiss parameters of a beam using a lumped element. This has the advantage of allowing `elegant` to vary or optimize such properties just as it could for a property of a quadrupole or any property of another traditional beamline element. Similarly, many local diagnostic outputs are obtained by inserting one of several diagnostic elements into the beamline.

`elegant` was the first accelerator code to make thorough use of self-describing data for input and output, starting originally with the Access With Ease (AWE) protocol [3] and transitioning in 1993 to the Self-Describing Data Sets (SDDS) protocol [4]. This feature is as important as the element dictionary in allowing incremental improvement and delivering new results to users in a consistent, usable fashion. With SDDS we can add new data to the output without disrupting users and applications that use the output files. We can also make use of general-purpose pre- and post-processing tools that are not `elegant`-specific.

In what follows, we discuss recent improvements in `elegant` and some of the programs distributed with it. We'll begin by discussing improvements of a general nature, followed by a discussion of new features that are specific to ring modeling. Next, we'll summarize the status of on-going parallelization of the code, then turn to a discussion of recent changes to related programs. Finally, we will briefly review some recent applications of `elegant` and plans for future development. This paper covers changes starting with version 16.0 and ending with version 22.1.

GENERAL IMPROVEMENTS

Although `elegant` ("ELEctron Generation ANd Tracking") was written for electron tracking, repeated requests were made to allow tracking of other particles. The new `change_particle` command allows to user to choose different particles by name or specify the charge and mass of the particle of interest.

Optimization is an important feature of `elegant` and perhaps one of its strengths, compared to other codes. `elegant`'s optimizer uses a single penalty function that is the sum of many terms, each of which is specified as an expression by the user. Essentially anything the program computes, including intermediate and final results of tracking, can be used in an optimization term. New features in optimization include the ability to define optimization terms from templates, so that many similar optimization terms may be added without much effort. We've also added

* Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

† borland@aps.anl.gov

the ability to read optimization terms from an SDDS file, so that they can be generated by external programs or scripts.

elegant features several elements for modeling wigglers and undulators, including the new UKICKMAP element, which implements undulator kick maps [5]. A script was written and distributed with elegant that translates RADIA [6] kick map output into SDDS for input to elegant. The CWIGGLER element (a canonically integrated wiggler using code from Y. Wu [7]) was improved to use more general field expansions, such as those resulting from certain helical or vertically polarized devices. The CWIGGLER element now also includes classical and quantum synchrotron radiation effects when tracking.

Another element that involves an undulator is LSRMDLTR, which simulates a laser/undulator beam heater. This element was upgraded to include a time-dependent laser profile as well as synchrotron radiation.

A number of methods of simulating synchrotron radiation effects are provided in elegant. One of these is element-by-element simulation during tracking. By default, modeling of quantum effects uses Gaussian energy scattering of simulation particles [8], which might not be accurate for beams with very small emittance or energy spread. To allow examination of these, the CSBEND element now allows modeling synchrotron radiation using the energy and angle distributions for the emitted photons.

Of course, one can use tracking with element-by-element synchrotron radiation modeling to compute the beam properties along a beamline or even at equilibrium in a storage ring. However, a more efficient method [9] is propagation of the beam envelope using matrix techniques, including the damping and diffusion effects of radiation. This is now available in elegant, both for storage rings and transport lines, using the moments_output command. The results of these computations at any number of points in the lattice may also be subjected to optimization. One application of this is coupling minimization.

Another source of beam size and energy spread is intrabeam scattering (IBS), which can be modeled in elegant using the IBSCATTER element. The algorithm behind IBSCATTER has been improved to include the effect of vertical dispersion [10], to allow multipole scattering locations, and to allow modeling of IBS with acceleration [11]. The program ibsEmittance, which is distributed with elegant, includes the same changes and can be used for computing equilibrium properties in storage rings.

In order to model IBS along a beamline using IBSCATTER, it is necessary to insert many IBSCATTER elements in the lattice. To make this easy, we added the insert_elements command, which allows inserting a new element at multiple locations in a lattice without editing the lattice definition file. A companion command, replace_elements, is also new. It allows replacing existing elements with new ones. In both cases, the result can be saved as a new lattice file.

Another example of using insert_elements would be to insert many WATCH elements in a lattice in order to get phase-space dumps at many locations. In the past, this wouldn't work as expected because all the output files would have the same name, so that only the last occurrence would be retained. In addition, in modeling a large beamline one could easily attempt to open more files than the operating system allows. Hence, we improved the WATCH element to allow versioning of the filenames and to make use of the SDDS library's file disconnection feature to allow an essentially unlimited number of output files.

elegant models lumped-element beam pipe apertures of various types. Using insert_elements provides a new way to add apertures to an existing lattice at specific locations; e.g., one could insert the same aperture downstream of all quadrupoles with the same name. To further improve aperture specification, we've added the aperture_data command, which allows providing the beam aperture as a function of position along a beamline using an SDDS file. This file can, of course, be plotted together with loss distribution data or beam size data, using sddsplot.

When modeling errors in long transport lines, the simulated beam may be completely lost on the apertures before making a full pass. This makes trajectory correction using traditional methods problematical. To address this, we added two new trajectory correction methods. The thread method attempts to thread the beam through the system looking only at transmission. The one-to-best method pairs each corrector with the downstream BPM showing the largest response. Both methods attempt to imitate what might be done in early-stage commissioning.

Singular-value decomposition (SVD) is a standard technique for orbit and trajectory correction in situations where the beam is fully transmitted. elegant's correction algorithm has been updated to use SVD, including various methods of downselecting the singular values that are used.

elegant is used frequently as part of a procedure for routine correction of lattice functions in the APS [12]. We know that many of our gradient errors result from orbit offsets in sextupoles. Under such circumstances, the sextupoles act like combined function sextupoles and quadrupoles. For convenience in the correction procedure, we added the KQUSE element, which is a combined canonically integrated quadruople and sextupole magnet. This magnet can also be used in advanced storage ring designs that posit combined function magnets of this type.

STORAGE RING MODELING

A topic of considerable recent interest in the storage ring community has been the use of pulsed sextupole magnets to perform injection [13]. To support modeling of such concepts, elegant now includes the MBUMPER element, which simulates a time-dependent multipole kicker. The waveform for the kicker is supplied using an SDDS file.

No matter what injection method is used, having sufficient dynamic aperture (DA) is an important consideration in obtaining high injection efficiency. We've improved the aperture search algorithms in elegant to include multiline scans from the origin as well as a "smart" calculation of the aperture area that ignores lobes that may indicate an unreliable result. In addition, the DA area may now be optimized using elegant's built-in optimization methods. Another method of improving dynamic aperture is minimization of resonance driving terms [14]. The computation and optimization of these quantities is now included in elegant, so that they can be optimized along with other linear and nonlinear properties of the lattice.

Particularly in light source rings, coupling has an important effect on DA. elegant now supports computation of coupled lattice functions, based on Ripken's method [15], and allows these to be optimized. This could be used, for example, to correct coupling in the presence of errors.

DA is just one aspect of storage ring optimization. Equally important is the position-dependent momentum aperture [16], which determines the Touschek lifetime. This computation is now included in elegant. It can be used for storage rings, of course, but has also been applied to single-pass systems like energy recovery linacs (ERLs) [17].

One of elegant's strengths in ring simulations is modeling collective effects. This includes transverse and longitudinal short-range wakes and resonant impedances. The former are computed turn-by-turn, while the latter persist over many turns. An effective way to avoid spurious transients in such simulations is to ramp the impedance gradually from zero[18]. This feature has been added for transverse and longitudinal wakes, impedances, and rf modes.

Typically when modeling a ring with impedances, one needs to track a large number of particles for many turns to get reliable results. Hence, one cannot afford element-by-element tracking. At the same time, one needs to include higher-order transport effects, e.g., chromaticity or tune shift with amplitude, as these may provide damping. This can be done with the new ILMATRIX element, which stands for Individualized Linear MATRIX. This element can stand in for an entire storage ring or a superperiod, for example. The user specifies the periodic lattice functions, the tunes, and the momentum- and amplitude-dependent tune shifts. Synchrotron radiation effects can also be included with the (pre-existing) SREFFECTS element, providing a very fast simulation with all the essential features.

In addition to impedance elements, elegant can now simulate transverse space-charge kicks in a storage ring [19]. This is accomplished using the insert_sceffects command, which inserts a number of SCMULT elements. Each of these elements imparts an effective space-charge kick that simulates the effect of weak space-charge forces over the intervening distance from the previous element. Using insert_sceffects, it is trivial to vary the number of elements used in order to verify convergence.

PARALLELIZATION

With the increasing emphasis on multicore processors in laptops and desktops, parallelization is essential to the future of any simulation code. Pelegant, the parallel version of elegant, has been successfully run on dual-core laptops and 1000-core supercomputers. The status of parallelization is detailed elsewhere in this conference [20, 21].

Parallelization of elegant is being performed gradually, concurrent with on-going improvements to the serial version, while maintaining a single set of source code files. The initial approach was to parallelize only those elements that involve "embarrasingly parallel" operations, then gradually parallelize the elements that involve interprocess communication. The code is capable of switching between parallel and serial mode automatically as required, based on information in the element dictionary. This approach resulted in a very useful parallel version in about 6 months, which was put to immediate use. At present, just under 90% of the elements have been parallelized for multi-particle tracking. Optimization that involves tracking also makes use of parallel computation.

Originally, the master node handled all input/output (I/O) and performed particle scatter/gather operations as needed. As a result of I/O and memory bottlenecks, this approach was limited to about 60M particles (for 16 GB of RAM on the master node). A significant recent improvement was the addition of parallel I/O using the parallel SDDS library [21] and subsequent elimination of the central role of the master processor. This has allowed simulation with hundreds of millions of particles with significantly improved performance.

In addition to basic tracking of multi-particle beams, which involves particle-based domain decomposition, several other operations were recently parallelized. These include frequency map analysis, dynamic aperture searching, and momentum aperture searching [20]. We believe Pelegant is the first parallel code to offer these features, although the elegantRingAnalysis script [22] provides equivalent functionality.

RELATED PROGRAMS

The consistent use of SDDS files makes it easy to deploy elegant as a component of a larger application. Examples of this abound in the use of elegant along with other accelerator codes and free-electron laser (FEL) codes to perform start-to-end modeling for FEL light sources. One necessary component of such simulations is the ability to translate phase-space conventions among codes. Recent additions of this type include a pair of programs to translate between elegant and ASTRA [23] conventions, and another pair to translate between elegant and TRACK [24] conventions. Several scripts are available to translate IMPACT-T [25] output, including phase-space output, into SDDS.

Modeling the effects of coherent synchrotron radiation (CSR) is of course important in linac-based light sources, but it is also of interest in storage ring design. A convenient way to model CSR in a storage ring is to use the steady state CSR impedance with shielding [26]. This impedance can be computed and placed in an SDDS file ready for use with elegant using the new program csrImpedance.

Different aspects of synchrotron radiation are handled by the new programs sddsurgent and sddsfluxcurve, which augment sddsbrightness in providing computations of synchrotron radiation properties. sddsurgent provides computation of flux distributions and spectra using code from the programs URGENT [27] and US [28]. sddsfluxcurve provides computation of flux tuning curves using code from US. All of these programs take beam distribution data from elegant output files.

Another recent addition is touschekLifetime [17], which allows computation of Touschek lifetime using optics and momentum aperture data generated by elegant.

elegant's commandline interface and use of SDDS strongly supports script-based automatation of simulations. Along these lines, we have written the graphical user interface script elegantRingAnalysis [22], to provide a convenient interface to many storage ring computations. elegantRingAnalysis is designed to take advantage of a computing cluster, but can run on a single processor. elegantRingAnalysis uses sddsfindresonances, another recent tool, to find resonances in frequency map data.

SOME RECENT APPLICATIONS

In this section we briefly highlight a few recent applications, some of which use the new features discussed above.

Short-pulse x-rays in rings: APS has investigated the application of Zholents' scheme [29] for crab-cavity-based short pulse x-ray production from a storage ring. Originally [30], we used serial elegant, but the studies later benefited immensely from the parallel version. In particular, the optimization of sextupole to reduce vertical emittance dilution[31] and the exploration of the use of pulsed cavities[32] both benefited from rapid turn-around with the parallel version. Figure 1 shows an example of using Pelegant to optimize the results of one-pass tracking to minimize the vertical emittance growth while maintaining the desired chromaticity. elegant was also used to investigate this scheme for the Diamond Light Source [33]. Several other short-pulse schemes, one based on a vertical kicker [34], another based on rf phase modulation [35], and a third based on circulation of a short injected pulse [36], were also investigated with elegant.

Storage ring optimization: Optimization of dynamic and momentum aperture is a challenging aspect of storage ring design. Using elegant and Pelegant, we implemented several highly successful direct methods [37] of optimizing dynamic and momentum aperture based on tracking and genetic optimization. Figure 2 shows an example of frequency map analysis performed with elegantRingAnalysis on 100 processors for an optimized APS lattice with 10 long straight sections. Earlier, elegant was used to develop potential replacement rings for the APS[38, 39] as well as ultimate storage ring light sources [40]. elegant is also being used to optimize the NSLS-II lattice in the presence of strong damping wigglers using both direct optimization and minimization of resonant driving terms [14, 41], to model injection into the ring[42], and to investigate instabilities [43].

ERL design and modeling: elegant has been used extensively for design and simulation of ERLs, both at APS and elsewhere (e.g., [44, 45, 46, 47]). The APS group has produced several designs [48, 49] for possible upgrades. This included simultaneous optics matching for beams of multiple energies in the same beamline, as in Figure 3, which shows an optics solution for a two-pass 7-GeV linac. Tracking with Pelegant was essential in evaluating concerns about the microbunching instability [50]. Other elegant-based investigations at APS are Touschek scattering simulation and loss minimization[19], intrabeam-scattering simulation[11], optics correction[51], and x-ray compression [52].

FEL design and simulation: The recent success of the Linac Coherent Light Source (LCLS) at SLAC [53] has demonstrated the power of modern simulation tools, including elegant, to accurately predict the performance of future accelerators [54, 55]. elegant was used for development of the LCLS design and is part of on-going work to develop new operating modes for LCLS [56, 57].

The FERMI project has made use of elegant for a wide variety of design and simulation problems. This includes study of the laser heater [58] and related diagnostics [59], trajectory correction [60], jitter [61], beam instabilities [62], bunch length diagnostics [63], and the microbunching instability [64]. Figure 4 shows the evolution of a density modulation in FERMI, modeled with Pelegant.

Many other FEL projects are using elegant, including efforts in Korea [65], the United Kingdom [66, 67], Italy [68], Sweden [69], Switzerland [70], the United States [71, 72], Germany [73, 74], Japan [75], and China [76].

International Linear Collider (ILC): The ILC is a proposed next-generation electron-positron collider based on superconducting technology. elegant has been used in several aspects of ILC design, including the electron source [77], positron source [78, 79], bunch compression [80], and damping ring [19, 81].

FUTURE DEVELOPMENT

While there is no formal plan for future work on elegant, we anticipate that in the not-too-distant future the following enhancements will be made available. Users are encouraged to send suggestions for additional features.

1. Simultaneous parallel optimization of dynamic and momentum aperture, to allow use of the built-in simplex optimizer to perform storage ring nonlinear optimization. This should be more convenient than the existing method using a genetic optimization script.

2. Upgrading of the CSR algorithm to include shielding, non-relativistic beams, and multiple magnet effects, using the method of Sagan *et al*. [82]. Although the existing algorithm seems to correspond very well to experiments on LCLS, this upgrade will extend the validity to longer

bunches and lower energies.

3. Addition of higher-order wakes and long-range resistive wall wakes is desirable.

4. Inclusion of IBS in moments computations is desirable in order to have a self-consistent result including radiation effects and IBS.

5. Improved coupling correction, using cross-plane response matrices and vertical dispersion correction. At present, coupling correction can only be done using somewhat artificial methods, such as correction of the coupled lattice functions or moments, or using an external script.

6. Built-in lattice correction using LOCO [83]. At present, this is performed by an external script. With parallel resources, it should be possible to quickly perform the simulated response matrix measurement and correction.

CONCLUSION

`elegant` and related tools are under continuous, incremental development for linac and storage ring simulation. We have reviewed some of the many new features added in the last three to four years, as well as highlighted some applications. Details of the features discussed in this paper may be found in the `elegant` manual, which is available on-line [84]. An additional source of information and assistance is the on-line forum [85].

ACKNOWLEDGEMENTS

We appreciate assistance from Y. Wu in understanding his symplectic integrator code, which is used by the `CWIGGLER` element. We also appreciate the many suggestions and questions from `elegant`'s users, which contribute significantly to the on-going improvement of the program.

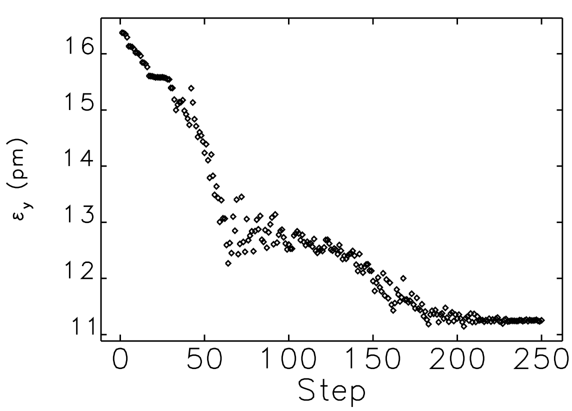

Figure 1: Vertical emittance optimization in the presence of crab cavities performed on 60 processors.

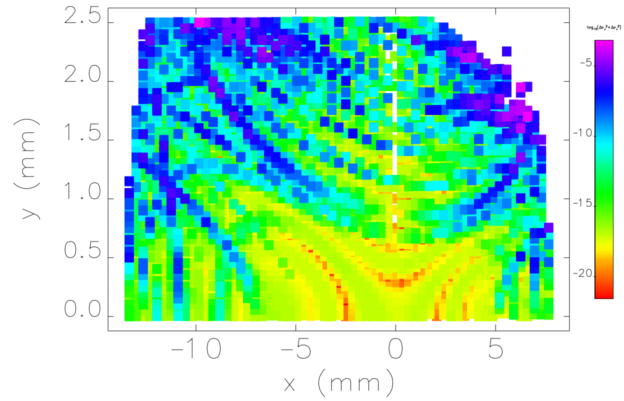

Figure 2: Frequency map analysis for a possible future APS lattice, using `elegantRingAnalysis` and 100 processors.

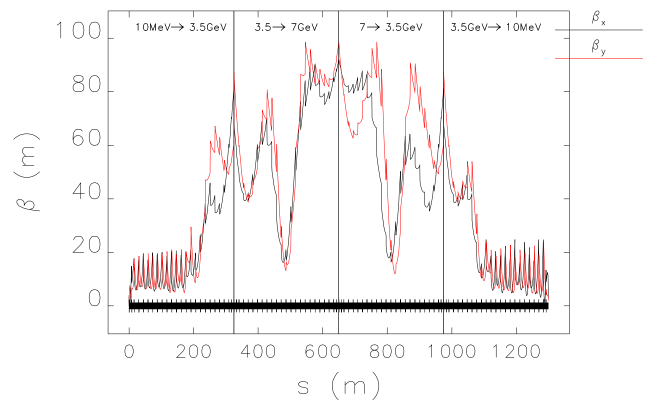

Figure 3: Two-pass energy recovery linac optics determined by simultaneous matching of four beams [49].

REFERENCES

[1] M. Borland, APS LS-287, September 2000.

[2] H. Grote and F. C. Iselin, CERN/SL/90-13(AP) (1991).

[3] M. Borland, Appendix A.2, SLAC-402 (1991).

[4] M. Borland, Proc. PAC95, 2184-2186 (2006).

[5] P. Elleaume, Proc. EPAC92, 661-663 (1993).

[6] P. Elleaume *et al.*, Proc. PAC97, 3509-3511 (1998).

[7] Y. Wu *et al.*, Phys. Rev. E **68**, 046502 (2003).

[8] M. Sands, SLAC-R-121, 1970.

[9] K. Ohmi *et al.*, Phys. Rev. E **49**, 751-765 (1994).

[10] F. Zimmerman, CERN-AB-2006-002 (2006).

[11] A. Xiao, Proc. Linac 2008, 296-298 (2009).

[12] V. Sajaev and L. Emery, Proc. EPAC02, 742-744 (2002).

[13] H. Takaki *et al.*, Proc. EPAC08, 2204-2206 (2008).

[14] J. Bengtsson, SLS Note 9/97 (1997).

[15] G. Ripken, DESY Report No. R1-70/04, 1970.

[16] M. Belgroune *et al.*, Proc. PAC03, 896-898 (2003).

[17] A. Xiao and M. Borland, Proc. PAC07, 3453-3455 (2007).

[18] M. Blaskiewicz, private communication.

[19] A. Xiao *et al.*, Proc. PAC07, 3456-3458 (2007).

[20] Y. Wang *et al.*, these proceedings.

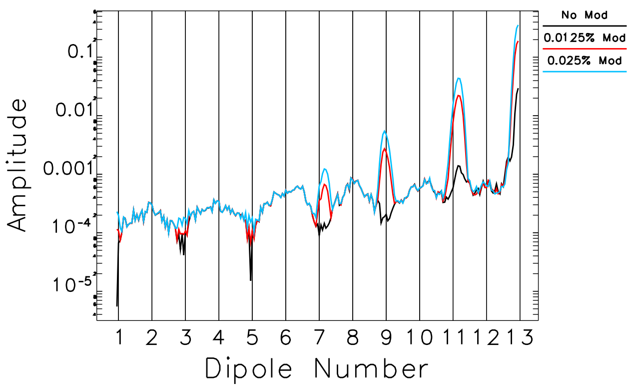

Figure 4: Modulation amplitude evolution in FERMI, for three initial levels of 25-μm modulation [64].

[21] H. Shang *et al.*, these proceedings.

[22] M. Borland, Proc. PAC05, 4200-4202 (2005).

[23] K. Flöttmann, www.desy.de/ mpyflo/Astra_dokumentation/

[24] V. N. Aseev *et al.*, Proc. PAC05, 2053-2055 (2005).

[25] J. Qiang *et al.*, J. Comp. Phys. **163**, 434 (2000).

[26] W. Warnock, SLAC-PUB-5375, 1990.

[27] R. Walker, private communication.

[28] M. Sánchez del Río and R. Dejus, SPIE Proc. **3448** (1998).

[29] A. Zholents *et al.*, NIM A **425** 385-389 (1999).

[30] M. Borland, Phys. Rev. ST Accel. Beams **8**, 074001 (2005).

[31] M. Borland and V. Sajaev, Proc. PAC05, 3886-3888 (2005).

[32] M. Borland *et al.*, Proc. PAC07, 3429-3431 (2007).

[33] R. Bartolini *et al.*, Proc. EPAC06, 160-162 (2006).

[34] W. Guo *et al.*, Phys. Rev. ST Accel. Beams **10**, 020701 (2007).

[35] G. Decker *et al.*, Phys. Rev. ST Accel. Beams **9**, 120702 (2006).

[36] X. Huang, Proc. PAC07, 1305-1307 (2007).

[37] M. Borland *et al.*, Proc. PAC09, TH6PFP062, to be published.

[38] A. Xiao *et al.*, Proc. PAC07, 3447-3449 (2007).

[39] V. Sajaev *et al.*, Proc. PAC07, 1139-1141 (2007).

[40] M. Borland, www.aps.anl.gov/asd/oag/presentations/borland-2008-06-03.pdf

[41] W. Guo *et al.*, Proc. PAC09, TU5RFP008, to be published.

[42] I. Pinayev *et al.*, Proc. PAC07, 1350-1352 (2007).

[43] S. Krinsky *et al.*, Proc. PAC07, 1344-1346 (2007).

[44] R. Hajima and R. Nagai, Proc. ERL07, 133-138 (2007).

[45] R. A. Bosch *et al.*, Proc. PAC07, 1287-1289 (2007).

[46] D. Angal-Kalanin, ERL09 Workshop, WG2, 138 (2009).

[47] N. Nakamura *et al.*, Proc. PAC09, TU5RFP084, to be published.

[48] M. Borland *et al.*, Proc. PAC07, 1121-1123.

[49] M. Borland *et al.*, Proc. PAC09, MO3PBI01, to be published.

[50] M. Borland, Proc. Linac08, 444-446 (2009).

[51] V. Sajaev, Proc. PAC09, TH6PFP007, to be published.

[52] V. Sajaev and M. Borland, Proc. Linac08, 447-449 (2009).

[53] P. Emma *et al.*, Proc. PAC09, TH3PBI01, to be published.

[54] H. Loos *et al.*, Proc. Linac08, 1095-1099 (2009).

[55] K. L. Bane *et al.*, Phys. Rev. ST Accel. Beams **12**, 030704 (2009).

[56] Y. Ding *et al.*, Proc. PAC09, WE5RFP040, to be published.

[57] D. Xiang and G. Stupakov, Phys. Rev. ST Accel. Beams **12**, 030702 (2009).

[58] S. Spampinati *et al.*, Proc. FEL07, 362-365 (2007).

[59] S. Spampinati *et al.*, Proc. FEL08, 342-344 (2008).

[60] S. Di Mitri *et al.*, Proc. EPAC08, 2993-2995 (2008).

[61] P. Craievich *et al.*, NIM A **604** (2009) 457-465.

[62] P. Craievich *et al.*, Proc. EPAC08, 1607-1609 (2008).

[63] P. Craievich *et al.*, Proc. DIPAC07, 168-170 (2007).

[64] M. Borland, Phys. Rev. ST Accel. Beams **11**, 030701 (2008).

[65] E. Kim and M. Yoon, Proc. FEL08, 125-127 (2008).

[66] P. Williams *et al.*, Proc. EPAC08, 145-147 (2008).

[67] C. Christou *et al.*, Proc. Linac08, 462-464 (2009).

[68] C. Vaccarezza *et al.*, Proc. EPAC06, 107-109 (2006).

[69] S. Thorin *et al.*, Proc. EPAC08, 139-141 (2008).

[70] Y. Kim *et al.*, Proc. Linac08, 483-485 (2009).

[71] M. Dunning *et al.*, Proc. PAC07, 1257-1259 (2007).

[72] A. Zholents *et al.*, Proc. Linac08, 501-503 (2009).

[73] E. Prat *et al.*, Proc. EPAC06, 1951-1953 (2006).

[74] M. Abo-Bakr *et al.*, Proc. FEL07, 220-223 (2007).

[75] H. Ohgaki *et al.*, Proc. FEL07, 390-393 (2007).

[76] Z. Liu, Proc. FEL06, 277-280 (2006).

[77] F. Zhou *et al.*, Proc. PAC07, 3127-3129 (2007).

[78] A. Drozhdin *et al.*, Proc. EPAC08, 2871-2873 (2008).

[79] W. Gai *et al.*, Proc. PAC09, WE6PFP068, to be published.

[80] E. Kim, Proc. PAC07, 2814-281 (2007).

[81] A. Xiao and L. Emery, Proc. PAC07, 3450-3452 (2007).

[82] D. Sagan *et al.*, Phys. Rev. ST Accel. Beams **12**, 040703 (2009).

[83] J. Safranek, NIM A **388**, 27 (1997).

[84] www.aps.anl.gov/asd/oag/manuals/elegant_latest/elegant.html

[85] https://www.aps.anl.gov/Accelerator_Systems_Division/Operations_Analysis/phpBB3/