

A MATLAB INTERFACE PACKAGE FOR ELEGANT SIMULATION CODE*

V. Smaluk, T. Shaftan[†], G. Wang, BNL, Upton, NY, USA

Abstract

A Matlab [1] interface package for Elegant [2] simulation code is under development. This package combines advantages of Elegant, which is one of the most advanced codes for accelerator simulations, with advantages of useful and effective Matlab functions for data processing, analysis, optimization, and real-time machine control using Matlab Middle Layer [3]. A number of functions have been already developed: calculation of lattice parameters and Twiss functions, linear and high-order chromaticity, amplitude-dependent tune shifts, modification of lattice elements, correction of betatron tunes and chromaticity, a set of functions for graphic representation. These functions have been successfully used at NSLS-II [4] for tracking and turn-by-turn simulations near the half-integer resonance, for maximizing tunability and dynamic aperture of NSLS-II Booster, and for calculating limits of top-up Booster energy interlock.

LATTICE PARAMETERS AND FUNCTIONS

A set of functions to calculate lattice parameters and functions using Elegant from Matlab environment has been developed. Input parameters and files for an Elegant run are prepared in Matlab scripts, then Elegant run is called using Matlab interface to the operation system. The elegant output data is imported back to Matlab from files or stream output.

A most universal function for calculation of lattice parameters and functions is `elegant_twiss`. This function requires a lattice file name, beamline name and beam energy as input parameters. There are also two Boolean parameters for optional calculation of radiation integrals and related beam parameters (emittance, damping times, etc.) and for calculation of resonant driving terms. This function calls Elegant with `twiss_output` command, reads the calculated parameters from the output file and stores them in a Matlab structure containing all the calculated data. This feature was found quite useful for comparison of several different lattices, because the output structures have the same fields and differ by their names only.

The structure contains the fields representing betatron tunes, chromaticities, central momentum, derivatives of beta functions with momentum offset, 2nd- and 3rd-order dispersion, coupling integral for difference resonance, distance from difference resonance, emittance ratio from coupling integral, 1st- and 2nd-order momentum compaction factor, radiation integrals (if requested), emittance, damping times and partition numbers, energy loss per turn. Also, there are array fields in the structure (for every element in the beam line): longitudinal position, name,

* Work supported by DOE under contract No.DE-AC02- 98CH10886.

[†] shaftan@bnl.gov

type, length, bending angle, K_1 , K_2 , K_3 , beta-functions, alpha-functions, betatron phase advances, dispersion functions and their derivatives. More fields are calculated inside the `elegant_twiss` function: circumference, revolution frequency, RF parameters, synchrotron tune, r.m.s. bunch length, synchronous phase and RF acceptance.

There are two functions for calculation of the 1st-, 2nd-, and 3rd-order chromaticity: `elegant_chrom_twiss` calls Elegant with `twiss_output` command and `elegant_chrom_track` calls Elegant with `track` command. The input parameters are: a lattice file, a beam line, synchronous energy, limits and step size for a scan of relative energy deviation $\Delta E/E$. The output includes 3 arrays: $\Delta E/E$, horizontal tune ν_x , vertical tune ν_y , and the 1st-, 2nd-, and 3rd-order horizontal and vertical chromaticities calculated by polynomial fit of $\nu_x(\Delta E/E)$, $\nu_y(\Delta E/E)$.

Since there is a recommendation in the Elegant manual to avoid the option of `twiss_output` command for calculation of betatron tune shift with amplitude, the Matlab function `elegant_tune_ampl` has been developed. This function scans oscillation amplitude calling Elegant with `track` command, the Elegant tracking output is processed in Matlab. The input parameters are: a lattice file, a beam line, synchronous energy, kick limits and steps for horizontal a_x and vertical a_y amplitude scans, number of turns for tracking. The output includes 6 arrays: a_x , $\nu_x(a_x)$, $\nu_y(a_x)$, a_y , $\nu_x(a_y)$, $\nu_y(a_y)$, and the detuning-with-action coefficients $\partial\nu_x/\partial J_x$, $\partial\nu_y/\partial J_x$; $\partial\nu_x/\partial J_y$, $\partial\nu_y/\partial J_y$ calculated by quadratic fit of $\nu_x(a_x)$, $\nu_y(a_x)$, $\nu_x(a_y)$, $\nu_y(a_y)$.

Matlab-Elegant interface functions have been also developed to calculate dynamic aperture, frequency map, momentum aperture and Touschek lifetime.

For accurate calculation, SBEN, QUAD, SEXT and WIGGLER elements are replaced by CSBEN, KQUAD, KSEXT and CWIGGLER ones in all the functions, which use tracking.

There is a function for modification of lattice elements (magnets, BPMs, etc.): `elegant_alter_elements`. This function changes a selected parameter of a list of elements using Elegant `alter_elements` command and writes the modified lattice configuration into a new file. Input parameters of the function are: name of the initial lattice file, name of the modified lattice file, name of beamline, synchronous energy, array of element names, string specifying a parameter to modify, array of the new values of the parameter.

Two functions have been developed to correct betatron tunes and chromaticity: `elegant_correct_tunes` and `elegant_correct_chrom`, respectively. These functions find proper settings of quadrupoles or sextupoles using corresponding Elegant commands and write new lattice configuration with corrected tunes or chromaticity into a new file.

Besides these functions used for lattice characterization and correction, a set of graphic functions has been developed to plot lattice functions, $v_{x,y}(\Delta E/E)$, $v_{x,y}(a_{x,y})$, etc.

SPACE CHARGE SIMULATION

A Matlab code has been developed for simulation of space charge effects on beam dynamics. A sample bunch distribution is generated in Matlab, and the space-charge impedance is calculated as well. Then, a “drift-kick” code is executed in Matlab and the output data is processed. The input data are generated from under Matlab for Elegant tracking with zlongit element. This element is used for simulation of particle-impedance interaction using a longitudinal space-charge impedance specified in a file generated by a Matlab function.

Energy-density distributions and their Fourier spectra calculated by the two codes are compared. Space-charge oscillation wavelength / phase are calculated. To plot the dependencies, a loop with various parameters determined the space-charge impedance $Z_{sc}=Z_{sc}(I, \sigma, \gamma, \Delta\gamma/\gamma)$: beam current I , r.m.s. bunch length σ , and Lorenz factor γ . Figure 1 shows an example of the space charge simulation.

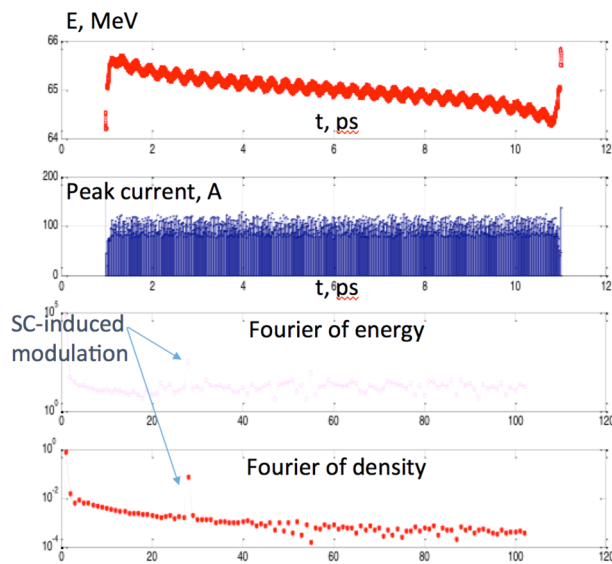


Figure 1: Space charge simulation.

BOOSTER QUADRUPOLE TUNABILITY AND DYNAMIC APERTURE

The Booster is a part of NLSL-II Injector, it accelerates the electron beam from the injection energy of 0.2 GeV up to 3 GeV and average beam current of 20 mA.

The Booster magnet lattice consists of four quadrants, each quadrant contains 8 combined-function defocusing dipole magnets, 7 combined-function focusing dipole magnets, 6 quadrupole magnets, and 4 sextupole magnets. The Booster quadrupoles are combined in 3 families: QF, QD, and QG.

As a part of preparation for the commissioning, optimization of the quadrupole settings has been done using a combined Matlab-Elegant code. The optimization procedure consists of the following steps. First, an Elegant

model with optimized linear optics is set up. Then, the strength of every quadrupole family (QF, QD, QG) is varying in a loop, and the lattice solutions are calculated by Elegant. Analysis of the data helps to determine a tunability range for each quadrupole family. This procedure is repeated iteratively with various lattice solutions until an optimum found.

Figure 2 illustrates the optimization procedure, it shows an example of quadrupole strength distributions (upper plots) and betatron tune diagram (lower plot).

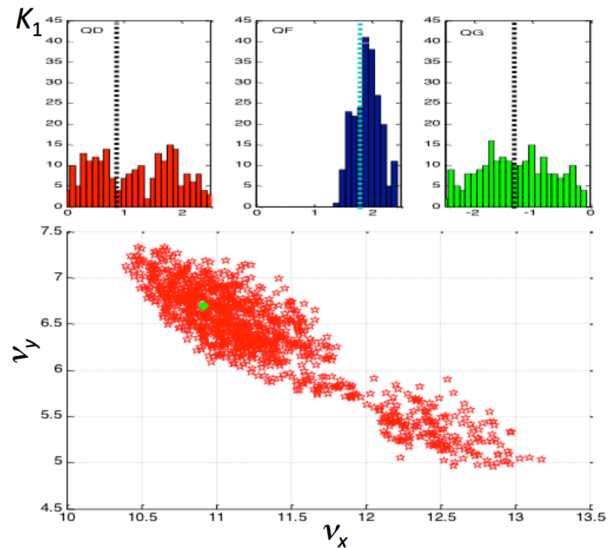


Figure 2: Quadrupole tunability optimization.

A similar optimization procedure has been developed to increase the Booster dynamic aperture (DA). The first step is set-up of Elegant model with optimized linear lattice. Then a 2D scan of betatron tunes is carried out using 2 out of 3 quadrupole families, dynamic aperture is computed by Elegant for every point, DA area is integrated in Matlab and plotted as a function of tunes (see Fig. 3). This procedure is repeated iteratively with various lattice solutions until an optimum found.

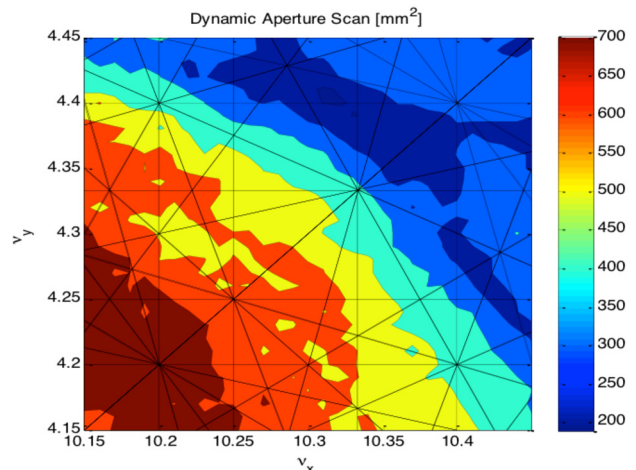


Figure 3: DA area as a function of tunes.

BOOSTER ENERGY INTERLOCK FOR TOP-OFF INJECTION

NSLS-II is operated now with top-off injection. This injection scheme provides a constant ($\pm 0.5\%$) beam current during user operation. For the commissioning of top-off injection, calculation of limits of Booster energy interlock is required. For safe operation, the current of dipole magnet power supply (PS) must not exceed a pre-determined window.

A combined Matlab-Elegant code has been developed to determine size of energy acceptance as a function of current of 3 dipole power supplies: BF, BD1 and BD2. Elegant is used for particle tracking, and Matlab is used for the problem set-up and output data processing. A Matlab function has been written to vary the PS current and scan particle energy as initial condition for tracking. Maximal and minimal initial energy give the energy acceptance that need to be constrained within $\pm 1\%$. If the dipole PS current is not constrained, Booster beam energy can exceed $\pm 2\%$.

Figure 4 illustrates a scan of 2 (BF and BD1) out of 3 dipole PS currents keeping the 3rd one (BD2) constant. In the figure, color is proportional to the energy deviation, and the footprint gives energy acceptance. White area corresponds to unstable lattice solutions.

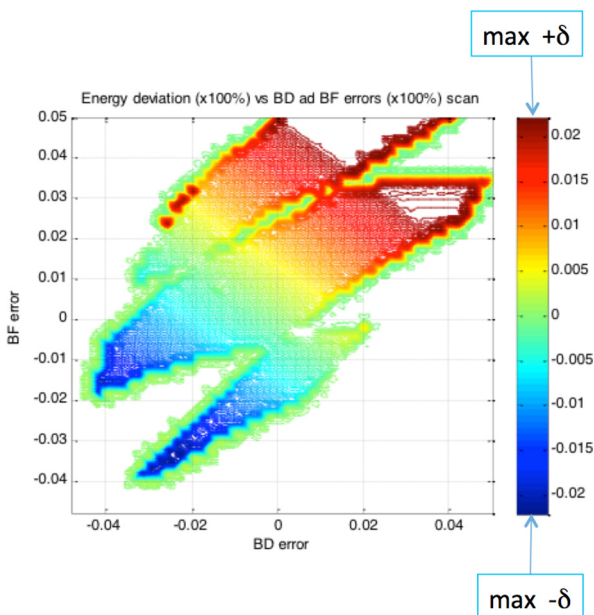


Figure 4: Scan of BF and BD1 PS currents.

ACTION-ANGLE CALCULATION

Analysis of betatron oscillations in action-angle variables was found very informative for studies of beam dynamics near half-integer resonance. To transform the turn-by-turn data from beam position monitors (BPMs), a combination of Matlab and Elegant function has been used.

Two watch points at the BPM locations with $180n+90$ degree phase advance were selected from a file produced by the `twiss_output` command of Elegant. These watch

points are inserted into the lattice using Elegant `insert_elements` command from Matlab. A tracking run is performed by calling Elegant with `track` command, the output is a set of turn-by-turn beam positions. The action and angle variables are calculated in Matlab with each BPM's position-angle and with paired BPM position.

Figure 5 shows an example of the action variable calculated for each and paired BPMs.

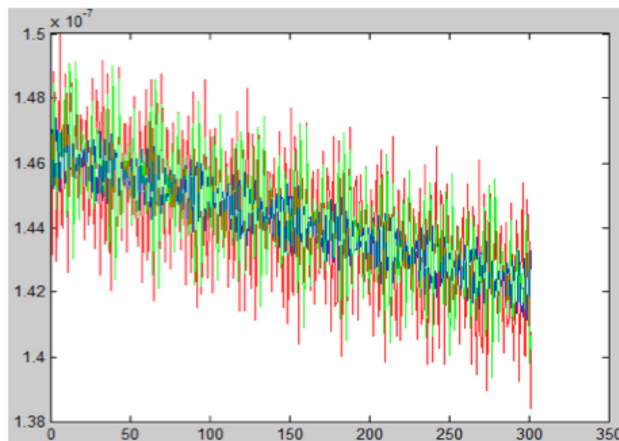


Figure 5: Action for each and paired BPMs.

REFERENCES

- [1] <http://www.mathworks.com/products/matlab/>
- [2] http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/manuals/elegant_latest/elegant.html
- [3] G. Portmann, J. Corbett, A. Terebilo, An Accelerator Control Middle Layer Using Matlab, Proc. of PAC-2005, Knoxville, 2005 (FPAT077).
- [4] T. Shaftan, "NSLS-II Accelerator Commissioning and Transition to Operations," presented at IPAC'16, Busan, Korea, 2016, WEPOY054, this conference.