A TOOL FOR LONGITUDINAL BEAM DYNAMICS IN SYNCHROTRONS*

J.-F. Ostiguy† and V.A. Lebedev, Fermilab, Batavia, IL, USA

Abstract

A number of codes are available to simulate longitudinal dynamics in synchrotrons. Some established ones include TIBETAN, LONG1D, ESME and ORBIT. While they embody a wealth of accumulated wisdom and experience, most of these codes were written decades ago and to some extent they reflect the constraints of their time. As a result, there is an interest for updated tools taking better advantage of modern software and hardware capabilities. At Fermilab, the PIP-II project has provided the impetus for development of such a tool. In this contribution, we discuss design decisions and code architecture. A selection of test cases based on an initial prototype are also presented.

INTRODUCTION

The longitudinal codes `ESME [1], LONG1D` [2], and `TIBETAN` [3] were developed in the late 1980s and early 1990s. They were implemented in `fortran`, and rely on the `namelist` mechanism for data input. `ORBIT` is somewhat more recent. It is principally a code for full 6D tracking with space charge that offers a limited longitudinal only mode. While some of the codes are still in use (e.g. `ESME` [4]), they are at best minimally supported and hard to maintain.

Recently, to respond to the needs of its LHC injector chain upgrade, CERN invested efforts in a new code (`BLOND`) which looks promising [5]. The implementation is mostly interpreted `python` with some computationally intensive functions in `C++`. The motivation for our own efforts was provided by the PIP-II project at FNAL. PIP-II involves replacing the existing warm 400 MeV linac with a new 800 MeV superconducting version which would serve as a platform for future improvements. A main goal is to deliver high intensity proton beams to a new generation of neutrino experiments. To do so, PIP-II requires various upgrades and improvements to the existing chain of Booster, Recycler and Main Injector synchrotrons. Some limitations in these machines are linked to longitudinal dynamics issues that need to be well-understood. In that spirit, after writing some narrowly focused code [6], it was decided that it would be beneficial to develop a more generic tool for longitudinal dynamics that we can fully control and adapt to our specific needs.

FEATURES

A considerable amount of experience and knowledge is encoded in the feature sets of the codes mentioned in the introduction. Much can be learned from them and they provide a template to define the functionality for a new code. `ESME`, for example, evolved over more than three decades and represents many man-years of work. A new implementation of an equivalent set of features is obviously a long term proposition. The initial priority is to implement the basic functionality which includes:

Magnetic ramp definition: provide a limited selection of predefined ramps and a provision for a user-defined ramp.

RF systems: single and multiple harmonic rf. Support for two simultaneous rf systems operating at different frequencies (required for for slip stacking)

Particle distributions: limited selection of predefined distributions. Ability to import and export distributions.

Impedance: predefined impedance models for resistive wall, pure inductive wall, "laminated" wall and low-Q broadband impedance. Provide a mechanism for user-defined impedance.

Narrow-band resonances : based on simple RLC model with wakes computed in the time domain.

Space charge: conventional space charge model based on analytical expression. Computed either in the frequency or the time domain.

gamma-t: control over phase-jump timing. Allow for non-linear and time-dependent momentum compaction factor (time-dependence is used to simulate a pulsed quadrupole γ_t-jump).

Monitors: losses, emittance, moments through the acceleration cycle. Line density for bunch shape evolution.

Basic beam feedback: Phase and radial loops. Dampers for supression of dipole and quadrupole oscillations.

IMPLEMENTATION CONSIDERATIONS

Implementation language(s) The code is primarily implemented in `C++`. This seems to be the best way to take advantage of the hardware capabilities; furthermore the language is popular, well-established and standardized. A subset of the functionality may be made available as python modules.

Parallelism A typical modern desktop supports 4 to 12 CPU cores and up to 128 Gb of memory. This makes it possible to simulate the complete acceleration cycle of a rapid cycling synchrotron with 10^5 macroparticles and space charge on a scale of minutes. Compiler support for `OpenMP` provides a straighforward and portable way to take advantage of multiple cores. A variety of parallelized opensource libraries (e.g. for FFTs) are also available and can be leveraged.

Interactive mode The simulation code is meant to be primarily used interactively. The ability to set up a problem quickly coupled with immediate user feedback makes work more efficient and promotes experimentation.

Batch mode The ability to run simulations in batch (console) mode remain useful and important for certain classes of problems. Such problems include for example, a parametric optimization or a high-statistics study of microwave instability after transition. The batch mode implies that simulations

* Operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

† ostiguy@fnal.gov

can be fully specified in a text file using a descriptive input syntax.

Plugins It should be possible for a user to extend or modify certain features through a plugin mechanism. Plugins are separately compiled modules that get dynamically loaded at runtime. They must adhere to a specific interface but otherwise may contain arbitrary code.

BASIC ALGORITHM

Differential Equations

Textbooks usually provide a simplified derivation of the longitudinal dynamics equations. A few subtleties tend to be overlooked and are worth mentioning.

Assuming one or more gaps providing a net periodic voltage V, the evolution –with respect to the synchronous particle – of a particle dynamical state is determined by the equations [7]

$$\frac{d}{dt}\left(\frac{\Delta E}{\Omega_s(t)}\right) = \frac{e}{2\pi}[V(\phi(t)) - V(\phi_s(t))] \tag{1}$$

$$\frac{d\phi}{dt} = -h(\Omega(t) - \Omega_s(t)) \tag{2}$$

ΔE is the energy deviation, h is the rf harmonic number, Ω is the angular circulation frequency and ϕ is the rf phase; the subscript s denotes quantities associated with the synchronous (reference) particle. The equations assume that changes in dynamical state are slow with respect to the circulation period. They also assume that the synchronous particle remains on a fixed trajectory throughout the acceleration cycle, as the guiding magnetic field is continuously ramped. Note in passing that in an actual machine, while the rf cavities deliver energy in discrete bursts, ramping is truly continuous.

A noticeable feature of equation (1) is the presence of Ω_s inside the differentiation operator. Technically, this is because $\Delta E/\Omega_s(t)$ and ϕ are the conjugate variables. Expanding the derivative and rearranging we get

$$\Delta\dot{E} = \frac{e}{2\pi}\Omega_s[V(\phi(t)) - V(\phi_s(t))] + \Delta E\frac{\dot{\Omega}_s}{\Omega_s} \tag{3}$$

Equation (3) suggests that the increase in particle energy is not exclusively due to the cavity(ies) – the first term on the right hand side. The second term actually accounts for the betratron acceleration, or more precisely the difference in betraton acceleration with respect to that experienced by the synchronous particle. The betatron acceleration arises from the time varying magnetic flux enclosed by the particle path. In a synchrotron, the synchronous orbit circumference is fixed and

$$\frac{\dot{\Omega}_s}{\Omega_s} = \frac{\dot{\beta}_s}{\beta_s} \tag{4}$$

which indicates that at each turn, as a fraction of the energy gained from the cavity(ies), the energy due to betratron acceleration is equal to the relative change in velocity. While the betatron contribution is small, ignoring it violates symplecticity. For a detailed discussion, see [8] and [9].

Map

For numerical simulation purposes, a spatial map is needed. A suitable one is obtained by integrating the system (1-2) over a complete turn i.e. over a period $T_{s,n}$. One then gets at step n

$$\frac{\Delta E_n}{\beta_{s,n}} = \frac{\Delta E_{n-1}}{\beta_{s,n-1}} + e\frac{[V(\phi_n) - V(\phi_s)]}{\beta_{s,n}} \tag{5}$$

$$\Theta_n = \Theta_{n-1} + 2\pi\frac{\Delta\Omega_n}{\Omega_{s,n}} \tag{6}$$

where Θ is the azimuthal position along the synchronous trajectory and $\phi = -h\Theta$. The frequency slippage $\Delta\Omega_n$ is caused by differences in velocity and path length with respect to the synchronous particle; both depend on the relative momentum deviation and one has

$$\frac{\Delta\Omega}{\Omega_s} = \frac{\Delta\beta/\beta_s - \Delta C/C_s}{1 + \Delta C/C_s} = \eta(\frac{\Delta p}{p_s})\frac{\Delta p}{p_s} \tag{7}$$

where Δp is the momentum deviation and C is the path length (circumference). Both $\Delta C/C_s$ and $\Delta\beta/\beta_s$ may be expanded as power series in $\Delta p/p_s$. The velocity deviation expansion is obtained analytically; the coefficients of the orbit length expansion (usually refered to as α_n) depend on the dispersion and can be computed using a separate optics code. At first order, (7) yields the familiar expression for the slip factor $\eta(\Delta p/p) = \eta_0 + \eta_1\Delta p/p_s + \cdots$

$$\eta_0 = \frac{1}{\gamma_s^2} - \alpha_1 \tag{8}$$

The higher orders terms become important at transition ($\eta_0 = 0$).

Collective Effects

Collective effects are incorporated by adding localized voltage kicks each representing the integrated effect of the collective fields. At fixed time, collective fields depend exclusively on the spatial distribution; the latter is obtained by projecting the phase space distribution. The collective voltage may be computed either in the time or frequency domain. In the frequency domain, the calculation assumes that the ring distribution is quasi-periodic; the voltage is obtained by multiplying the impedance by the periodic extension of the longitudinal distribution and represents a steady-state solution. If the fields persist for more than a turn, it becomes necessary to store the response over multiple turns and this introduces a significant amount of complexity. Fortunately, the most relevant practical case of persistent wake is that of narrow-band resonant modes in rf cavities. A narrow resonance can be modeled by a simple passive RLC circuit. The impulse response of the circuit (a damped sinusoid) can be expressed in terms two variables, the capacitor voltage and the inductor current. These variables completely characterize both the current state and the future response of the circuit; therefore at turn n the response is the transient response due to the excitation during the current turn plus the response associated with the decaying fields from the state at $n-1$. In practical terms this implies that two floating point numbers are sufficient to account for all past history. The caveat is that the calculation must be done in the time domain.

EXAMPLE

A substantial fraction of the functionality outlined above has been implemented into what might be best described as an advanced prototype dubbed `LDYN`. A first public version, is expected in the coming months. The basic tracking algorithm is based on the map defined by (5-6). The collective effects are incorporated in a conventional way, as previously described. The basic capability and performance of the code (in interactive mode) is best illustrated with an example involving a complete cycle of the Fermilab Booster synchrotron. 10^6 particles are tracked for 20,000 turns; on a 6-core, 3 GHz CPU, the simulation runs in approximately 15 mins. The total beam charge was set to 5.0×10^{12}e. The left hand side of Fig. 1 shows the voltage profile for the cycle. On the right hand-side is a plot of the magnet "laminated wall" impedance. The latter is accounted for, together with the space-charge impedance. Fig. 2 shows the beam phase space distribution at different key moments: injection, capture, shortly below and above transition and final energy. The plots on the right-hand side of the phase space display are from top to bottom: the line densities in azimuth and energy deviation, the beam induced voltage and the beam spectrum. Of note is the high induced voltage at transtion, which reaches over 200kV. Fig. 3 shows the rms bunch size and bunch emittance evolution during the cycle. The small initial bump in the emittance plot is due to uncaptured (and eventually lost) particles. Note that, as expected, while β varies from 0.71 to 0.97, the emittance remains constant before transition once the distribution has reached steady-state. In this simulation, feedback was applied to limit both dipole and quadrupole oscillations. Transition crossing causes the rms emittance to nearly double, from an initial value of about 0.015 eV-s (rms) due to filamentation. The core emittance is on the order of 0.1 eV-s.

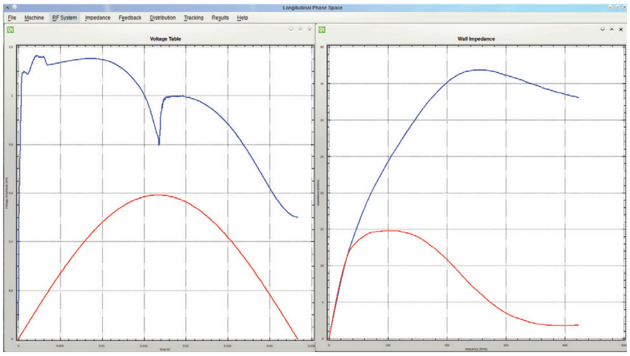

Figure 1: Voltage program (blue : total, red: acc.). The peak voltage is slightly above 1.1 MV.Left: wall impedance (blue: real, red: imag.).

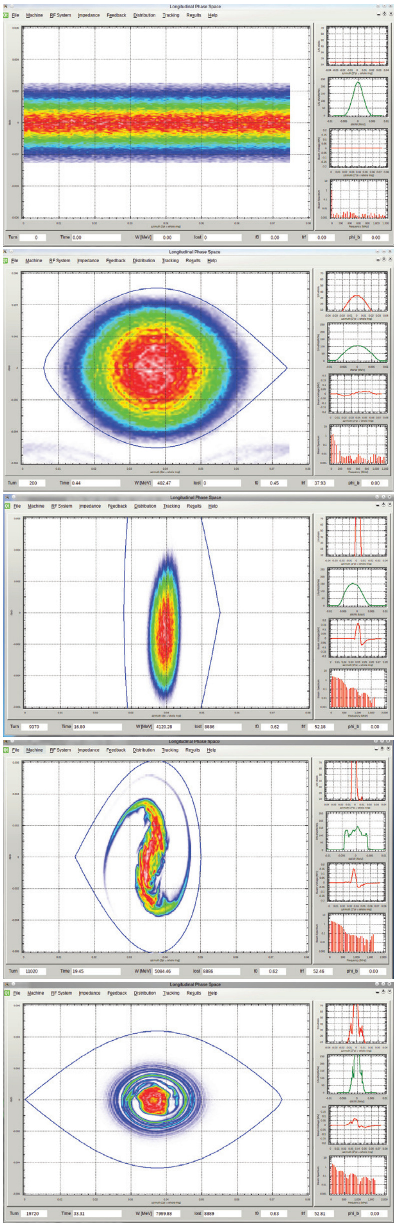

Figure 2: Phase space distribution. Top to bottom: initial, after capture, near (below) transition, above transition, final energy.

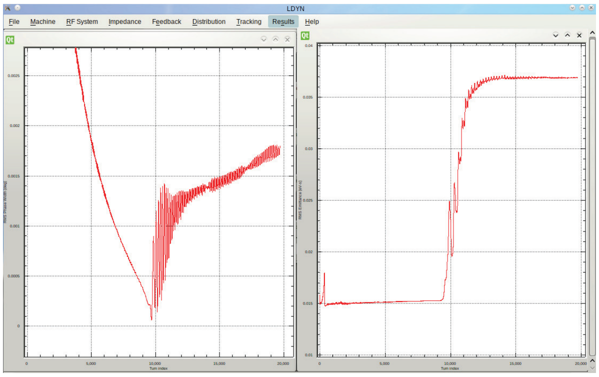

Figure 3: Beam moments evolution during a Booster cycle. Left: rms phase extent. right: rms emittance.

REFERENCES

[1] J. MacLachlan, "Particle Tracking in E-Phi Space for Synchrotron Design & Diagnosis", 12-th Int'l Conf. on Appl. of Acc. in Res. and Ind., Denton TX, 4 Nov 1992.

[2] S. R. Koscielniak, "The `LONG1D` Simulation Code", TRIUMF Report TRI-PP-88-41.

[3] J. Wei, "Longitudinal Dynamics of the Non-Adiabatic Regime on Alternating-Gradient Synchrotrons", Ph.D. Thesis, SUNY/Stony Brook (1990).

[4] ESME Web Site: `http://esme.fnal.gov`

[5] H. Timko et al. "Benchmarking the Beam Longitudinal Dynamics Code BLOND", in Proc. of IPAC16, Busan, Korea.

[6] V.A. Lebedev et al., "Beam Acceleration and Transition Crossing in the Fermilab Booster" in Proc. of HB2016, Malmo Sweden.

[7] E.D. Courant and H.D. Snyder Ann. Phys., 3., 1-48 (1958).

[8] H.G. C Hereward, "What are the Equations for the Phase Oscillations in a Synchrotron" CERN PS report 66-6.

[9] P.J. Bryant K. Johnsen, "Lagrangian/Hamiltonian Derivation of the Phase Equation", CAS and NIKHEF-H, Amsterdam, CERN-1992-001, pp. 51-61.