

# Studies of the stability of the Dispersion Free Steering method

Paul Lebrun

December 15, 2006

## Abstract

The Dispersion Free Steering (DFS) method is studied in the Tesla Linac. Although obsolete, this linear accelerator is very similar to either the second stage of the International Linear Collider (ILC) bunch compressor, or the ILC Main Linac. The CHEF tool-kit is used and has been bench-marked against Merlin. Although most of the calculations have been done on a static lattice, algorithms discussed here are required to convergence towards a stable solution. It is shown than in absence of cavity tilts (rotations on the YZ plane), for the seed studied at least, DFS provide a unique and stable solution with negligible emittance growth. This is true in the static limit, independently of the BPM unknown offsets and the BPM resolution. If cavity tilts are about 200 to 300 micro-radian, the DFS solution is no longer unique and significant emittance occurs as well. In presence of perturbations such as ground motion, these DFS solutions will be more difficult to stabilize. Also, such DFS solutions do not allow us to determine the BPM scale factor and offset with good accuracy. Therefore, it is suggested to consider using movers on quadrupole/BPM assemblies, and the support system for the cavities. Detailed justifications for these movers are given.

The preservation of the transverse emittance from the damping ring to the interaction point (I.P.) has long been recognized as a critical item in the list of ILC challenges[1]. In this note, the Dispersion Matched Steering (DMS) is studied in detail. If the design Dispersion is zero, this is known

as Dispersion Free Steering (DFS). Since the Tesla Main Linac design was straight, DFS will be used. However, the new code will handle the more general case, DMS, as the Baseline for the ILC Main Linac is curved and has a typical Dispersion of 1.4 mm.

While the basic implementation for Linacs has already been simulated in various frameworks, it also has been shown tricky to implement in reality. The path towards robust Beam Based Alignment (BBA)<sup>1</sup> methods is tedious but necessary. One must add to our simulation codes numerous parasitic effects, such as beam jitters, instrumentation defects. Moreover, these beam based steering techniques will operate on an imperfect machine and are literally "moving target" problems: Ground motion must be taken into account while handling an imperfect beam typical of the commissioning phase.

Traditionally, low emittance transport are studied in two distinct phases: first, the beam is "statically" steered through a physically misaligned machine, ignoring all dynamical effects. In the second phase, ground motion, technical noise and other time dependent effects are studied. The response of feedback systems are simulated and optimized. The distinction between the "static" and "dynamical" simulation is conceptually easy and saves computing time: "static" features can be modeled more simply when the much more CPU demanding dynamical studies are considered. However, a static steering algorithm might not be of practical value once dynamical effects are considered. While the study of such dynamical effects is far from over, this note explores the possible pitfall of the DFS algorithms when the static solutions are not unique and a bit unstable. The hallmark of this new simulation framework is its ability to handle both static and dynamical problem at once, with a potentially more faithful representation of the real machine.

Most of the calculations presented here have been done in the CHEF[2] framework. This code has not been used for Low Emittance Transport (LET) studies prior to this work. Hence, we will first present CHEF-utility code and benchmark results, where Merlin[3] recent results are compared to those obtained using CHEF. Merlin was the code used by the Tesla collaboration. The second section is devoted to the tedious, albeit necessary, description of the BBA algorithms. The third section emphasizes stability issues and the importance of cavity tilts in the low energy ( $< \approx 20$  GeV/c) section of the Main Linac or in the second stage of the bunch compressor. In the next

---

<sup>1</sup>This acronym is often used for Ballistic Beam Alignment. Apologies for the confusion!

section, results of the performance of DFS algorithm in presence of beam jitter and ground motion is studied, as well as tentative fixes.

## 1 Relevant CHEF/Merlin Benchmarks

As Merlin, CHEF is a C++ tool-kit, i.e., a collection of class and libraries for Accelerator Physics simulation. Both frameworks started more than a decade ago and are based on the same ideas and same principle. They evolved along parallel but distinct paths. CHEF has been used to simulate proton synchrotron (Fermilab Booster, Tevatron and CERN-PS) basic design, while Merlin has been mostly used in the context of the Tesla and ILC collaboration. Since the tracking section have somewhat similar designs, and they share the same language, it has been possible to compare and even exchange bits of code between the two frameworks, allowing a detailed studies of the beam physics in both packages.

More recently, new interfaces to CHEF and other codes were build to support detailed studies of collective effects, such as space charge. [6]. Such collective effects have to first order little to do with emittance preservation in a Linear Collider. (However, wake fields in r.f. structures, that are highly relevant for the ILC, can be considered as a collective effect!). The cost of introducing this new LET code, i.e., LET problem benchmarking, is small compared to the advantages CHEF/Synergia had to offer: strong local support and the ability to resolve software integration issues. For instance, because of the clean architecture of CHEF, we will be able to run jointly the Geant4 framework and CHEF, to be able to simulate energy deposition in the liquid Helium due to beam halo induced by wake field excitations, which are partly driven by structure misalignments. Software Integration is the hallmark of the Synergia project and is key to success for LET problems.

### 1.1 Applying the CHEF to the ILC LET problems

There are two ways of working with CHEF: either as C++ (or Python) class library, or as a pre-build executable that includes a GUI. The later provides immediate and easy access to relatively common problems, while the former allows for in-depth comparison of codes and algorithms, and build upon the framework with new beam physics. These two methodologies do not exclude

each others. This second approach was chosen, mostly because additional functionality was required, such as wake field on control steering code. In addition to the mandatory “user-written” main program, the following C++ classes have been written:

- **myPositronBunch** Derived from the CHEF class **PositronBunch**, this container class offers methods to generate the single bunches of macro-particles matched to a particle cell in a lattice, and to compute emittances. A macro-particle is usually just a positron, except when considered by the wake field process, where it’s electric charge is multiply by the realistic total charge of the bunch (ILC bunches contains  $\approx 2^{10}$  particles) divided by the number of simulated macro-particles in this bunch.

Throughout this work, transverse emittance are defined and computed in this class as follow. They are solely based on the second moments of the distribution of macro-particles. That is, the normalized vertical emittance is:

$$\varepsilon_y = \gamma * \sqrt{\sigma_y^2 * \sigma_{yp}^2 - \sigma_{y,yp}^2}$$

where  $\gamma$  is the Lorentz boost factor,  $\sigma_y$  is the 2nd order moment <sup>2</sup> of the vertical position distribution. Similarly,  $\sigma_{yp}$  is the 2nd order moment of the  $Py/|P|$  distribution, where P is the macro-particle momentum.  $\sigma_{y,yp}$  is the off-diagonal 2nd order moment, i.e.,

$$\sqrt{\sum (y - y_0) * (yp - yp_0) - y_0 * yp_0}$$

The normalized vertical emittance corrected for first order dispersion,  $\varepsilon_{yND}$  is obtained in a similar way, except the contributions due to correlation between the vertical position and the momentum is taken out. Same for the correlation between  $Py/|P|$  and momentum  $|P|$ . The linear correlation coefficients between momentum and these transverse phase space coordinates are computed. New vertical coordinates are defined, where this correlation is exactly canceled out. Lastly, the

---

<sup>2</sup> $\sigma_y = \sqrt{\sum (y - y_0)^2 / (n - 1)}$ , where  $y_0$  is the centroid, n is the number of macro-particle

normalized vertical emittance on these new vertical coordinates is computed as described above.

Positron bunches generated by the Merlin package can also be read and propagated by CHEF, ensuring that the initial condition on the beam are the same when comparing performances<sup>3</sup>

Although the analysis and plots are done separately from this CHEF-user package **myPositronBunch** also provide a small data visualization method to represent the 2D vertical or horizontal phase space distribution, as a 2D color coded bitmap.

- **SteerUtil** This is a set of LET utility class to:
  1. Misalign the machine based on flat ASCII file with beam element displacement and/or rotations. Such misalignments can be “dynamical”, e.g., they can be changed prior to propagate a bunch or a particle through the affected element(s), on a pulse by basis, without having to re-instantiate any classes, or re-initialize CHEF.
  2. Set or adjust dipole correctors. Also manage the list of broken correctors.
  3. Allow the ground to move, interfacing to the ground motion package (described later, not used in the benchmark!)
  4. Control the voltage on the cavities.
  5. Contain the beam parameters for matched bunches.
  6. Send pilot bunches (one macro particles, neglecting wake field effects) or real bunches down the Linac, collect BPM data. Optionally, compute emittances. The class keeps track of the elapsed time, as one can also send multiple bunches separated by 200 ms (the ILC repetition rate is 5 Hz). Since long range wakes are currently not yet implemented, only one bunch per klystron pulse (or “train”) is simulated.

---

<sup>3</sup>Merlin uses a slightly different 6D phase space coordinate system, beside a trivial re-ordering of the components. In Merlin, the second coordinate on the horizontal or vertical plane is  $y' = P_y/P_z$ . Although it make very little difference at positron energies of 5 GeV, proper conversion from one system to the other has been implemented.

The functionality of this class is currently too broad, one will have to think splitting it into a dynamical geometry class, control, and beam operation.

- **WakeFieldProcess** and ancillary classes. These classes have been lifted from the Merlin[3] package and adapted to the CHEF core data structures. Wake fields do not have a big impact on proton machine, so this beam physics got never implemented in CHEF. The port of these class from Merlin to CHEF has been checked looking at macro-particle distribution prior to study does not have this The **WakePotential** and **cavWakes** class reads the tabulated wake field potential functions[7]. These functions depend on the exact cavity shape and have been recalculated over the years. For sake of simplicity and consistency, the benchmark described below is using the “old” 2003 Tesla wake field value. These values are overestimated by  $\approx 20\%$ . However, one expect more higher gradient cavity design, where the wake field will be more pronounced. Making conservative assumptions while benchmarking is wise.

Once these functions are known<sup>4</sup>, they are used along with the macro-particle positions to compute the electromagnetic field applied to each particle in the bunch. This field is computed by slicing the bunch longitudinally (100 slices, by default) and estimating the slice centroids and width.

- **MLBPMs** Main Linac Beam Position Monitor simulation class. As the bunch or single particles are propagated in the lattice, and such beam line element are recognized by the CHEF framework, response of the BPMs are simulated and recorded. A bit of terminology is in order: the BPM response  $y_m$  as a function of the true vertical position  $y$  of the particle, or bunch can be written as:

$$y_m = bpm_o + bpm_s y + GaussRandom(0., \sigma_b)$$

where  $bpm_o$  is the BPM position offset,  $bpm_s$  is the BPM scale (the deviation from unity is often referred to as the scale error). The resolution

---

<sup>4</sup>They are assumed to be constant for a given cavity type. Fluctuations in cavity shapes are ignored, as well as higher order effects. This is not a true, self-consistent calculation.

function is approximated by a Gaussian, with a r.m.s. of  $\sigma_b$ . Most of the simulation have been done assuming  $bpm_s = 1.$ , and  $\sigma_b < 0.1 \mu m$ . The former assumption assume that the BPM calibration can be done accurately enough prior to the installation of the BPM, and does not change. The latter assumption implies that, in the static limit, we run as many pulse as needed such that the random fluctuations due to BPM resolution is no longer a concern.

Only the vertical BPM are currently working in the simulation. Extending the code to the horizontal is obviously trivial, it hasn't been done since the focus of the study is the preservation of the vertical emittance, ignoring XY coupling terms

The Higher Order Mode r.f. absorbers are not yet simulated. Such devices could be of great values if the cavity position and tilts (rotation in the YZ plane, called "pitch" in CHEF terminology) can be adjusted during BBA running.

The similarities in design and the common language (C++) in Merlin and CHEF allowed the exchange of propagator algorithms. For instance the quadrupole propagator from Merlin has been successfully inserted in CHEF, in order to diagnose possible differences. It was found that the small subtleties in modeling the integration of the equations of motion through these optical elements do not drive the emittance growth.

## 1.2 Emittance Growth Benchmark on the Tesla Lattice

This benchmark has been proposed at the February 2006 LET meeting at CERN, and the results from other codes have been presented at the Vancouver meeting[4]. A corresponding note is being written. For sake of completeness, results on Merlin and CHEF are also given here. Relevant properties of the Tesla Main Linac are summarized on table 1. Not the ELoss for the cavities is crudely simulated in CHEF: we simply re-adjusted the nominal gradient in such a way that the average energy for the bunch matches the one obtained running Merlin. Further detail, such as the quadrupole setting in the matching section between the high and low energy section are provided in the MAD file that can be obtained from the ILC/LET benchmark web site[8].

The lattice is static but misaligned, and a list of best value for the corrector settings were submitted to the Merlin and CHEF code. The LET performance for this problem are shown on figure 1. While numerically and statistically significant differences are obvious, the dependency of  $\varepsilon_v$  on S is very similar. The relative emittance growths do agree with  $\approx 15\%$ . However, there are significant caveats to this encouraging result:

1. While the LET performance shown on figure 1 is within our emittance budget, it should be noted that the first 9 cryo-modules (108 cavities, covering  $\approx 150$  m) are assumed to be perfectly aligned. The reason behind this “cheat” will become apparent when the DFS algorithm is explained.
2. Among other simplification, let us mention the absence of XY coupling, either in the inject beam, or in the lattice. Also, no long range wake field since these are single bunch simulations.
3. On Cavity Rotation: The S coordinate of the axis of rotation was never clearly specified in the files we received. In CHEF, this axis of rotation can be located anywhere along the beam element being rotated. For lack of a better idea, early and unsuccessful CHEF calculations were done assuming the axis of rotation is located in the middle of the cavities. Much better performance is reached assuming it was located at the end of the structure. This is documented on figure 2
4. On Dipole Corrector Settings: Significant improvements to the DFS algorithm implemented in the TAO package[9] were implemented while benchmarking results were collected, resulting in a new set of dipole corrector values. This two sets of dipole corrector values are compared on figure 3. Unfortunately, the performance of this solution is not as good as the one obtained before, as shown on figure 4

Obtaining different solutions that give similar performance is not necessarily a good thing: it could be an indication of unknown sensitivity to various parameters in the problem and lead to operational uncertainties. Let us stress that the “solution” of the DFS problem is a set of corrective measures, such as a list of dipole corrector settings, or, if movers are available, as set of physical displacements or rotations of specific beam line elements. At

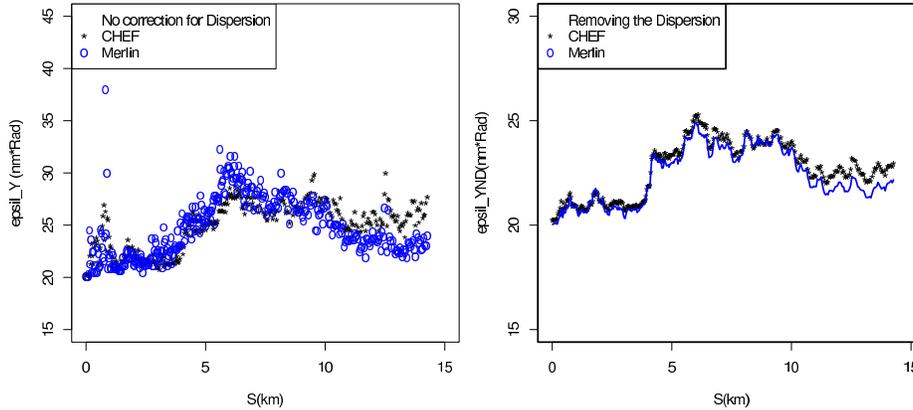


Figure 1: Transverse Emittances  $\varepsilon_y$  versus distance for the Benchmark2 problem, using the dipole corrector settings found in the file “nick23p4\_misxy\_ycor\_1.txt”, distributed in February 2006.

the risk of being overly pessimistic, in this narrow sense, our DFS code on this benchmark problem don’t really agree, as given corrector setting, locally, can differ values by as much as 100%. At this point, such issues can not be understood without a full implementation of DFS in CHEF.

### 1.3 Dispersion Free Steering Benchmark, Merlin vs CHEF

This benchmark consists of deducing the best (i.e., smallest vertical emittance growth) set of dipole corrector settings for the misaligned Tesla lattice described above. As the other codes don’t always agree of what these setting should be for a given set of misalignments, it is clear that multiple solutions to this problem exists. Realizing this, instead of implementing the simple, “static” version of DFS, a bit more software has been written in the context of CHEF, such that such seemingly minor differences in various approaches can be studied, including DMS, and with or without dynamical effects. This software is described in more details in the next section. Here, a somewhat simplified version of the DFS algorithm is used The recently

Table 1: Relevant properties of the Tesla Main Linac to the LET problem

Machine length	14,277.095 m.
Two sections	Low/High Energy ( $E_l < 115 < E_h$ GeV )
Injection Energy	5 GeV
Exit Energy	252.18 GeV
Acceleration	
Cavity Length	1.03 m
Cavity Gradient	23.4 MeV/m
Phase, first 12 bet. cells	5 deg ahead of crest
Phase, other	27 deg behind crest
Number of cavities per Cryo-modules	12
Number of cavities per Klystrons	24
Focusing system	
Lattice type	FODO, 60 degree betatron phase advance
Quadrupole Length	0.666 m
Quad Strength low energy sect.	0.045231
Quad Strength high energy sect.	0.03033
Approximate distance between quads	38 m.
$\beta_{max}$ , low energy sect.	115.57 m.
$\beta_{max}$ , high energy sect.	172.07 m.
Total number of quads	355
Total number of vertical correctors	355
Total number of BPMs	355
Twiss Parameters at injection	
$\beta_x$	89.309 m.
$\beta_y$	50.681 m.
$\alpha_x$	-1.451 m.
$\alpha_y$	0.873 m.
Beam Parameters	
$\varepsilon_y$	20 nanometer-radiant
Bunch length	300 micron
$\Delta_P/P$	2.78 %
Misalignments	
Aver Quad. vert. displacements	350 $\mu$ m
Aver Cavity. vert. displacements	350 $\mu$ m
Aver Cavity. Tilt	200 $\mu$ radiants

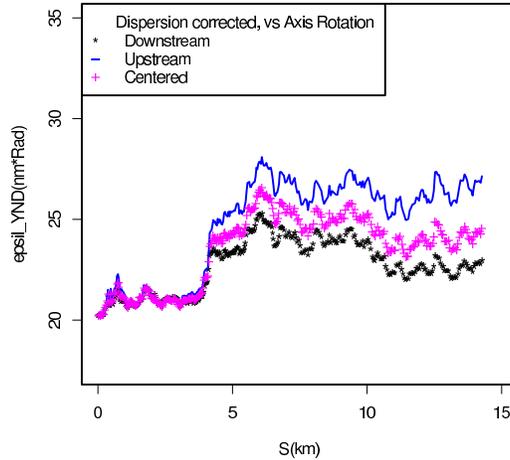


Figure 2: Transverse Emittances  $\varepsilon_{yND}$ , corrected for Dispersion, versus  $S$  for the dipole corrector settings labeled “Merlin” in the previous figure, for three assumptions on the longitudinal location of the cavity rotation  $YZ$  axis.

written DMS/CHEF code has been tuned to reproduce the relevant features of the Merlin example “ILCDFS” [5]. This Merlin example has been “reversed engineered” into the CHEF Beam Steering package, after most of the ChefSteering package has been written.

Among the relevant modification or simplifications made on this ChefSteering code, let us mention:

1. Response matrices<sup>5</sup> are computed once and are based on the perfectly aligned machine.
2. Dipole corrector setting for a given section of the lattice are computed once, and applied with a corrective gain of 100%. No iterations are performed. No proofs of convergence are given.
3. Perfectly static problem, no beam jitter.
4. BPM resolutions are exquisite (nanometers) in both Merlin and CHEF.

---

<sup>5</sup>see next section

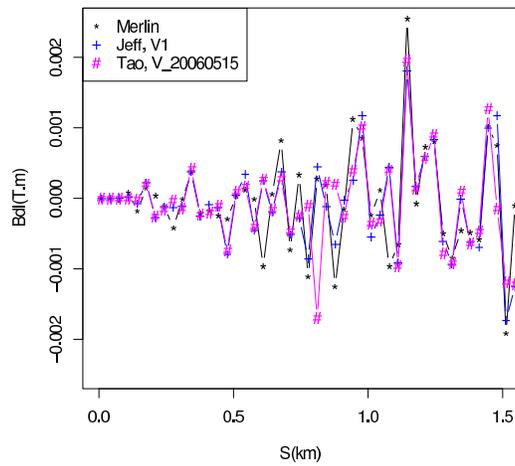


Figure 3: The Dipole Corector settings in the beginning of the misaligned Tesla Linac, as computed by the preliminary DFS-LIAR, and as adopted by the Merlin proponents (labeled Merlin on the figures), as found by two releases of the DFS-Tao program.

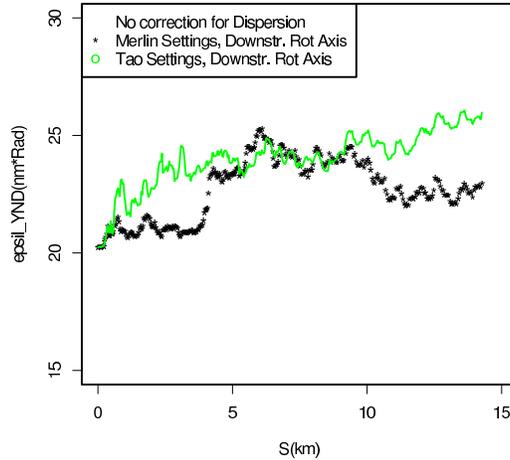


Figure 4: Transverse Emittances  $\varepsilon_y$  versus distance for the Benchmark2 problem, using the dipole corrector setting found by the Tao program, latest release.

5. In DFS, one needs to run an upstream section of the Linac at a reduce accelerating gradient. Both CHEF and Merlin code ran with a gradient reduction of 20%, upstream and in the DFS section being steered.
6. A DFS section comprises 40 quadrupoles/dipoles. The overlap between section is 20 quadrupoles.
7. Both codes are not minimizing the Dispersion, strictly speaking. As shown later, in presence of cavity rotation, such a “pure DFS” algorithm is unstable. Instead, a combine merit function that includes orbit deviation and Dispersion is minimized. The later term is spoiled by uncertainties in the BPM offsets. The former is in principle what “pure DFS” should do. The best set of relative weights orbit vs DFS has been used for both frameworks, corresponding to an effective trajectory error of 360 microns and a Dispersion error of  $\approx 14$  microns.
8. Only iteration per DFS section, with a corrective gain of unity.

As shown on the previous benchmark,  $\varepsilon_v$  grows relatively rapidly in the

first few misaligned section, then settles down as the Dispersion is being corrected. There is quite a bit of disagreement between codes on how fast the  $\varepsilon_v$  spikes up at  $S$  0.85 km. Since our BBA algorithms are sequential and always start from the beginning of the lattice, might as well focusing on the first one or 2 km. The mechanism for emittance growth later on could be different, such small lattice mismatches between subsequent Main Linac major sections. We'll get to them in due course. Thus, results shown on figure 5 are limited to  $\approx 2$  km. Note that, unlike the previous benchmark, matched bunches are generated by each package separately, causing some fluctuations in the starting emittance. The agreement in the projected emittance, growth, uncorrected for Dispersion, for this critical section is far from perfect. Moreover, as feared from the previous benchmark, the dipole corrector settings obtained by this simplified version of DFS differ substantially. Although there is good agreement in the range of the correction to be applied, the sign of the correction can change locally<sup>6</sup>. This starts to occur at  $S \approx 0.6$  km,  $\approx 2$  betatron wave length, when the  $\varepsilon_v$  rises quickly.

The emittance growth presented by the Merlin authors at EPAC06 for the ILC Main Linac are definitely smaller than those shown on figure 5. This is because this Linac starts at 15 GeV instead of 5 GeV, and with a momentum spread of only one percent. Spurious Dispersion effects on the transverse emittance do vanish if the momentum spread vanishes. Improving both the agreement between the various DFS implementations and, as important, the LET performance - requires a better understanding of the root cause of these emittance growth.

## 2 DFS algorithms and code in CHEF

### 2.1 Algorithms

Dispersion Free Steering (DFS) is a Beam Based Alignment technique to be used when the all beam line element offsets, BPM in particular, are unknown. One relies of a know momentum difference to measure the Dispersion function. If this Dispersion has to vanish at all BPM location, then this technique does not rely on the knowledge of the BPM scale factors, as one attempts to zero-out the Dispersion. The steps for a basic, “pure and simple” DFS are:

1. Define the section -the DFS section - of the Linac where the beam is

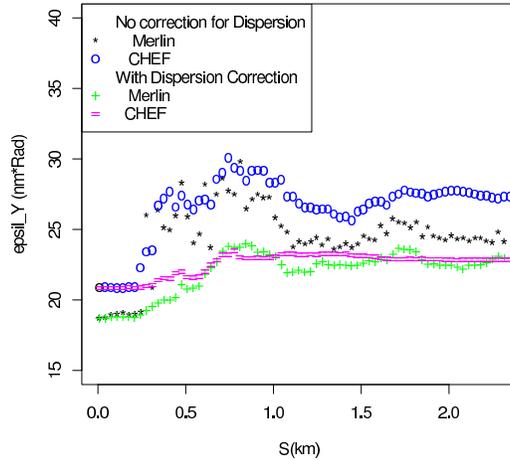


Figure 5: Transverse Emittances  $\varepsilon_y$  and  $\varepsilon_{yND}$  versus  $S$  found by the DFS benchmarks ILCDFS, when applied to the old Tesla Linac design, and the CHEF version of the same (or nearly the same!) DFS algorithm.

being steered. That is, the section where the dipole corrector setting will change as a result of the steering process.

2. Run the beam through the Linac, including the DFS section, at nominal acceleration gradient. Record BPM readings in this DFS section.
3. Run the beam through the Linac at a reduce accelerating gradient. Record the BPM values. Compute the Dispersion, assuming the BPM scales and cavity voltages are known.
4. Repeat the process changing dipole correctors, one at a time, throughout the DFS section. This allows us to compute a Response Matrix, or, a “transfer function”, dipole corrector settings to Dispersion at given BPM.
5. Compute the pseudo-inverse of the Response Matrix, and apply the required correction to minimize the Dispersion.

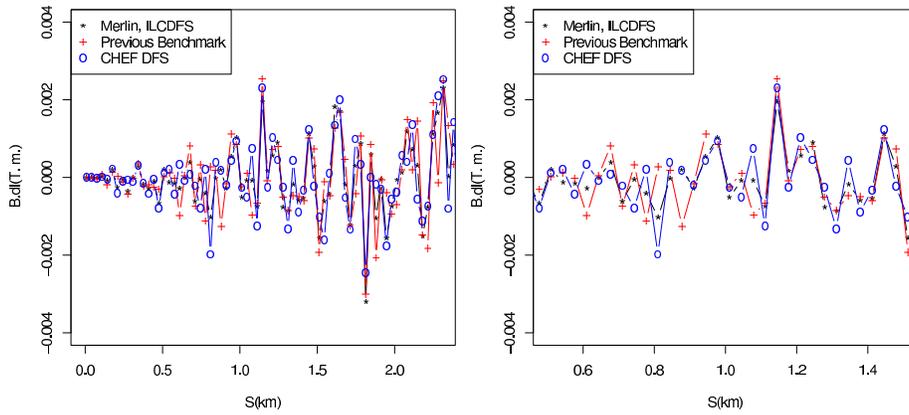


Figure 6: The DFS solution corresponding to the performance shown above, i.e., the Dipole Corrector Settings. Also shown are the dipole corrector setting found for the same misaligned lattice found about in February 2006 and used in the previous benchmark. Left and right plots have the same data, longitudinal scale  $S$  is expanded on the right.

6. For a given DFS section, repeat the process if needed. In Merlin, and the above benchmarks, the above steps are done only once.
7. Move to subsequent DFS section. The DFS section can overlap each other.

In practice, in addition to minimize Dispersion, it is best to also minimize beam displacements throughout the Linac. Since the uncertainty in the BPM offsets are dictated by well known mechanical tolerance, these offsets are not completely unknown. This means a contribution to the Response Matrix can be computed based on the BPM readings at nominal accelerating gradient. This is usually refer to as the "displacement" or "orbit" term. The relative weight - or errors - assigned to each BPM and to each Dispersion measurement is tuned for best LET performance.

Numerous DFS variants have been proposed. The CHEF implementation now supports:

- Matched Dispersion Steering (DMS) vs DFS: The Response Matrices (RM) are computed based on the designed Dispersion, obtained via standard Linear Optic calculation. Thus, support for the ILC curved Linac is provided. The above benchmark for a straight Linac, target Dispersion of zero everywhere.
- DFS sections: The length and overlap between successive DFS section are both independently adjustable.
- The RMs can either be computed for:
  1. Different machine alignment state. The RM can be computed for the perfectly aligned machine, or conversely, after the beam-line elements are displaced or rotated, and at each iteration within a DFS section, as the relative position of the beam with respect to the beam line elements changes.
  2. For a realistic bunch, including beam offsets, jitter, with finite emittance. Or, conversely, for a virtual bunch, with zero emittance and no simulated wake field. (In simulation, a BPM can record the position of a single positron).

The above Benchmark used RMs corresponding to a perfectly aligned Linac and with single positrons.

- The RM scale factor. To compute the RM, one has to gauge the response from an actuator change, i.e., propagate a bunch or a positron down the Linac with/without a change in a given dipole corrector setting. If the problem is linear, the RM values and the final outcome of the optimization do not depend on the amount by which the value of the actuator changes. However, if non-linearities are suspected, this is no longer true. Thus, an RM scale factor parameter has been provided.
- Iteration within a DFS section. In order to study the robustness of the algorithm, and in real life where operational care matters, it is rarely recommended to apply 100% of the suggested correction. Instead, only a fraction of it (the “corrective gain”) is applied to the actuator. One then iterate, applying successive corrections. Convergence is reached if two successive changes to any actuator (in this case, Dipole Correctors) are smaller than a prescribed number. In our case, it is expected that the anticipated average of the dipole correction strength increases with momentum if the quadrupole position tolerance stay constant over the length of the machine. Thus, the successive change of the dipole corrector setting  $\delta_{cd}$  are renormalized at the injection energy of 5 GeV, such that this parameter is uniform across the DFS sections.
- Multiple iterations over the entire lattice. That is, after moving from one DFS section to next and reaching the end of the Linac being studied, we can start all over again, retaining the set of dipole corrector values. The two distinct motivations for such macro-iterations are:
  1. Perhaps a 2nd, or third pass can improve the performance for the static problem. It could be true if changes are made upstream of the DFS section.
  2. More importantly, in case of time-dependent defects, e.g., the “dynamical” problem, one has to go back to the beginning and check that the solution did not changed too badly, by going back and re-doing the problem.
- The Energy reduction scheme. While either computing the RMs or measuring the corresponding Dispersion function, one has to decide

where along the Linac the accelerating gradient is reduced. We do not have complete flexibility here since multiple cavities are powered by a single klystrons. However the number of contiguous klystrons with a reduced gradient upstreams of the DFS section, or in the DFS section can be freely adjusted. Note that the energy can be also reduced in the DFS section itself.

- Extra BPMs: In principle, additional information can be gain from the BPM downstream of the DFS section. The RMs are in general non-diagonal, an arbitrary number of such responders can be used.
- Pre-Steering option. Prior to DFS steer in a given section, it can be useful to make sure that the incoming beam trajectory parameter  $y_0$ ,  $y'_0$  entering the DFS section does not depend on the bunch momentum. The last three dipoles of the previous, or upstream, DFS section are used to steer the beam injected to the current DFS section in such a way that the beam centroid  $y_0$ ,  $y'_0$  are identical with running the Linac at reduced gradient.
- Crude Visualization. Both core C++ code in CHEF and Merlin have no provision for analysis nor visualization. In CHEF, this is done via the GUI and the Synergia[6] Python interface. However, since LET jobs are CPU intensive tasks, interactive options are not practical. Thus, some simple 2D color plots representing the vertical phase space at BPMs are optionally generated.
- Support for missing dipoles, broken BPMS, etc.. Successful runs with such machine defects are not always possible for all algorithm or variations of them.

Emphasis is placed on robustness studies of the DFS algorithms. The goal is to study the stability of future feedback systems. If enough information about impeding motion or problems, feed-forward control systems will be used. Again, instead of writing two distinct packages, one for “static” steering and a second one for “dynamical” studies, a single set of classes have been prototyped to support both goals.

## 2.2 Code Organization

In addition to the utility classes described above, a new set of objects and free-standing functions<sup>6</sup> have been written:

- **ResponseMatrix.** Since we plan to study many BBA algorithms, **ResponseMatrix** is actually a C++ template. A list of algorithms is provided, each corresponding to a set of actuators. An actuator can be a dipole corrector, a quadrupole mover, a gismo that tilts a cavity and so forth. Currently, the only responders are the BPMS, but that can be extended. The dimension of these matrices are the number of constraints (or actuators) times the number of BPM in a given DFS section. Since most of these algorithms are related to a momentum change along the Linac, a **setEnergy** method must be provided for each instance the **ResponseMatrix** template. These RMs can be generated and computed at any stage of the steering or re-alignment of the lattice. In the most successful “static” implementation, they are computed for the perfectly aligned lattice. Thus, to save CPU time, utilities to dump these matrices to files has been provided.
- **DMSPlan** In addition to these response matrices, additional parameters for the steering algorithm must be provided, such as the length of the DFS section, the overlap between section, the number of cavity per klystrons and so forth. Instead of building very long argument lists, it is best having a container for such parameters. A detailed list of the data member is outside the scope of this document. Optionally, attached to this Steering plan is the required collection of RMs created from the perfect lattice.
- Top level Steering Functions These are free-standing functions that can be called from the Main program. For sake of brevity, only one of them is described here. The other in the package are pre-prototypes.
  1. **SteerWeightedDMS.** Takes basically a DMS plan as argument. Loops over the required macro-iteration, and for a given macro-iteration, over the DFS section. If the problem is “dynamical”, for each pulse (fixed ILC rep rate is 5 Hz), the initial bunch is

---

<sup>6</sup>All members of the namespace **Steering**

recreated based on a random “jitter” value for  $y_0$  and  $y'_0$ . In between each such pulse, each active beam line element is moved based on ground motion parameters.

Finally, an interface class, **GroundMotionInterf**, to connect the ground motion simulation sub-package to the pre-existing utilities in **SteerUtil** has been written.

### 3 Results for the front-end of the Tesla Linac

Although this design is now obsolete, this lattice basically has the basic features of the ILC Baseline<sup>7</sup>. The number of cavities per cryo-module is different, but the number of cavities per klystrons is the same. The curvature and the higher accelerating gradient are the recent modification to the original Tesla design. Both make the LET task a bit more difficult, but this is mitigated by the higher injection energy 15 instead of 5 GeV<sup>8</sup> and the reduced momentum spread. For historical reasons, this was our starting point. Yet, it is still a good start to study the robustness of DFS.

The DFS performance shown on figure 1 has a complex dependence on S. The agreements among the various codes on the spikes in the projected emittance which occurs about 1.5 to 2 betatron wavelength after the first misalignment, where the energy is still relatively low ( $\approx 18$  GeV), remains poorly understood. This is not a bad place to start investigating.

#### 3.1 On the Robustness of static DFS

The DFS solution for the benchmark described above is not unique. For a given set of misalignment, one can have similar performance, but different corrector currents. To see this, let us follow a phenomenological approach, playing with the parameters of the DFS algorithm.

---

<sup>7</sup>Winter 2006 version

<sup>8</sup>This is related to the cavity tilt issue

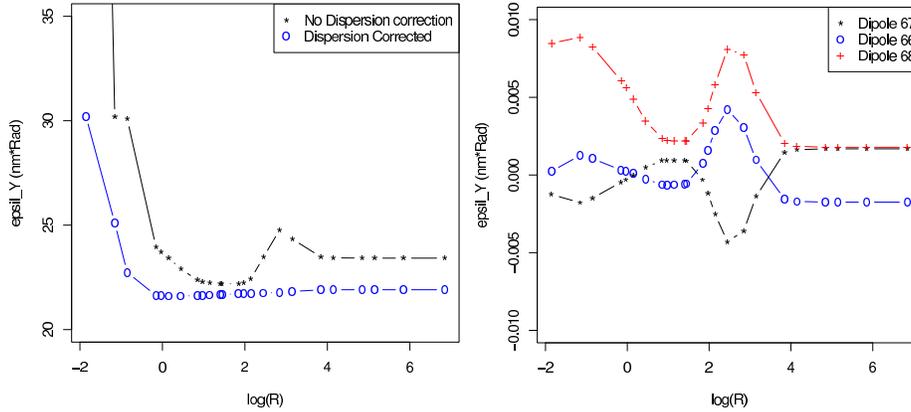


Figure 7: On the horizontal axis is the log of the ratio  $R$  between the weight of the Displacement term and the Dispersion.  $R \rightarrow \text{inf}$  corresponds to the pure DFS algorithm. Left graph shows the performance, i.e., the vertical emittance at the end of the last DFS section, and the right graph shows the setting for three dipole correctors.

### 3.1.1 On the relative weight between Dispersion and Displacement

The performance of a weighted DFS/Displacement  $\epsilon_v$  is shown on figure 7, versus the ratio between the weight of the Dispersion and the weight of Displacement. The dipole corrector settings for the three dipoles in the last DFS section are also shown.  $\epsilon_v$ , left uncorrected for Dispersion, is minimum for a finite ratio Dispersion/Displacement. Settings for Dipole 67/68, located at  $S = 2.28\text{km.}$ , change signs not too far away from this optimum, while  $\epsilon_v$  corrected for Dispersion stays flat. If this quantity is our merit factor, it is no surprise that different codes give different sets of dipole corrector values.

### 3.1.2 Other features of the DFS algorithms

Furthermore, three simple exercises using the above benchmark case document the sensitivity to the DFS solution to the DFS parameter.

1. The Dependency of the performance on the corrective gain and the

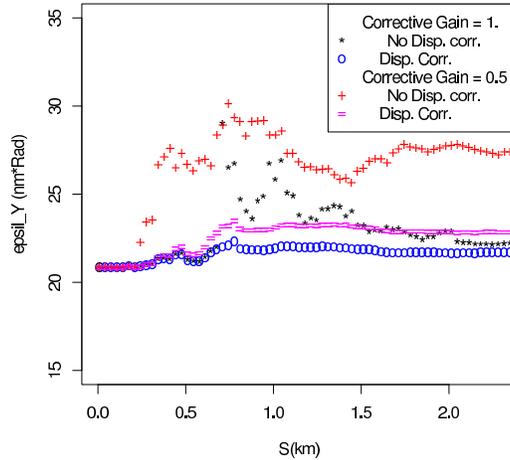


Figure 8: The DFS performance - the emittance vs S - for different corrective gain factors.

number of iteration within a given DFS section is illustrate on figure 8. The criterion for convergence is  $\delta_{cd} = 2.5 \cdot 10^{-7} \text{ T.m}^9$ , at 5 GeV. The good news is that we reach convergence relatively quickly, approximately 10 steps for a gain of 0.5. Much shorter, 1 or 2 steps if the gain is one. The bad news is that we actually got a different, less performant, solution than just run with a gain of 1.0, one and only one iteration. The casual optimist might think reaching the “right” answer in one step is neat. However, just confirming the solution, i.e., run a few more iterations change changes the answer significantly, as shown on figure 9.

## 2. Dependency on the length of the DFS section.

This dependency is documented on figure 10. In this case, a corrective gain of 0.5 was chosen. Convergence is reached in about 10 steps per DFS section. Smaller DFS section lengths require a few more steps. In all cases. the overlap is half of the length.

---

<sup>9</sup>Or 0.025 Gauss for a 10 cm. long corrector. This is a small field, one order of magnitude lower than the earth field!

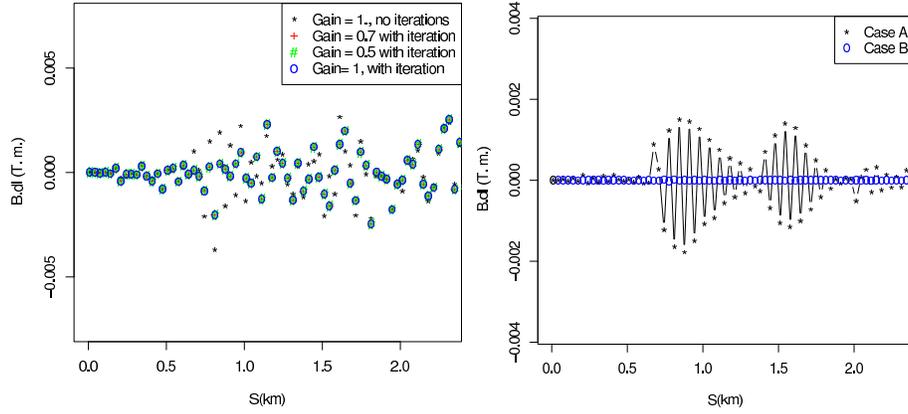


Figure 9: The DFS solution for difference corrective gain factors. Left is the actual value of the corrector setting, right is the difference between (A) Gain of 1, for one iteration vs letting iterating until convergence is obtained (B) Gain of 0.5 vs 1.0, with iteration in both cases.

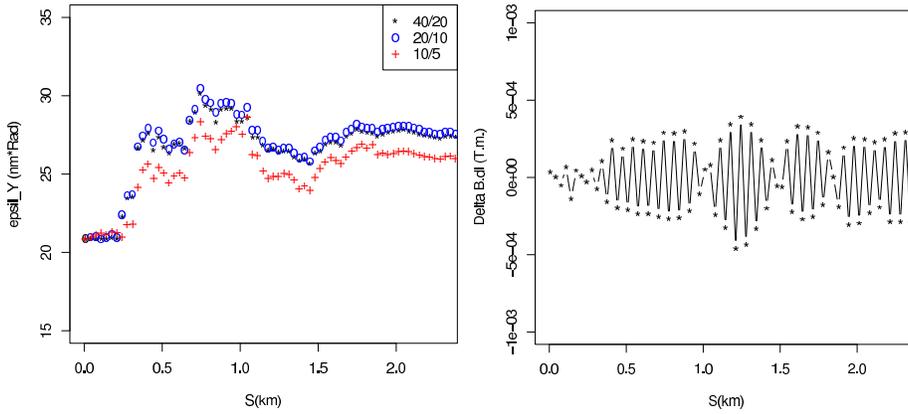


Figure 10: Left plot: the DFS performance for three different section lengths. Convergence at  $\delta_{cd} = 2.5 \cdot 10^{-7} T.M$  is reached in all cases. Right plot: and the difference in the solution between the 20/10 and 10/5 section lengths.

3. Response Matrices accuracy: In the above cases, the RMs are computed based on the perfectly aligned machine (“perfect RMS”) The knowledge of the lattice function is therefore implicitly assumed with infinite accuracy. Two distinct approaches to make the simulation more realistic can be considered: quadrupole strengths errors are introduced<sup>10</sup>, or, conversely, we recompute the RMs, on each iteration (“imperfect RMs”). In the later case, the RMs are measured after the machine is assembled. If the corrective gain is set to a small value, the dipole setting are found to be equal to those obtained with perfect RMS, within a few percent. Performance are virtually identical. This is also true for a corrective gain of 1. Other codes found that better performance were obtained with perfect RMs than imperfect ones[10]

We have clearly established that multiple solutions to the problem can be found. This of course based on only one set of misalignment. However, this set was chosen randomly, so it is very unlikely that this set is peculiar. In the static limit, this is just a feature. However, it does confuse the beginner, and might create operational confusion.

### 3.1.3 Towards dynamical simulation.

So far, only one macro-iteration was considered. That is, we went through the Linac once, DFS section after DFS section, from upstream to downstream, and called it done. Now, let us turn one dynamical effect: and compare solutions and performance after going through a few of these macro-iterations. The RMs are the perfect ones, computed only one time at the beginning of the job. The DFS section length is set to 40, overlap 20. For each DFS section, let us iterate until convergence with  $\delta_{cd} = 2.5 \cdot 10^{-7} T.M$ . The corrective gain of 0.5. For each such iteration, we send 20 pulses, each bunch jitters by 1 micron ( $\approx 0.1\sigma$ ) in vertical offset, and 0.2 micro-radian for  $y'$ .  $y$  and  $y'$  are uncorrelated. The DFS performance and solutions are shown of figure 11 and figure 12. Differences in emittance growth are observed, corresponding to a few % relative differences in dipole settings. These results are encouraging since no large jump in the solution space occurred in these three iterations. However, this simulation took place in less than one minute, real time, as one requires about 20 pulses (4 seconds) per DFS section. Note also that

---

<sup>10</sup>Errors on distance between optical elements are negligible

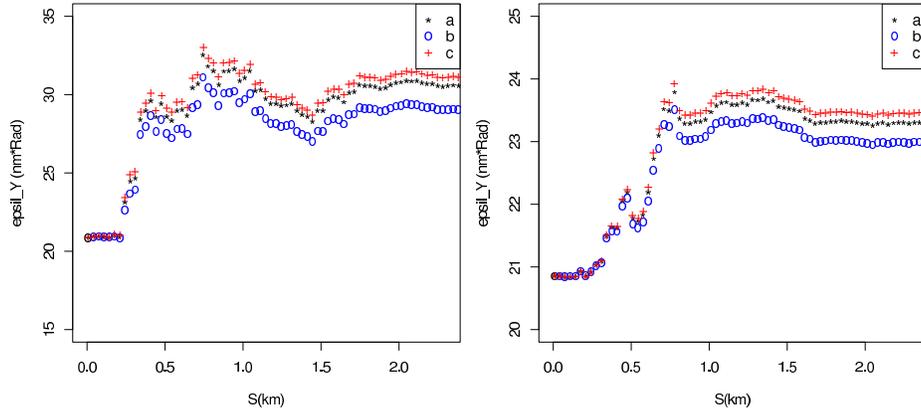


Figure 11: The DFS performance in presence of a  $0.1 \sigma$  beam jitter, with 20 pulses per trajectory measurements. The three curves correspond to three successive macro-iterations. Left:  $\varepsilon_v$ , left uncorrected for Dispersion. Right: same runs,  $\varepsilon_v$  is corrected for Dispersion.

the BPM resolution in all these exercises was also assumed to unrealistically small (nanometer). More runs will be required...

### 3.1.4 Ground motion, without fast inter-pulse feedback correction

A C++ version of the Ground Motion code used in previous studies[11] has been integrated into this steering package. It is based on the ATL model, extensively used in previous simulation[12]. This model can be compared to the data taken in the Fermilab MINOS experimental hall with a few hydrostatic sensors(HLS). Since we are not interested in an overall tilt of the entire machine with respect to the average earth gravity field, the simulated position variation is compared to the measured difference of HLS readings  $L_i$ . For instance:

$$d_2 = L_2 - (L_1 + L_3)/2.$$

The same simple calculation is also made on the output of the ATL model. This comparison is shown on figure 13. The MINOS site, located in the

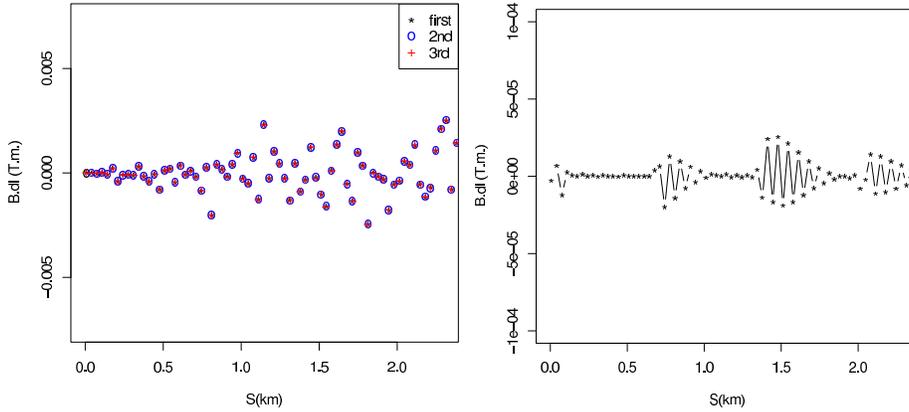


Figure 12: The DFS Solution in presence of a  $0.1 \sigma$  beam jitter, . Same case as the figure above. Left are the dipole corrector settings, right the difference between the first and third macro-iteration.

Maqoketa shale sometimes compares favorably to a “quiet” site. However, when the water table around the site change its level, the site becomes more noisy. The fluctuation due to the tides is approximately 0.5 to 1 micron per hour.

More work is planned on both the experimental front, as we need 5 Hz data, and over longer distance ( few km.)<sup>11</sup> In addition, one could make a more systematic analysis of the MINOS data and extract the ATL parameters.

Even for the “quiet” site, a convergence criterion  $\delta_{cd} = 2.5 \cdot 10^{-7}$  T. m. is perhaps a bit unrealistic. Indeed, a tenth of  $\sigma$  of beam jitter makes it unnecessarily too tight. The maximum dipole corrector setting change during the steering through the first 40 dipoles is shown on figure 14. “Convergence” is reached, almost by luck after 93 iterations without ground motion, and not reached after a maximum (arbitrarily set) of 100 iterations. Also shown on this last figure is the performance, significantly worse then in the static case. Loosening the convergence criteria by an order of magnitude,  $\delta_{cd} = 2.5 \cdot 10^{-6}$ ,

<sup>11</sup>As pointed out by V. Shiltsev, this might not be necessary. If the ATL is correct, one could use data from various periods, basically exchange events spread out in time to events spaced out in physical space

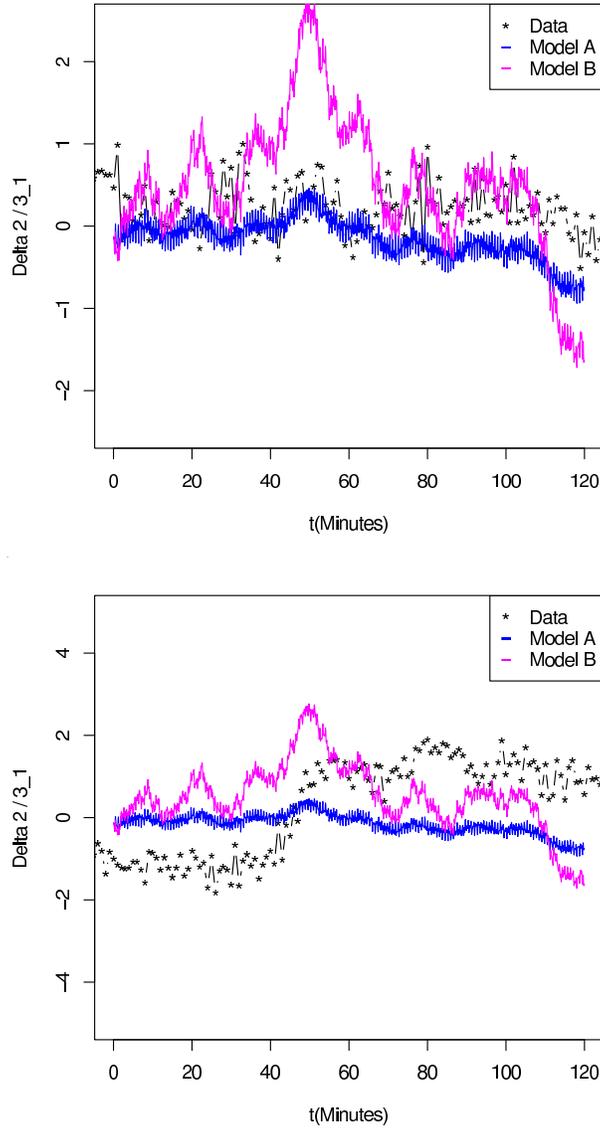


Figure 13: Ground Motion comparison, data  $d_2$  at MINOS, and the ATL model, for a limited time of two hours. The top plot corresponds to a relatively quiet time in the site, starting on Oct 8 2006, 10:04 A.M., and (bottom plot) a more active time, starting on Oct 3 2006, 4:00 A.M.. Data is recorded every minutes, while the model works at the ILC repetition rate, 5 Hz.

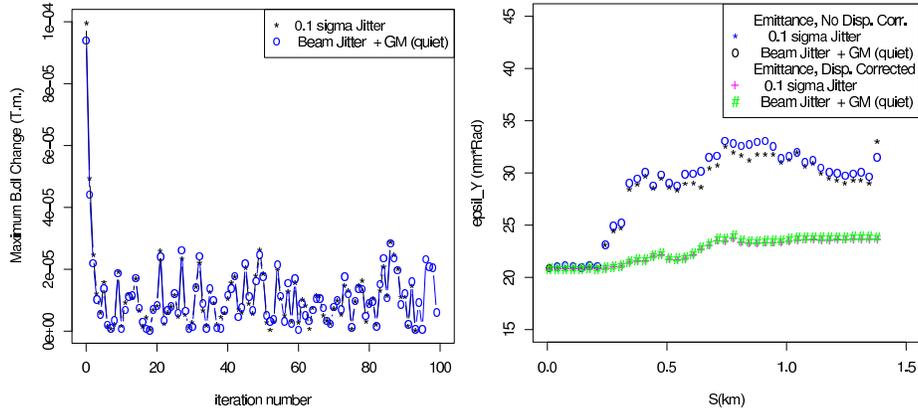


Figure 14: Left plot: the maximum change of the dipole corrector settings as we iterate and steer through the first 40 dipole. Random changes are made after less than 10 iterations. Right: the Performance at the end of the steering through this section. The last dipole in the section is not optimum, this will be taken care in the next DFS section.

speeds up the process greatly, without losing too much performance.

Under these circumstances, for the quiet site, negligible emittance growth at the end of 3 macro-iterations is observed, with respect to the static case. Even with a noisy site, corresponding to the ATL parameters of  $A = B = 1.0^{-16}$ , convergence to seemingly manageable emittance is reached (shown on figure 15). Beam Jitter was turned off for this run, and BPM were averaged for only 5 pulses. However, at the end of each steering section (40 dipole long), a waiting period of 1000 pulses ( $\approx 3$  minutes) was applied.

Evidently, there is quite a bit of arbitrariness in these parameters. A systematic study will be undertaken based on the current ILC Main Linac lattice. Meanwhile, these results are encouraging. However, the following caveats must be kept in mind:

1. If extrapolated linearly, an emittance, corrected for dispersion, growth of  $\approx 2.5$  nm.radiant per 2.5 km. of Linac is our total budget for emittance growth. No safety margin left.
2. The projected emittance, left uncorrected for Dispersion, is unaccept-

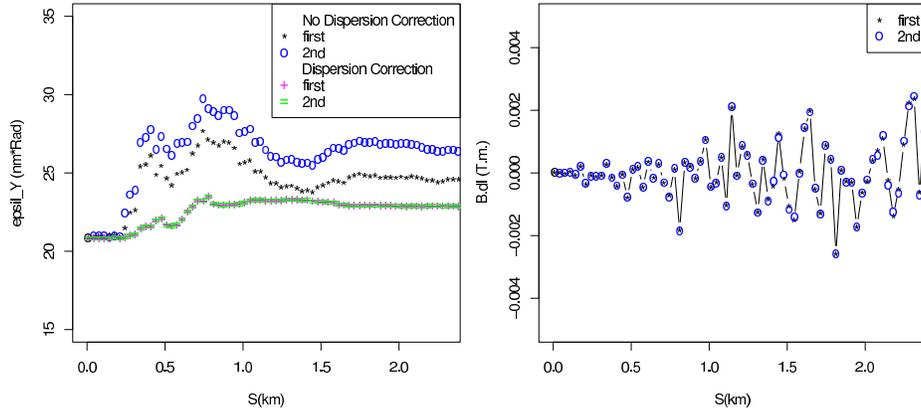


Figure 15: The DFS performance (left plot) in presence of a “noisy” ground motion, after the first two macro-iterations. Right plot shows the corresponding dipole corrector settings

ably large. In principle, correcting for Dispersion is always possible, but often cumbersome and difficult.

3. The Linac was perfectly aligned for the first 150 meters.
4. The DFS solution we obtain does not give us the BPM offsets, because the trajectory departs from the straight line by one mm, or so, 3 to 10 times the anticipated BPM offset with respect to this straight line. The Dispersion deviates significantly from its designed value (zero in our case). Consequently, the simple, the local algorithm described in reference [11] can not work and will require modification(s).
5. A possibility is too keep using the DFS method quasi-continuously. This mean running the Linac at reduced gradient (typically, by 20 %). Evidently, these pulses can not to used for HEP. This mean reduced luminosity.
6. The ATL model might be wrong. For instance, we know that the gravitational tides implies strictly periodic (as opposed to random) perturbation.

7. No quadrupole field error.
8. Emittance growth due to other machine defects, such XY coupling, klystron LLRF jitter, or long range wake fields have not been considered yet.

Other improvements of the DFS algorithm have been proposed:

1. Increase ( or decrease ) the overlap between DFS sections. Not much improvement: is the overlap is maximized (e.g., move from one DFS section to the next by incrementing the first dipole index by one), the algorithm converges to very similar dipole corrector settings to those obtained in the previous DFS section, for the upstream part of the section. The final solution might end up being different than from the case where the dipole increment is half of the number of dipole per DFS section, but not necessarily better in terms of emittance growth.
2. For a given DFS section, the beam is “pre-steered” to a trajectory in such a way that the apparent Dispersion in the first few BPM is zero. This is achieved using dipole corrector situated just in front of this DFS section. Based on a first few runs, the performance improvement is not very significant. Moreover, there might be a bit of an ambiguity in the algorithm: once the settings in the upstream DFS section are modified and the beam steered through the current DFS section, should we go back to the upstream corrector, and set them to the value obtained when steering through that section, or leave them as set to steer through the current section? A bit more clarification is needed here.
3. Kick Minimization: To the Dispersion and Displacement constraints, one adds a third one: the absolute value of the dipole corrector settings must also be somewhat minimized. This prevents reaching a solution where the successive correctors are “fighting each others”, i.e., the correction is large, but changes signs at each dipole. Such an oscillator pattern is also correlated with oscillatory pattern in the trajectory. Thus, a re-adjustment of the Displacement weight also works.

## 3.2 LET Without Cavity tilts nor displacement.

Let us consider a much simpler problem by drastically reducing the potential sources of emittance dilution. The cavities centers are now assumed to be placed exactly on the beam line, and on axis. More importantly, there are no cavity tilts. The BPM offsets are still unknown, and the quadrupoles are displaced vertically. In this case, a unique solution can in principle be obtained, because the number of perturbation (quadrupole displacement) is the same as the number of actuators, or “knobs” (dipole correctors) one can adjust. In this case, the “pure DFS” algorithm can be applied, with optimum performance. That is, the weight assigned to the apparent trajectory are set to zero. The optimization simply consists at minimizing the Dispersion. The solution is fairly unique, provided the DFS section is short enough, or the overlap between successive section is large compared to the section length. To illustrate this, figure 16, 17 and 18 correspond to the following cases:

- a: Pure DFS (weight for orbit set to 0.), DFS section length = 10 dipoles, overlap = 5 dipoles.
- b: Pure DFS, DFS section length = 20 dipoles, overlap = 10 dipoles.
- c: Pure DFS, DFS section length = 40 dipoles, overlap = 20 dipoles.
- d: Weighted DFS, trajectory error of 360 microns, DFS length = 40 dipoles, overlap = 20 dipoles.
- e: Almost pure 1-to-1 steering, with trajectory error of 4 microns, DFS section length = 40 dipoles, overlap = 20 dipoles.
- h: Pure DFS, DFS section length = 10 dipoles, overlap = 8 dipoles.
- i: Pure DFS, DFS section length = 20 dipoles, overlap = 18 dipoles.
- j: Pure DFS, DFS section length = 40 dipoles, overlap = 38 dipoles.

In all cases, the convergence criterion was set to  $\delta_c = 2.5 \cdot 10^{-6}$  T.m.

The reason for this good performance are:

1. The beam offset with respect to the center of the cavity is typically less than 100 microns. This greatly mitigate the effect of the short range wake fields.

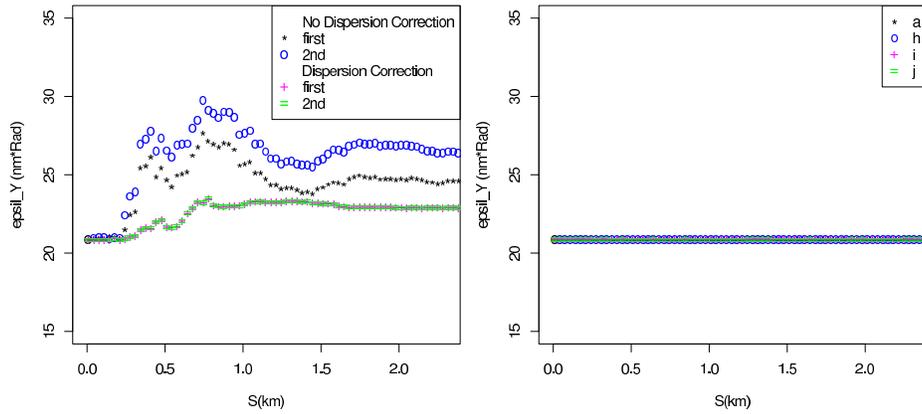


Figure 16: The DFS performance with perfectly aligned cavities. Only quadrupole and BPM offsets are unknown. See text for details.

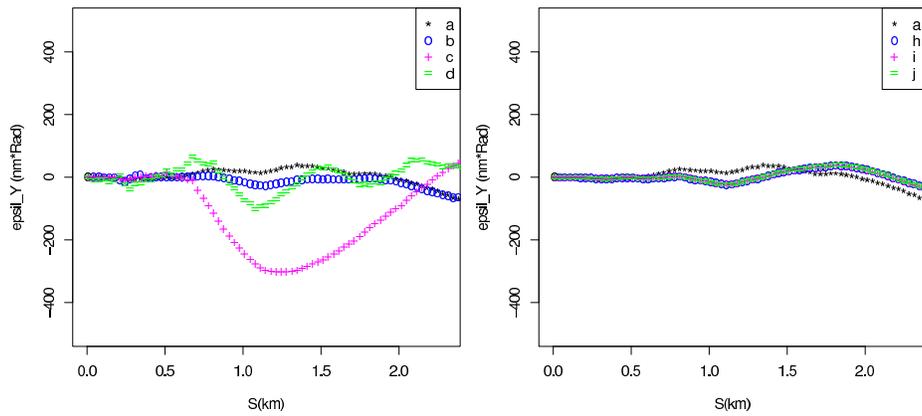


Figure 17: Trajectories if the cavities are perfectly aligned, on the same cases as specified in the previous figure.

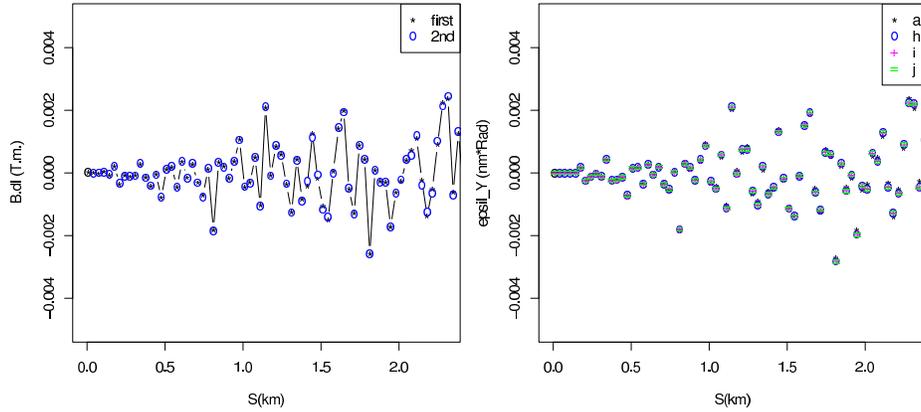


Figure 18: DFS solution corresponding to cases where the cavities are not tilted nor displaced.

2. One important source of Dispersion has been entirely removed. The cavity no longer disperse the beam as the accelerate the beam on axis. To see this, let us go back to the original displacements and tilts. The vertical kicks  $y' = P_y/|P|$  are measured just after and before the BPM/Quadrupole/dipole/ packages, and their differences are directly compared to the change in the same  $y'$  after/before a quadrupole. At 5 GeV, the individual vertical kicks due to 24 cavity tilts of  $\approx 200$  micro-radian is comparable to those induced by quadrupoles displaced by  $\approx 300$  microns. See figure 19

### 3.3 On Correcting Cavity Tilts.

Having a near straight trajectory through the cavity has the following benefits:

1. The LET performance is undoubtedly better.
2. It will mitigate the harmful effects of long range wakes, namely, emittance growth and dynamical load on the cryogenic system.

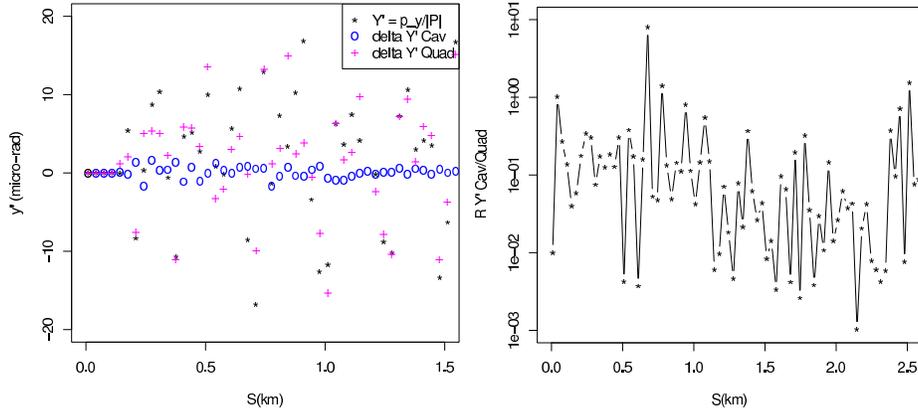


Figure 19: Left Plot: The transverse momentum  $y'$  versus  $S$ , for the misaligned and steered Tesla Lattice discussed in the benchmark section. Shown also is the change in  $y'$  before/after quadrupoles and a continuous string of cavities. Right: the ratio between the changes in  $y'$ , cavities/quadrupoles.

3. If the DFS solution corresponds to a perfectly straight trajectory with respect to the laser straight reference line, the value of the BPM offsets can be trivially deduced. Unfortunately, residual trajectory offsets of about 50 to 100 microns over a km are still there. This is with a convergence criterion of  $2.5 \cdot 10^{-6}$  Tesla meters. Dynamical effects and realistic tolerance on time dependent stray field are likely to prevent us from reaching a  $\delta_{cd}$  much lower than this. Yet, some valuable information about these offsets can be obtained if one get closer to the nominal trajectory.

The effect on the LET performance can easily be understood based on linear optics. Leaving aside wake field, the kick due to a tilted cavity of length  $L$  can be approximated by

$$y' \approx \theta \int_0^L g/E \approx \theta \Delta E/E$$

where  $g$  is the gradient and  $\Delta E$  is the energy gain in the cavity. Thus, the  $y'$  depends on the cavity gradient. Reducing this gradient change the

dispersion, and a perfectly aligned cavity generate no dispersion. This opens the possibility of a Beam Based Alignment method, i.e., a variant of the DFS method, where a local change to the momentum is applied to detect and correct deleterious Dispersion. Straightforwardly, two types of actuators can be proposed:

1. Adding a small dipole corrector magnet in each cryo-module, to locally compensate for deleterious dispersion created by these cavity tilts.
2. Tilting the cavity, i.e., mechanically re-position them, at operating temperature.

However, in either case, the beam based alignment tuning method relies on the capability to change the accelerating gradient in a single cryo-modules. Changing the gradient in one r.f. unit, consisting of two cryo-modules in the Tesla Design, or three of them in the ILC design, means affecting the Dispersion anywhere in between each quadrupole. There is one and one only quadrupole per r.f. unit. Thus, this DFS variant will not be able to distinguish between correcting cavity tilts integrated over the r.f. unit, or correcting the quadrupole displacements.

Although the current r.f. distribution system does not allow to control the gradient over a specific cavity nor a cryo-module, the following algorithm has been implemented in the simulation. Prior to DFS steer a given section, each cryo-module is tilted, attempting to cancel the trajectory difference between running the cryo-module at reduced voltage (by 80%) and at full voltage. Four BPMs located immediately downstream of the affected r.f. unit/quadrupole are used. This is done sequentially for cryo-modules belonging to the DFS section, starting from the most upstream modules. Then, DFS steer through this section. This is done for each DFS section. Also, multiple macro-iterations are performed, basically alternating between cavity tilt corrections and quadrupole displacement corrections.

The results of this algorithm are shown on figure 20. They correspond to the same lattice and same misalignment (same seed) used in all previous cases. The DFS section length is set to 20, with an overlap of 10. While steering with dipole correctors, the trajectory error is set to 360 microns, with a convergence criterion of  $2.5 \cdot 10^{-6}$  Tesla meters, and while steering with cavity movers, the convergence criterion was set to 4 micro-radian. Little

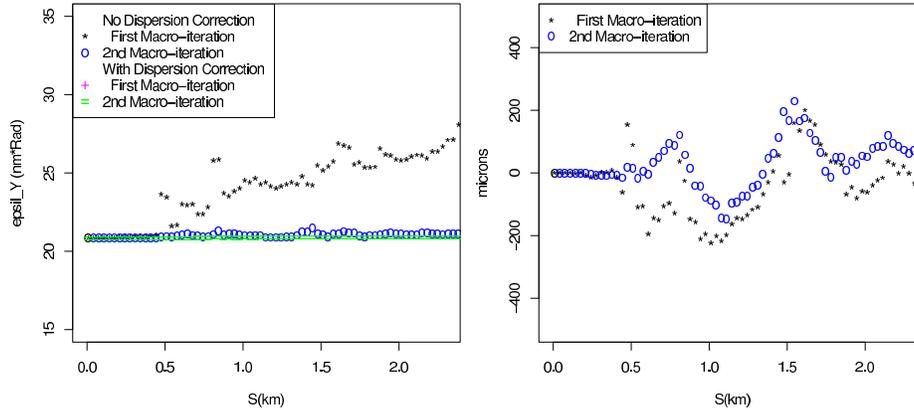


Figure 20: Left Plot: The performance for the variant of DFS, re-aligning cavities and compensating for quadrupole displacements with dipole correctors. Right Plot: the corresponding trajectory, showing excursion of less than 300 microns through the critical low energy section of Linac.

or none optimization for these parameters have been done in the context of this algorithm, yet, it seems to work.

## 4 Conclusions

A technical review of the DFS algorithm for the Tesla Main Linac lattice has been conducted, by re-implementing the method in the CHEF framework. While some algorithm have been borrowed from other package (short range wake field from Merlin, in particular), the macro-particle tracking is done by the CHEF core beam line code and is therefore an independent check. The controls and basic loops for the beam based steering algorithms has been entirely re-written. The Singular Value Decomposition method, required for linear optimization, has been recently improved in CHEF, based on feedback from this project.

As observed by many authors, the DFS method “basically works” for Linac. However, its most simple implementation - i.e., without down-weighting large trajectory excursion, does not give stable and unique solutions.

In the static limit, the root cause for the sudden increase in the projected emittance at the beginning of the Linac has been isolated, and can be cured if one is allowed to either physically re-align the cavities, or add more dipole correctors, and as controversial, be able to control the gradient in a single cryo-module. Since this source of spurious transverse kicks drops like  $1/E$ , only the front-end of the Linac would require this capability.

Ground Motion and beam jitter have been simulated, and preliminary results show that one can DFS-steer while the machine is moving, albeit with some performance loss for the noisy sites. Although encouraging, this last result must be considered as very preliminary.

For consistency and brevity, all these studies have been done on the obsolete Tesla design lattice, with only one misalignment seed. This is clearly far from sufficient. Meanwhile, CHEF can parse the new ILC lattice, and bunches have been propagated through it with negligible emittance growth, in absence of misalignment and dynamical effects. Previous studies have establish that the Linac curvature has little impact on the DFS performance. The plan is therefore obvious: briefly verify this claim, and move onto further studies of the DFS with ground motion and other dynamical defects.

## 4.1 Acknowledgments

None of this work would have been possible without the dedicated support of the CHEF team (Leo Michellotti, Francois Ostiguy and Lynn Garren). Having access to the Merlin code was also extremely useful. The author also acknowledges very useful discussions with Kirti Ranjan, Alexis Valishev, Jeffrey Smith, Peter Tennenbaum, Nikolay Solyak and other members of the ILC Global Design group responsible for Beam Physics studies.

## References

- [1] G. Loew *et al* ILC-TRC 2003 Report  
*<http://www.slac.stanford.edu/xorg/ilc-trc/2002/2002/report/03rep.htm>*
- [2] L. Michelotti and J. F. Ostiguy, FERMILAB-CONF-05-090-AD *Proceedings of Particle Accelerator Conference (PAC 05), Knoxville, Tennessee, 16-20 May 2005*

- [3] **Merlin**, A C++ Class Library for Accelerator Simulation  
*<http://www.desy.de/~merlin>*
- [4] J. C. Smith *Talk presented at the Vancouver DGE meeting, 19-22 July 2006* *<http://ilcagenda.cern.ch/contributionDisplay.py?contribId=210&sessionId=2&confId=316>*
- [5] F. Poirier and D. Krücker, N. Walker, *An ILC Main Linac Simulation Package based on Merlin* Proceedings of European Particle Accelerator Conference (EPAC 06), Edinburgh, Scotland, 26-30 June 2006
- [6] J. Admundson, P. Spentzouris  
*[http://cd-amr.fnal.gov/aas/Advanced\\_Accelerator\\_Simulation.html](http://cd-amr.fnal.gov/aas/Advanced_Accelerator_Simulation.html)*  
and reference therein.
- [7] T. Weiland and I. Zagorodnov  
*The Short-Range Transverse Wake Function for TESLA Accelerating Structure* *[http://flash.desy.de/reports\\_publications/tesla\\_reports/tesla\\_reports\\_2003/index\\_eng.html](http://flash.desy.de/reports_publications/tesla_reports/tesla_reports_2003/index_eng.html)*
- [8] J. Smith,  
*[http://www.lns.cornell.edu/~js344/LC/bba\\_benchmarking](http://www.lns.cornell.edu/~js344/LC/bba_benchmarking)*
- [9] D. Sagan and J. Smith, *The TAO Accelerator Simulation Program*, Contribution to the 2005 Physics Accelerator Conference, May 16-20, Knoxville, Tennessee, USA  
*<http://accelconf.web.cern.ch/AccelConf/p05/PAPERS/FPAT085.PDF>*
- [10] J. Smith, Private communication, October 2006
- [11] V. Ivanov, *Dynamic Alignment in Presence of Ground Motion and Technical Noise* Talk presented at the 9th International Workshop On Accelerator Alignment (IWAA 06) 26-29 Sep 2006, Menlo Park, California
- [12] A. Seryi, A. Mosnier, Phys. Rev. E, 56, 3, 1997.-pp.3558-3571. See also A. Seryi, L. Hendrikson, P. Raimondi, T. Raubenheimer, P. Tenenbaum, Simulation Study of the NLC with Improved Ground Motion Models, Proc. XX Int. Linac Conf., August 21 - 25, 2000, Monterey, California.
- [13] J. Volk *et al* *fermilab-conf-06-334-ad*