OPTIMIZATION ALGORITHMS FOR ACCELERATOR PHYSICS PROBLEMS*

B. Mustapha[#] and P. N. Ostroumov
Physics Division, Argonne National Laboratory, IL 60439, U.S.A.

Abstract
Optimization tools are needed in every step of an accelerator project, from the design to commissioning to operations. However, different phases have different optimization needs that may require different optimization algorithms. For example, a global optimizer is more appropriate in the design phase to map the whole parameter space whereas a local optimizer with a shorter path to solution is more adequate during operations to find the next best operating point. Different optimization algorithms are being used in accelerator physics, we mention in particular standard algorithms such as least square minimization and evolutionary algorithms such as genetic optimization. Over the years, we have developed several optimization tools for beam tracking codes to include 3D fields and SC effects. Including particle tracking in the optimization process calls for parallel computing. We will review the different algorithms and their implementation and present few highlight applications.

OPTIMIZATIONS IN ACCELERATOR PHYSICS

Optimizations are heavily used in the design phase of an accelerator project, but they are much less used to support the commissioning and operations once the machine is built. During the design phase, optimizations are used in the design of the different beam line elements: magnets, rf cavities, etc. They are also used for the lattice optimization to find the appropriate sequence of elements and drift spaces. Once the lattice is defined more optimizations are used to determine the appropriate element settings for optimal beam dynamics and beam quality. This is often iterated with the lattice design. Once the accelerator is built, more effort is dedicated to hardware problems than to developing a realistic model of the machine. We believe that using the appropriate optimization tools during the commissioning should help better understand the machine's behaviour and expedite the delivery of the first beam. Fits to reproduce the experimental data using a model should significantly improve the predictability of the model to use for real-time machine operations. Often, simplified models (1D, single particle) are used to support daily machine operations [1]. Simple models have the advantage of being fast and able to describe the overall behaviour of the machine while detailed 3D models are slow and still cannot reproduce the details seen in the data [2]. We believe that a significant effort should be dedicated to developing more realistic 3D models before being able to use them to support real-time machine operations. Once such models are fully developed large scale parallel computing could be used for fast turn-around simulations and optimizations.

ELEMENTS OF AN OPTIMIZATION PROBLEM

The first important step is the proper definition of the optimization problem. An optimization problem has one or more objectives which are the important quantities or qualities characteristics of the problem that you would like to optimize. These objectives depend on the parameters of the problem which are the variables affecting the outcome or the solution to the problem. It is usually a good practice to choose the parameters to which the solution is more sensitive. These parameters could be subject to constraints and or correlations which define the limits of the parameter space. The simplest case is where the parameters are independent with lower and upper bounds. If the parameters are correlated, it is usually recommended to reduce them into a set of independent parameters. The last and most important element of an optimization problem is the choice of the appropriate optimization algorithm. Depending on the nature of the problem, the most appropriate algorithm could be a local optimizer, a global one, a standard or an evolutionary.

LOCAL VERSUS GLOBAL OPTIMIZATION ALGORITHMS

It is usually not hard to find a local minimum of an objective function. What is hard is to prove that the minimum found is a good one and it is even harder to prove that a minimum is an absolute or a global one. A local optimizer usually starts with a first guess then finds a direction that minimizes the objective function and moves one step in that direction. The procedure is repeated iteratively until no more progress could be made. A local algorithm is usually fast because it explores the parameter space along a single path defined by the minimization direction adjusted at every step. In contrast, a global optimizer should explore the entire parameter space and eventually find all local minima before finding a global one. It should also prove that the minimum found is a global one which makes it much slower than a local optimizer. Luckily, not all problems or applications require global optimizations. A global optimizer is more appropriate to use in the design phase of an accelerator project to map the whole parameter space and make sure not to miss the best set of design parameters. Such a global optimizer should also find all feasible solutions to

* This work was supported by the U.S. Department of Energy, Office of Nuclear Physics, under Contract No. DE-AC02-06CH11357.
[#] Corresponding author: brahim@anl.gov

help make compromises if needed. On the other hand, a local optimizer with a shorter path to solution is more adequate to support accelerator operations. In this case, we usually start from a good starting point to find the next best operating point by returning few elements on the accelerator. The time to solution is a very important parameter in the choice of the appropriate optimization algorithm. A good optimizer should also optimize the path to the best solution.

STANDARD VERSUS EVOLUTIONARY OPTIMIZATION ALGORITHMS

Standard optimization algorithms [3] are the most common and widely used. They usually have a single objective function to optimize which could be a weighted sum of multiple objectives. Derivatives of the objective function with respect to the optimization parameters are usually required. A single trial solution is evaluated at every iteration. Evolutionary algorithms are more recent [4] and are based on the "Theory of Evolution and Natural Selection", where only the best survive. Multiple objective functions could be included in the optimization without the requirement of knowing their derivatives. Multiple trial solutions are evaluated at every iteration which gives the global nature of evolutionary algorithms.

Examples of Standard Optimization Algorithms

The first example of standard optimization algorithms is the simplex method [5], which does not require the function derivatives. It starts by building a simplex consisting of the first guess and a base of feasible solutions in the parameter space. Iteratively, the worst solution is replaced by a better one built from the base. The iterations stop when no more progress could be made or at a predefined cut-off error on the solution. The second example is the gradient descent method [6] which require the first derivatives of the objective function on the optimization parameters. The derivatives are used to determine the minimization direction p_i at every iteration i: $p_i = -B_i^{-1}\nabla f_i$ where f is the objective function and B is a symmetric non singular matrix. In the case where B is the matrix identity I the method is called the steepest descent method. The third example is the Newton method [7] which requires both the first and second derivatives of the objective function. In this case B is the matrix of second derivatives also known as the Hessian matrix H. The fourth example is the Quasi-Newton method [8] which uses the first derivatives but keeps updating the matrix of second derivatives at every iteration. In this case B is an approximation to the Hessian matrix and does not require extra computing effort for the second derivatives as for the Newton method. Each of these methods is actually a class of methods with a variety of implementations.

Example of Evolutionary Algorithm:A Genetic Optimizer

A genetic optimizer [9] starts with a set of solutions randomly generated inside the parameter space. The solutions are evaluated and ranked based on the objectives and constraints of the problem to select a subset of best solutions. The selected solutions are then used to generate the next population of solutions by crossover, mutation or other using predefined rates. The evaluation and ranking procedure is repeated for the new set of solutions until no progress could be made or a stable set of best solutions is obtained. For a given solution, the array of parameter values plays the role of a gene. Figure 1 shows the different ways of generating the next generation of solutions from the selected set of best solutions, namely crossover, mutation and random. The random generation of new solution could be turned on at the beginning for better sampling of the parameter space then turned off.

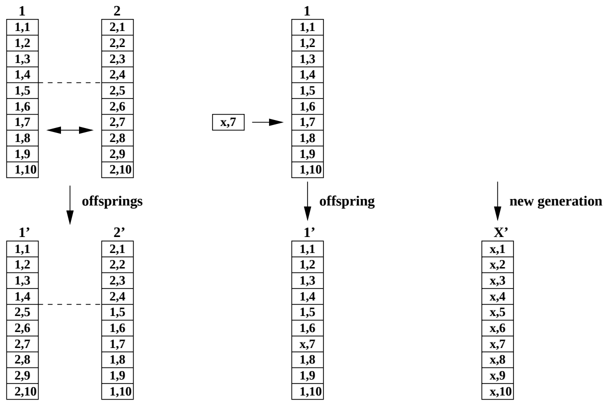

Figure 1: Three mechanisms to generate off-springs from selected best solutions. Crossover of two solutions on the left, mutation of a single solution in the middle and new random solution on the right.

PARALLEL OPTIMIZATION ALGORITHMS

Standard algorithms are serial in nature because the direction of the next iteration is decided based on the outcome of the current one. The fact that a single solution is evaluated per iteration makes these algorithms parallelizable only at the level of the objective function and derivatives evaluation. For example, a least square minimization where the objective function is of the form:

$$f = \sum_{i=1}^{N} \frac{\Delta X_i^2}{\varepsilon_X^2}$$

could be parallelized by parallelizing the sum for large N. Optimizations using multi-particle tracking could be parallelized by parallelizing the tracking, the Poisson solver and the statistics calculations for large number of particles. A global optimizer may be parallelized by subdividing the parameter space and assigning the different sub-spaces to different processors.

In contrast Evolutionary algorithms are parallel in nature because at every iteration multiple solutions are evaluated independently which makes them well suited for parallel processing with minimal communication. It is however not easy to parallelize the ranking, selection and offspring generation which are usually assigned to the master process. This makes evolutionary algorithms more suitable for optimization using multi-particle codes with realistic 3D external and space charge fields. Usually no parallel particle tracking is required unless a very large number of particles is needed for the optimization problem. In practice, any tracking code with space charge calculations could be used. A higher level parallel layer is often used to manage the generation, ranking and selection of trial solutions and every process calls the code when needed. In our beam dynamics code TRACK [10] this layer was built-in within the code.

APPLICATIONS IN BEAM DYNAMICS OPTIMIZATION

Automatic longitudinal tuning of a multiple charge state ion beam before a stripper

In this application, a longitudinal fine-tuning procedure was developed specifically for a multiple charge state beam to minimize its longitudinal emittance right before a stripper [11]. The beam should reach the stripper in the form of an up-right ellipse in the (Δφ, ΔW) plane to minimize the emittance growth from the energy straggling effect in the stripper. This could be realized by matching the beam centroids and Twiss parameters of the different charge state beams. The objective function in this case is:

$$F=\frac{(W_{q0}-W_0)^2}{\varepsilon_w^2}+\sum_{qi}\frac{\Delta W_{qi}^2}{\varepsilon_{\Delta w}^2}+\sum_{qi}\frac{\Delta\phi_{qi}^2}{\varepsilon_{\Delta\phi}^2}+\sum_{qi}\frac{\alpha_{qi}^2}{\varepsilon_{\alpha}^2}+\sum_{qi}\beta_{qi}$$

where W_0 is the desired beam energy and ε_W is the associated error. $\varepsilon_{\Delta W}$, $\varepsilon_{\Delta\varphi}$ and ε_α are the allowed errors on the relative energy, phase and α shifts of the individual charge state beams from the central beam. The fit parameters in this case are the RF cavities phases and amplitudes in the section up-stream of the stripper. Figure 2 shows the results of the fit for a five charge state uranium beam in the medium energy section of the RIA driver linac. This optimization reduced beam losses in the high-energy section of the linac by a significant factor as seen on figure 3.

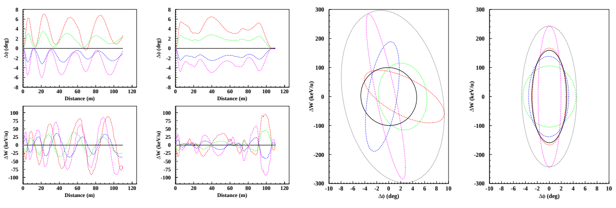

Figure 2: The left 4 plots show the phase and energy oscillations of the five charge states around the central charge state before and after applying the tuning procedure. The right 2 plots show the corresponding beam ellipses on the stripper before and after tuning.

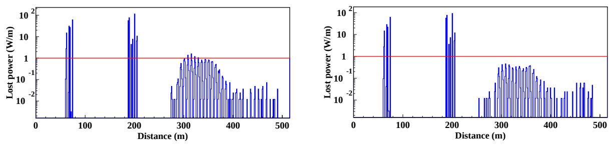

Figure 3: Beam loss in the RIA driver linac before and after applying the longitudinal tuning procedure. The two peaks correspond to the location of the strippers and the scatter loss is in the high energy section which has reduced significantly after fine tuning is applied.

A realistic corrective steering algorithm

We have recently developed a realistic corrective steering procedure [12]. A simplified algorithm is presented in figure 4.

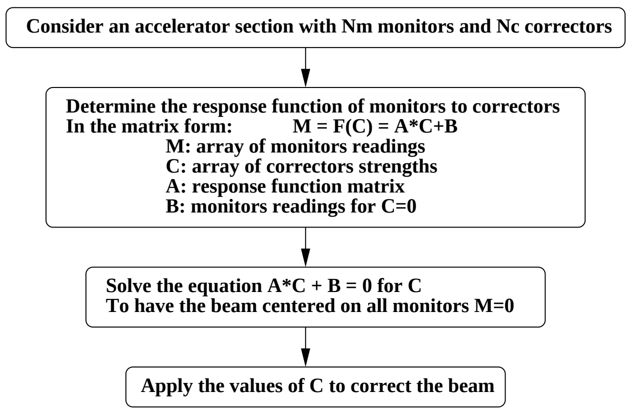
Consider an accelerator section with Nm monitors and Nc correctors
Determine the response function of monitors to correctors
In the matrix form: M = F(C) = A*C+B
M: array of monitors readings
C: array of correctors strengths
A: response function matrix
B: monitors readings for C=0
Solve the equation A*C + B = 0 for C
To have the beam centered on all monitors M=0
Apply the values of C to correct the beam

Figure 4: Algorithm for the corrective steering procedure.

In TRACK implementation, instead of solving the matrix equation A*C+B=0 for the array of corrector field strengths C, with A being the response function of monitors to correctors and B the monitors readings for C=0, we perform a least square minimization of the equivalent function given below:

$$f(C_{i_c},i_c=1,N_c)=\sum_{i_m=1}^{N_m}\frac{(\sum_{i_c=1}^{N_c}A_{i_c i_m}*C_{i_c}+B_{i_m})^2}{\sigma_{i_m}^{\ 2}};for\left|C_{i_c}\right|\le C_{\max}$$

In this way, we can include the monitor precision σ_{im} and the maximum corrector field strength C_{max} in the solution. Monitors with different precisions will have different weights in the minimization procedure. The minimization should lead to an approximate solution in the case of an over-determined problem (more equations than variables) and to the best solution in the case of multiple solutions (under-determined problem). This realistic corrective steering procedure could very well be applied in the design phase of an accelerator project to determine the monitors and correctors requirements for an effective beam center correction as well as in the control room of an operating accelerator. The results of its application during the design of the HINS project front-end linac [13] being built at Fermilab are shown on figure 5. After multiple iterations, the optimum numbers and locations of correctors and monitors for an effective correction were determined.

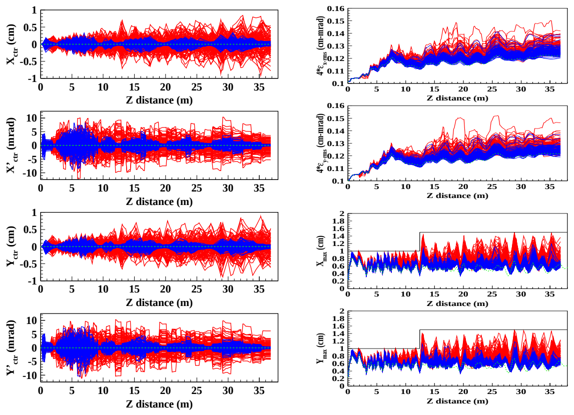

Figure 5: Beam position and angle centers (left) and beam emittances and envelopes (right) before (red curves) and after (blue curves) applying the correction procedure for 100 randomly misaligned linacs. In green is the ideal case without misalignment and in black is the aperture.

Design optimization of a chicane in an ultra-low emittance electron injector

We have recently implemented a parallel genetic optimizer into the beam dynamics code TRACK. As a first application we used it for the design optimization of the first chicane of an electron injector for an X-FEL-O linac [14]. The objective was to minimize the transverse emittance at the end of the chicane. The parameters were the quadrupoles and solenoids strengths as shown in figure 6. We were able to reduce the transverse emittance by about 10% from a manually optimized case which is very critical for an ultra-low emittance injector.

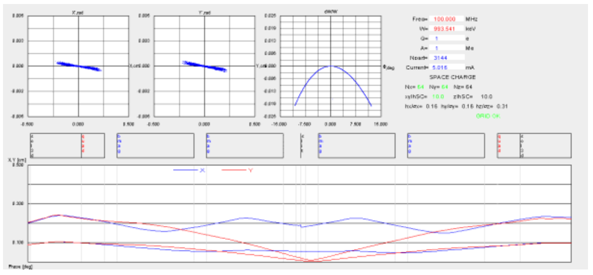

Figure 6: Optimized beam in the first chicane of an electron injector for an X-FEL-O linac.

To study the convergence of the genetic optimizer we compared the corresponding results to the results from a 2D map in the case of two parameter optimization. Figure 7 shows the results for both cases proving the convergence of the genetic optimizer in 5 iterations corresponding to a total of 500 trial solutions. Large scale optimizations are underway for a larger section of the linac.

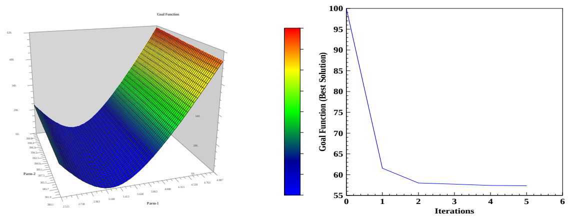

Figure 7: Comparing the result of a 2D map (left) with the result of the genetic optimizer (right) for a two parameter optimization problem.

POTENTIAL APPLICATION: THE MODEL DRIVEN ACCELERATOR

Concept and Motivations

The concept of the model driven accelerator is to develop and use a computer model to support real-time accelerator operations. Presently, no accelerator in the world could fully rely on a computer model for its operations. The main reason is a discontinuity between the design and operation phases of an accelerator project. Among the factors contributing to this discontinuity are: 1- Simulations in the design phase assume almost perfect conditions and cannot reproduce the real machine, 2- Actual elements specification and performance are usually different from their original design and in most cases 3- Not enough diagnostic devices to characterize the machine. The lack of a realistic model to support the commissioning and operations results in significant delay in the deployment of a new machine and a lot of time spent on machine tuning during operations. This usually leads to low availability and high operating cost of the machine. For example, a complex system such as the proposed FRIB facility [15], where primary beams from proton to uranium up to 600 MeV/u are used to produce beams of rare isotopes all over the map, cannot afford not to have a computer model to support its operations.

To bridge the gap between the design and operation phases we propose to develop a realistic model of the machine. Among the benefits of such a model is fast tuning for the desired beam conditions and fast retuning to restore the beam after a failure. This should significantly improve the availability of the machine and reduce its operating cost.

Requirements for the realization of the Model Driven Accelerator

The main requirements for the realization of the model driven accelerator could be summarized in the development of a 3D beam dynamics code with the appropriate set of optimization tools and large scale computing capabilities. A multi-particle beam dynamics code is more realistic than matrix-based and single-particle codes because it supports 3D fields, includes fringe fields and appropriate space charge calculations. It also allows more detailed simulations necessary to study eventual beam loss and produce data similar to the measured data. Such a code should also include a large set of optimization tools. Optimization tools are needed not only for design optimization but also to tailor the computer model to the actual machine to be useful for real-time operations. Multi-particle optimizations usually involve tracking a large number of particles for large number of iterations which is very time consuming and requires large scale parallel computing. Therefore the beam dynamics code should have parallel computing capabilities.

A small scale realization of the Model Driven Accelerator

We have recently succeeded to extract, accelerate, analyze and recombine a two-charge state DC bismuth beam from an ECR ion source [16]. The beam is perfectly combined at the point of injection into a subsequent RF accelerator. We consider this as a small scale realization of the concept of the model driven accelerator. The beam dynamics code TRACK was used first to design the beam line and then to support the operation by predicting the elements settings required to recombine the two charge state beam at the end of the LEBT. Figure 8 is a general 3D view of the prototype 2Q-LEBT beam line.

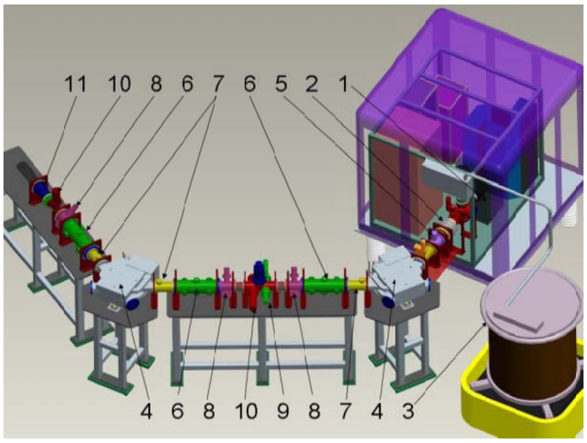

Figure 8: General 3D view of the 2Q-LEBT

For a realistic beam dynamics simulation, 17 beams (O and Bi) are tracked simultaneously from the ion source through the LEBT with a total current at the source of about 2 mA. We assumed a 4D water-bag initial distribution for all beams and a 50 % charge compensation factor in non-electric devices and 0% in electric devices. Realistic 3D models were developed and used for all beam line elements. In order to tailor the TRACK model to the actual beam line we had first to determine the initial beam parameters at the source. To do so we had to develop a new procedure to fit the beam profiles measured at the middle plane by varying the beam parameters at the source. Figure 9 shows the result of the fit for a two-charge state 75-kV bismuth beam (20+, 21+). The transverse emittances and Twiss parameters obtained by fit show that despite the axial symmetry of the extraction region, the beam is not axial symmetric which could be explained by a non symmetric plasma boundary inside the ion source.

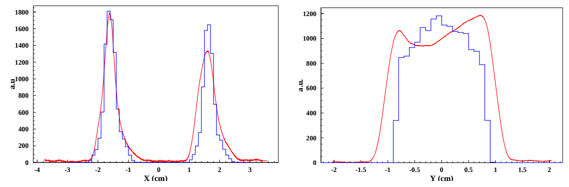

Figure 9: Horizontal (left) and vertical (right) beam profiles. The curves are the measured profiles and the histograms are the result of the TRACK fit.

Once the initial beam conditions are known, we used the computer model to find the element settings for the desired operation mode. A new fit procedure was developed to produce symmetric beam dynamics between the two bending magnets as this is a necessary condition to recombine the multiple charge state beams. Another fit was also used to find the setting of the last triplet for a perfect combination at the end of the LEBT where a beam profile monitor and a Pepper-Pot emittance meter are installed. Figure 10 shows the measured beam profiles and figure 11 shows the Pepper-Pot images at the end of the LEBT. A comparison of the element settings predicted by TRACK and the actual setting to combine the two beam shows maximum deviation of ~ 10 % which could be improved by checking the assumptions made in the simulations. We notice that the two charge state beams are almost perfectly combined.

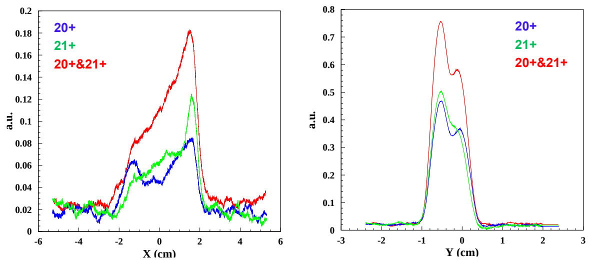

Figure 10: Measured beam profiles at the end of the LEBT for the individual Bi 20+ and 21+ beams and the combined beam.

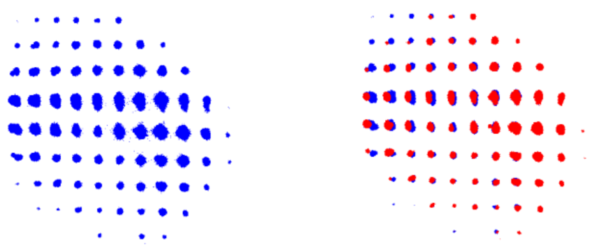
Figure 11: Pepper-Pot images of the combined beam (left) and the individual beams (right). Bi 20+ is in blue and 21+ is in red.

Further developments towards the Model Driven Accelerator

To be able to use a realistic 3D model online for real-time machine operations, we should be able to perform large scale optimizations on large number of processors (32768 processors or more). More optimization tools need to be developed for the commissioning phase to tailor the computer model to the actual machine by fitting the measured data. For this purpose, interfaces between the beam diagnostic devices and the computer model are needed to calibrate and analyze the data to input to the code. Numerical experiments could be used to test and fine tune the tools before implementation to the real machine by producing detector-like data. Only after all these developments, that a realization of the model driven accelerator will be possible. As a full scale application, we are proposing to apply this concept to the superconducting linac ATLAS at Argonne and to other existing machines.

SUMMARY

Optimization tools and methods are needed in every phase of an accelerator project, namely the design, commissioning and operations. No single algorithm could satisfy all these optimization needs. Different algorithms are being used in accelerator physics: local, global, standard and evolutionary. We have briefly reviewed and compared different classes of optimization algorithms and presented few applications in beam dynamics optimization. The ultimate goal of realizing the concept of the "Model Driven Accelerator" will require the development of a realistic 3D beam dynamics code with the appropriate set of optimization tools and large scale parallel computing capabilities. For new machines, we should take advantage of the commissioning phase to bridge the gap between the original design and the actual machine by tailoring the computer model to the machine. Obviously, a significant development effort is still needed for a full scale realization of the concept of the "Model Driven Accelerator".

REFERENCES

[1] "Using Online Single Particle Model for SNS Accelerator Tuning", A. Shishlo, Proceedings of HB-2008, Nashville, Tennessee, USA.

[2] "Measurement and Simulations of the J-PARC Linac", M. Ikegami, Proceedings of HB-2008, Nashville, Tennessee, USA.

[3] "Numerical Optimization", a book by J. Nocedal and S. J. Wright, Springer Series in Operations Research.

[4] "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", E. Zitzler and L. Thiele, IEEE Transactions on Evolutionary Computation 3 (1999) 257–271.

[5] "A simplex method for function minimization", J. A. Nelder and R. Mead, Comp. J. 7 (1965) 308.

[6] "Multiplier and gradient methods", M. R. Hestenes, Journal of Optimization Theory and Applications, 4 (1969), p. 303-320.

[7] "Newton's method for constrained optimization", J. Goodman, Mathematical Programming, 33 (1985), p. 162-171.

[8] "Quasi-Newton methods, motivation and theory", J. E. Dennis and J. J. More, SIAM Review, 19 (1977), p. 46-89.

[9] "Multivariate optimization of a high brightness dc gun photoinjector", I. V. Bazarov and C. K. Sinclair, Phys. Rev. ST Accel. Beams 8, 034202 (2005).

[10] "TRACK: The New Beam Dynamics Code", V. N. Aseev et al, Proceedings of PAC-2005 Conference, Knoxville, Tennessee.

[11] "Automatic longitudinal tuning of a multiple-charge-state heavy-ion beam", B. Mustapha and P. N. Ostroumov, Phys. Rev. ST Accel. Beams 8, 090101 (2005).

[12] "A Realistic Corrective Steering Algorithm: Formalism and Applications", B. Mustapha et al, Proceedings of PAC-2009, Vancouver, BC, Canada.

[13] "Front-End Design of a Muli-GeV H-minus Linac", P. N. Ostroumov et al, Proceedings of PAC-2005, Knoxville, Tennessee.

[14] "Optimized Design of an Ultra-Low Emittance Injector for Future X-ray FEL Oscillator", P. N. Ostroumov et al, Proceedings of PAC-2009, Vancouver, BC, Canada.

[15] "FRIB: A New Facility for Rare Isotope Beams", R. C. York, Proceedings of PAC-2009, Vancouver, BC, Canada.

[16] "Analysis and recombination of multiple-charge-state beams from an electron cyclotron resonance ion source", P. N. Ostroumov et al, Phys. Rev. ST Accel. Beams 12, 010101 (2009).