A SIMPLE REAL-TIME BEAM TUNING PROGRAM FOR THE KEKB INJECTOR LINAC

J.W. Flanagan, K. Oide, N. Akasaka, A. Enomoto, K. Furukawa,
T. Kamitani, H. Koiso, Y. Ogawa, S. Ohsawa, T. Suwada,
KEK, 1-1 Oho, Tsukuba, Ibaraki 305-0801 Japan

Abstract

A simple real-time beam tuning program has been developed for use at the KEK B-Factory injector linac. The program features the ability to adjust an arbitrary combination of parameters (e.g. magnet currents and RF phases) in order to optimize an arbitrary combination of monitor values (e.g. the ratio of a downstream beam current to gun current). Based on the downhill-simplex method[2][**?**], it requires no knowledge of beamline details. An additional "persistence" parameter is used to adjust the treatment of pulse-to-pulse variations of monitor values while mapping the parameter-space terrain, and controls the peak-holding performance in the presence of both statistical fluctuations and long-term drift. Preliminary results from the commissioning of the KEKB injector linac and plans for the future are discussed.

1 INTRODUCTION

The KEKB injector linac went through the first phase of commissioning from October 1997 to June 1998, following extensive upgrades to the prior facility.[1] During this time several techniques for tuning the beam were employed, such as the introduction of local bumps, energy feedback via klystron phase tuning (using orbital radius in the arc section as a diagnostic), and automatic orbit smoothing with the use of BPMs. These techniques were very successful in achieving the target beam currents. In addition to the above techniques, daily fluctuations in the machine state and the rise of non-linear behavior at higher target currents suggested that a non-model-based optimization tool might prove useful. As a result, a beam feedback tool based on the downhill simplex method was developed.

2 SOFTWARE

2.1 Environment

Figure 1 shows the software environment in use at KEKB, including the linac and the beam transport line between the linac and the KEKB storage rings. The SADscript environment[4] uses a Mathematica-like syntax with object-oriented extensions and provides easy access to all monitor and control records in the EPICS machine-state database, in addition to the beam optics and tracking functions of SAD. The user interface, shown in Figure 2, is built from a GUI toolkit based on Tcl/Tk.[5] During the Spring commissioning some linac components were accessed via stand-alone commands, though from Fall 1998 all instrumentation controls are based on EPICS, using a uniform method of access to monitors and controls and simple extensibility to beam-transport and KEKB ring components.

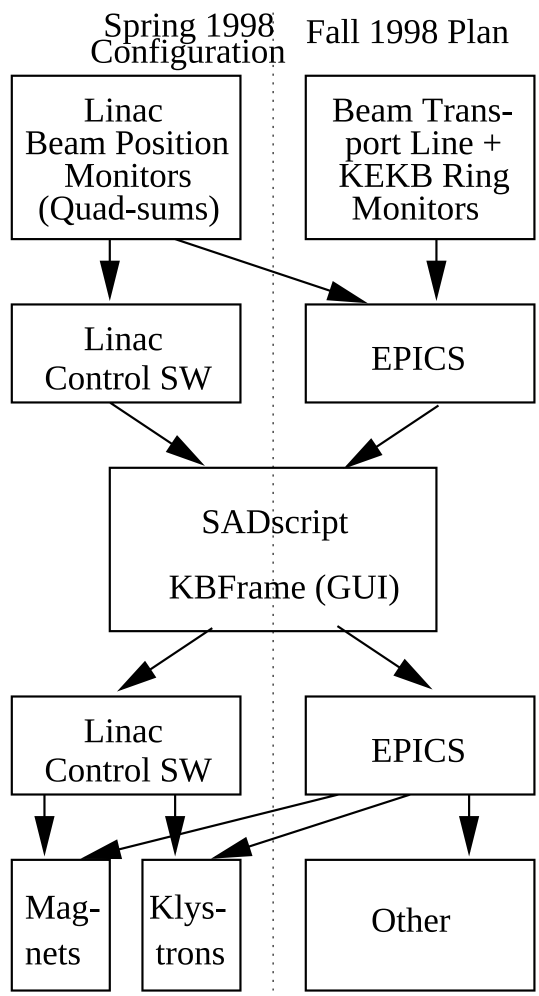

Figure 1: Beam Feedback Software Environment

2.2 Downhill Simplex Method

The Downhill Simplex optimization method is well known in the scientific community, being extensively discussed in Reference [2]. Given a function F of N parameters, an "amoeba" (simplex) with N+1 "feet" measures the height of the N-dimensional terrain, and attempts to walk down the slope in search of the bottom of a valley. When the amoeba seems to be near a minimum, it will resize to contract around the suspected minimum. It can also perform other operations depending on the terrain.

The method is quite general, and can find a minimum of

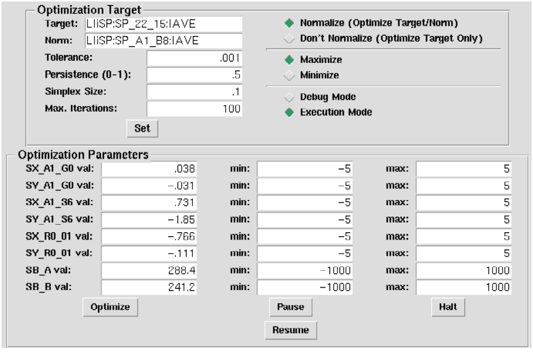

Figure 2: Optimization Tool Control Panel

an arbitrary function in N-dimensions without knowledge of derivatives or any other information about the function. This makes it applicable to cases where the functional dependence on the input parameters is difficult to calculate but relatively constant, making it a good choice as a beam fine-tuning tool. The method was applied to the minimization of vertical emittance during operation of the Tristan ring at KEK.[3] A drawback is that while the downhill-simplex algorithm expects F to be arbitrary, it does not expect its value at one point in parameter space to fluctuate from measurement to measurement. Our solution to this problem is the introduction of a parameter called "persistence."

2.3 Persistence

There were two goals for the program. One was to fine-tune a beam that had already been tuned using other methods, to correct for non-linear or non-calculable effects remaining. For this purpose, the program is largely operating like a human operator with faster reaction time. The second goal is to maintain a good machine state, continuously adjusting for drifts in machine performance over longer-term intervals.

As mentioned, the simplex consists at any one time of a list of measured values for N+1 locations (feet) in parameter space. Based on the heights of these feet in the terrain, it will attempt to feel out new and better locations and relocate the least-optimally located feet to better positions. In principle, the most reliable measurement for each foot would be the result of averaging over many pulses. For our initial experiments, we used a 10-pulse average at each location. (Such averaged measurements are available in the EPICS database, along with single-shot measurements.) When the amoeba is near a suspected optimum, it will contract around in size around the most optimally placed foot, and continue. At this point, before the contraction, the value of the optimizing function is remeasured at the current optimal location. The value F at that location is then assigned a weighted average of the old and new values:

$$F = [PF_{old} + (1 - P)F_{new}],$$

where $0 \leq P \leq 1$ is the persistence. A persistence of 0 means that the algorithm retains no memory of old values, and 1 means that it never remeasures the optimum location twice as long as another foot does not find a better location. The simplex is then resorted to verify that the optimum is still the optimum, or reassign the optimum to a different foot if necessary. This periodic remeasurement only at suspected optima, instead for example of remeasuring all feet, is a time-saving compromise. Also, since each remeasurement of the optimizing function at a point in parameter space requires physically changing the machine parameters to the corresponding state in order to make the measurement, this procedure avoids putting the machine into states that are already believed sub-optimal.

2.4 Performance

The performance of the algorithm was tested during the commissioning of the 3.5 GeV positron line. Electrons were accelerated to 1.5 GeV in the first two sectors ("A," "B") of the linac, then pass through a $180\deg$ arc section, and are accelerated to 3.7 GeV by the end of the next two sectors ("C," "1"), at which point a target is inserted when the linac is in positron mode. The positrons from the target are then accelerated to 3.5 geV in the remaining 4 sectors ("2"-"5") for transport to the KEKB low-energy storage ring.

For the case of positron production, the quantity being optimized was the ratio of the beam current downstream of the positron target to the beam current just after the gun. The parameters being varied were one pair of x and y steering magnets just after the gun, another pair after the buncher section, and another pair just at the entrance to the arc section, in order to minimize emittance growth due to transverse wake effects on the beam from injection and beam pipe misalignments. In addition, the RF phases of the first two sectors were varied to minimize the energy spread. The energy acceptance of the arc section is $\pm 1.5\%$

The results were encouraging, with the highest positron current being recorded for the commissioning period at several percent over previous performance. Typically most of the improvement was seen within the first 50 iterations. The persistence setting was found useful in keeping the machine near an optimum – a persistence of zero led to drift away from optimal settings even after the optimum had been found, and a persistence of one tended to result in the simplex becoming hung up on a false peak due to fluctuations. Figure 3 shows the optimization curve for a persistence setting of 0.5, which shows good peak-holding performance. The speed of the algorithm largely depends on the rates at which the machine state can be changed and the monitor values can be read.

3 SUMMARY

A real-time beam optimization tool has been developed for the KEKB injector linac. Performance was found to be enhanced by the introduction of a persistence parameter.

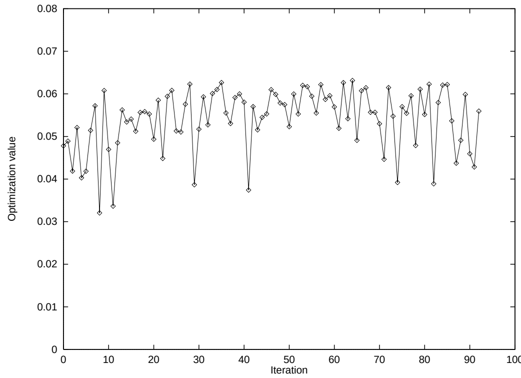

Figure 3: Example Optimization Curve for persistence$=0.5$, showing the value of the ratio of the current downstream of the positron target to that just after the gun, as a function of iteration.

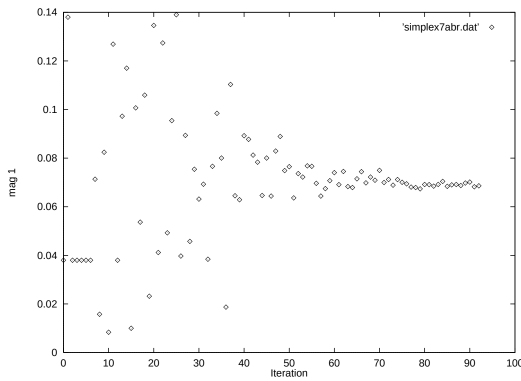

Figure 4: Diagnostic plot of parameter value history during an optimization run. In this example the parameter is the first steering magnet, which stabilizes at a new level by the 70th iteration.

Further testing and extension of applicability to the beam transport line and KEKB rings is underway.

4 ACKNOWLEDGEMENTS

The authors are indebted to the efforts of Dr. Noboru Yamamoto and the KEKB Control Group, who established the EPICS environment at KEK, and Prof. Shin-Ichi Kurokawa for his continual support. Work supported by the Japanese Ministry of Education, Science, Sports and Culture (Monbusho) and the Japan Society for the Promotion of Science.

5 REFERENCES

[1] Y. Ogawa *et al.*, "Commissioning of the KEKB Linac", KEK Preprint 98-125, and proceedings of Linac98. (1998).

[2] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, "Numerical Recipes in C", Cambridge University Press, pp. 408-412 (1988). Note that they refer primacy to: J.A. Nelder and R. Mead, *Computer Journal*, vol. 7, pp. 303-313.

[3] M. Kikuchi, private communication.

[4] http://www-acc-theory.kek.jp/SAD/sad.html

[5] http://faraday.kek.jp/KEKBFrame/contents.html
Akasaka *et al.*, "Operation Software for Commissioning of KEKB Linac Programmed with SAD," proceedings of APAC98.