USE OF A GENERAL-PURPOSE OPTIMIZATION MODULE IN ACCELERATOR CONTROL*

L. Emery†, M. Borland, H. Shang, Argonne National Laboratory, Argonne, IL 60439, USA

Abstract

The SDDS EPICS toolkit has recently been enhanced by the addition of a general-purpose optimizer. The tool `sddsoptimize` is easily configured and has features that make it robust. The sddsoptimize program has been incorporated in many new Tcl/Tk applications used for various Advanced Photon Source tune-up operations, such as injection tune-up for the storage ring, coupling minimization of the storage ring, and the automatic phasing of the linac.

INTRODUCTION

The development of high-level software for operations and machine physics measurements at the Advanced Photon Source (APS) has progressed steadily since the start of operations. The foundation of our software is the robust self-describing data set (SDDS) file protocol [1], and the accompanying SDDS Toolkit of applications. In addition an SDDS-compliant EPICS toolkit [2] acts as the intermediary between the EPICS control system and SDDS-protocol data files.

A generic optimization program `sddsoptimize` has been added to the EPICS toolkit. The ability to do optimization complements the generic feedback tool `sddscontrollaw`, with both applications handling the set of adjustment problems likely to be seen in accelerators. The features of `sddsoptimize` and recently added features of `sddscontrollaw` are covered in reference [2].

A feedback process relies on several readbacks being (at least approximately) linearly dependent on a set of actuators. Using a correction matrix to evaluate new set points for the actuators, feedback usually takes a few steps to converge. In optimization, there is only one readback (or a sum of squares of readbacks), there is no matrix to give an initial direction for actuator changes, and the readback is most likely a nonlinear function of the actuator changes. As a result, one can expect the optimization process to take many more steps to converge, especially with a large number of variables. This is a drawback, but there is no other choice in these cases.

As mentioned earlier, feedback and optimization are applicable to separate sets of problems. However, a feedback problem with one or two variables can be formulated into an optimization problem if the convergence time is not an issue. Optimization obviates a correction matrix and its configuration files are easier to maintain.

In accelerators, optimization is usually done manually or with software written for a particular optimization problem. This approach may cost tune-up time (e.g., inefficient tweaking) and effort (maintenance of customized software). As far as we know, no one has written an optimization control system application that is configurable for any problem. In this paper we will list the features of `sddsoptimize` and give examples of its use at the APS.

FEATURES

The optimization criterion is the rms value of one or more EPICS process variables (PVs), or else the value obtained by running a script. In the former case, the PVs are listed in an SDDS file with optional values for weights, target values, and tolerances. In the latter case, the measurement script can be used to perform more general operations, which may or may not involve accessing PV values. The variable PVs are listed in an SDDS file as well with a range defined by lower and upper limits data. Setting the values of control variables can be replaced by running a “variable script” (given by the varScript option) so that the program can effectively set PVs in an arbitrarily complicated fashion, or even perform optimizations that do not involve PVs. There are no obvious IOC calls if both variable and measurement scripts are provided.

Two optimization methods are provided: Simplex and successive 1D optimization (also called 1D scan). Simplex is a multidimensional minimization method that requires only function evaluations. It is frequently the best method if the computational burden is small. By default, our Simplex method makes explicit use of a one-dimensional minimization algorithm as a part of the computational strategy, since this often will make the optimization proceed faster; this can be disabled in cases where it is found not to help. The successive 1D scan method allows minimizing of the target with respect to each parameter separately and in turn. The main disadvantage is that if the optimal changes of the parameters are mutually dependent, this method may converge very slowly toward the minimum. Nevertheless, it runs efficiently when the variations are quasi-independent.

The program performs minimization by default and will perform maximization if the “-maximize” option is given.

`sddsoptimize` can be used to adjust knob PVs, which are predefined linear combinations of PVs. Examples are knobs for orbit bumps or ganged timing control for a set of kicker magnets.

The control variables for readback and variables are specified by SDDS files. The knobs PVs are configured by SDDS files.

To make the optimization robust, a series of validity tests on PV values are implemented by means of an additional SDDS file containing the names of PVs and their corre-

---

* Work supported by U.S. Department of Energy, Office of Basic Energy Sciences under Contract No. W-31-109-ENG-38.

† emery@aps.anl.gov

sponding limit values. The optimization is suspended if one of the tests fails. This can be used to avoid processing invalid data and to terminate the program if it adjusts settings beyond a safe or reasonable range.

The optimization can be stopped at will by the user using ctrl-c (i.e., the UNIX SIGINT signal). The best settings obtained so far will then be implemented before the program terminates.

`sddsoptimize` optionally logs settings and results to an SDDS file. This file can be used to view results during or after an optimization and also to set up a new optimization.

APPLICATIONS

We have configured several instances of `sddsoptimize` for various optimization problems at APS. They are usually wrapped within a customized graphical user interface (GUI) that uses standardized Tcl/Tk widget libraries. The controls available on the GUIs are: an entry for the log file directory, a button to set up the accelerator condition, a button to start optimization, and a button to plot the progress of optimization. For simplicity, most of the optimization parameters in the GUI are hard-coded. However, where flexibility is required, some parameters are available for modification in the GUI. A pop-up window displays the output of `sddsoptimize` with an abort button.

Averaging of the readbacks is used in all cases, as the simplex search algorithm is sensitive to noise. Several cycles of optimzation is recommended to ensure that noise does not cause a false minimum.

Maximizing Injection Efficiency

The efficiency is particularly difficult to optimize at APS since the available aperture of the storage ring (SR) is relatively small, the booster beam is relatively large, and the trajectory in the transfer line jitters to some degree. The optimization problem is that of steering with unknown aperture coordinates and beam absolute positions (at least to the accuracy desired). If we had a beam position monitor (BPM) in the transfer line where the aperture is smallest, then a feedback process would be used to reproduce the trajectory there.

A measurement script calculates an average efficiency from the charge stored in several pulses divided by the total charge injected. The variable PVs are the two SR septums and a pure x-coordinate “Entrance” knob that uses the last two horizontal plane correctors of the transfer line. Though Simplex optimization finds the best direction for the variables after testing many directions, we occasionally help the method by creating knobs that might speed up the search initially. In this case, the first SR septum PV variable was replaced with an “Exit” knob that has the two SR septum PVs combined to give a pure x-coordinate at the end of the downstream septum. The aperture is known to be small there, so an adjustment of the trajectory at that aperture is thought to make the optimum search efficient.

The optimization usually finds a peak rather than a broad optimum. Depending on the SR lattice, one may obtain a peak efficiency as low as 80% (for the low-emittance lattice). Figure 1 shows an example of optimization of the injection efficiency for the low-emittance lattice. Some of the actuator set points produce a bad result. This is necessary in the optimization process, as negative directions for the simplex are found and excluded.

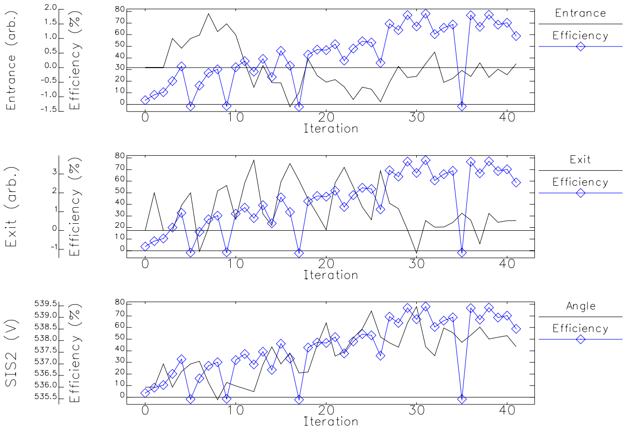

Figure 1: Injection efficiency optimization. Two variables “Entrance” and “Exit” are knobs mentioned in the text. The downstream septum is named S1S2 in the control system and functions as a trajectory-angle adjustment.

The injection optimization is done during the machine studies period that precedes top-up operation. The injection rate is 2 Hz, so the optimization proceeds relatively quickly. The optimization has not been implemented during top-up operation yet, given the low repetition rate of injection (once every two minutes), and the possible long periods of low efficiency when the variables are searching for the best path. If the injection efficiency during top-up operation drops significantly, we presently do a 1D tweaking on one septum to prevent getting too far from reference set points.

SR Beam x-y Coupling Minimization

SR beam x-y coupling minimization uses measurements of the vertical beam size of the X-ray pinhole beam image as readbacks. Knobs for sine and cosine harmonics of ten skew quadrupoles are the variables.

Note that minimizing the coupling could be handled by a complicated feedback process where coupling matrix elements at various points around the ring would be measured with several experiments. However, we already know the knobs selected have the largest influence on the x-y coupling. Because optimization is faster, we selected this method to minimize the coupling. Figure 2 shows a good example of coupling reduction during optimization.

On-Axis Injection Set-Up

To set up on-axis injection for some machine physics experiments, the kickers are adjusted so that the first-turn

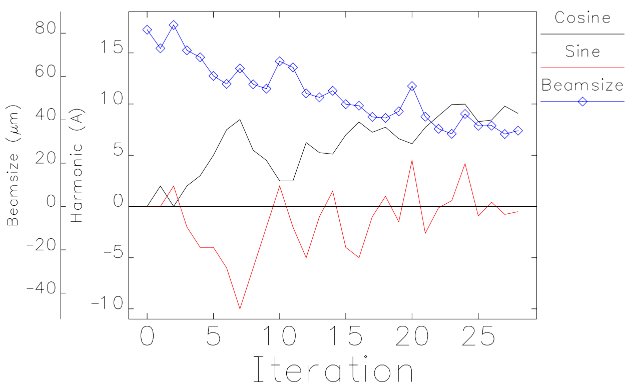

Figure 2: Coupling minimization.

trajectory coincides with the closed orbit. The readbacks are the BPM readbacks in the horizontal plane, sampled on the first turn and averaged for 32 injections. The range of sectors of BPMs is selectable from the GUI. Limiting the range of BPMs was useful when a local perturbation was present and sectors downstream of the perturbation needed to be ignored. The variables are the two injection kickers downstream of the injection area.

A feedback could have accomplished the same thing, since the BPM readbacks are linearly dependent on the kicker strength. However, this would have required a lattice-dependent linear correction matrix for each lattice type, which implies extra maintenance on trajectory configuration files. The implementation for optimization happens to be simpler in this case, and the optimization files work for all lattices. Figure 3 shows an example of trajectory correction starting with the kickers in a nominal accumulation configuration that produced an rms of 8 mm, and finally converging with an rms of 0.5 mm.

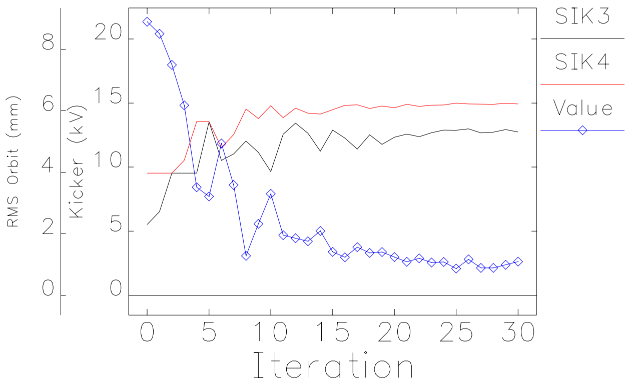

Figure 3: Steering the first-turn trajectory on axis.

Injection Closed-Bump Set-Up

A closed bump is created by a set of four symmetrically-placed kickers in the injection area. A closed bump optimization is required to make best use of the SR aperture at the septum, i.e., bring the stored beam as close as possible to the septum wall without losing particles.

The closed bump condition is obtained by minimizing the betatron amplitude caused by the kickers' pulses. A script reads the turn-by-turn history of a selected BPM, calculates an amplitude, and averages over many pulses. The first kicker of the group is fixed by the user setting the overall amplitude of the bump. The variables of the optimization are the two downstream kickers. The second kicker is essentially equivalent to the third one, and both are equalized in amplitude after an optimization cycle.

Booster-to-SR rf Phase Adjustment

Booster-to-SR rf phase adjustment is optimized for centering the injected beam in the rf bucket. A script downloads BPM turn-by-turn history and determines the amplitude of the data. The variable is an rf phase PV.

Beam-Based Optimization of rf Phase and Power in the APS Linac

The injector has a four-magnet-chicane bunch compressor after the first set of accelerating structures. A fluorescent screen at the center of the chicane allows viewing of the linac beam energy and energy spread. A Tcl/Tk application was written that periodically inserts the screen and then runs `sddsoptimize` to reduce the energy spread while keeping the energy constant using `sddscontrollaw`. The application was found to be robust and is quick enough to run between top-up events and thus keep the upstream part of the linac properly phased. We also envision using it as part of the start-up procedure to obviate the need for manual rf phasing by operators.

Maximizing Capture Efficiency of the Particle Accumulator Ring

Another injector application is to maximize capture efficiency of the particle accumulator ring (PAR), which is downstream of the linac. The PAR is sensitive to energy drift in the linac. We have found that this is not completely compensated by restoring the trajectory, even though we have BPMs in dispersion regions and allow the feedback to change rf power levels. Hence, a Tcl/Tk application was written that uses `sddsoptimize` to adjust the power and phase in the last linac structure, along with the set points for the trajectory feedback. At present, this is a work in progress, as we find the algorithm does not always result in an improvement, probably due to the amount of noise in the readbacks.

REFERENCES

[1] M. Borland, "A Self-Describing File Protocol for Simulation Integration and Shared Postprocessors," 1995 PAC, Dallas, Texas, 2184 (1996).

[2] H. Shang et al., "New Features in the SDDS-Compliant EPICS Toolkit," these proceedings.