USPAS 2015 June – Beam Based Diagnostics

Computer Class – Session 4

Xiaobiao Huang (6/9/15)

Goals: Online optimization with RCDS and other algorithms.

Part 1: SPEAR3 coupling correction with RCDS

Enter subdirectory ‘optimization’ and open the script “run_RCDS_test.m” and follow the instructions.

Cell 1: Initialization

Load the initial lattice and calculate coupling with ‘calccoupling.m’. The initial coupling is 0.88%, generated by skew quadrupole components on 29 of the 72 sextupole magnets.

Cell 2: Set up optimization parameters

The optimization parameters are 13 skew quadrupoles (also on the sextupole magnets). These 13 skew quads are set to zero in the initial lattice.

In the simulation the ranges of the skew quad gradient are $[-0.3, 0.3]\ m^{-2}$. The initial solution is with all 13 skew quads at zero. For the objective function, the parameters are all normalized to the range $[0, 1]$. Therefore, the initial solution is 0.5 for all variables.

Cell 3: Evaluate random noise level

The objective function is defined in ‘func_obj.m’. It first map the normalized solution to the actual parameter values. Then the solution is applied to the model by setting the skew quad values. The coupling ratio is calculated using ‘calccoupling.m’. The objective is the loss rate, which is calculated from the coupling ratio. Noise is introduced to the objective function by adding a random error to the beam current. The noise level for the beam current measurement (DCCT) is assumed to be 0.003 mA.

The noise level of the objective function is evaluated by repeating the measurement 20 times and calculating the standard deviation. This noise level is needed by the RCDS algorithm.

Cell 4: Run the RCDS algorithm

To launch the RCDS algorithm, call the function ‘powellmain’ and specify the objective function, the initial solution, the initial step size, the initial direction set, and a few other optional arguments.

Here the initial direction set is derived from SVD of the orbit response matrix Jacobian matrix with respect to the 13 skew quads. This is approximately a conjugate direction set of the coupling correction problem.

After the algorithm is launched, it will print out the progress in the command window. Run data are also saved to the global variable ‘g_data’. The initial objective function is roughly -0.6 mA/min. You can terminate the program (with ‘Ctrl + C”) after about 500 function evaluations or when the objective function reaches -2.5 mA/min. After termination, save the workspace with, for example,

```
>> save tmpRCDSdata
```
Change the file name if you want to avoid it from being overridden.

In an experiment, it is often advisable to use the ‘diary’ function of Matlab to keep a record of the commands and output in Matlab.

Cell 5: Analyze data and apply the best solution
Each raw in ‘g_data’ represents an evaluated solution, which include the parameter values, the objective function, and additional information. In this case, the actual coupling ratio (no noise) is included. The history of the evaluated solution can be analyzed in the actual, or normalized parameter space.

The function ‘process_scandata.m’ is used to sort and plot the history data. Results of a test run are shown in Figure 1 below.

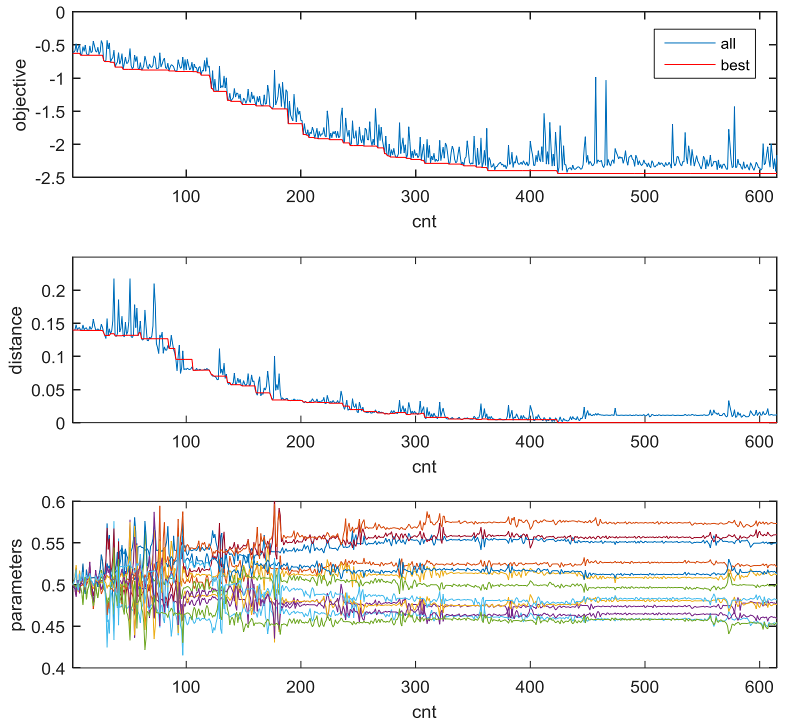

Figure 1: History of a test RCDS run for the coupling correction optimization. Top plot is the objective function, middle plot is the distance to the best solution, and the bottom plot shows the history of all 13 parameters.

Note that the best solution is not the last solution. After the best solution is located and applied to the model. We calculate the coupling ratio. In the test run shown above, the coupling ratio for the best solution is 0.0297%.

Repeat the RCDS run after changing the initial direction set to the unit matrix.
Repeat the RCDS run after changing the objective noise level (by changing the value of ‘duration’ in ‘func_obj.m’.). Save data to different files.

Part 2: SPEAR3 coupling correction with Downhill Simplex (Nelder-Mead) algorithm

We now try to solve the same problem with the Downhill Simplex method. Open the script 'run_Simplex_test.m' and follow the instructions.

Cell 1&2: Initialization and parameter range setup

This is the same as for the RCDS method.

Cell 3: Run the Downhill Simplex method

Using the Matlab implementation in function 'fminsearch', we run the algorithm with

```
>> [x, fv] = fminsearch(@func_obj,x0);
```

Terminate the run after about 300 (or more if you want) function evaluations. When the noise level is as for the case with a duration of 10 s, the objective function stopped gaining after less than 50 evaluations in one test run.

With 'duration=10' and beam current noise level 'rms_dcct=0.003', save data to file

```
>> save data_Simplex_10s_3uA
```

Cell 4: Analyze and plot data

Use 'process_scandata' to sort and plot the history data. With noise, the algorithm is stuck when the objective function is reduced to about $-0.8\sim-1.0$ mA/min.

Repeat Cell 3 and 4 run after setting the beam current noise to zero with 'rms_dcct=0.0'. Save data to

```
>> save data_Simplex_0uA
```

Part 3: SPEAR3 coupling correction with NSGA-II algorithm (MOGA)

NSGA-II is a popular multi-objective genetic algorithm. We apply it to the coupling correction problem (single objective). Go into the subdirectory 'NSGA-II' and open the script 'run_MOGA_test.m'.

Cell 1: Initialization

Lattice and parameter setup are the same as before.

The population size (number of solutions in one generation) is 60. The crossover probability is set to 90%. The initial population is drawn from a hyper-cube with size of 0.2 (in the normalized parameter space) which is centered on the solution with all skew quads set to zero.

The global variable 'mu' and 'mum' control the width of the probability distribution function for the crossover and mutation operations, respectively.

Cell 2: Launch the NSGA-II algorithm

The algorithm is launched with the ‘nsga_2m.m’ function. All data are saved to a global variable ‘g_data’ and also written to a file for each generation. It can be terminated anytime with ‘Ctrl+C’. But one should save ‘g_data’ to a file immediately.
The algorithm can be re-launched from the last saved generation.

Cell 3: Process and analyze data.
Data saved in the ‘g_data’ variable can be processed and plotted as before. One can also plot the evolution of data through the generations. In a test run, the objective function is reduced to only about −1.2 mA/min in about 5800 evaluations.

Part 4: SPEAR3 coupling correction with Particle Swarm Optimization (PSO) algorithm

Particle swarm optimization algorithm is a powerful stochastic optimization algorithm. It has demonstrated better performance than MOGA for many problems. We use this method on the coupling correction problem.

Cell 1: Initialization
Lattice and parameter setup are the same as before.

Cell 2: Launch the PSO algorithm
The initial population is centered on the solution with all skew quads off (which is also included), randomly selected from the hyper-cube of size 0.2. The population size is 60.

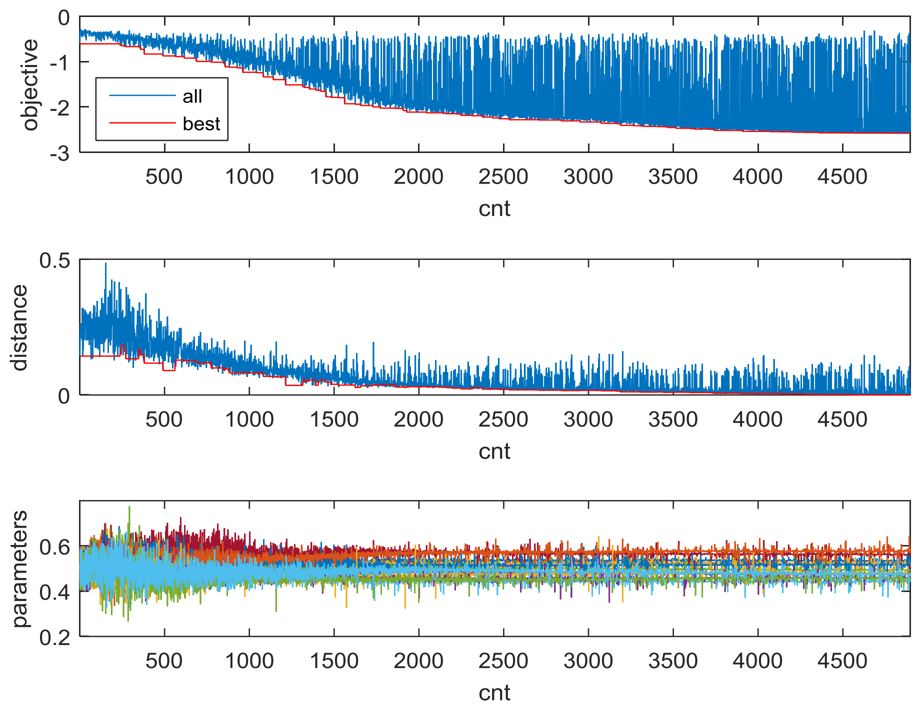

Figure 2: history of a PSO run for the coupling correction problem.

Cell 3: Process and analyze data
Sort and plot the history data. One can also plot the evolution of the global best solutions. For a test run, the coupling ratio of the best solution is down to 0.025% (objective function is −2.57 mA/min).

Cell 4: Compare solutions from PSO and RCDS
The parameter values of the best solutions of the PSO and RCDS are compared. The two algorithms basically find the same solution. PSO is able to find a better solution through fine search, while RCDS converges much faster.

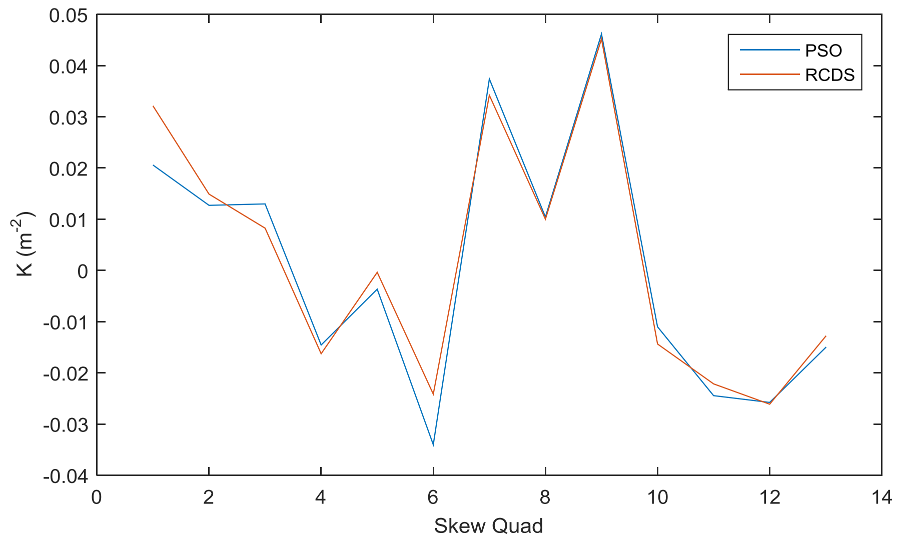

Figure 3: Comparison of parameters for the RCDS and PSO best solutions.

Part 5 (optional): Coupling correction with LOCO
One can load the initial lattice (with coupling), generate LOCO data with ‘plotfamily’, and perform LOCO fitting and optics/coupling correction. Compare the coupling ratio and parameter values for the corrected lattice to the RCDS and PSO solutions. Iterate the correction if necessary.