Development and Application of Online Optimization Algorithms

Presented at NAPAC 2016

Xiaobiao Huang
SLAC National Accelerator Laboratory
October 14, 2016

U.S. DEPARTMENT OF ENERGY
Office of Science
SLAC NATIONAL ACCELERATOR LABORATORY

Outline

- Motivation
  - Beam based correction vs. beam based optimization
  - Manual tuning vs. automated tuning
- Development and test of the RCDS algorithm
  - The RCDS algorithm
  - Test with SPEAR coupling correction: simulation and experiment
  - Usage of RCDS package
  - Performance stabilizer
  - AutoTuner
- Applications of the RCDS algorithm
- Other online optimization algorithms
- Summary

Achieving optimal accelerator performance

SLAC

The process:

The ideal scenario:

The process		The ideal scenario
Accelerator design	←	All physics principles are known and included in the model. Optimization in design is thorough and complete.
Build the machine according to design	←	No mechanical (machining and alignment) and magnetic errors.
Set set-points to design values	←	Calibration is accurate. Monitors are accurate. No variation with time or environment.
Machine in reality	←	Machine as built is identical to the design.

However, the reality is never ideal.

Solutions: (1) Beam-based correction.
(2) Beam-based optimization (tuning).

Beam based correction

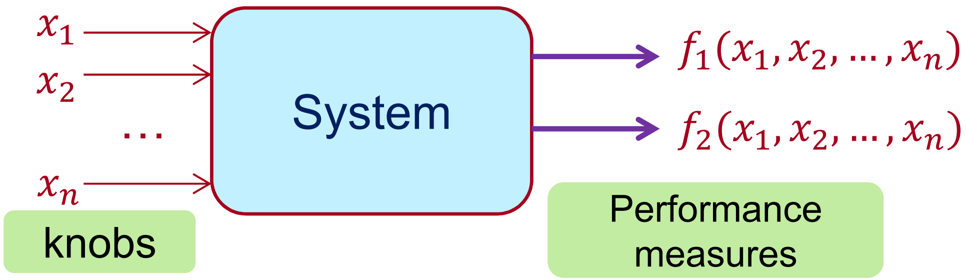
Beam based correction: correct the operating condition of a subsystem toward the ideal (design) condition through beam based measurements and a deterministic procedure.

	Actuators (knobs)	Diagnostics (monitors)	Deterministic method	Target
Orbit correction	Orbit correctors	BPMs	Orbit response matrix	Ideal orbit
Optics correction	Quadrupole correctors	Beta, phase advance, orbit response matrix	Response (Jacobian) matrix	Design optics

What if any of diagnostics, deterministic method, or ideal target is missing?

Beam based optimization – tuning

Beam based optimization (tuning): adjust the operating condition to optimize machine performance directly.



We know the system works – changing input leads to performance responses. But we don't know **exactly** how it works – the functions are unknown.

Machine tuning is a multi-variable and (potentially) multi-objective optimization process. The function(s) is evaluated through the machine.

Manual tuning vs. automated tuning

Manual tuning
Knob changing by human hands, data processing and decision making by human brain.

Automated tuning
Knob changing, data processing, and decision making all by computer.

Manual tuning	Automated tuning
Slow	Fast
Human dependent	Human independent
Limited to small problems (few knobs)	Scalable to large problems

Why isn't automated tuning popular yet, long after machines are completely computer controlled?

Probably because of the lack of reliable, effective optimization algorithm.

Challenges to automated tuning algorithms

- Noise – functions evaluated on machine have noise.
  - Most of the traditional methods are designed for smooth functions.
- Efficiency
  - Need to converge to the optimum fast.
- Safety, reliability, robustness
  - Survive occasional outliers.
  - Cause no disaster when machine mal-functions.
- (previous) Common auto-tuning algorithms
  - Iterative 1D scan, Downhill simplex*, Random tries

Robust conjugate direction search (RCDS) is ideal for automated tuning.

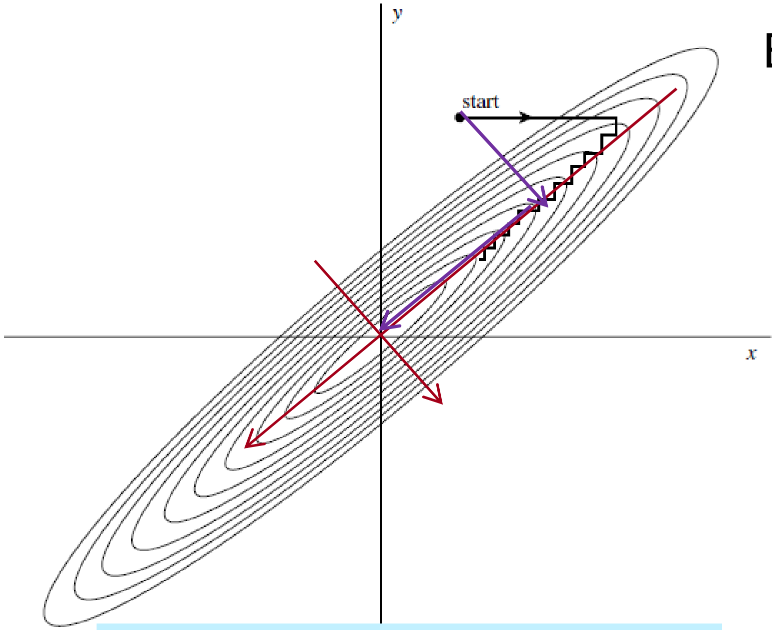
*L. Emery et al, PAC2003, implemented 1D scan and the downhill simplex method.

The development of the RCDS algorithm

- The development was motivated by the need to optimize storage ring nonlinear beam dynamics.
  - Correction of nonlinear dynamics is difficult – lack of direct diagnostics, deterministic method, and even target.
- Robust conjugate direction search (RCDS)* performs iterative search over conjugate directions with a robust (against noise), efficient line (1D) optimizer.
  - The conjugate direction set may be updated with Powell's method.
  - The 1D robust optimizer is designed to deal with noise.

*X. Huang, J. Corbett, J. Safranek, J. Wu, "An algorithm for online optimization of accelerators", Nucl. Instr. Methods, A 726 (2013) 77-83.

Search over conjugate directions



Inefficient search directions

It takes many tiny steps to get to the minimum when searching along x and y directions.

Efficient search directions: conjugate directions

A search over conjugate direction does not invalidate previous searches.

Directions $\mathbf{u}$ and $\mathbf{v}$ are conjugate if

$$\mathbf{u}^{\mathrm{T}} \cdot \mathbf{H} \cdot \mathbf{v} = 0$$

with $\mathbf{H}$ being the Hessian matrix of function $f(\mathbf{x})$, $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$.

Around the minimum

$$f(\mathbf{x}_m + \Delta\mathbf{x}) = f(\mathbf{x}_m) + \frac{1}{2}\Delta\mathbf{x}^{\mathrm{T}} \cdot \mathbf{H} \cdot \Delta\mathbf{x}.$$

Powell's method can update the directions using past search results to develop a conjugate set.

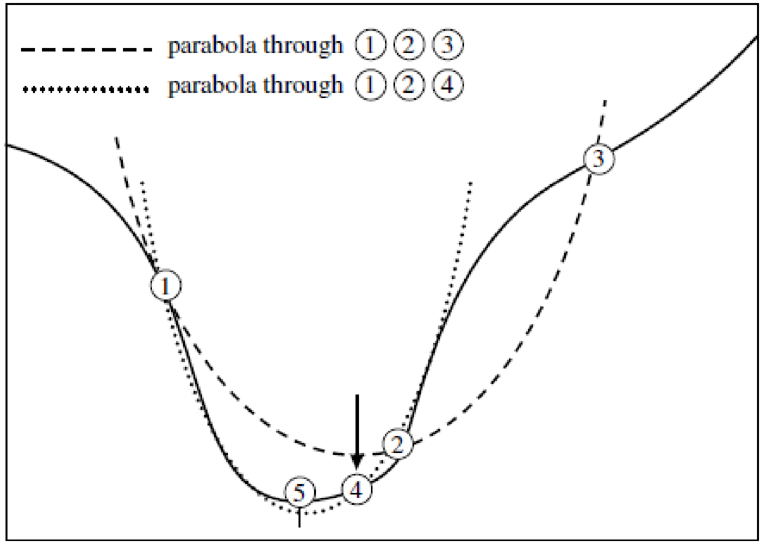
*W.H. Press, et al, Numerical Recipes

*M.J.D. Powell, Computer Journal 7 (2) 1965 155

Anatomy of a line optimizer that is sensitive to noise

Line optimizer – Brent's method

Step 1: Initially bracketing the minimum.
Step 2: Successive interpolation to converge to the minimum.



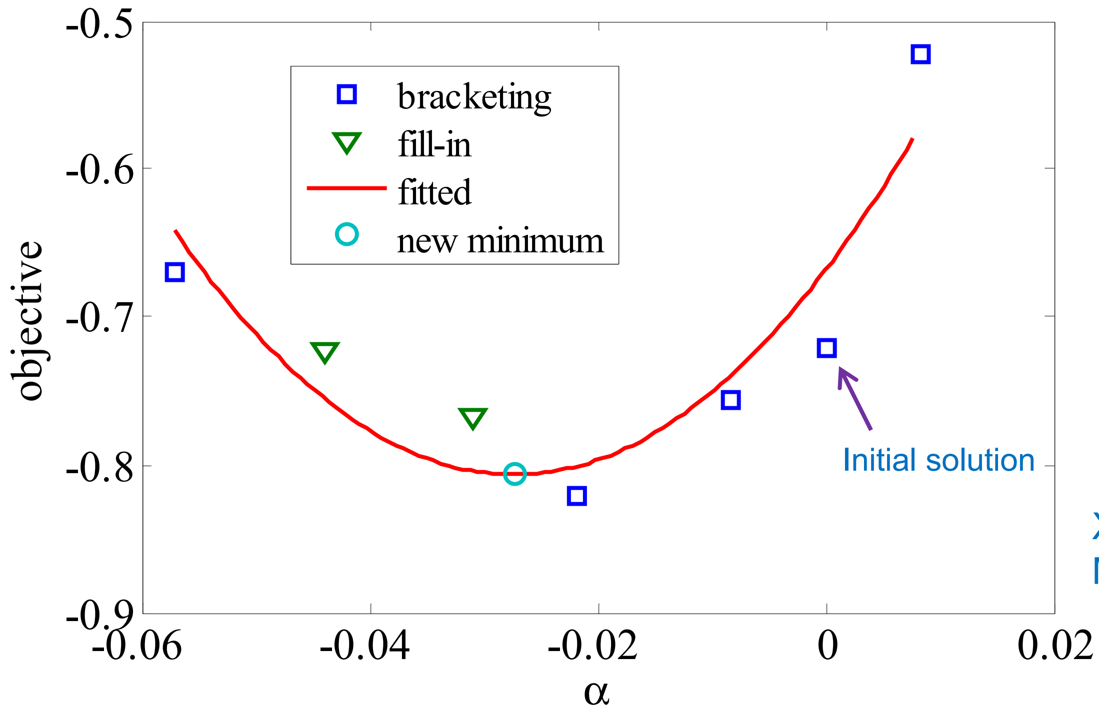
Inverse quadratic interpolation (figure from Numeric Recipes*.)

With noise, the comparison of values in both steps can go wrong and the algorithm won't converge.

*W.H. Press, et al, Numerical Recipes

The robust 1D optimizer

The robust optimizer is aware of noise in bracketing and uses noise level to filter out outliers. Noise level is detected before optimization.



X. Huang et al, Nucl. Instr. Methods, A 726 (2013) 77-83.

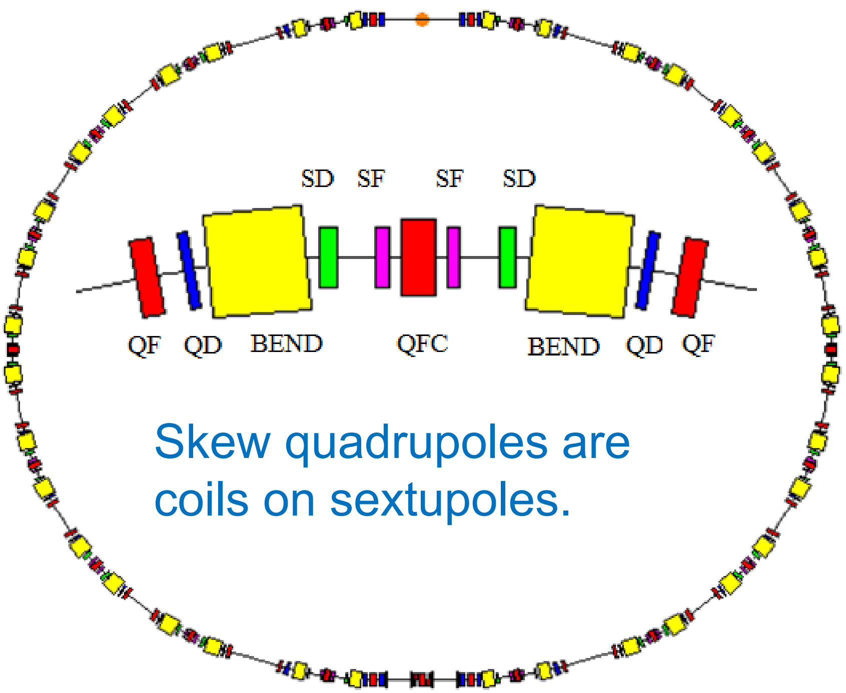
Bracketing: step size is increased in the search. Bracket ends are higher than minimum by 3 noise sigma.
Fitting: fill in additional points when necessary to better sample within the bracket and then fit a parabola.

Implementation of RCDS

- Parameters are bounded and normalized to [0, 1]
  - Parameters in online optimization always have limited ranges.
  - Keeping parameters within pre-defined ranges is a safety measure.
  - Normalizing parameters makes algorithm code independent of actual problems
- Powell's method of automatic updating of conjugate direction set is implemented.
  - In real life problems usually only a few directions are replaced before terminating. So we hardly benefit from this procedure for online problems.
- The interface between the algorithm and a particular application is the objective function and a simple setup script.

Testing the algorithm with a simulation problem

Testing problem: coupling correction for the SPEAR3 storage ring with skew quadrupoles.



Skew quadrupoles are coils on sextupoles.

The SPEAR3 storage ring

Objective: maximize beam loss over 6 seconds (Touschek loss rate $\propto 1/\sigma_y$).

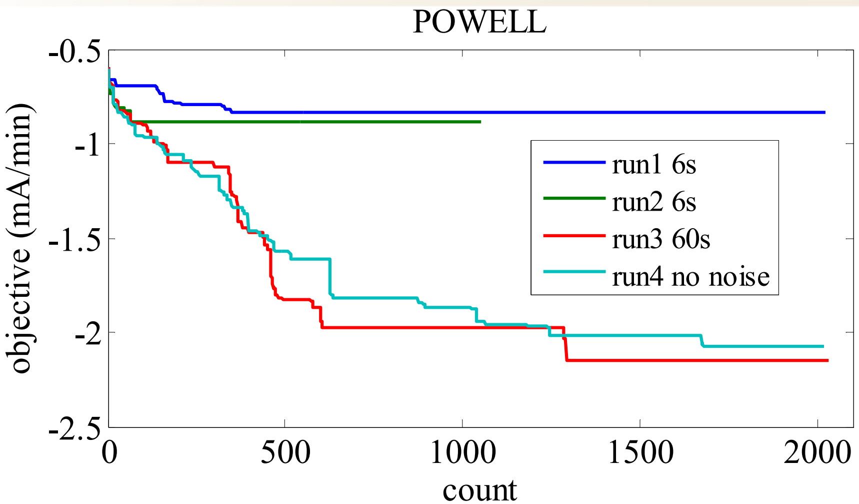
Knobs: 13 skew quads

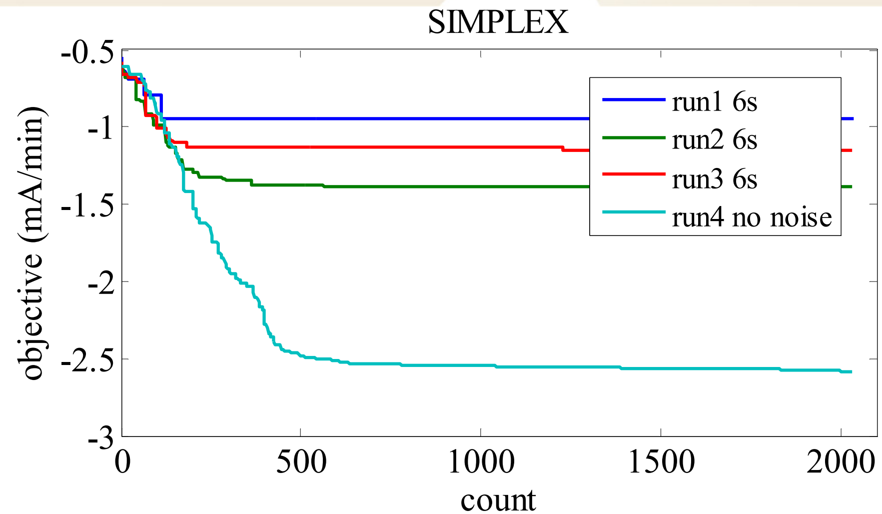
Setup: (1) errors are added to 42 skew quadrupoles. Initially all 13 correcting skew quads are off, with coupling ratio of 0.9%.
(2) Noise level for loss rate is about 0.06 mA/min, with initial loss rate at 0.6 mA/min.
(3) Initial conjugate direction set is from SVD of the Jacobian matrix of the orbit response matrix w.r.t. skew quads.

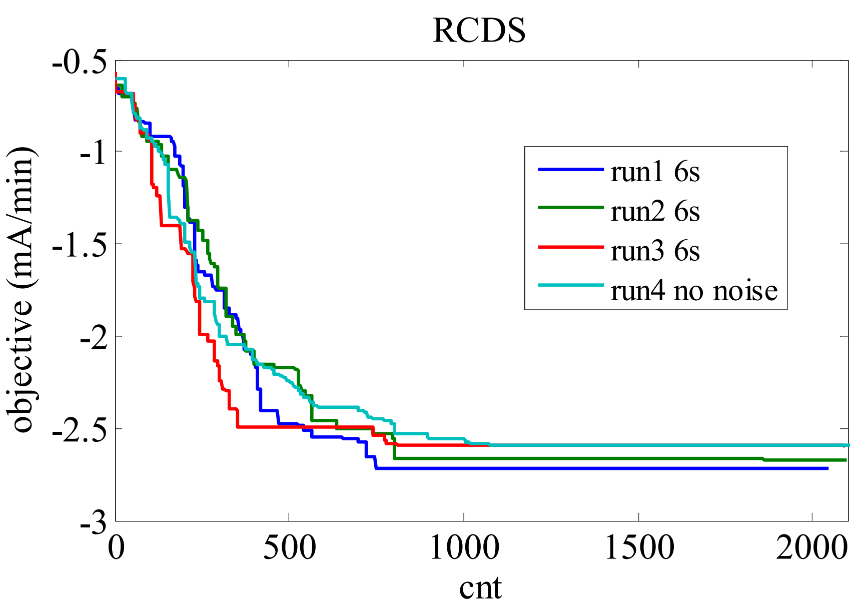
$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Each column in $\mathbf{J}$ is for a skew quad. Conjugate directions are represented by columns in $\mathbf{V}$.

Simulation results for three direct search methods



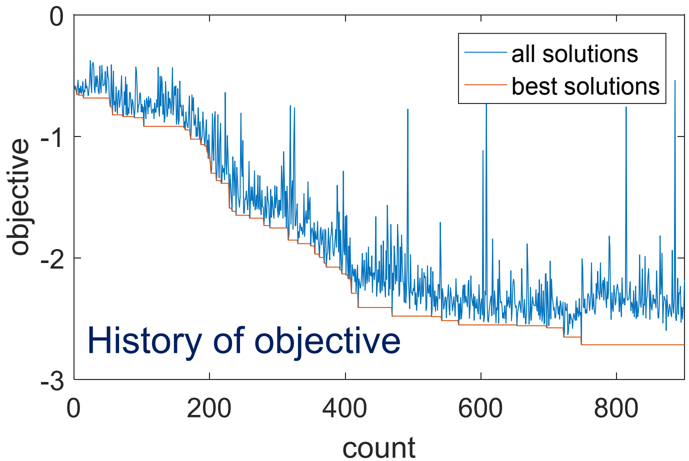


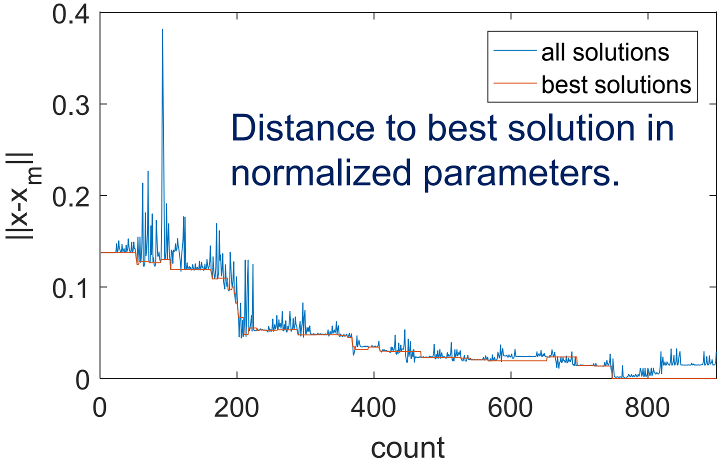


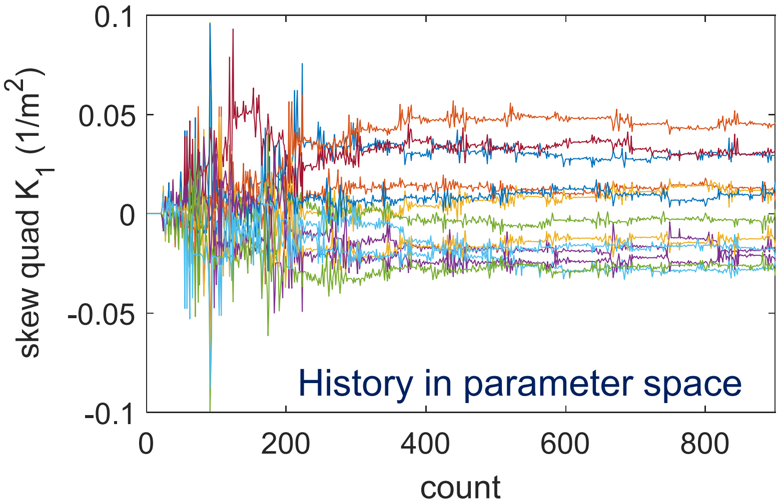
(1) Showing history of the best solution.
(2) The simplex method is efficient without noise, but fails to reach the minimum with noise.
(3) Powell's method works without noise, but fails with noise. The initial direction set are individual skew quads.
(4) The RCDS method is efficient with or without noise.

The performances of algorithms for noisy problems depends on the problems.

Detailed look of an RCDS run





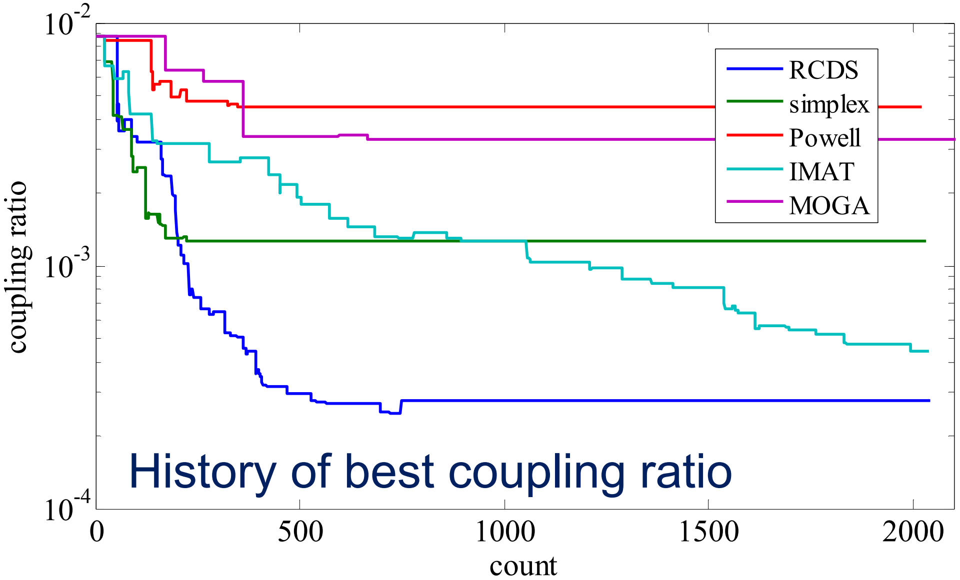


The algorithm converges fast but it does not stay right at the minimum – it keeps probing around.

So usually we need to sort the solutions and apply the best one to the machine.

Comparison of algorithm performances

Best performance for several algorithms



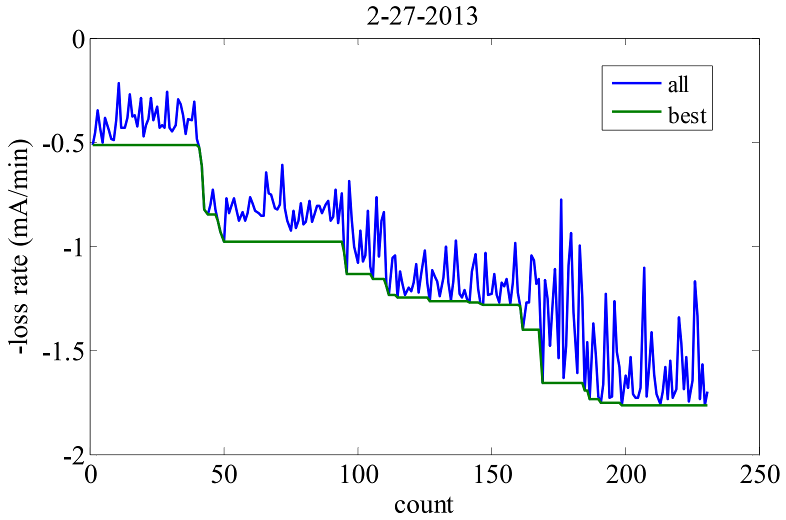
"IMAT": iterative scan of each skew quad with the robust 1D optimizer.

The difference between "IMAT" and "RCDS" clearly shows the power of using conjugate direction set for problems with highly coupled parameters.

Only "IMAT" and "RCDS" have steady gains toward the minimum – a manifest of the noise-resistance feature of the robust 1D optimizer.

Coupling correction experiments on SPEAR3 with RCDS

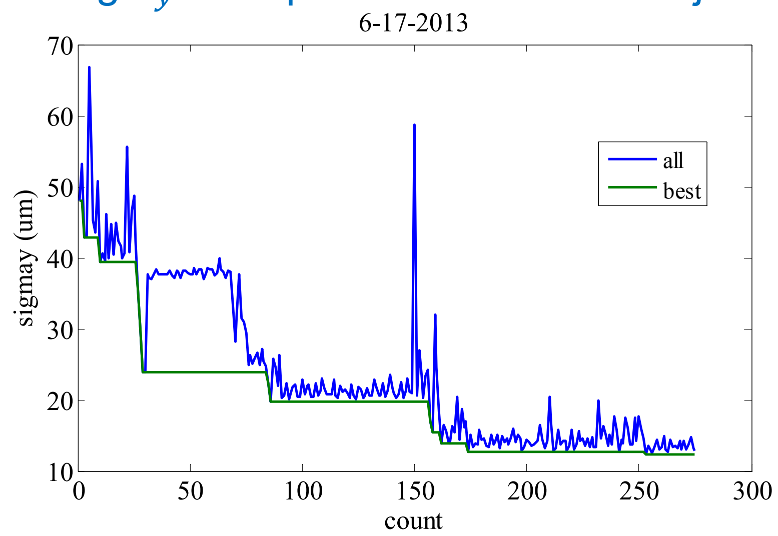
Using loss rate (normalized) as objective



Beam loss rate is measured by monitoring the beam current change on a 6-second interval (no fitting). Noise sigma 0.04 mA/min. Data were taken at 500 mA with 5-min top-off.

Initially all 13 skew quads were off.
At 500 mA, the best solution had a lifetime of 4.6 hrs. This was better than the LOCO correction (5.2 hrs)

Using σ_y from pinhole camera as objective

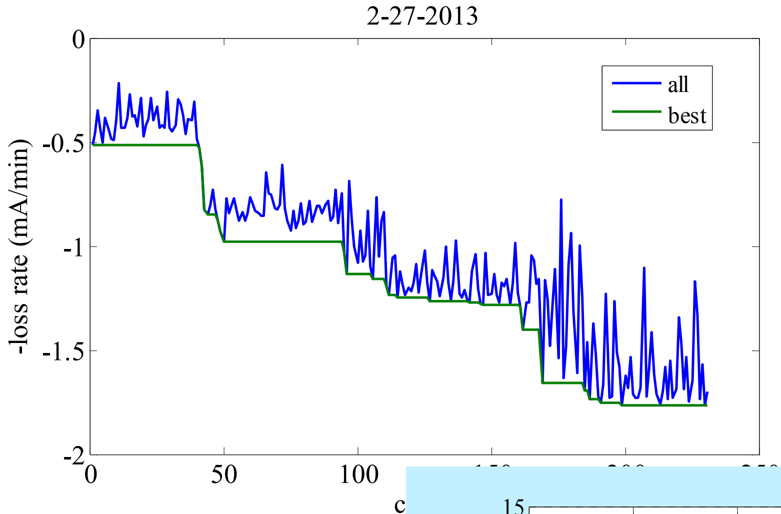


σ_y noise level at 0.3 micron.
All 13 skew quads were off initially.

Pinhole camera resolution is limited.

Coupling correction experiments on SPEAR3 with RCDS

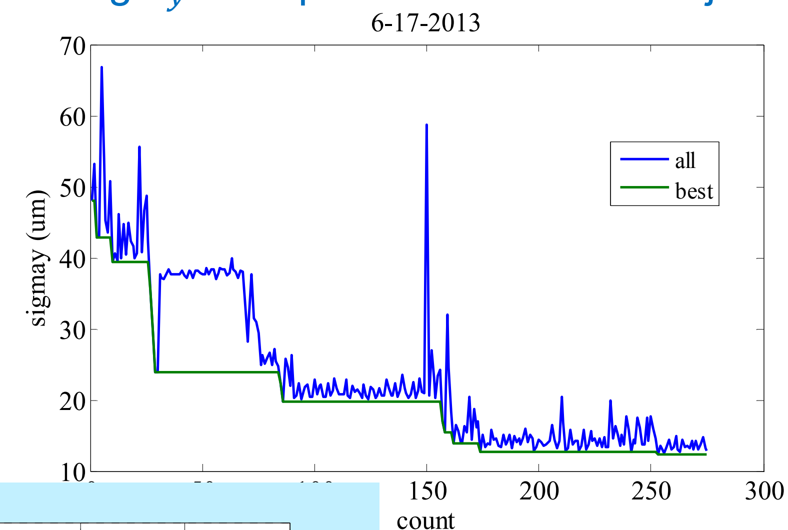
Using loss rate (normalized) as objective



Beam loss rate is m
beam current chang
(no fitting). Noise sig
were taken at 500 m

Initially all 13 skew q
At 500 mA, the best
4.6 hrs. This was be
correction (5.2 hrs)

Using σ_y from pinhole camera as objective



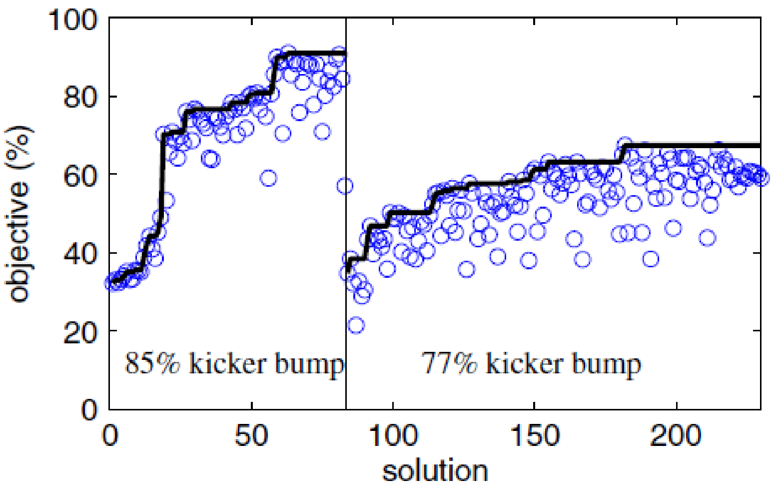
at 0.3 micron.
ads were off initially.

a resolution is limited.

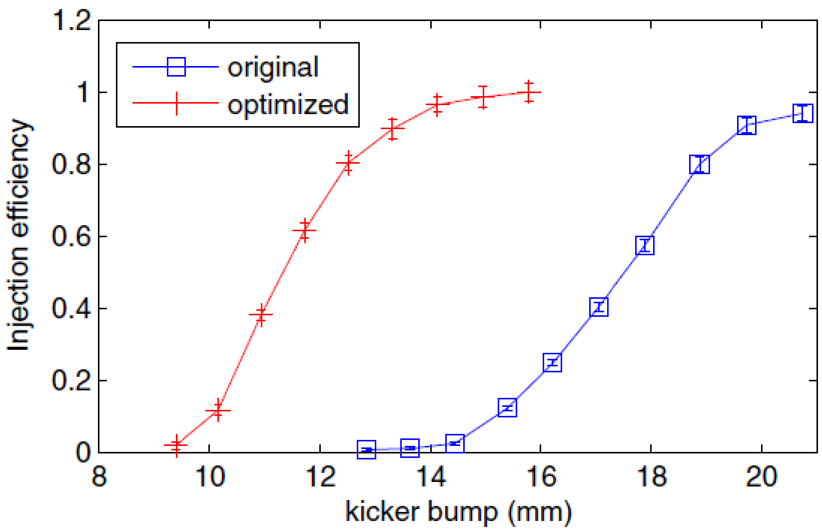
Applications of RCDS on real-life problems

- SPEAR3
  - Kicker bump matching
  - Transport line optics
  - Transport line steering
  - GTL steering and optics
  - Injection efficiency w/ sextupoles — X. Huang, J. Safranek, PRSTAB 18, 084001 (2015)
- LCLS
  - Undulator taper optimization — J. Wu, K. Fang, X. Huang, 2014-2016
- BEPC-II luminosity optimization
  - Steering and coupling — H. Ji, et al, Chinese Physics C 2015 Vol. 39 (12)
  - Interaction point beta
- ESRF — S. M. Liuzzo, et al, IPAC’16, THPMR015
  - beam lifetime w/ sextupoles
  - Injection steering

Online dynamic aperture optimization for SPEAR3



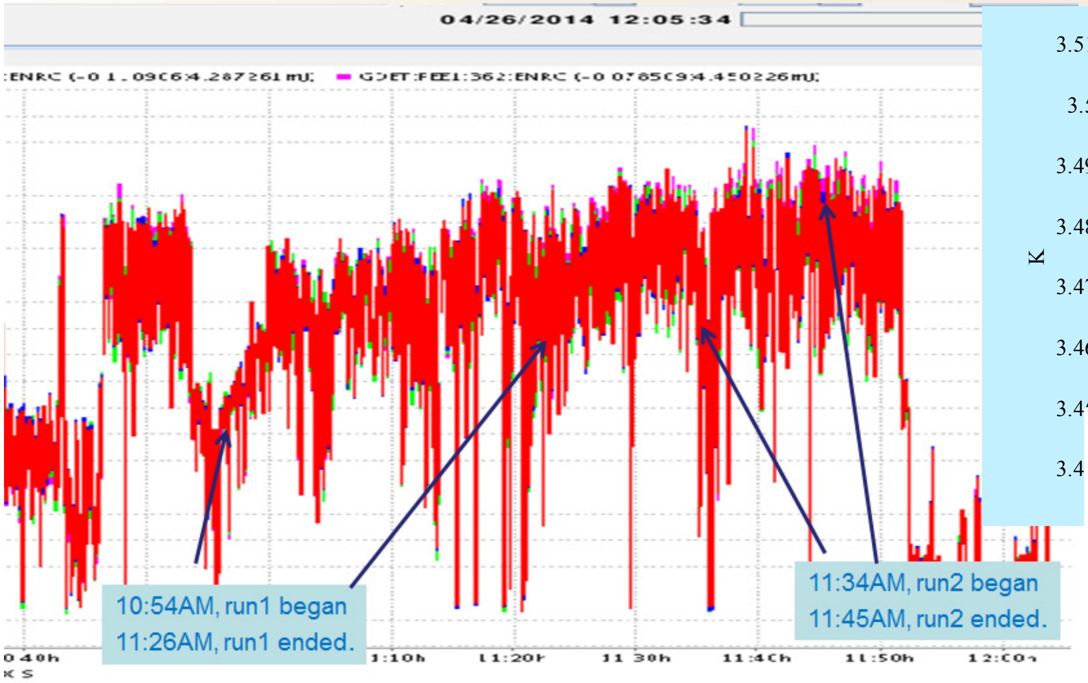
Optimizing injection efficiency with reduced kicker bump.
Knobs: 8 sextupole knobs – each knob is a pattern of 10 sextupole families that do not change chromaticities.

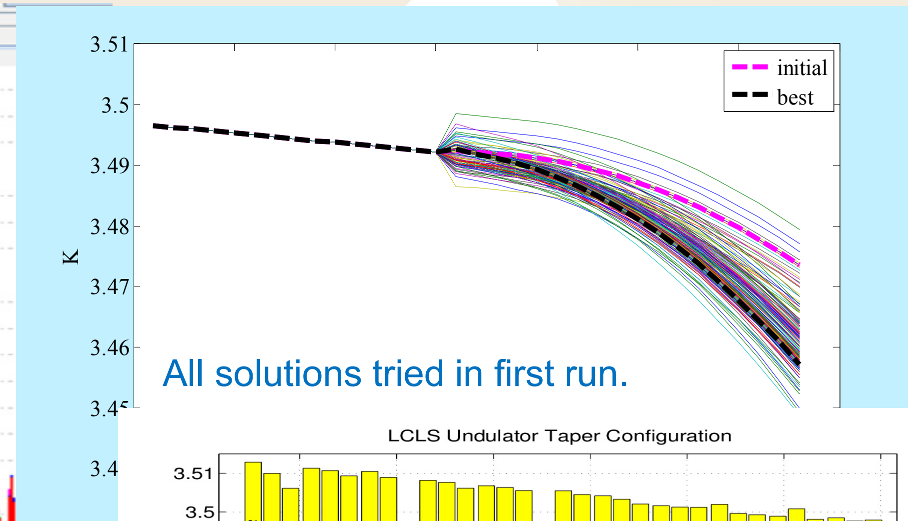


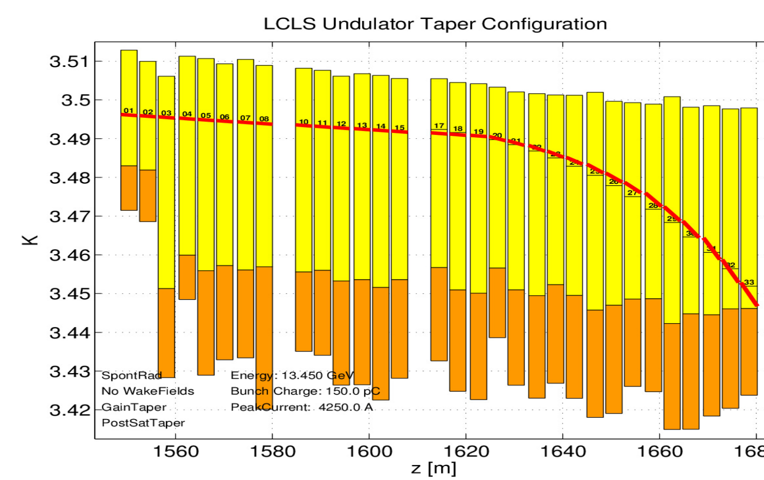
DA was increased from 15.1 mm to 20.6 mm by optimization.
Momentum aperture (MA) was not affected.

X. Huang, J. Safranek, PRSTAB 18, 084001 (2015)

LCLS taper profile optimization







Knobs: 4 parameters that control the taper profile, two phase shifters.

For U1-U8, and U10-U15: $K_j = K_0 (1 - a_0 j)$ with j = 1,…,15

For U17-U33: $K_j = K_1 [1 - a_1 (j - 16) - a_2 (j - z_2)^2]$ with j = 17,…,33.

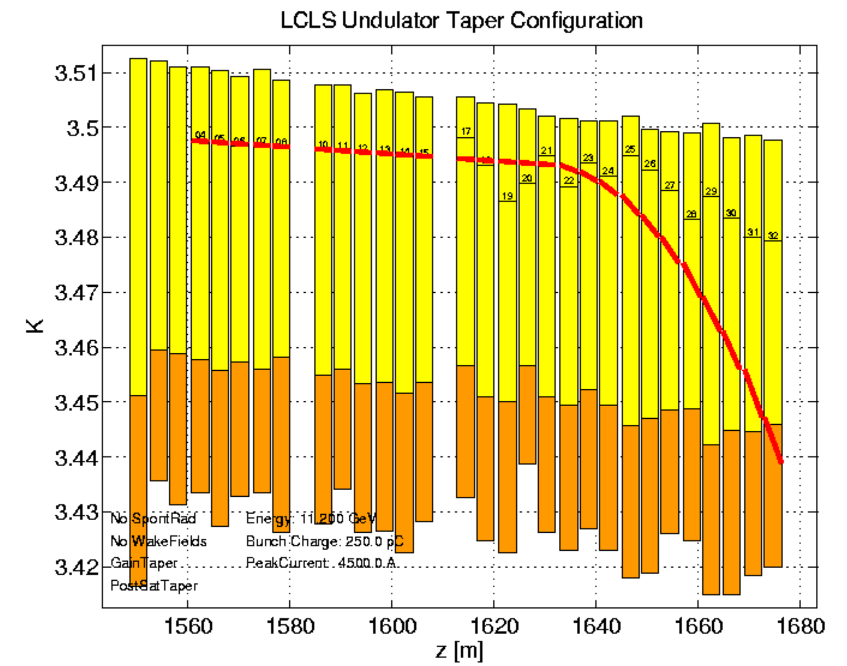
Objective: FEL photon beam intensity.

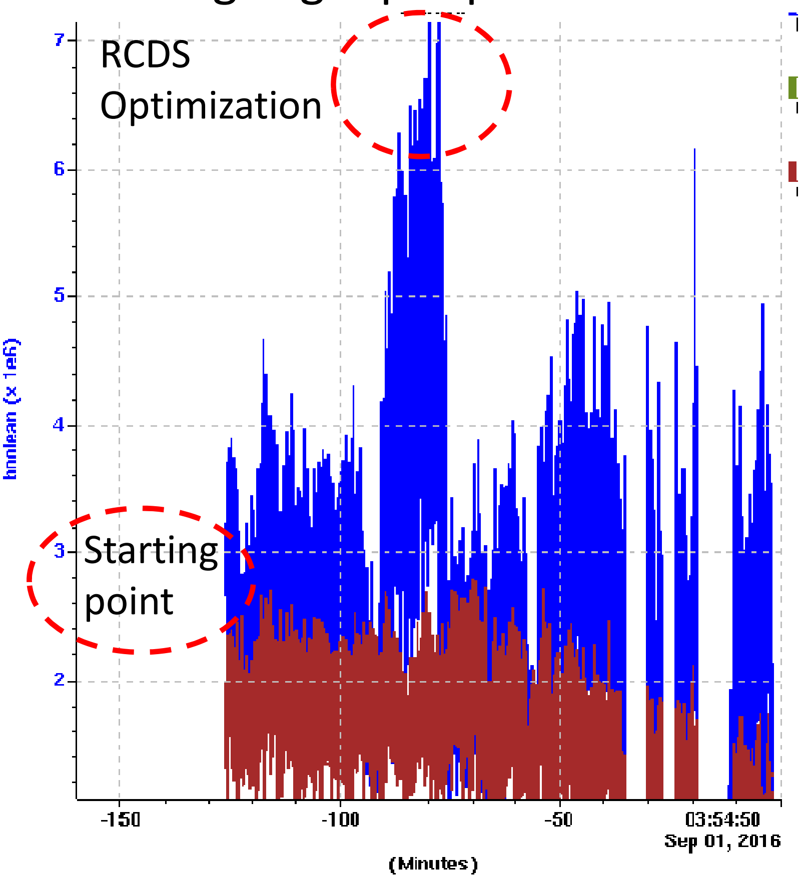
J. Wu, K. Fang, X. Huang, 2014

Recent result by Juhao Wu, 9/1/2016

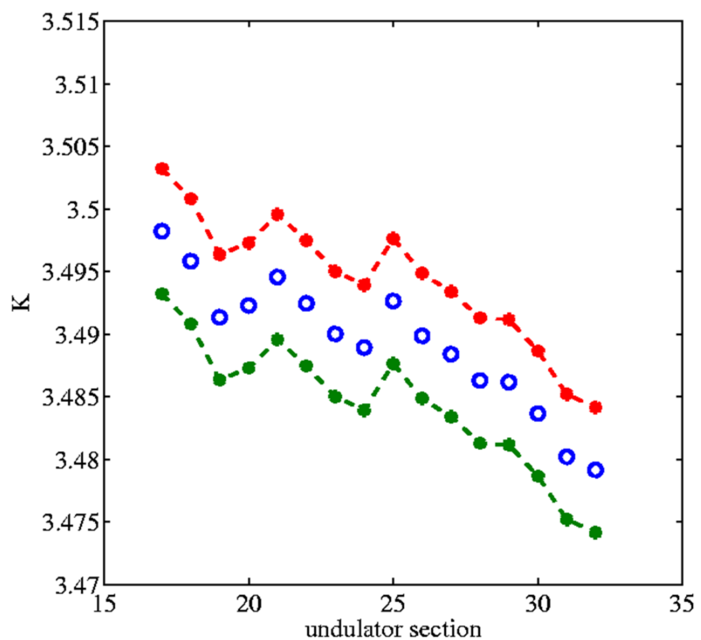
SELF-SEEDING FEL OPTIMIZATION

- 5.5 KeV Self-seeding FEL
  - More than doubled
  - U17-U32 continuous function: does not work well
  - Zig-zag taper profile: ~ 1 mJ in 10 fs





Knobs: 16 parameters that control the taper profile. For U17-U32: *each K is freely optimized with bounds*.
Objective: FEL photon beam intensity.



ESRF optimization of beam lifetime with sextupoles

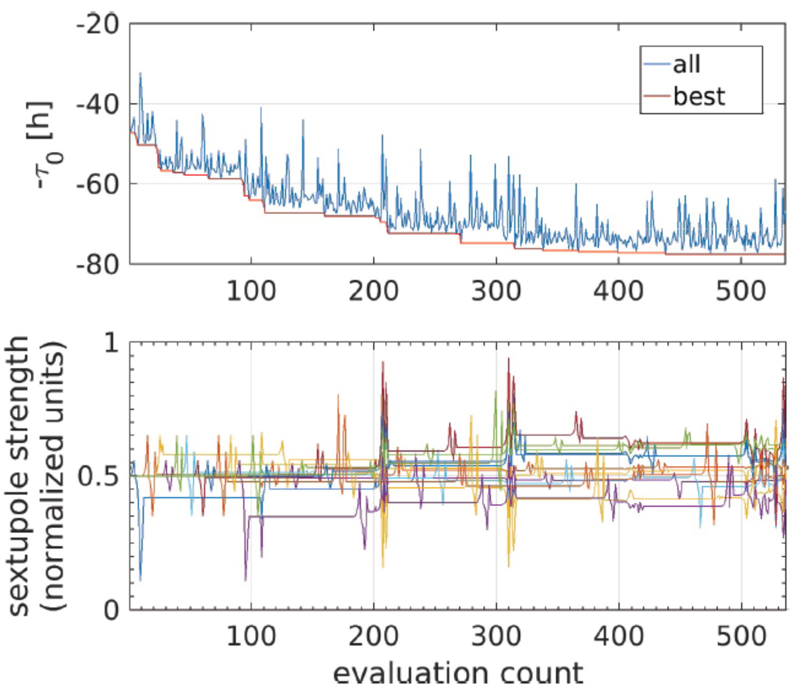
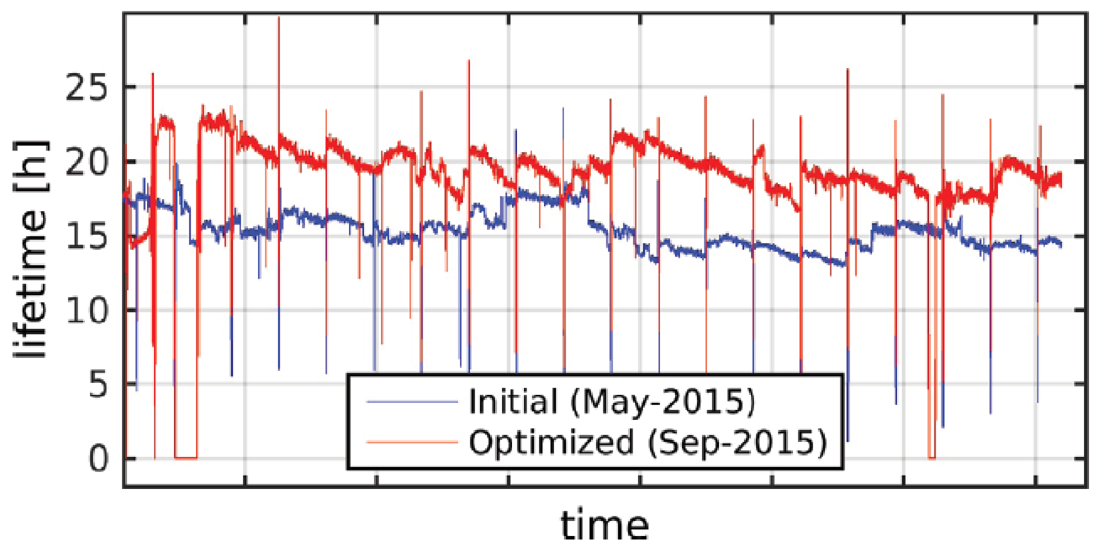


Figure 2: Optimization of lifetime using 12 sextupole correctors in 7/8+1 mode.



Lifetime for the 16-bunch mode in one month before and after optimization.

Objective: lifetime normalized by current, bunch length, and vertical size (average over 13 beam size monitors)

$$\tau_0 = \tau \frac{I}{I_0} \frac{BL(I_0)}{BL(I)} \frac{\sigma_{y,0}}{\sigma_y}$$

S. M. Liuzzo, et al, IPAC'16, THPMR015

The usage of the Matlab RCDS code

- A Matlab RCDS package is available, with instructions and examples. A Python version has also been developed and is available.
- The setup for a new problem is extremely simple:
  - Modify an objective function template
    - Make changes to knobs and take measurement of performance
    - Record data
  - Modify a setup and launch script
    - House keeping: record initial parameters, set parameter ranges
    - Measure and specify noise level (only needed once)
    - Launch RCDS
    - Sort solutions and apply the best solution.

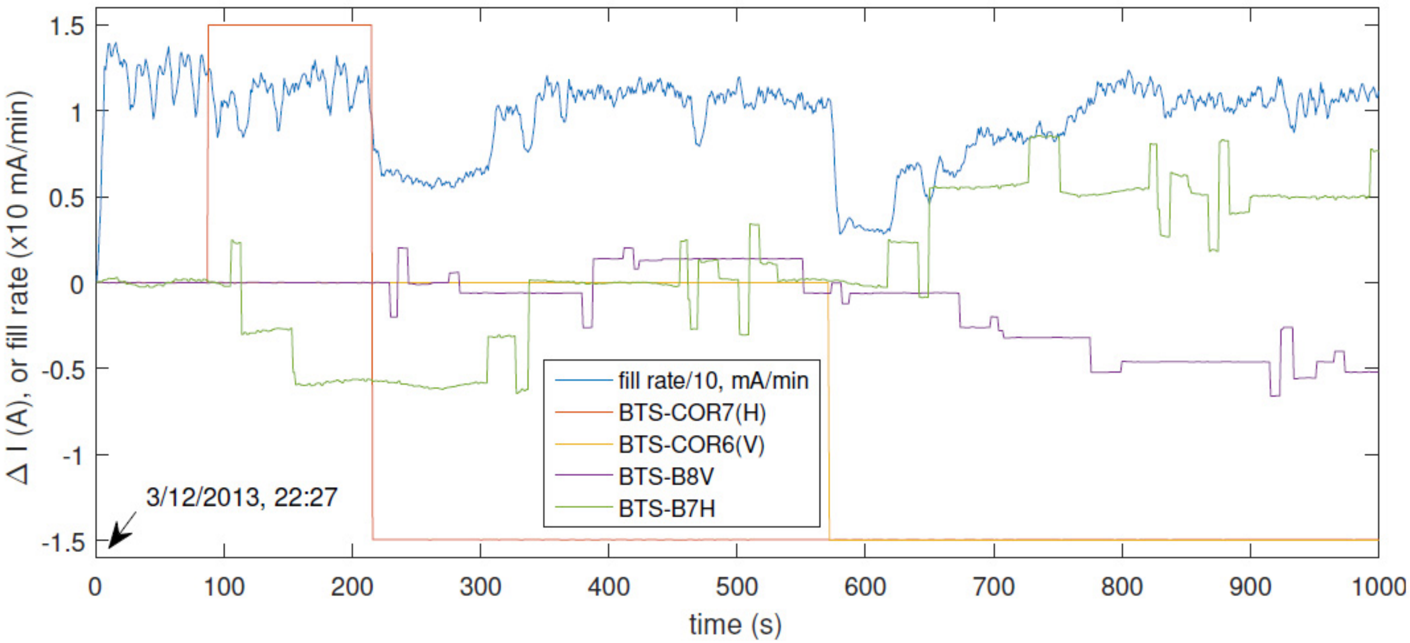
This test was performed very rapidly thanks to the clear and user friendly implementation of the RCDS Matlab code. --- S. M. Liuzzo, et al, IPAC'16

Some comments on RCDS

- RCDS is not simply a variation of Powell's method
  - Yes, RCDS is implemented as Powell's method with the new robust line optimizer.
  - But in online application one seldom benefits from conjugate direction update because only limited directions are replaced.
  - It is the **robust line optimizer** that gives rise to the effectiveness of RCDS.
- RCDS is not simple iterative parameter scan
  - It works with combined knobs.
  - Parameter scan usually have fixed scan ranges and pre-determined, uniform step sizes. Choice of step size (or # of steps) is problem dependent.
  - RCDS uses bracketing, variable step size, and quadratic fitting – a lot more efficient.
  - RCDS algorithm does not need problem-dependent setup.

A variant of RCDS to stabilize performance

- We saw the need to stabilize performance for drifting systems and developed and tested an RCDS stabilizer for it.

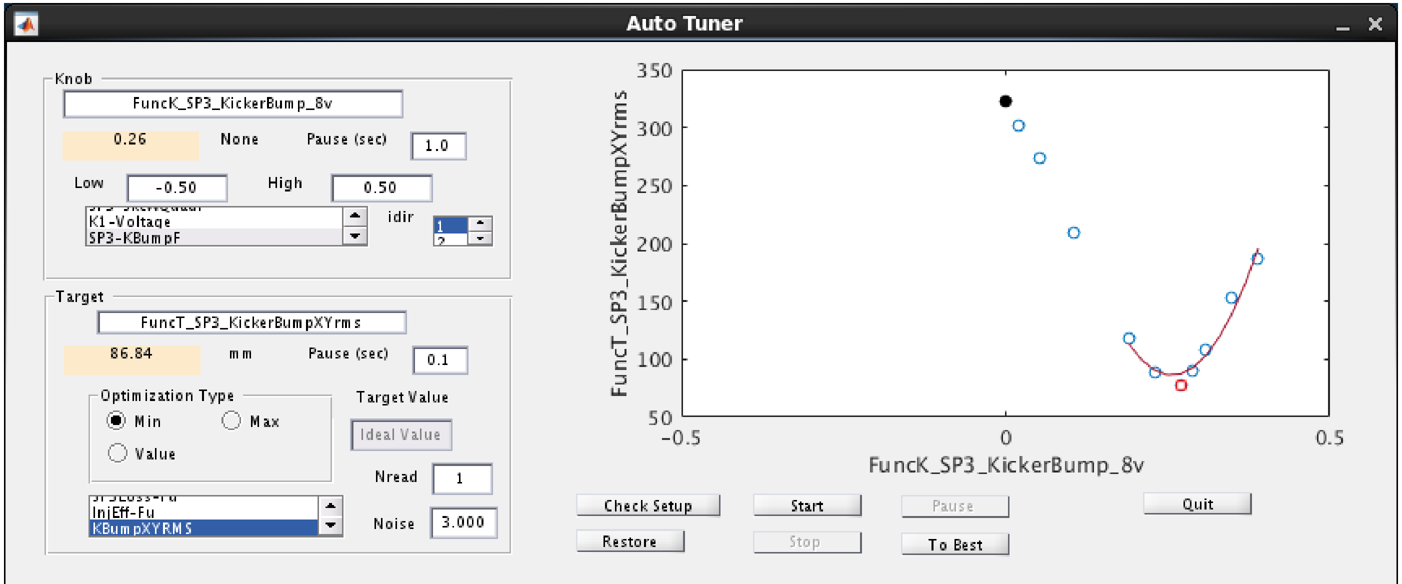


In this test the stabilizer were tuning four steering magnets at the end of the BTS.

When upstream steering magnets were manually changed, the stabilizer responded and brought injection efficiency back.

AutoTuner – An interactive GUI based on RCDS

- A GUI is substantially easier to use – increased productivity and reduced training requirements.
- The code is completely re-written to allow interruption.

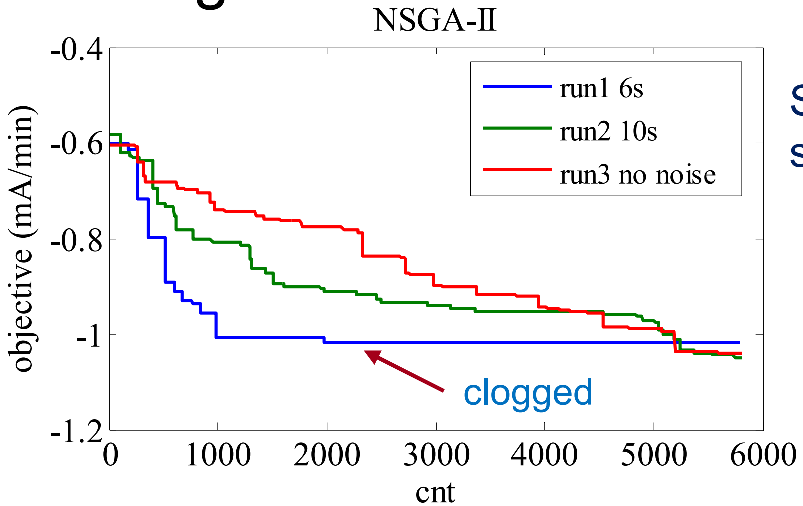


It supports single knob (PV or function) and multi-knob (combined knobs).

AutoTuner has been tested on SPEAR3 and its injector with many knobs.

Other algorithms - Genetic algorithm (NSGA-II)

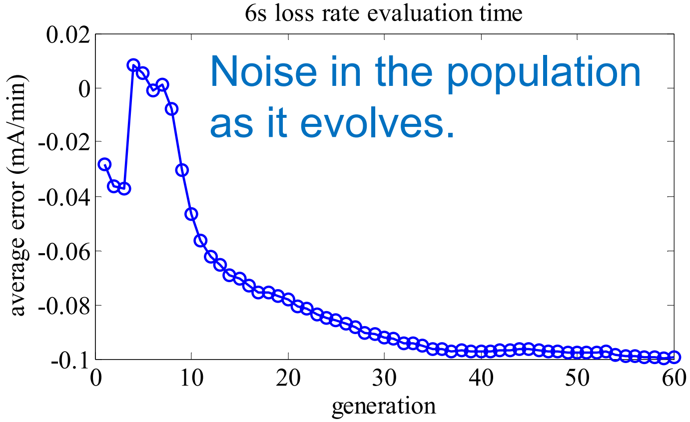
- Genetic algorithm is inefficient even without noise.



Same SPEAR3 coupling correction simulation problem.

Population: 100; Ran 60 generations; 10% mutation, 90% crossover.

- Noise gives a bias to the selection operation.

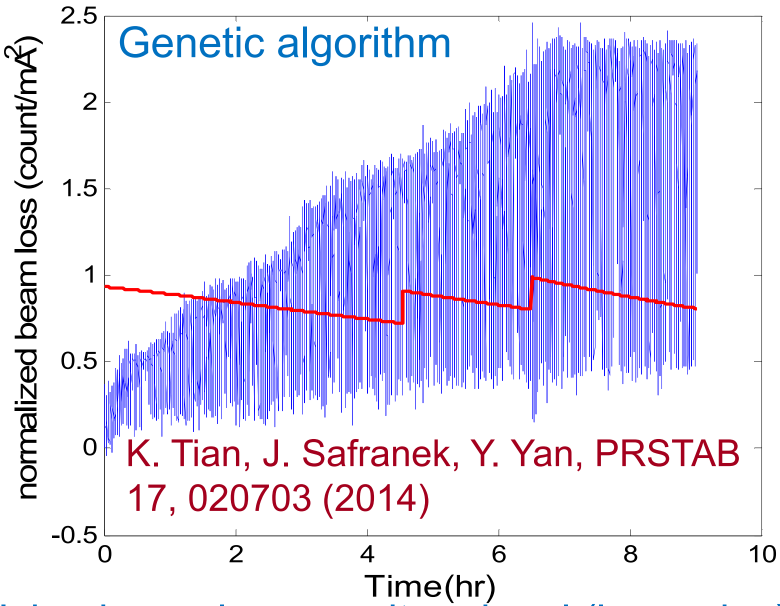


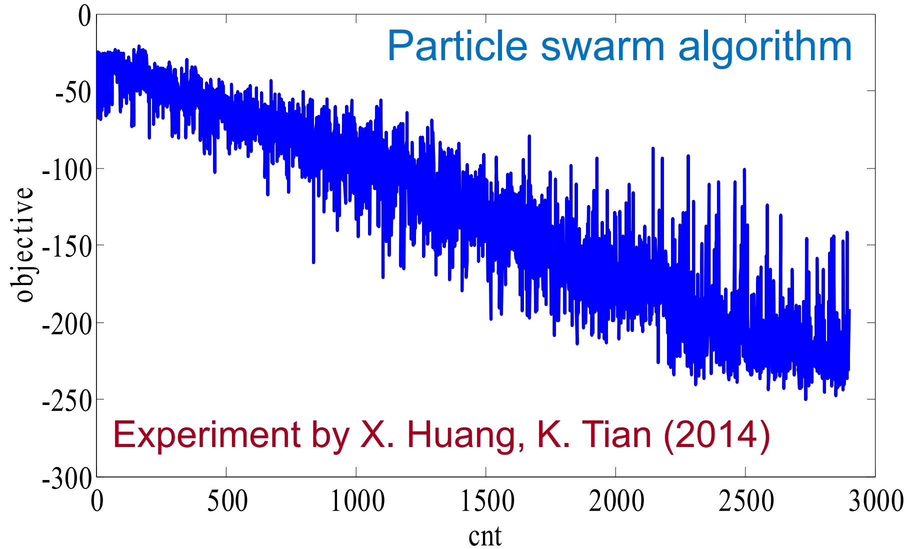
Bad guys (solutions with favorable random errors) tend to enter the next generation. This prevents converging to the true minimum.

X. Huang et al, Nucl. Instr. Methods, A 726 (2013) 77-83.

Genetic algorithm and particle swarm algorithm

Online coupling correction with genetic algorithm





Using beam loss monitor signal (low noise) as objective. It took **20,000** evaluations.

Same setup as the genetic algorithm experiment. It took **3,000** evaluations.

… while RCDS only took **200** evaluations (see slide 17) for a much noisier setup.

When online global search is desired, it seems the particle swarm algorithm is a better choice: (1) more efficient; (2) no bias introduced by noise.

The Extremum Seeking (ES)* method

- The ES method is theoretically elegant.

$$p_1(n+1) = p_1(n) + \Delta\sqrt{\alpha\omega_1}\cos\left(\omega_1 \Delta n + k\hat{C}(n)\right)$$

$$p_2(n+1) = p_2(n) + \Delta\sqrt{\alpha\omega_2}\cos\left(\omega_2 \Delta n + k\hat{C}(n)\right)$$

$$\vdots$$

with $\hat{C}(n) = C(\mathbf{p}(n), t) + \nu(t)$ ← noise

The optimization parameters (knobs) are rotated with various frequencies and amplitudes, and subject to modulation by the cost function.

At the high frequency limit, the behavior approaches that of a gradient descent method

$$\frac{d\mathbf{p}(t)}{dt} = -\frac{k\alpha}{2}\nabla C^T(\mathbf{p}(t), t)$$

Pros: (1) noise is averaged out; (2) a simple and general framework; (3) can dynamically track the optimum.

Cons: (1) algorithm control parameters are problem specific and need tuning;

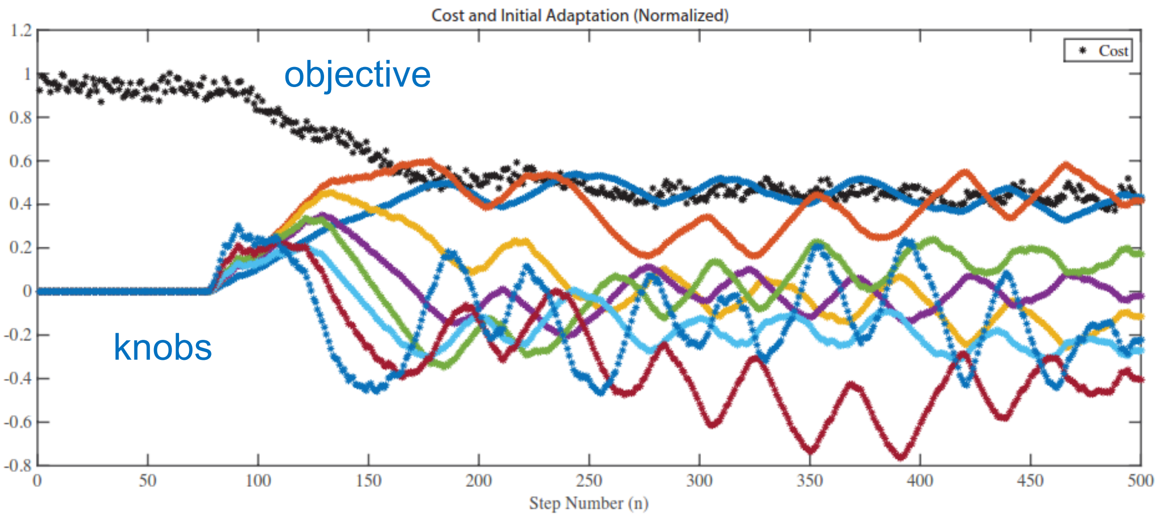
(2) may not be as efficient as other direct search method (e.g. RCDS, simplex);

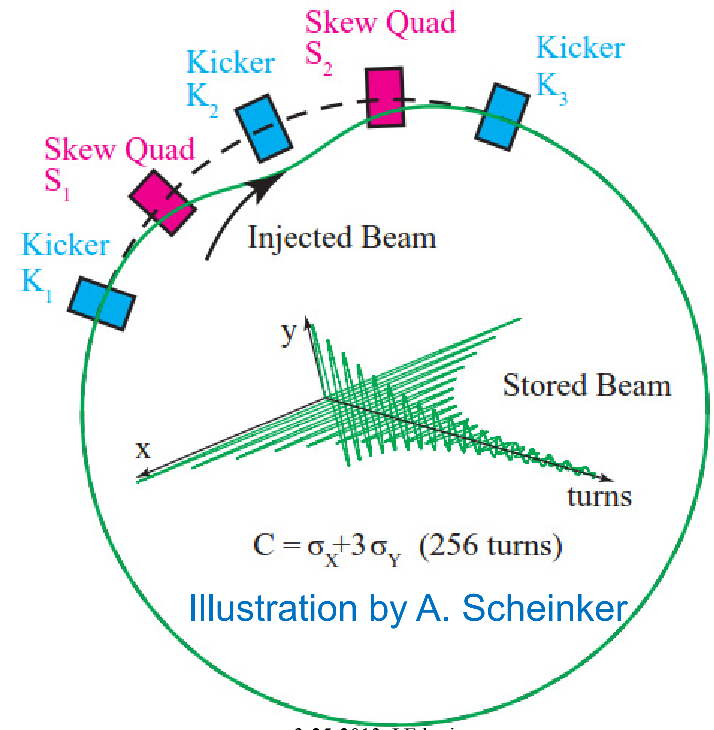
(3) Parameter update rate is bounded, but parameters are not.

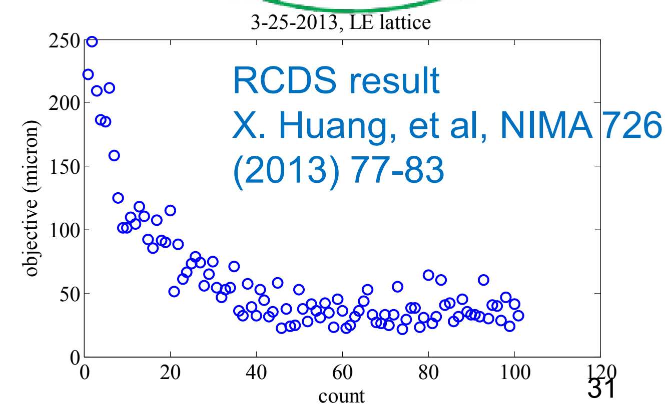
*A. Scheinker, M. Krstic, IEEE Trans. Automatic Control, 58, 1107 (2013).
A. Scheinker, M. Krstic, Systems & Control Letters, 63, 25 (2014)

Test of the ES method on SPEAR3*

The problem: injection kicker bump matching
Knobs: pulse amplitude, width, and delay of K1 and K2, and two skew quads – 8 knobs total.
Objective: residual oscillation of stored beam



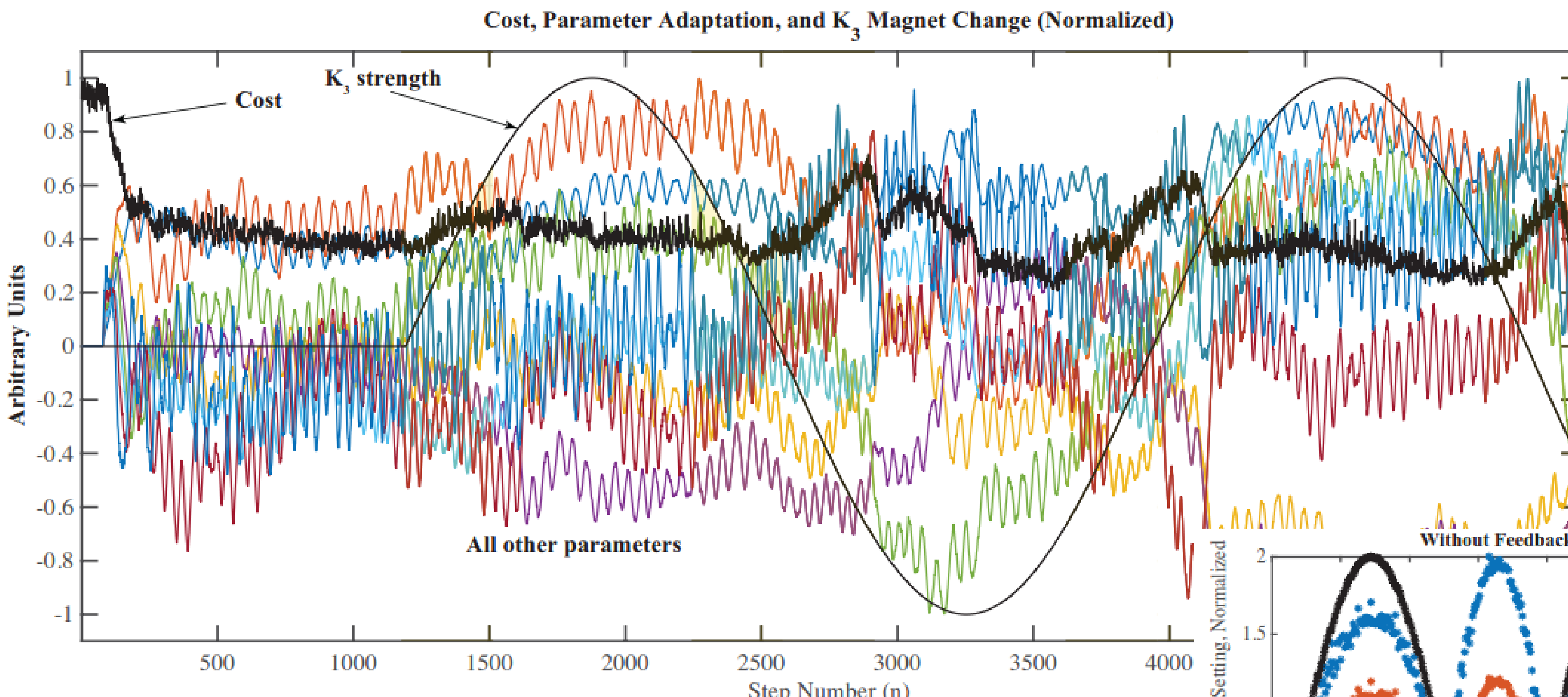




*A. Scheinker, X. Huang, J. Wu, SLAC-PUB-16508 (2016)

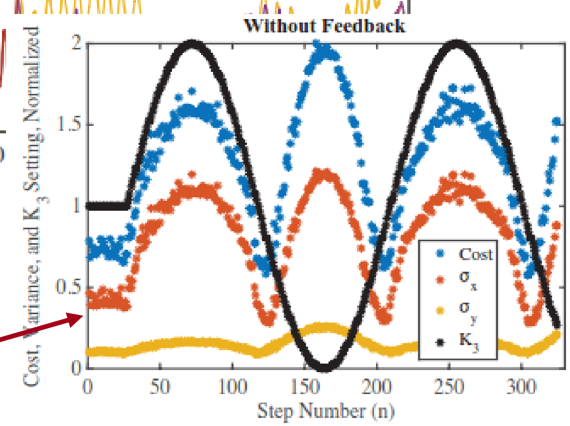
ES – dynamic tracking of the objective

In this test one parameter (K3 voltage, not an optimization variable) is varied, while the ES algorithm serves as a feedback to make compensation.



The ability to maintain performance with a drifting system is important.

If there is no ES feedback



Summary

- Computer controlled systems can be optimized online without a model or knowledge of system interior.
- The RCDS algorithm is a robust and efficient method for online optimization, tested on many accelerator problems.
- Automatic tuning GUI and performance stabilizer based on RCDS have been developed and tested.
- Other algorithms were also tested for online optimization.

Acknowledgements

- Thanks to James Safranek for many helpful discussions.
- Thanks to users of RCDS that helped demonstrate the method, especially
  - Juhao Wu (SLAC), Yi Jiao, Hongfei Ji et al (IHEP), S. M. Liuzzo et al (ESRF)