

Markov chain optimisation for energy systems (MC-ES)

John Moriarty
Queen Mary University of London

19th August 2016

“Approaches to the stochastic optimisation of power systems are not mature at research scales, and at industrial scales they are absent.”

- Workshop report, *US Department of Energy (2013) [1]*

Aim: To transform high-dimensional stochastic optimisation problems (hard) into high-dimensional simulation problems (easier)

MCMC optimisation

- ▶ Consider the problem of maximising an expectation (or ‘aggregated performance measure’)

$$U(g) := E[u(g, W)] \quad (1)$$

where u is a function, W a random vector, and g a control policy

- ▶ Suppose that direct sampling from the distribution of W is difficult: for example, W may arise from a complex spatio-temporal (multidimensional) model of wind power conditioned on forecast data

Modelling uncertainty

- ▶ The distribution of W could be specified directly, eg. using a set of scenarios
- ▶ To decompose its complexity, W may be modelled hierarchically as $W = (X, Y)$, where X is a set of unknown parameters with prior density $p(x)$ and Y are the observable quantities with conditional density $p(y|x)$. Their joint density is then $p(x, y) = p(y|x)p(x)$
- ▶ Example: wind power modelling
 - ▶ W may be a large set of historic scenarios for the trajectory of wind output
 - ▶ Or X may represent a set of key (but unobserved) weather variables at various points in time and space, while Y represents the trajectory of observed output at a set of wind farms, *given* X

Example: UK day-ahead generation scheduling

- ▶ The problem of choosing an optimal operational policy (or 'schedule') for the generation units in a power system subject to network and power balance equations, plus operational and risk constraints
- ▶ In the UK there are on the order of 200 large controllable power stations, scheduled half-hourly
- ▶ W may represent wind power output ($\sim 10\%$ of generation)

Example: UK day-ahead generation scheduling

- ▶ So in this example $T = 48$, $K = 200$, and

$$\begin{aligned}g &= (g_1, \dots, g_T) \in G, \\g_i &= (g_i^1, \dots, g_i^K), \quad i = 1, \dots, T,\end{aligned}$$

where g_i^j represents the chosen control policy for generator j at time i .

- ▶ Simple linear example: let the observables be

$$y = (y_1, \dots, y_T)$$

and we take affine control policies:

$$g_i^j(y) = A_i^j + B_i^j y_i.$$

Here A_i^j is the 'base' generation schedule for generator j at time i , and $B_i^j y_i$ is the real time correction for generator j based on the observation y_i at time i .

- ▶ More complex parametric, and bounded, control policies may also be considered

Proxy model for power system trajectory

- ▶ Fix a suitable state space \mathcal{Z} for the power system, so $Z \in \mathcal{Z}$ represents a ‘snapshot’ at a fixed time
- ▶ We wish to model the *trajectory* of the power system over the optimisation horizon, ie. the sequence of snapshots:

$$z := (Z_t)_{t=1, \dots, T} \in \mathcal{Z}^T$$

- ▶ Suppose that z is given by a function $v(g, x, y)$ which passes the control policy g , parameters x and observations y through the power system’s controlled dynamics, and outputs the resulting states of the power system at times $t = 1, \dots, T$
- ▶ The trajectory $v(g, x, y)$ is thus a *proxy* for the real-time controlled behaviour of the power system
- ▶ The proxy trajectory v should be computationally inexpensive to evaluate, eg. a sequence of DC power flow calculations with power injections depending on g and y .

Optimising the performance function

- ▶ Having evaluated the proxy trajectory $v(g, x, y)$ we then take a performance function $u = u(g, x, y, v(g, x, y)) = u(g, x, y)$. Note that u is not assumed to be convex, nor even continuous
- ▶ The performance function u may be eg. socio-economic, or alternatively a measure of reliability; it may be directly specified or learned
- ▶ Recall that we wish to maximise the expected (ie. aggregate) performance function U : we seek $g^* := \arg \max_{g \in G} U(g)$, where

$$U(g) = E[u(g, W)] = E[u(g, X, Y)] \quad (2)$$

$$= \int_{\mathcal{X} \times \mathcal{Y}} u(g, x, y) p(x, y) dx dy. \quad (3)$$

Efficient optimisation through sampling (Müller et al. [2])

- ▶ The idea involves treating the policy g as a random variable G and simulating a Markov chain $(G_k, X_k, Y_k)_k$
- ▶ Not every sample ('proposal') is used – this allows us to control the distribution of the sampled Markov chain
- ▶ By accepting proposals with a certain probability (the 'Metropolis-Hastings algorithm') which depends on the performance function u , the chain has stationary distribution

$$h(g, x, y) \propto u(g, x, y)p(x, y) \quad (4)$$

- ▶ Integrating out x and y , then G has the marginal density

$$\int_{\mathcal{X} \times \mathcal{Y}} h(g, x, y) dx dy \propto U(g) \quad (5)$$

- ▶ So the optimal g is the mode of this marginal density: the **most frequent simulated value of G**

Incorporating constraints

- ▶ Chance constraints can be included using Lagrange multipliers as in [3] (increases computational complexity; more on this later if time allows)
- ▶ Some operational (hard) constraints can be incorporated in the simulation procedure, see eg. [4] in the power systems context

Advantages of MCMC optimisation

- ▶ Simultaneously performs simulation and optimisation, more efficient than tackling these aspects separately (eg. if uncertainty has a complex, non-Gaussian distribution)
- ▶ Hierarchical uncertainty modelling is Bayesian – allows uncertainty (prior distributions) over unknown parameters X . Thus conservatism may be adjusted by the flexible choice of the prior density $p(x)$
- ▶ Can be parallelised by running N independent simulations – results in more peaked density $U^N(g)$ for g so mode easier to estimate from the sample
- ▶ Opens up stochastic optimisation problems to the vast mathematical literature on MCMC, eg. convergence results and variance reduction methods
- ▶ MCMC has proved effective and scalable in a wide variety of difficult problems, eg. in genetics

Scalability?

- ▶ Noise, and hence the trajectory z , are simulated in our MCMC optimisation procedure rather than exhaustively evaluated
- ▶ The unknown parameters x and observables y are also simulated
- ▶ Nevertheless, the theory of MCMC guarantees convergence of the result (eventually!)
- ▶ This means that the ‘curses of dimensionality’ associated with large state spaces and detailed models of randomness, and suffered by approaches such as MILP and approximate dynamic programming, are mitigated in MCMC optimisation
- ▶ The space G of policies is still potentially large, so the estimate for the mode may have high variance. Here the MCMC literature can help
- ▶ For example, there are MCMC variance reduction techniques loosely related to scenario reduction

The 'Markov chain optimisation for energy systems' project (MC-ES)

- ▶ Funded by EPSRC [5]
- ▶ At Queen Mary University of London (Principal Investigator: JM)
- ▶ Runs from 2017 to 2020
- ▶ Three industrial project partners
- ▶ Open to collaboration!

References

[1] F. Alexander. A multifaceted mathematical approach for complex systems. 2013.

<http://escholarship.org/uc/item/9930z832>.

[2] P. Müller, B. Sansó, and M. De Iorio. Optimal bayesian design by inhomogeneous markov chain simulation.

Journal of the American Statistical Association, 99(467):788–798, 2004.

[3] J. Gonzalez, Y. Kitapbayev, T. Guo, J. Milanovic, G. Peskir, & JM. Application of sequential testing problem to online detection of transient stability status for power systems, 55th IEEE Conference on Decision and Control, to appear.

[4] T. Satoh and K. Nara. Maintenance scheduling by using simulated annealing method [for power plants].

IEEE Transactions on Power Systems, 6(2):850–857, 1991.

[5] EPSRC Grants on the Web.

<http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/P002625/1>

Quickest transient stability classification

John Moriarty

Joint work with J. Gonzalez, T. Guo, Y. Kitapbayev, J.
Milanovic, G. Peskir (Manchester)

Quickest transient stability classification problem

- Suppose we **know** a disturbance has cleared in a power system with several generators
- We **observe** the rotor angle of each generator in real time
- **Objective:** predict the post-disturbance 'stability class' [1] *as quickly as possible under chance constraints*
- Assume a **tradeoff** exists: Earlier predictions are preferable, but later predictions are more accurate
- Need to decide **when** to make prediction (τ), and **which stability class** to predict (d)

Training a PNN at each time point

Idea: use probabilistic analysis to design a **flexible** classification time which minimises the **average** time taken for a classification, while respecting the chance constraints

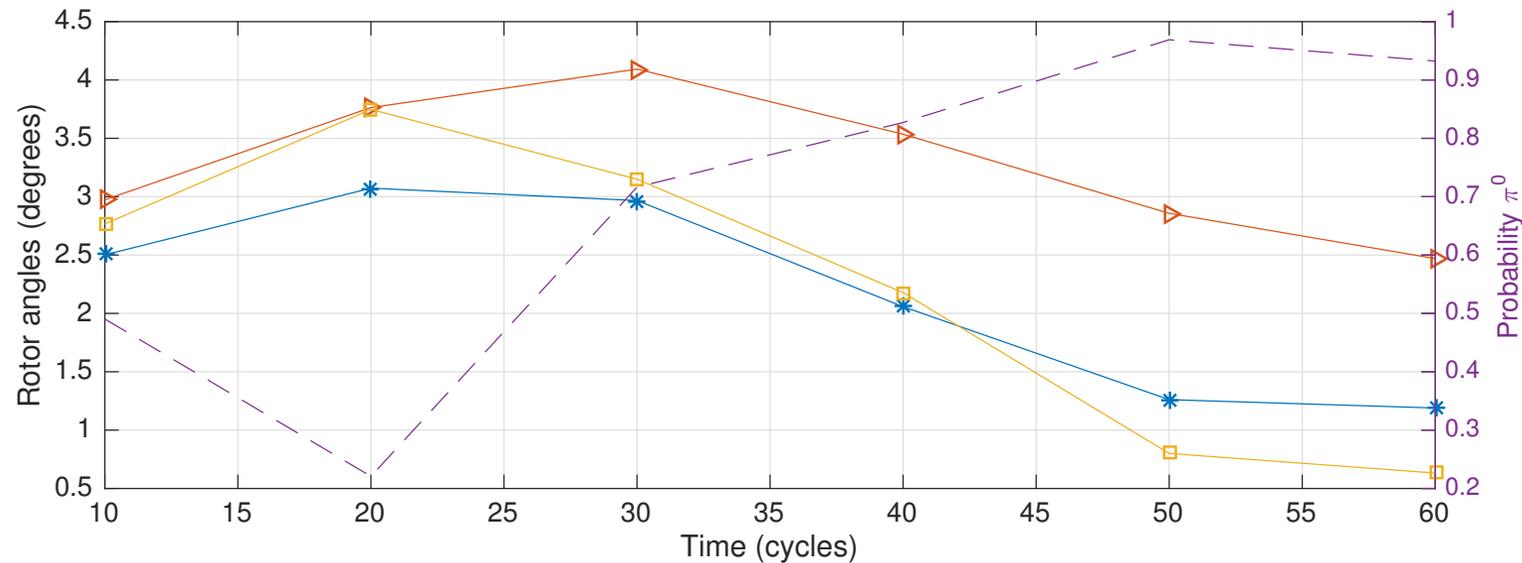
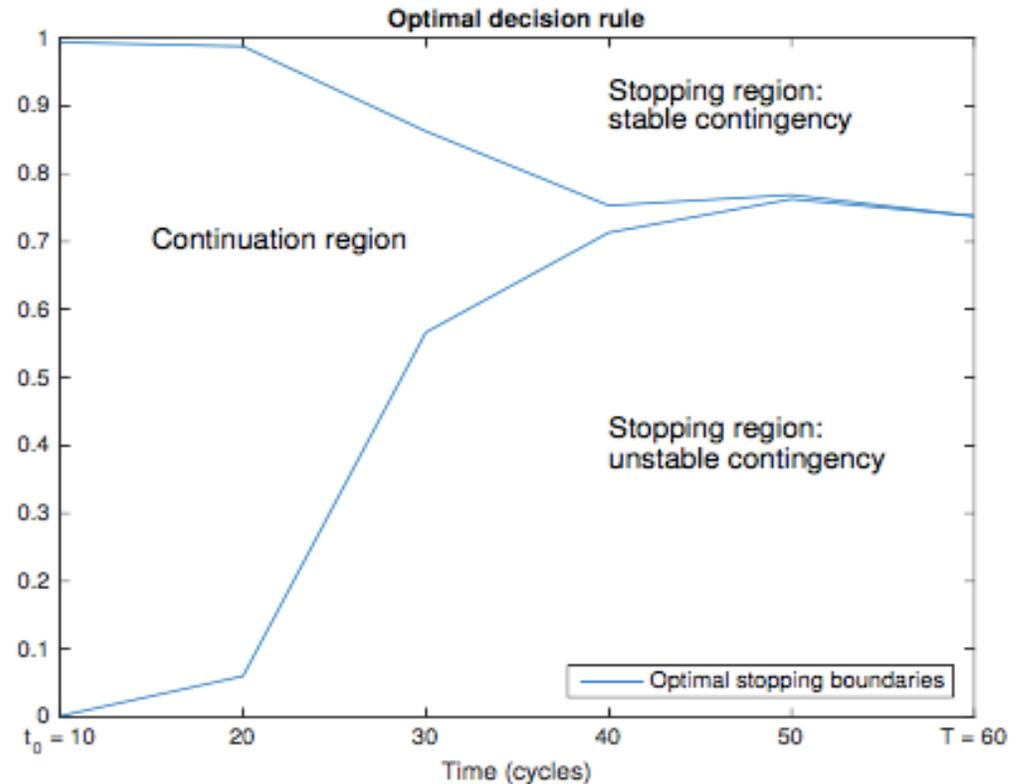


Fig. 2. Illustrative training and test data for a probabilistic neural network in the one-rotor example of Section IV-A. Stable training trajectory (blue), unstable training trajectory (red), test trajectory (yellow) and test output (purple dashes).

The decision rule: solution boundaries

- We train PNNs at 10,20,... ,60 cycles using a learning set of simulations
- PNN output π is multi-dimensional (one dimension per stability class)
- Use a **second** learning set to obtain **optimal boundaries** which characterise the flexible classification time
- This approach is known to be optimal in general for 1-d Markovian problems [2]
- Use π_0 (probability of stable state) to determine **prediction time τ**
- Use the remaining π_i to predict the **stability class d** at time τ



Optimal upper and lower boundaries as function of time

Unconstrained reformulation using Lagrange multipliers

(RCP) Risk-constrained sequential testing problem:
Choose the pair of decision variables (τ, d) in order to minimise the average time $\mathbb{E}[\tau]$ taken to make the prediction d , subject to the following risk constraints:

$$\mathbb{P}[A_1] \leq p_1, \quad (1)$$

$$\mathbb{P}[A_2] \leq p_2, \quad (2)$$

$$\mathbb{P}[A_3] \leq p_3. \quad (3)$$

- 1) If our prediction is $d = 0$ (stable) when the contingency is actually unstable (of any type), we say that *error A_1 occurs*,
- 2) If our prediction is $d > 0$ (unstable) when the contingency is actually stable, we say that *error A_2 occurs*,
- 3) If our prediction is $d > 0$ when the contingency is actually unstable but of a different instability type, we say that *error A_3 occurs*.

(UP) Unconstrained problem:

Find a nonnegative multiplier vector λ^ and feasible boundaries v^* such that*

$$\lambda_i^* \mathbb{E}[\mathbf{1}_{A_i} - p_i] = 0, \quad (6)$$

for $i = 1, 2, 3$ and the boundaries v^ minimise the Lagrangian:*

$$L(v^*; \lambda^*) = \inf_v L(v; \lambda^*). \quad (7)$$

Performance: approx. three-fold increase in prediction speed vs. pre-committed (fixed) prediction times

TABLE I
ERROR RATES WITH PRE-COMMITTED STRATEGY AND
SEQUENTIAL TESTING SOLUTION

Time (cycles)	Error rate A_1 (out of 171)	Error rate A_2 (out of 1829)	Error rate A_3 (out of 171)
10	25	5	19
20	13	5	19
30	9	2	18
40	5	2	18
50	6	3	19
60	6	3	19
Optimal:11.25	7	7	22

Thanks for your attention!

References:

[1] Guo, T., & Milanović, J. V. (2016). Online Identification of Power System Dynamic Signature Using PMU Measurements and Data Mining. *IEEE Transactions on Power Systems*, 31(3), 1760-1768.

[2] Peskir, G., & Shiryaev, A. (2006). *Optimal stopping and free-boundary problems* (pp. 123-142). Birkhäuser Basel.

Presentation based on: J. Gonzalez, Y. Kitapbayev, T. Guo, J. Milanovic, G. Peskir, & JM. Application of sequential testing problem to online detection of transient stability status for power systems, 55th IEEE Conference on Decision and Control, to appear.