A nonlinear filtering tool for analysis of hot-loop test campaings

Enso Ikonen* Jenő Kovács*,**

* Systems Engineering Laboratory,
Department of Process and Environmental Engineering,
University of Oulu, Finland
Enso.Ikonen@oulu.fi www.oulu.fi/pyosys

** Foster Wheeler Energia Oy, Varkaus, Finland

Outline:

- Introduction
  - model calibration,
  - experiments & hot-loop model
- Global optimization
- Sequential optimization
  - PF algorithm
  - filtering & smoothing
- Simulation studies
- Discussion & conclusions

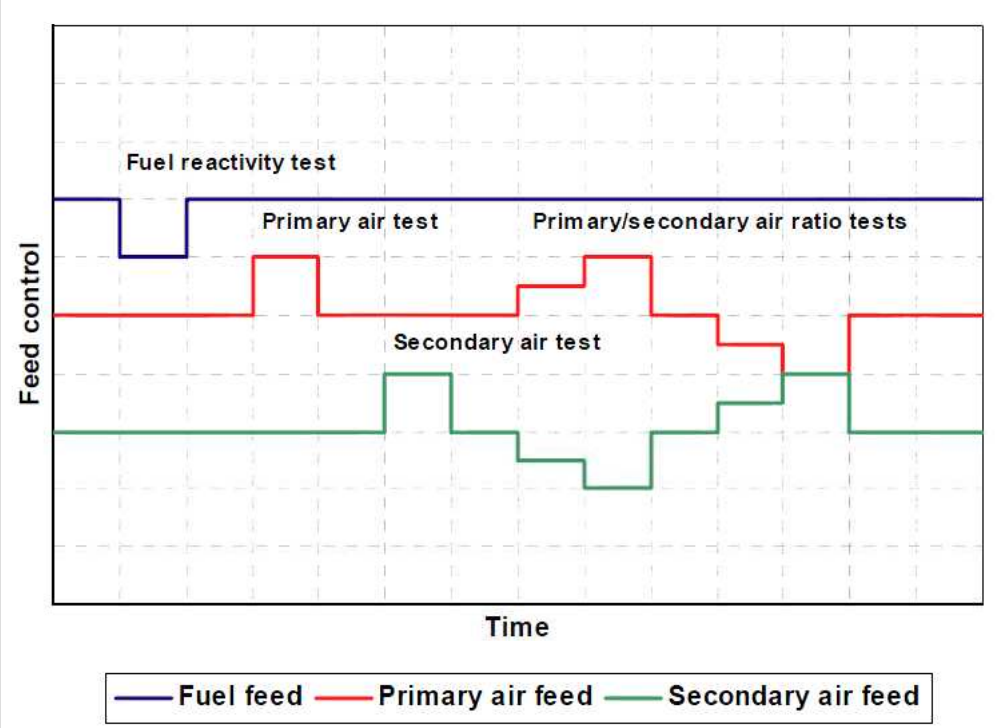
Introduction

- model *vs*. data
  - coherence
    - validates the model (mechanisms, parameters, ...)
  - mismatch
    - inadequate model
    - inaccurate measurements
    - unmeasured disturbances during tests
  - fault diagnosis
    - errors, residuals
    - banks of models
  - bayesian reasoning
    - stochastic framework
- model calibration & PE
  - effective values of physical parameters
  - tuning:
    - measurements,
    - physical understanding
    - engineering sense
  - parameter values & their uncertainties
  - identification,
- state estimation
  - SE for PE
    - parameters belong to the (unknown) system state
    - reason the proper value of the state based on a system model & measurements from it
  - Kalman filter with extensions
  - monitoring, control (MPC)

A large amount of modelling work has and is being conducted at the industry and academia to model various industrial processes. Often, the bottleneck is not in the lack of models but in how to make maximal use of the ones that already exist. The research work reported in this paper aims at extracting more out of the modelling investments, via improved understanding of the plant (model analysis), improved design (model tuning) and improved on-line control (design of on-line process state estimation).

Dynamic tests

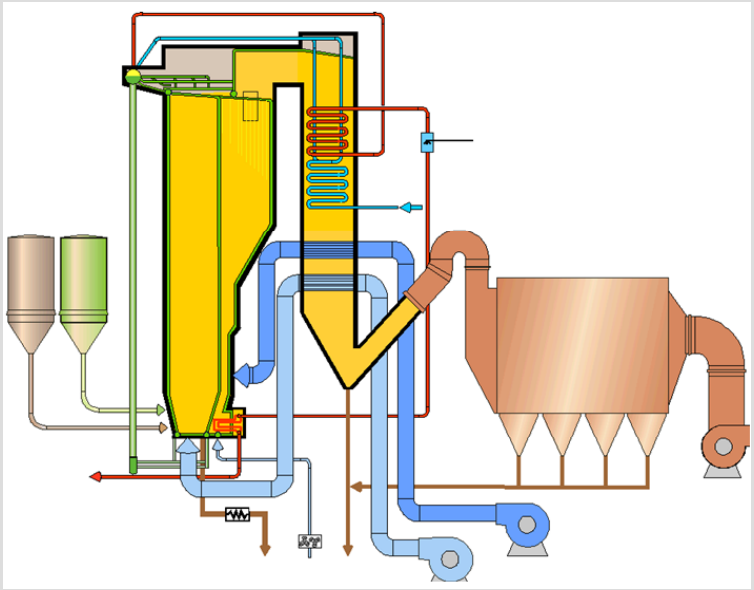
- dynamic models
  - becoming commonly used in the industry
  - execution of dynamic tests is time-consuming & expensive
- examination of test outcomes
  - tools for assessment
    - model vs measurements
    - measurements vs model
  - hypothesis evaluation
- state estimation
  - monitoring,
  - automatic control



- CFB test series.
  - In addition: closed loop steps, ramps, ...

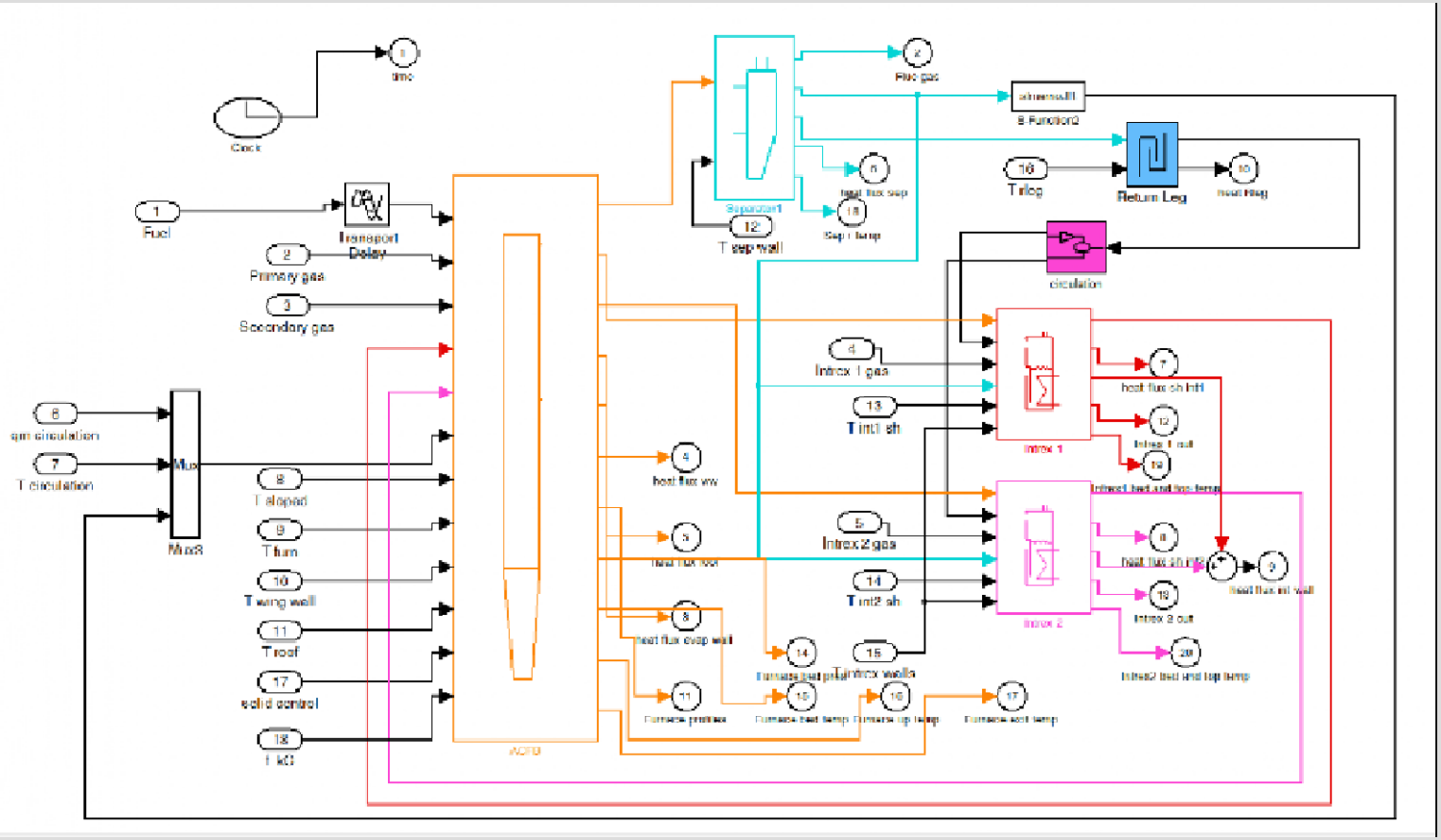
CFB boilers

- Circulating fluidized bed boilers
  - mixed in inert material, fluidized by air
  - closed circle via cyclone
  - heat transfer in furnace & from flue gases
  - multifuel capabilities
  - oxyfuel options
  - once-through designs
- Control
  - fuel & air flows
  - stoichiometric conditions, fluidization, temperatures, load changes
  - emissions, corrosion,..
- Unmeasured states
  - fuel inventory, inert mass & distribution,
  - fuel characteristics (heat value, fuel mix, particle size distribution), ...



- FW's CFB with INTREX.
- hot-loop model
  - semi-physical
  - furnace, separator, Intrex
  - no steam-side
  - validated with real plants

hot-loop model



- furnace, separator, Intrex, return

HOPE-project

- HOPE = HOt-loop Parameter Estimation
- allow originally constant parameters to vary with time, $\boldsymbol{\theta}_k$..
- ..in order to match simulations with measurements
- search for optimal $\boldsymbol{\theta}_k$
- assess feasibility of estimated $\boldsymbol{\theta}$

- considered variables:
  - char affinity
  - heat transfer coefficients
    - wing wall
    - roof
    - furnace wall
  - fuel moisture
- data from reactivity tests
- minimize squared deviation in O_2 & T_{bed}

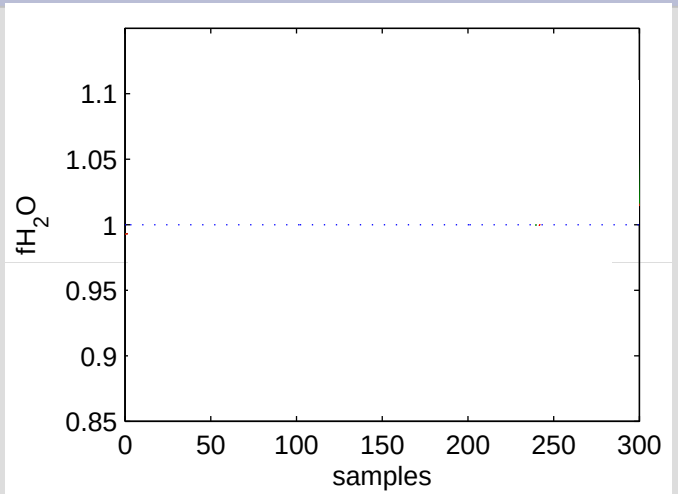
Initial studies (fixed)

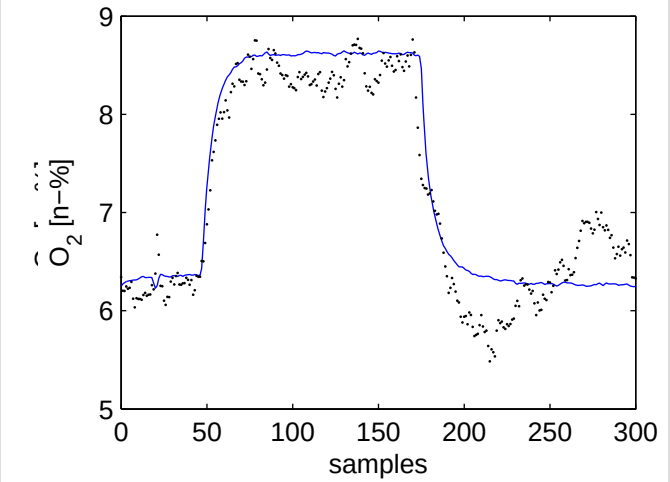
- Simulate with fixed **θ**
  - nominal value at 1
  - simulate with {0.8, 0.9, 1, 1.1, 1.2}
  - static gains

	O_2	T_{bed}	T_{roof}	T_{exit}
fkc	-0.001	-0.268	-0.198	-0.197
$fHTC1$	0	-34.158	-40.778	-44.006
$fHTC2$	0	-2.885	-5.744	-4.760
$fHTC3$	0	-117.020	-131.180	-135.530
$fH2O$	2.459	-64.161	-60.625	-58.721

TABLE 1. Static gains. Rows: parameter θ^p, columns: output y^o.

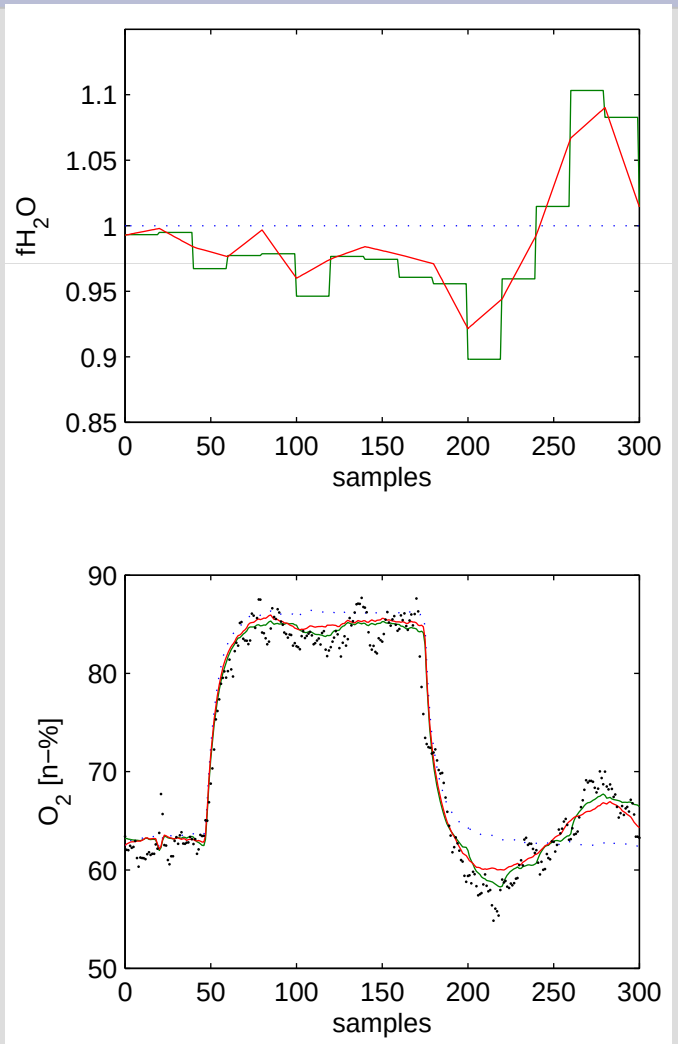
The table shows the gain $\Delta y^o / \Delta \theta^p$ around the nominal value $\boldsymbol{\theta} = \mathbf{1}$.





Initial studies (intervals)

- global optimization
  - simple gradient search
  - 0/1st order hold
- simulations follow measurements well
  - with feasible parameter change amplitudes
- speed & # evaluations, accuracy, robustness
- potential directions:
  - advanced gradient methods,
  - random search,
  - parameterized trajectories,
  - sequential methods



Sequential search

- View PE as a state estimation problem
- Solve problem sequentially
  - 'natural' when new data becomes available in time

- Monte Carlo
  - rely on random sampling
- Bayesian reasoning
  - Kalman filter, EKF
  - Particle filter, UKF,...

Particle filtering

- Describe unknown pdf with *N* particles
  - approximation, $N \rightarrow \infty$
  - propagate particles using model
  - update population with measurements (death/survival)
- no linear/Gaussian limitations, can handle complex dynamics
- computationally heavy
  - requires 'cheap' computing power + memory

Particle filtering (markovian)

A dynamic model describes how the state vector evolves with time, a measurement equation relates the received measurement to the state vector:

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{w}_k) \quad (2)$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (3)$$

where

- k is the sampling instant ($t = kT$, where T is the sampling interval and t is the real time);
- $\mathbf{x}$ is the state vector to be estimated, the pdf of $\mathbf{x}_0$ is assumed to be known;
- f and h are known (possibly non-linear) functions;
- $\mathbf{w}$ is a white noise sequence (the process noise), the pdf of $\mathbf{w}$ is assumed to be known;
- $\mathbf{y}$ is the vector of received measurements;
- $\mathbf{v}$ is a white noise sequence (the measurement noise), the pdf of $\mathbf{v}$ is assumed to be known and independent of $\mathbf{w}$.

Equation (2) defines a Markov process. An equivalent probabilistic description of the state evolution is $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$, the transition density. An equivalent probabilistic model for (3) is $p(\mathbf{y}_k|\mathbf{x}_k)$. With initial conditions $p(\mathbf{x}_0)$ the specification is complete.

Particle filtering (bayesian)

In the Bayesian approach, one attempts to construct the posterior pdf of the state vector: $p(\mathbf{x}_k|\mathbf{Y}_k)$, where $\mathbf{Y}_k$ denotes the set of all measurements recieved up to and including instant k, $\mathbf{Y}_k = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$. The initial condition is given by $p(\mathbf{x}_0|\mathbf{Y}_0)$ where $\mathbf{Y}_0$ is the empty set. The formal Bayesian filter consists of prediction and update operations. The prediction operation propagates the posterior pdf at instant $k-1$ to a prior at k:

$$\underbrace{p(\mathbf{x}_k|\mathbf{Y}_{k-1})}_{\text{prior at } k} = \int \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1})}_{\text{dynamics}} \underbrace{p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})}_{\text{posterior from } k-1} d\mathbf{x}_{k-1}$$

The prior pdf may be updated with the new measurement $\mathbf{y}_k$:

$$\underbrace{p(\mathbf{x}_k|\mathbf{Y}_k)}_{\text{posterior}} = \underbrace{p(\mathbf{y}_k|\mathbf{x}_k)}_{\text{likelihood}} \underbrace{p(\mathbf{x}_k|\mathbf{Y}_{k-1})}_{\text{prior}} \Big/ \underbrace{p(\mathbf{y}_k|\mathbf{Y}_{k-1})}_{\text{normalization}}$$

where $p(\mathbf{y}_k|\mathbf{Y}_{k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{Y}_{k-1})\, d\mathbf{x}_k$. The measurement likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$ is regarded as a function of $\mathbf{x}$ given $\mathbf{y}$.

Particle filtering (SIR)

1. Suppose that a set of N random samples from the posterior pdf $p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})$ are available. Denote thes particles by

$$\{\mathbf{x}_{k-1}^i\}_{i=1}^N$$

i.e., a set of N particles indexed with i from 1 to N.

2. The *prediction* phase consists of passing each of these posterior particles (from instant

3. The *update* phase consists of calculation of a weight for each particle, normalization of weights, and resampling according to normalized weights. A weight $\overline{\omega}_k^i$ is calculated for each particle, based on the measurement likelihood (density function) evaluated at the value of the prior sample (and measurement):

$$\overline{\omega}_k^i = p\left(\mathbf{y}_k|\overline{\mathbf{x}}_k^i\right). \tag{4}$$

The weights are then normalized so that they sum to unity: $\omega_k^i = \frac{\overline{\omega}_k^i}{\sum_{j=1}^N \overline{\omega}_k^j}$. The prior particles are resampled (with replacement) according to the normalized weights to produce a new set of particles

$$\{\mathbf{x}_k^i\}_{i=1}^N \text{ such that } \Pr\left\{\mathbf{x}_k^i = \overline{\mathbf{x}}_k^j\right\} = \omega_k^j \text{ for all } j \text{ and } i$$

4. The new set of particles are samples of the posterior pdf at k. The cycle of the algorithm is complete, and we continue from Step 2 with this new set

$$\{\mathbf{x}_k^i\}_{i=1}^N.$$

Particle filtering (interpretations)

The measurement likelihood (4) can be intepreted as an indicator of those regions of the state-space that are plausible explanations of the observed measurement value [19]:

- If the value of the likelihood function is high, these state values are well supported by the measurement
- If the likelihood is low, these state values are unlikely
- If the likelihood is zero, these state values are incompatible with the measurement model

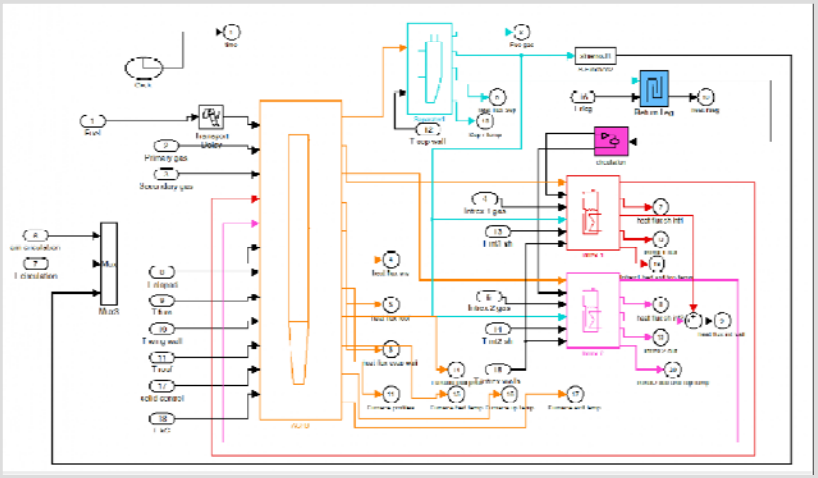
As with the Kalman filtering algorithm, we can add a deterministic system input $\mathbf{u}_{k-1}$ (plant control manipulations) to Step 2:

$$\overline{\mathbf{x}}_k^i = f_k\left(\mathbf{x}_{k-1}^i, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}^i\right).$$

The rest of the algorithm remains intact.

Simulations (hot-loop model)

- discrete-time state-space form
  - hot-loop states (660)
  - past inputs (20)
    - interpolation
  - unknown parameters (5→2)
    - random walk model, $N(0, 0.01^2)$
    - $½ < \theta_k < 2$
  - outputs (108→2)
    - measurement noise (gaussian)
    - O_2: $N(0, 0.2^2)$ %-vol
    - T_{bed}: $N(0, 5^2)$ ^oC



$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{y}_k &= h(\mathbf{x}_k, \mathbf{v}_k) \end{aligned}$$

- 1-step-ahead simulations
  - sequence length x # particles (=lots of..)
  - assuming initially in steady state
- *N*=500 (# particles)

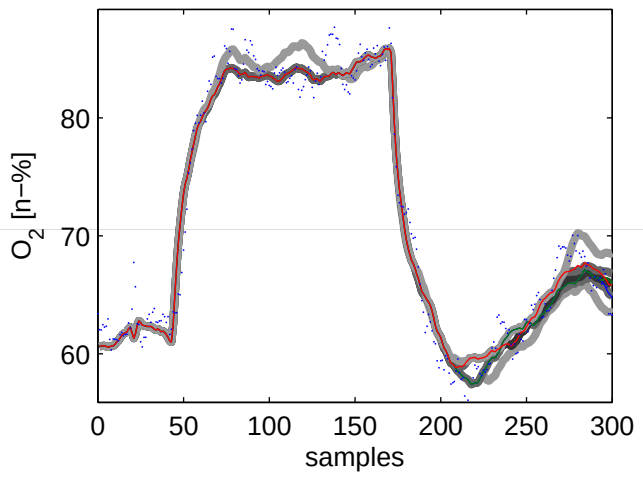
Simulations (algorithm)

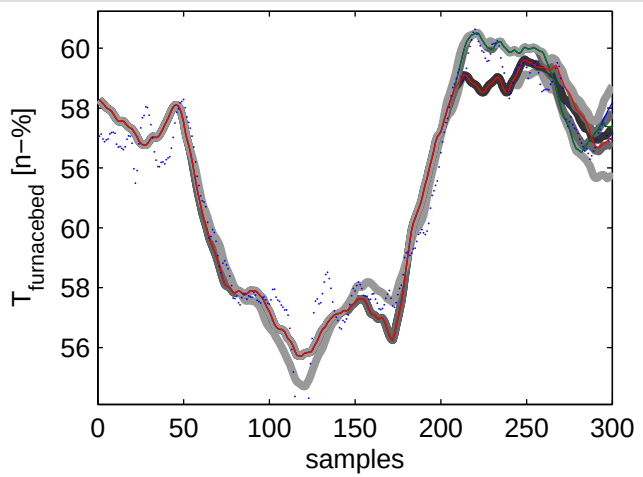
Algorithm 1 HOPE PF parameter estimation algorithm

Initialize algorithm parameters (N, $\boldsymbol{\Sigma}_{\mathbf{x}}$, $\boldsymbol{\Sigma}_{\mathbf{y}}$) and particle states $\mathbf{x}_0^i$ ▷ prior knowledge
for k=1...K **do** ▷ for each sample
  Read measurement data: $\mathbf{u}_k$ and $\mathbf{y}_{k+1}$
  for $i = 1...N$ **do** ▷ for each particle
    Read particle state $\mathbf{x}_k^i$ and extract $\mathbf{x}_k^{\mathrm{h},i}$, $\mathbf{u}_{k-1}$ and $\boldsymbol{\theta}_{k-1}^i$
    Generate $\mathbf{w}_k^i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}})$ ▷ exploration
    Compute $\boldsymbol{\theta}_k^i = \boldsymbol{\theta}_{k-1}^i + \mathbf{w}_k^i$
    Ensure parameter bounds $\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta}_k^i \leq \boldsymbol{\theta}_{\max}$
    Simulate hot-loop model $\mathbf{x}_{k+1}^{\mathrm{h},i} = f(\mathbf{x}_k^{\mathrm{h},i}, \mathbf{u}_k, \mathbf{u}_{k-1}, \boldsymbol{\theta}_k^i, \boldsymbol{\theta}_{k-1}^i)$ ▷ simulation[1]
    Append and store particle state $\mathbf{x}_{k+1}^i$
    Simulate hot-loop model measurement equations $\mathbf{y}_{k+1}^{\mathrm{h}} = h(\mathbf{x}_{k+1}^{\mathrm{h},i})$
    Weigh particle by evaluating $\overline{\omega}_{k+1}^i = \mathcal{N}(\mathbf{y}_{k+1} - \mathbf{y}_{k+1}^{\mathrm{h}}, \boldsymbol{\Sigma}_{\mathbf{y}})$ ▷ measurement
  end for
  Normalize particle weights ω_{k+1}^i
  Resample particle population with replacement ▷ SIR
  Compute on-line statistics of interest. ▷ application
end for
Compute statistics of interest. ▷ application

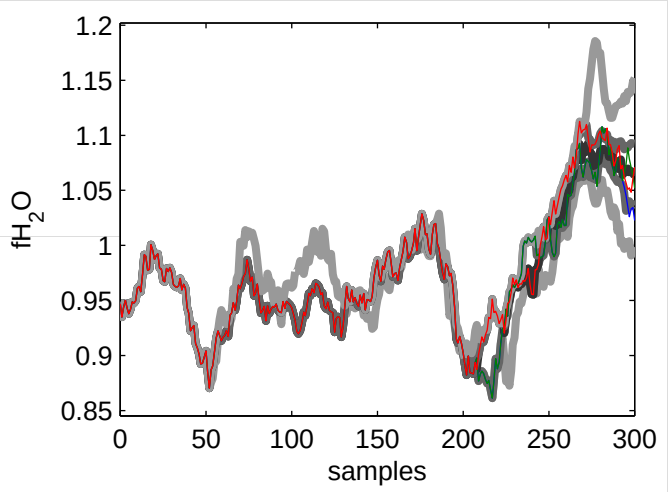
[1] Hot-loop model simulation uses 1st order interpolated inputs.

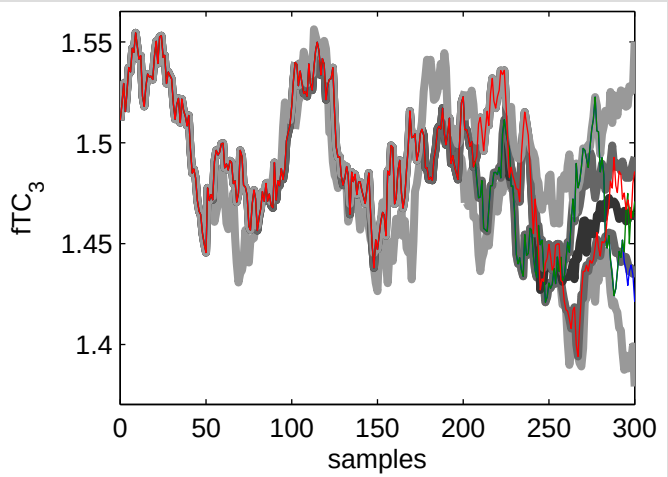
Simulations (PF)



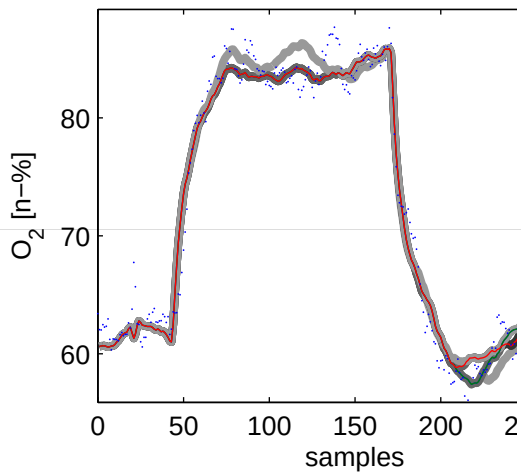


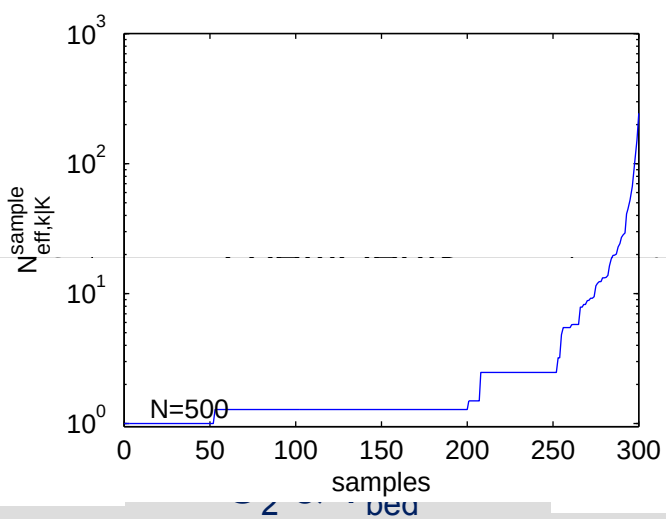
- estimated moisture & heat transfer coefficients
- predicted O_2 & T_{bed}
- measured O_2 & T_{bed}
- quantiles in MC
- a random trajectory
- a trajectory
- a random trajectory

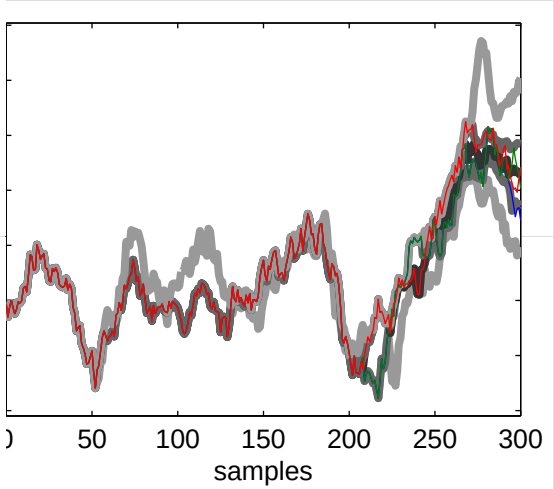


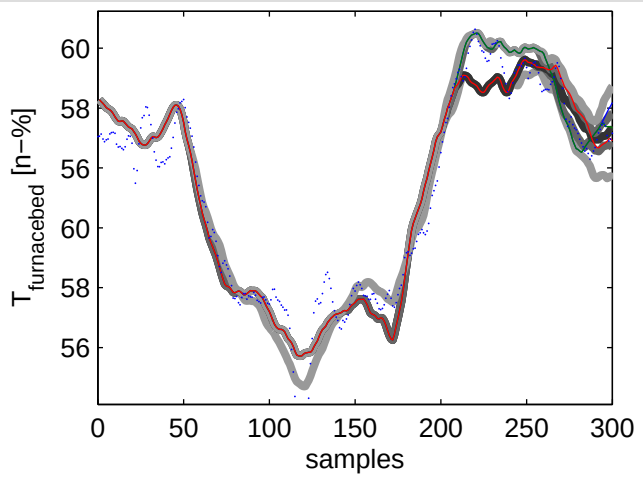


Simulations (smoothing)

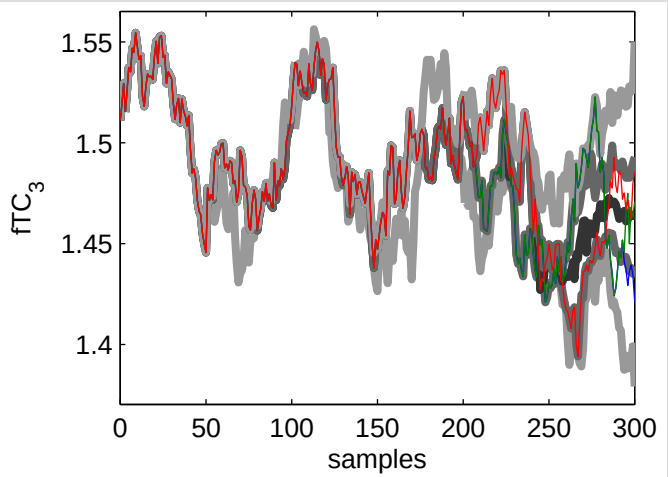




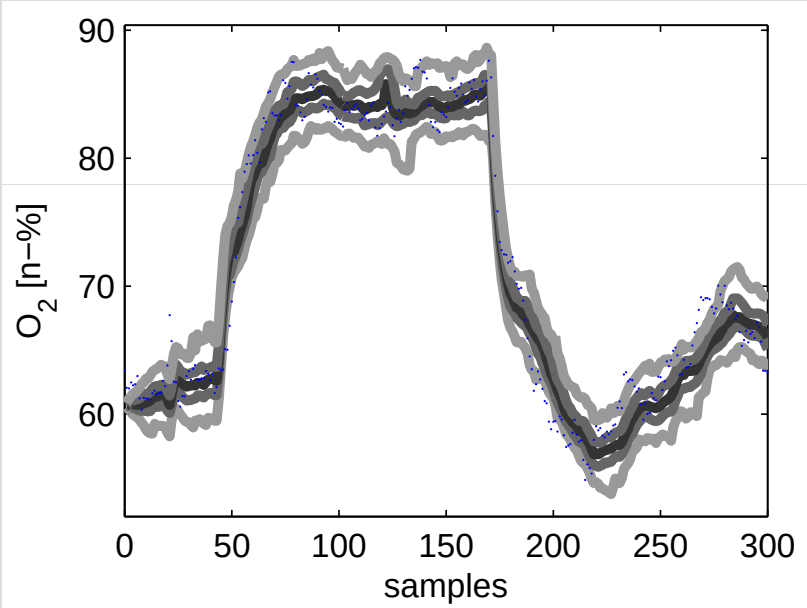


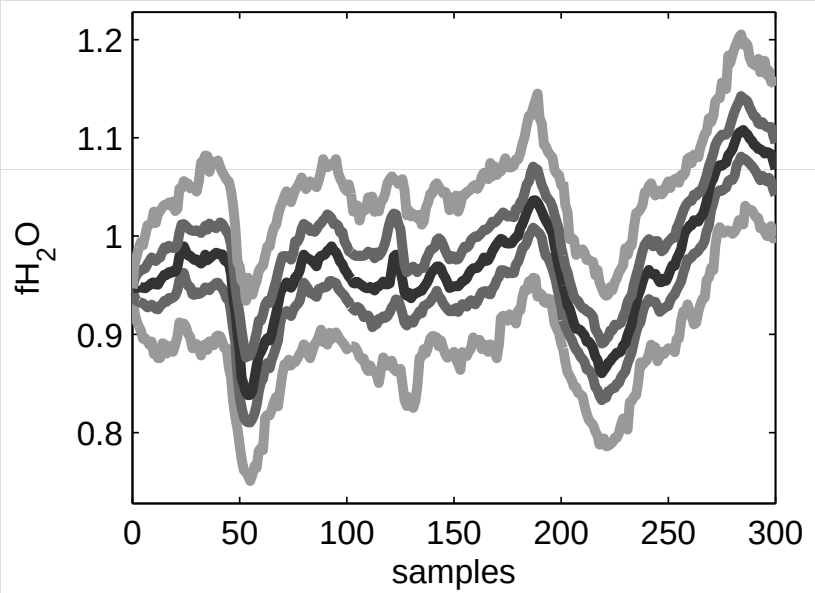


- quantiles in MC
- a random trajectory
- a trajectory
- a random trajectory



Simulations (filtering)

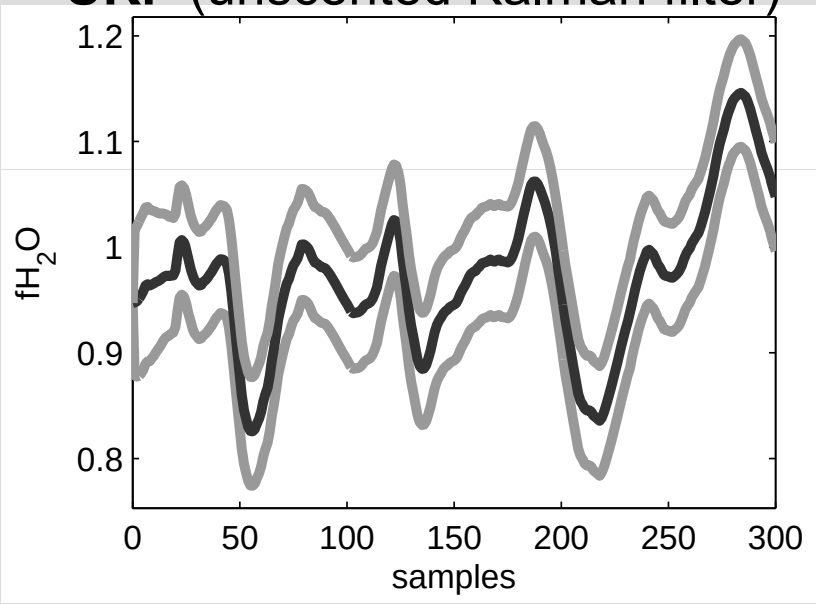




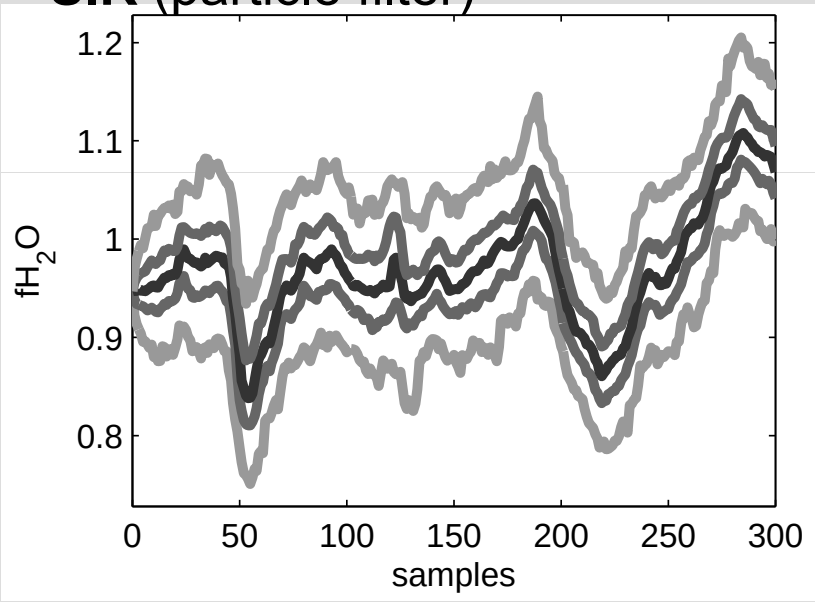
filter distributions $k|k$
using data up to (but not exceeding) k

Simulations (UKF)

UKF (unscented Kalman filter)



SIR (particle filter)



PF – 500 particles *(500 simulations in parallel)*
UKF – 5 sigma-points *(5 model simulations in parallel)*

Discussion & Conclusions

- Bayesian state estimation for model calibration / experiment test assessment
  - both model & measurements are efficiently used, the role of the two can be transparently interpreted
  - theoretically solid
  - estimates/predictions are not limited to samples, expectations, or prior distributions
  - simple to implement for any simulation model

Discussion & Conclusions ii

- For more info, see:

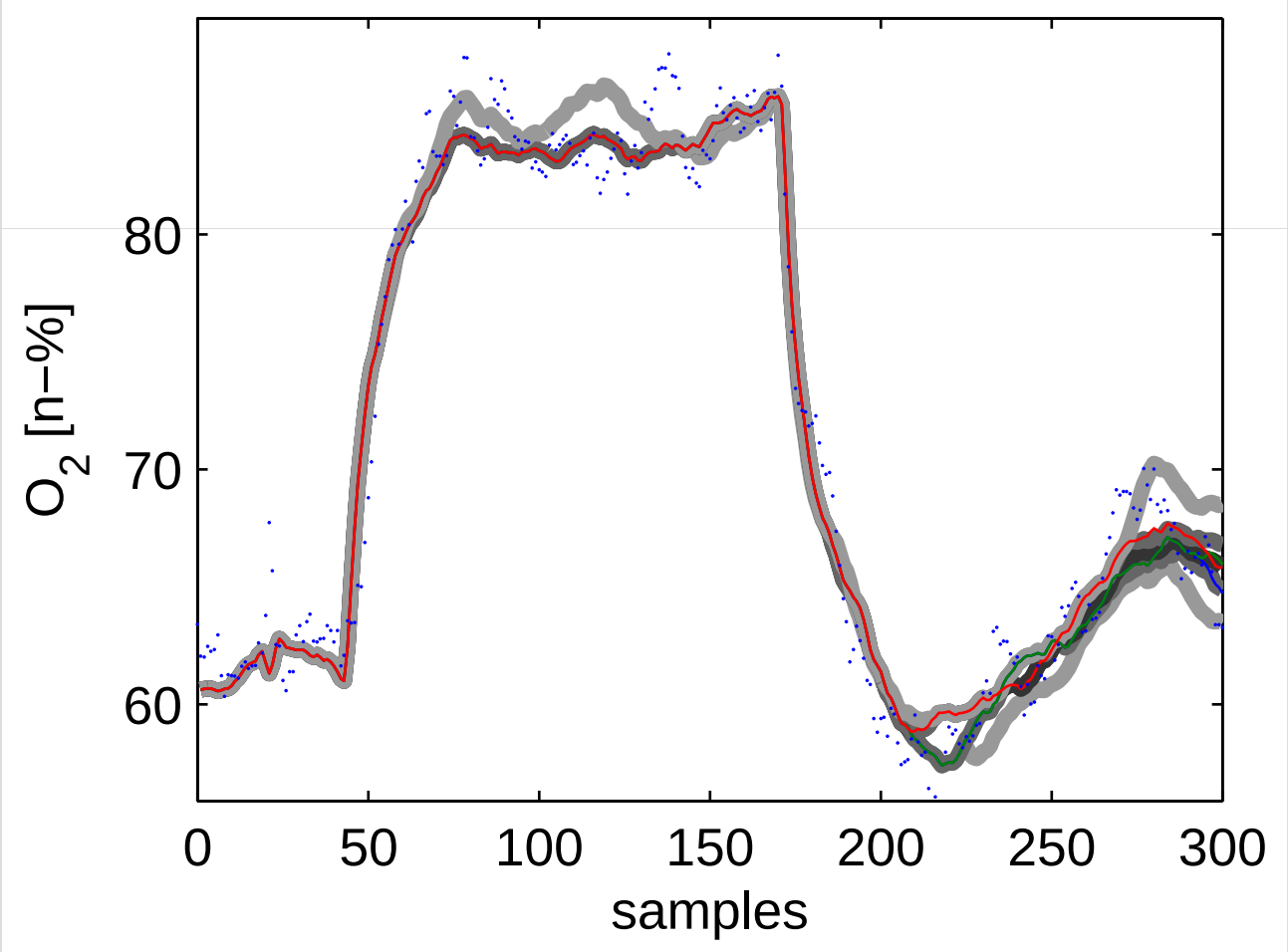
  Ikonen, E., J. Kovacs & J. Ritvanen (2013) Circulating fluidized bed hot-loop analysis, tuning, and state-estimation using particle filtering. *International Journal of Innovative Computing, Information and Control*, **9** (8), pp 3357 – 3376.

- Future directions
  - user feedback from engineers => further developments
  - smoothing UKF algorithms?
  - how to illustrate/use sequences of multidimensional distributions?

Thank you!

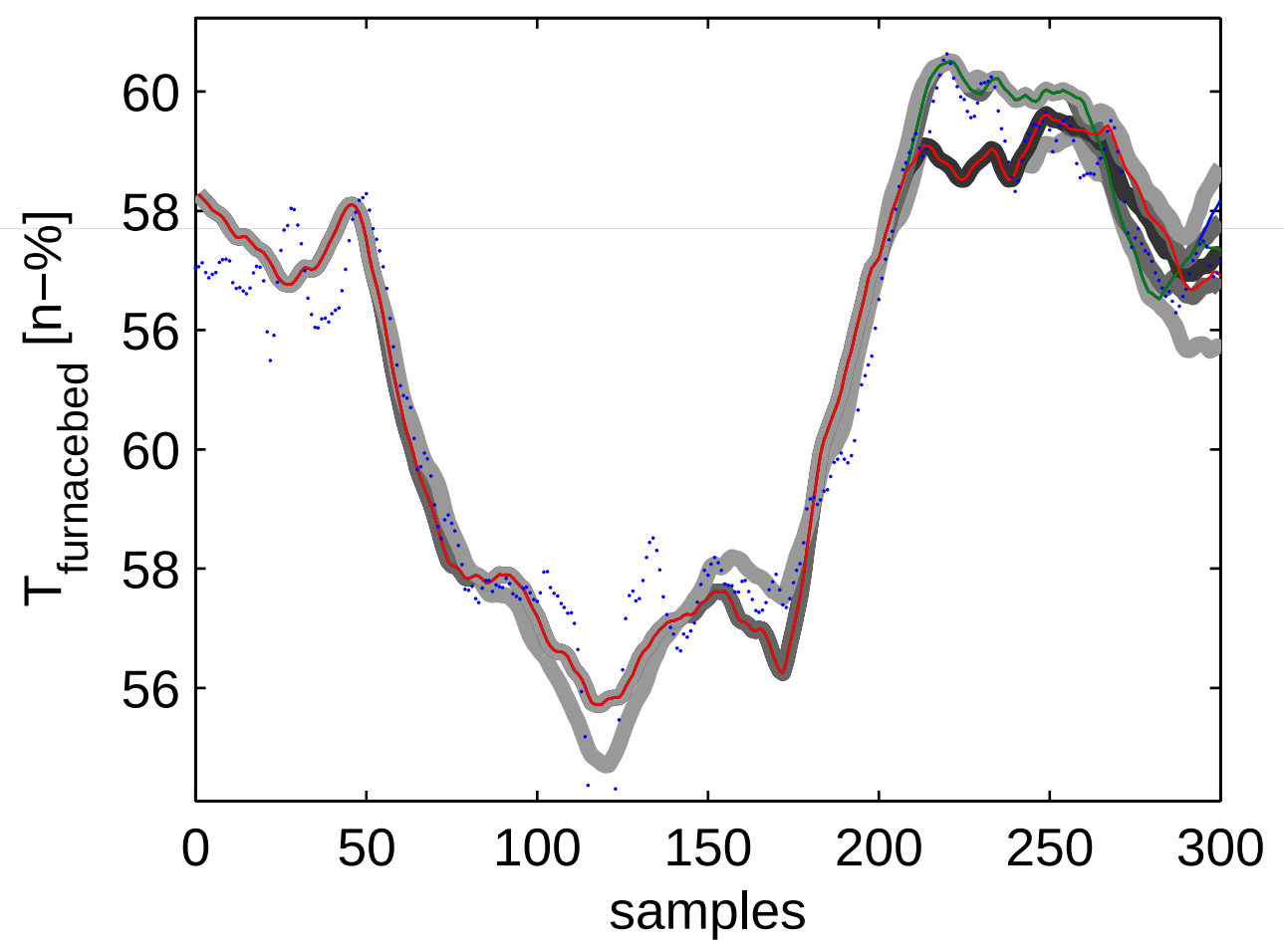
SYTE / Department of Process and Environmental Engineering
http://www.oulu.fi/pyosys
Enso.Ikonen@oulu.fi

Simulations (smoothing)



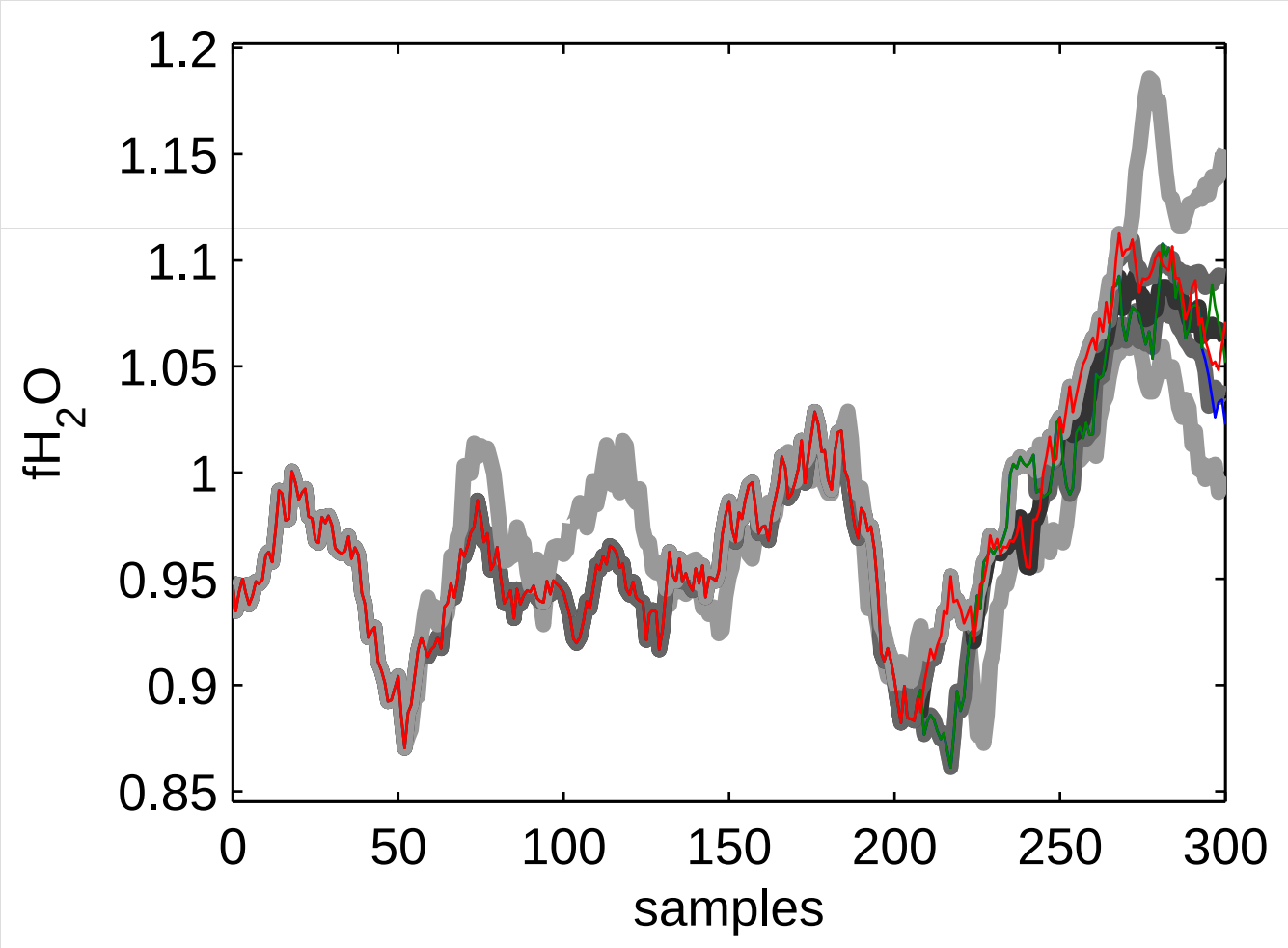
back

Simulations (smoothing)



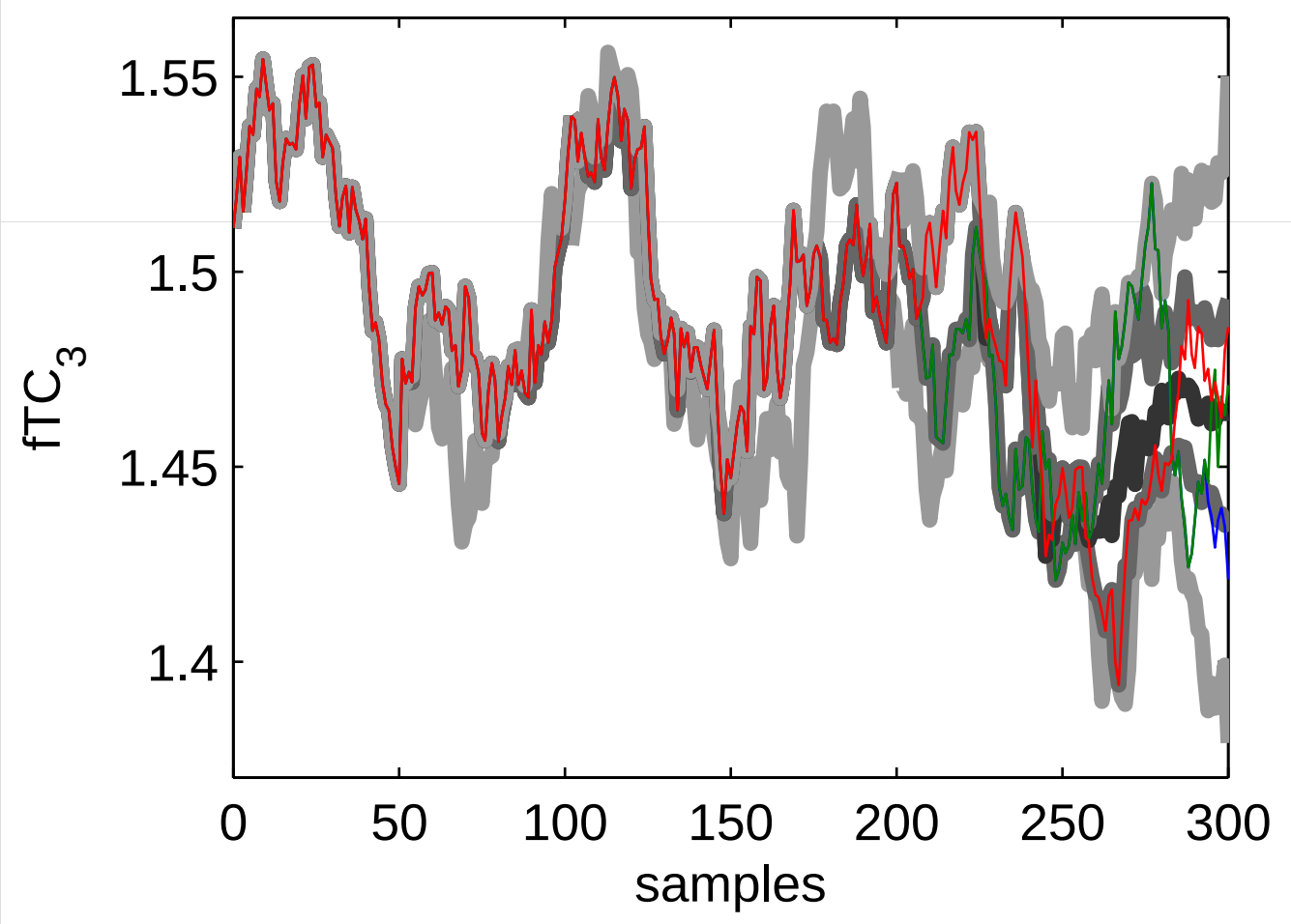
back

Simulations (smoothing)



back

Simulations (smoothing)



back