

Learning to Control

State Estimation

Cambridge, 30th April 2012
Roger Frigola

What is Control?

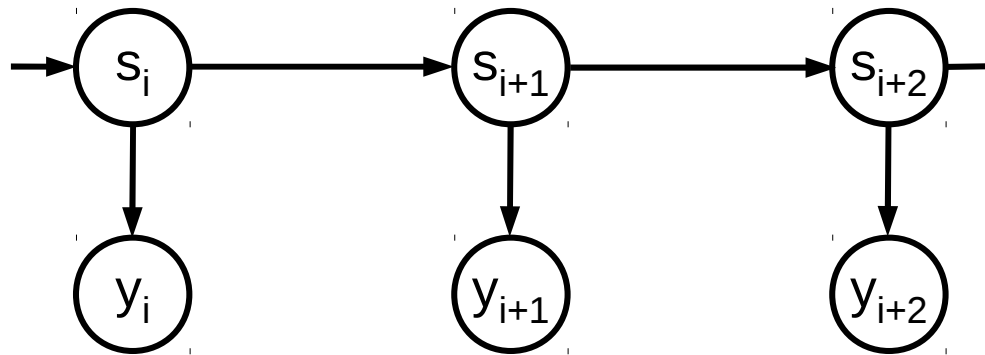
- Sensing + Computation + Actuation
- Goals: Performance, Stability, Robustness.
- [Demo](#)

Objective

- The control policy needs to run in real time: find a policy that computes control actions based *solely on the current estimate of the state*.
- The model of the dynamics is stochastic. *Minimise the expected loss over a horizon*.

State Space Models (SSMs)

- As opposed to Hidden Markov Models, the states are continuous.

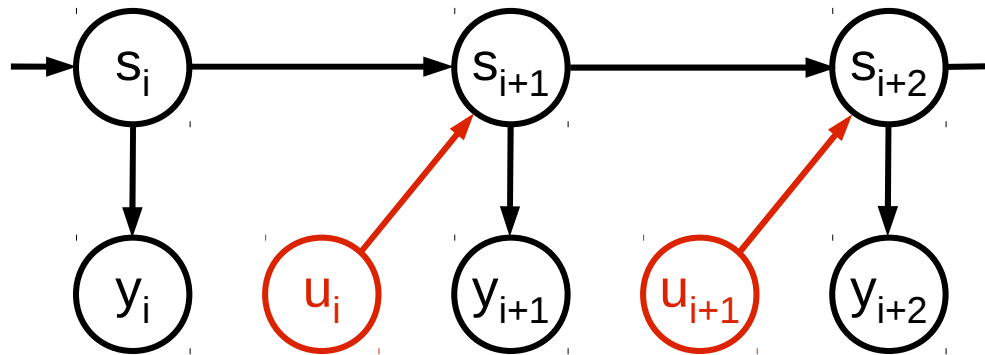


$$s_{i+1} = f(s_i) + \varepsilon_i$$

$$y_i = g(s_i) + \delta_i$$

Control Inputs

- We can influence the state transitions.

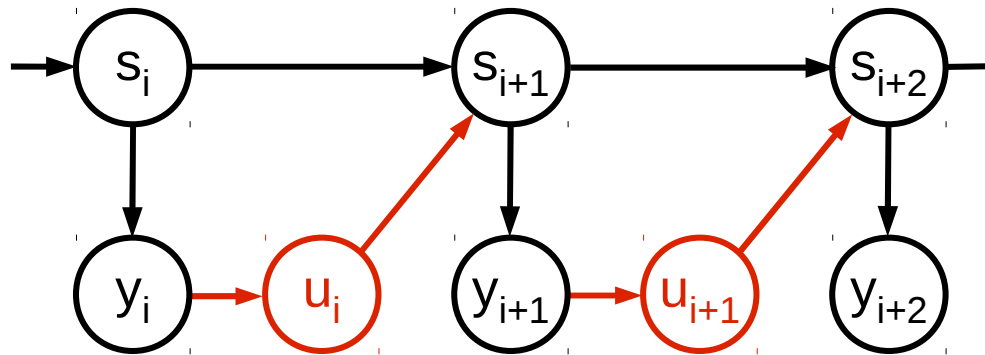


$$s_{i+1} = f(s_i, u_i) + \varepsilon_i$$

$$y_i = g(s_i) + \delta_i$$

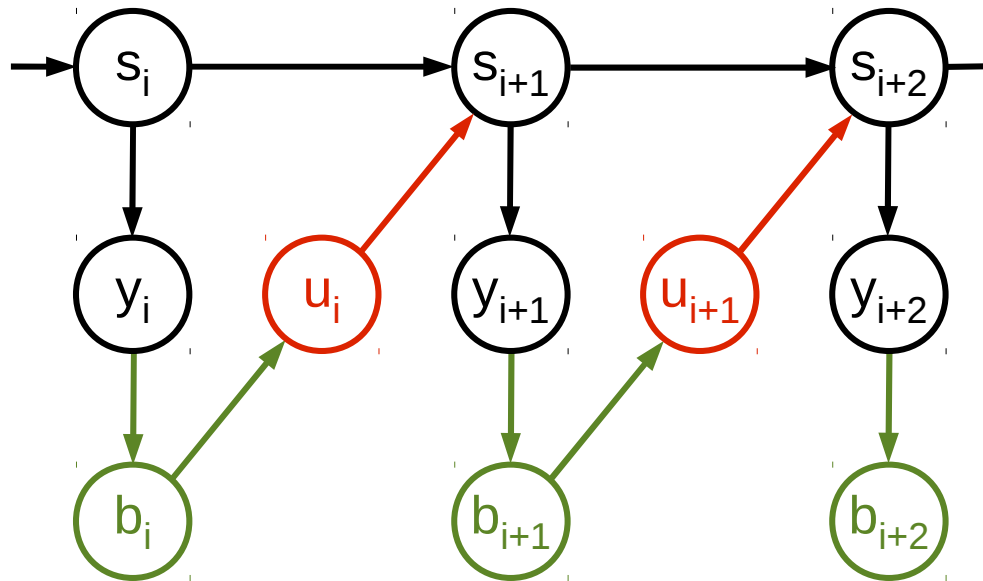
Feedback

- The input is a function of the measurement: output feedback.



Feedback with State Estimation

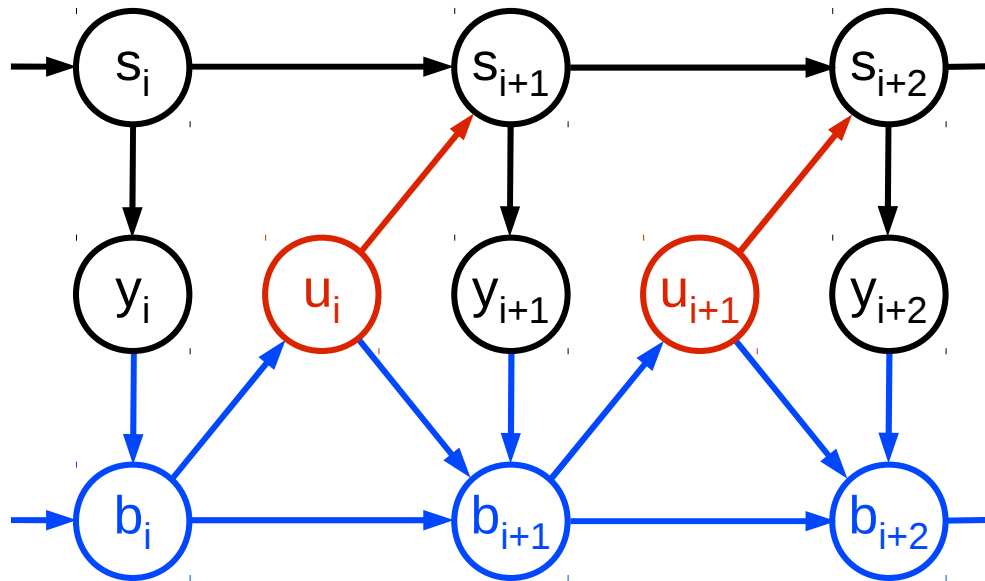
- We may have a rough idea about s and use it as a prior: $p(s)$
- Inference is explicitly represented as part of the model!



$$b_i = p(s_i | y_i) \propto p(y_i | s_i) p(s_i)$$

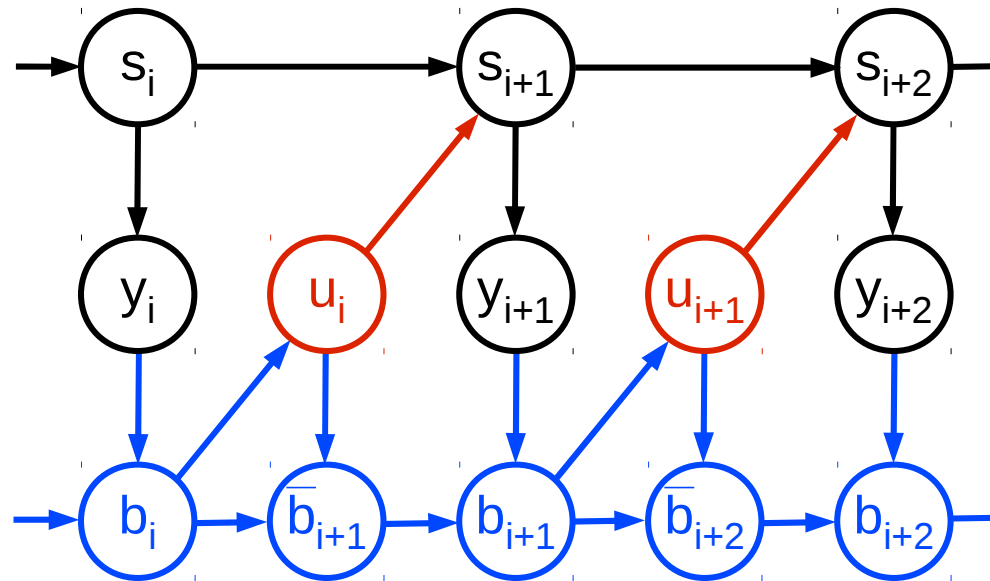
State Estimation Using a Model of the Dynamics

- Using a model of the transition dynamics can be VERY useful.



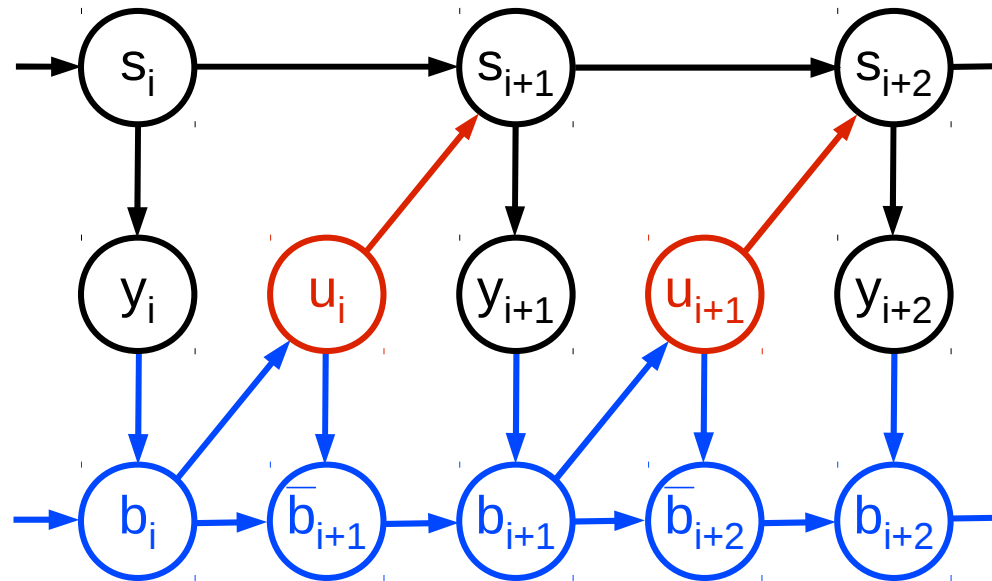
Bayesian State Estimation (Bayesian Filtering)

- Step 1 - Prediction: the current belief is propagated forward using the dynamics model.
- Step 2 - Update: the propagated belief is used as a prior to compute the new belief.



Bayesian State Estimation (Bayesian Filtering)

- Step 1 - Prediction: the current belief is propagated forward using the dynamics model.
- Step 2 - Update: the propagated belief is used as a prior to compute the new belief.



$$\overline{bel}(s_{i+1}) = p(s_{i+1} | y_{1:i}, u_{1:i})$$

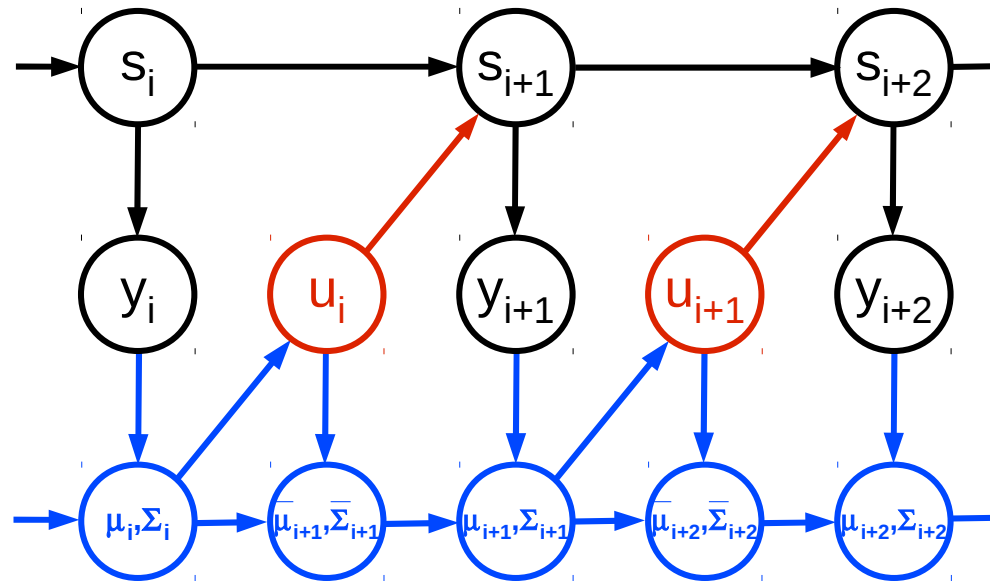
$$bel(s_{i+1}) = p(s_{i+1} | y_{1:i+1}, u_{1:i})$$

$$\overline{bel}(s_{i+1}) = \int p(s_{i+1} | s_i, u_i) bel(s_i) ds_i$$

$$bel(s_{i+1}) = \eta p(y_{i+1} | s_{i+1}) \overline{bel}(s_{i+1})$$

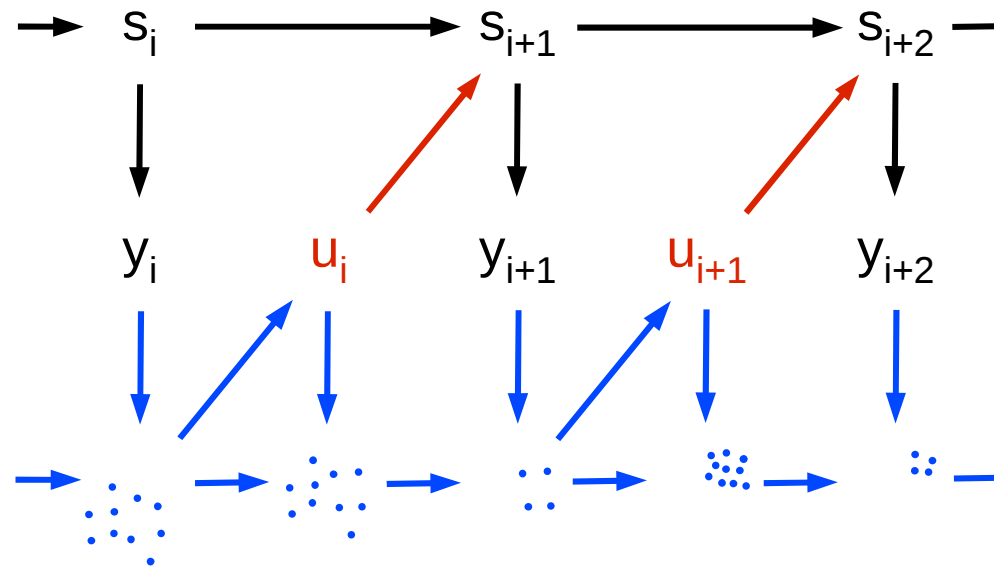
Kalman Filtering (Linear-Gaussian Model)

- The Kalman filter is the analytical Bayesian Filter solution for Linear-Gaussian State Space Models.



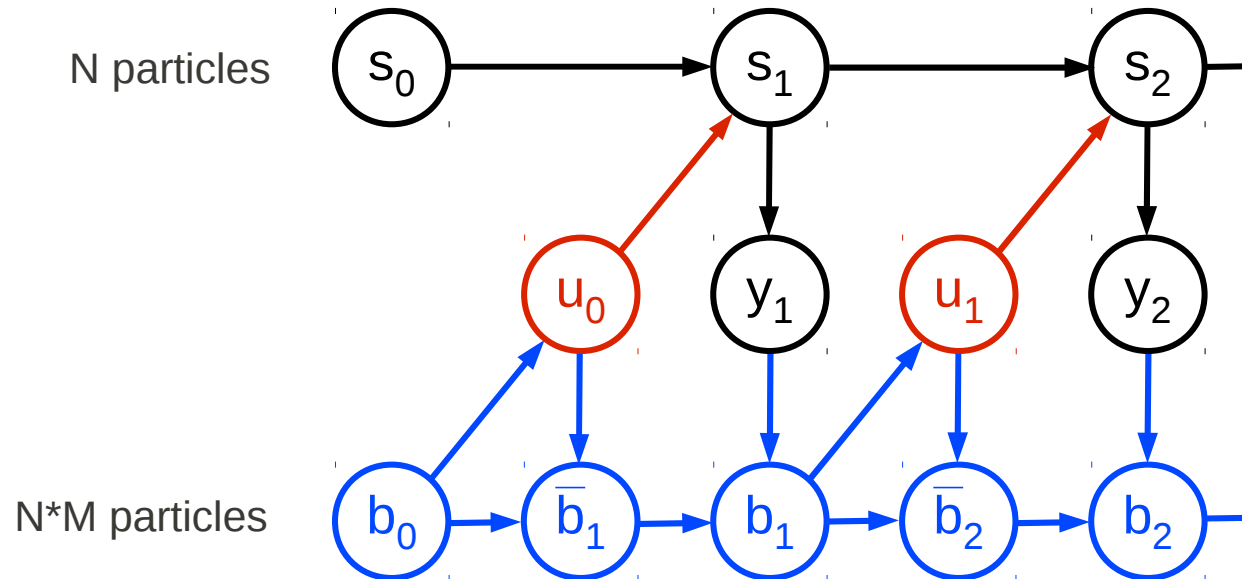
Particle Filtering

- Sequential Monte Carlo method for arbitrary systems, e.g. nonlinear and non-Gaussian.



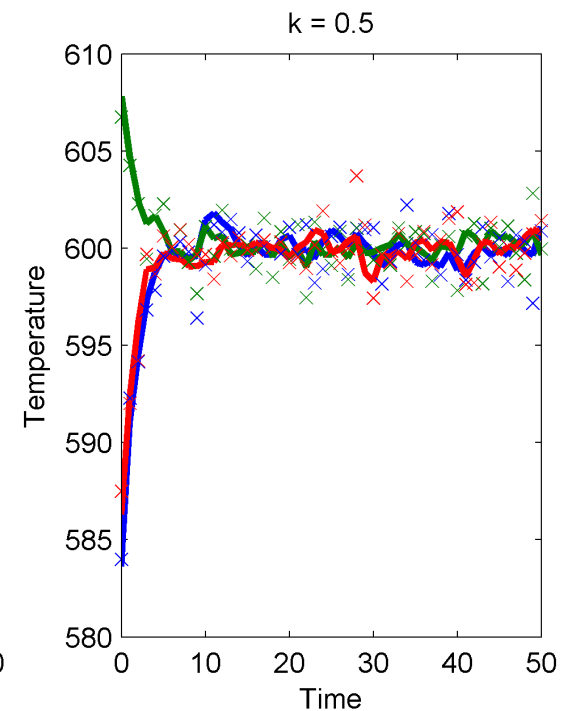
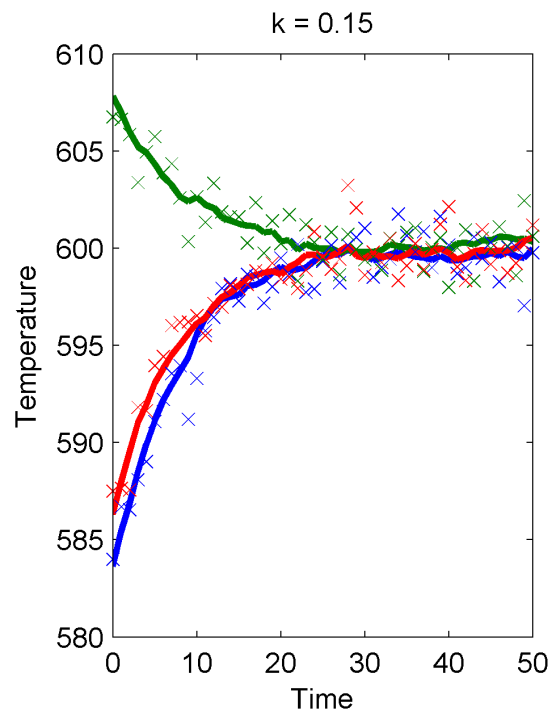
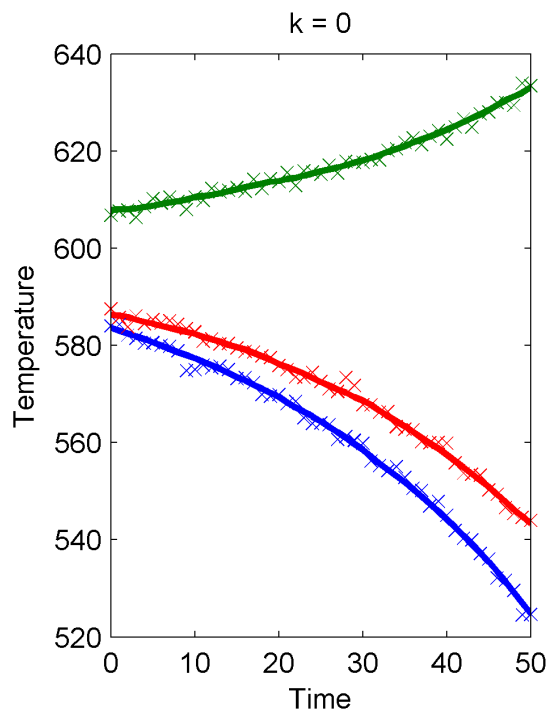
Sampling from the Generative Model

- We sample N independent chains of states.
- Each of those N chains has its own particle filter with M particles.



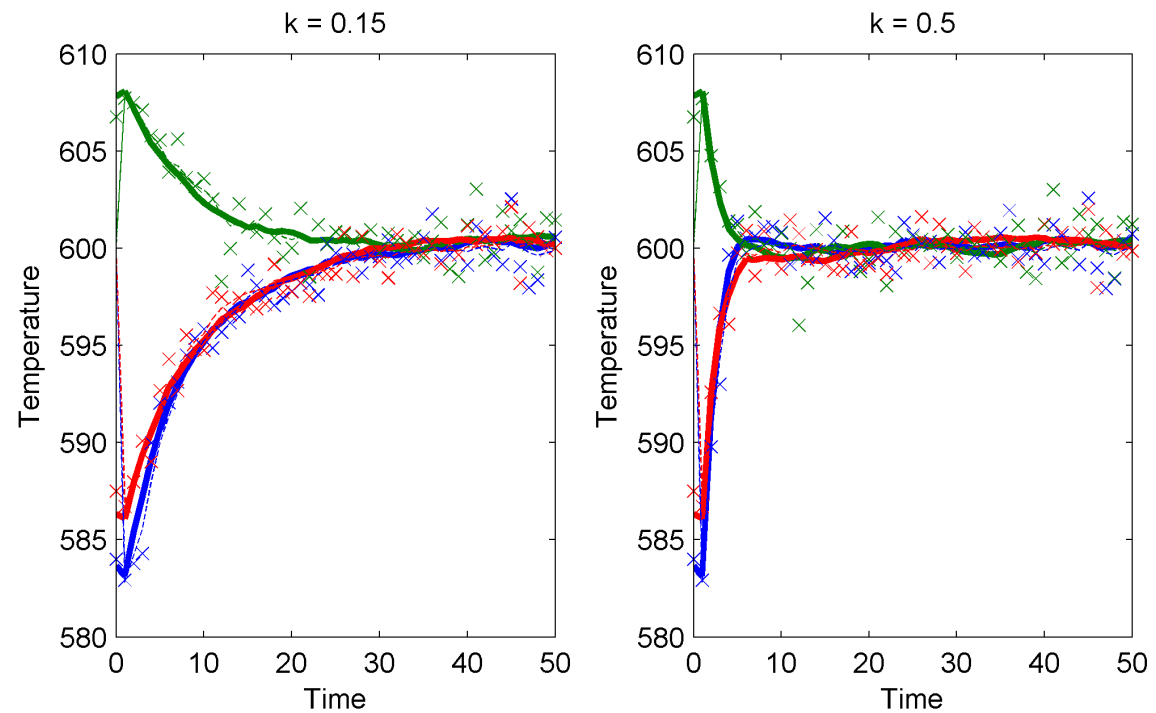
Sampling Example (1/3)

- A 1D example: control of temperature in a nuclear reactor.
- $N=3$ particles. Each of those has its own particle filter with $M = 100$ particles.



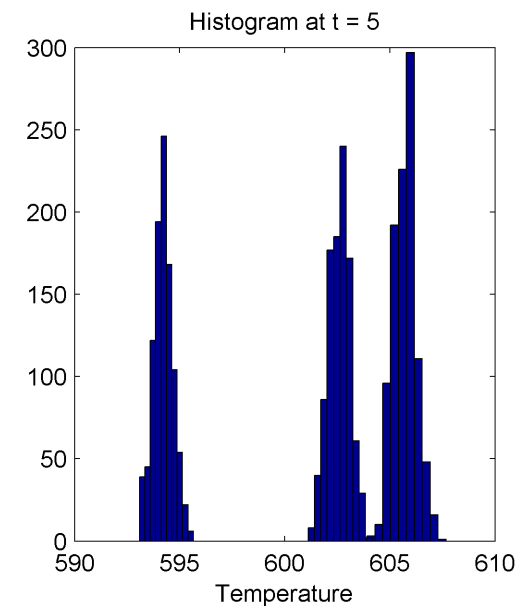
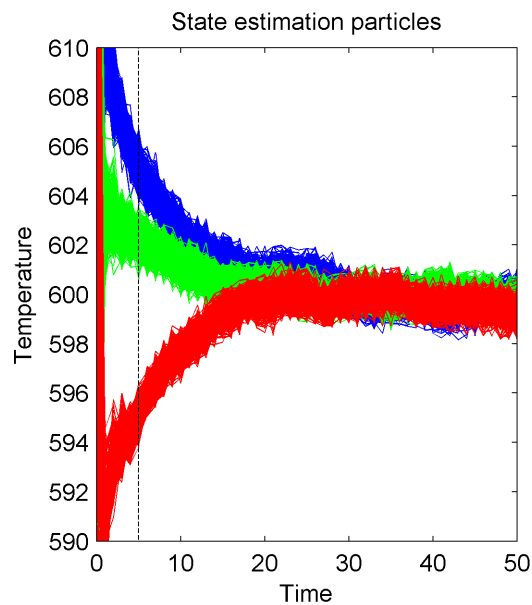
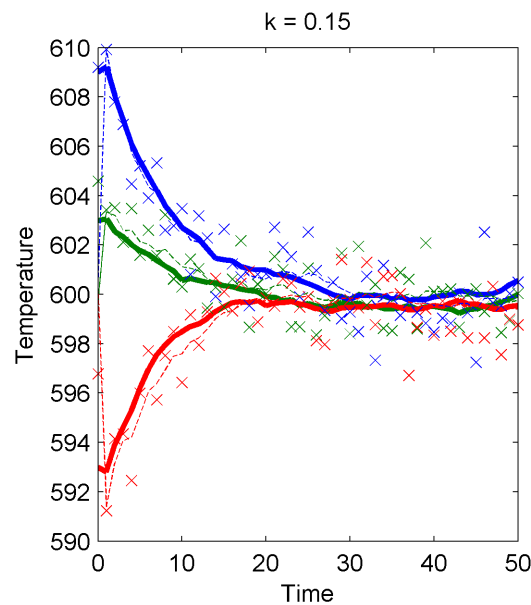
Sampling Example (2/3)

- Feedback using the state estimated by a Particle Filter.
- Control signal is proportional to the mean of the particles in the filter.

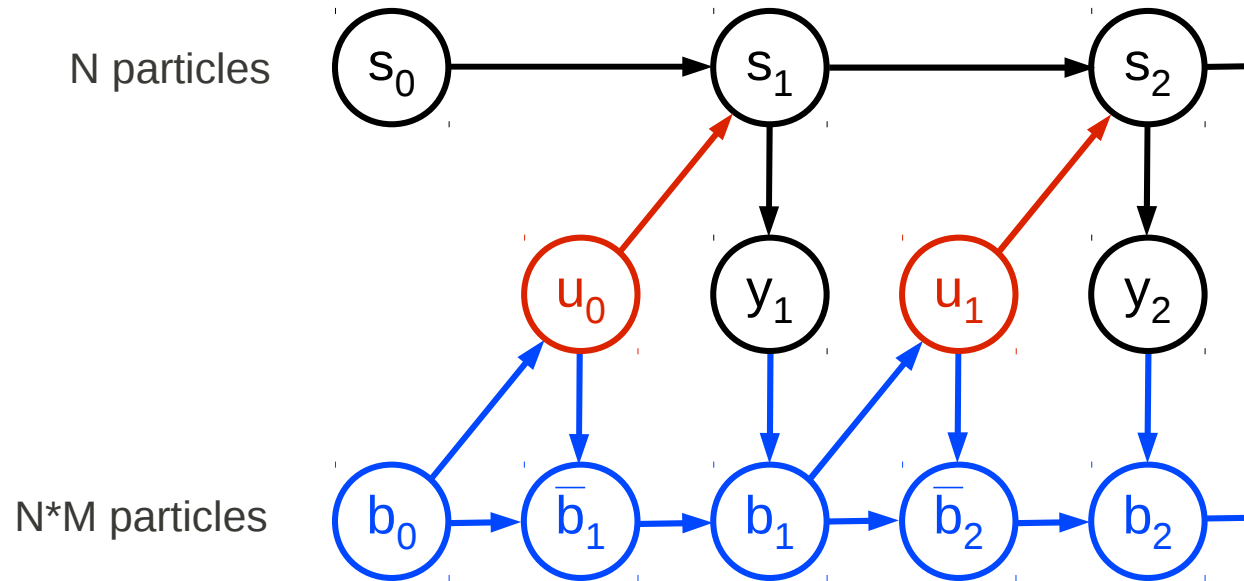


Sampling Example (3/3)

- Each of the three sampled trajectories has its own particle filter with $M=1000$ particles.
- This is a Linear-Gaussian model: the belief is Gaussian.



Recap



Future Work

- Do we need to model distributions over belief distributions? (probably not)
- Effect of imperfect knowledge of the dynamics model.
- Is it beneficial to train a policy that uses the variance in the belief (in real time)?
- How well can we deal with partial observability of the state?
- Representation of the state in a latent space.