Estimating Parameters and Hidden States in Biological Networks with Particle Filters

Nicolas Brunel, Minh Quach and Florence d'Alché-Buc

IBISC FRE CNRS 2873
University of Evry and Genopole, France

4th September 07 / PMNP workshop

Outline

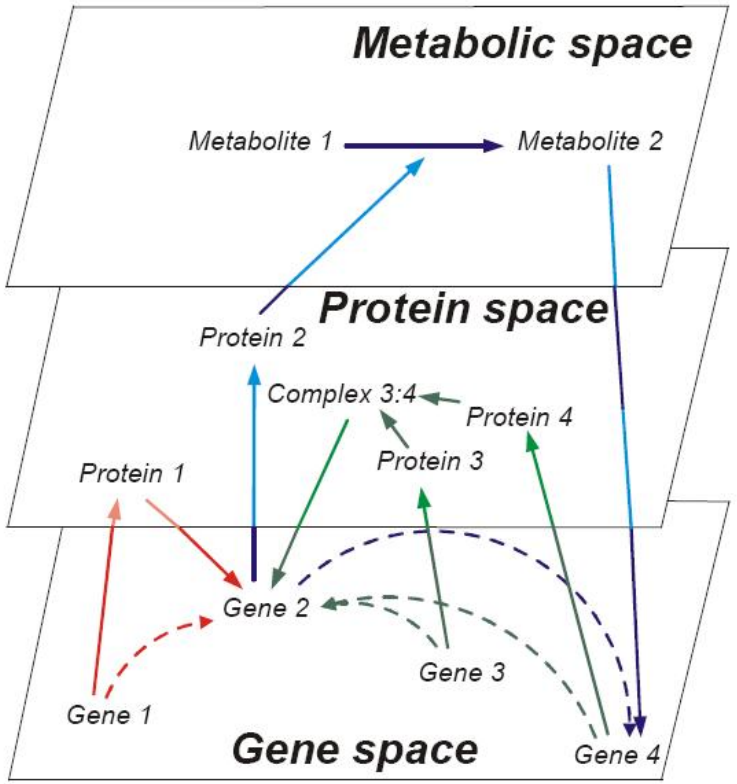
Outline

Problem: Reverse Engineering of Biological Networks

ODE Model

$$\frac{d\mathbf{x}(\mathbf{t})}{dt} = \mathbf{f}(\mathbf{x}(t);\boldsymbol{\theta})$$
$$\mathbf{y}(t) = \mathbf{H}(\mathbf{x}(t);\boldsymbol{\theta}) + \epsilon(t)$$

- $\mathbf{x}(t)$: state variables at time t (protein, mRNA, metabolite concentrations)
- $\mathbf{f}$: nonlinear function, derived from biochemical reactions.
- $\boldsymbol{\theta}$: parameter set (kinetic parameters, rate constants,...)
- $\mathbf{H}$ is a nonlinear observation function
- $\epsilon(t)$ is a i.i.d measurement noise



Problem

- Given: A sequence of observed data : $\mathbf{y}_{1:K} = \{\mathbf{y}_1, ..., \mathbf{y}_K\}$ at time $t_1, t_2, ..., t_K$
- Goal: Estimation of parameters $\boldsymbol{\theta}$ and states $\mathbf{x}(t)$

Outline

Nonlinear State-Space Model

Continuous time ODE model

$$\begin{aligned} \frac{d\mathbf{x}(\mathbf{t})}{dt} &= \mathbf{f}(\mathbf{x}(t);\boldsymbol{\theta}) \\ \mathbf{y}(t) &= \mathbf{H}(\mathbf{x}(t));\boldsymbol{\theta}) + \boldsymbol{\epsilon}(t) \end{aligned}$$

Nonlinear State-Space Model

Continuous time ODE model

$$\begin{aligned} \frac{d\mathbf{x}(\mathbf{t})}{dt} &= \mathbf{f}(\mathbf{x}(t);\boldsymbol{\theta}) \\ \mathbf{y}(t) &= \mathbf{H}(\mathbf{x}(t));\boldsymbol{\theta}) + \boldsymbol{\epsilon}(t) \end{aligned}$$

The corresponding discrete-time augmented state-space model

- Augmented states:

$$\begin{aligned} \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k \\ \mathbf{x}(t_{k+1}) &= \mathbf{F}(\mathbf{x}(t_k);\boldsymbol{\theta}_k) \end{aligned}$$

with

$$\mathbf{F}(\mathbf{x}(t_k);\boldsymbol{\theta}) = \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}(\tau);\boldsymbol{\theta})d\tau$$

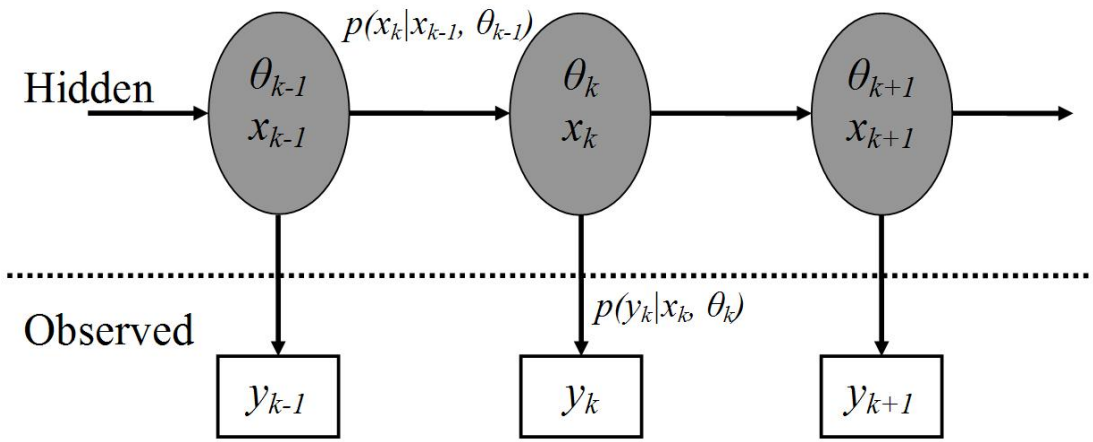
- observation model:

$$\mathbf{y}(t_k) = \mathbf{H}(\mathbf{x}(t_k);\boldsymbol{\theta}_k) + \boldsymbol{\epsilon}(t_k)$$

Bayesian estimation: Filtering

Given:

- Prior distribution over the initial state and parameters: $p(\mathbf{x}_1, \boldsymbol{\theta}_1)$
- A state transition model: $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1})$
- An observation model: $p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}_k)$
- A sequence of observations: $\mathbf{y}_{1:K} = \{\mathbf{y}_1, ..., \mathbf{y}_K\}$



Estimating the posterior distributions

$p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ $\quad k = 1, 2, \ldots, K$

Recursive Filtering Algorithm

Two steps

1. Prediction: $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k$

Recursive Filtering Algorithm

Two steps

1. Prediction: $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k$
2. Update:

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}) = \frac{p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})}{p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k})}$$

where:

$$p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})d\mathbf{x}_{k+1}$$

Recursive Filtering Algorithm

Two steps

1. Prediction: $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k$
2. Update:

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}) = \frac{p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})}{p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k})}$$

where:

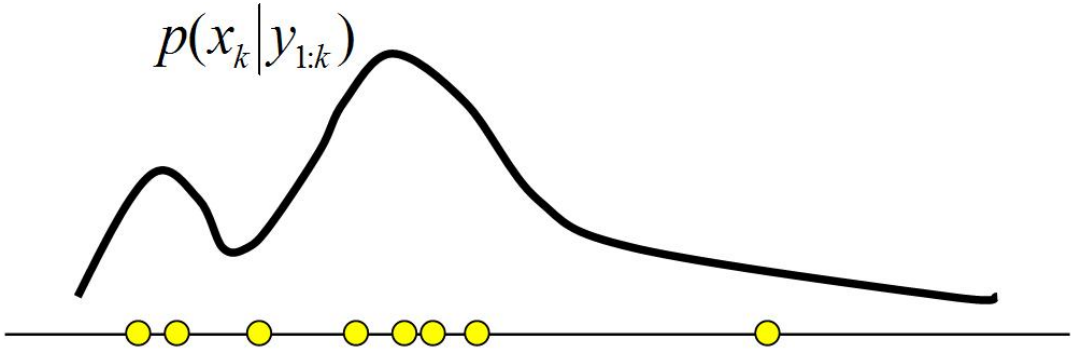
$$p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})d\mathbf{x}_{k+1}$$

- Analytical solution obtained only when $\mathbf{F}$, $\mathbf{H}$ are linear and $p(\mathbf{x}_1)$ and ϵ are Gaussian $\longrightarrow$ Kalman Filter
- When $\mathbf{F}$, $\mathbf{H}$ are nonlinear, the integrals are usually intractable $\longrightarrow$ Approximate solutions
  - Gaussian Approximations: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF).
  - Sequential Monte Carlo Methods or Particle filters [Gordon 1993, Doucet 1998]

Outline

Sequential Monte Carlo Methods or Particle filters

- Map intractable integrals of optimal Bayesian solution to tractable discrete sums of samples drawn from the posterior distribution.



$$\mathbf{x}_k^{(i)} \overset{i.i.d}{\sim} p(\mathbf{x}_k|\mathbf{y}_{1:k}) \qquad \hat{p}(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})$$

- So, any expectation of the form

$$E[g(\mathbf{x}_k)] = \int g(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k$$

may be approximated by:

$$E[g(\mathbf{x}_k)] \approx \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}_k^{(i)})$$

Importance sampling

- Often impossible to sample directly from true posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.
- Use importance sampling:

$$\int g(\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:K})d\mathbf{x}_k = \int g(\mathbf{x}_k)\frac{p(\mathbf{x}_k|\mathbf{y}_{1:k})}{q(\mathbf{x}_k|\mathbf{y}_{1:k})}q(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k \approx \sum_i w_k^{(i)} g(\mathbf{x}_k^{(i)})$$

where $\mathbf{x}_k^{(i)}$ is sampled from a proposal distribution $q(\mathbf{x}_k|\mathbf{y}_{1:k})$:

$$\mathbf{x}_k^{(i)} \overset{i.i.d}{\sim} q(\mathbf{x}_k|\mathbf{y}_{1:k})$$

and the importance weights given by:

$$w_k^{(i)} = \frac{p(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})}{q(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})}$$

Sequential Importance Sampling with Resampling

- Recursive implementation of sequential importance sampling.

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k})}$$

- Most popular choice of proposal distribution is the transition prior:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

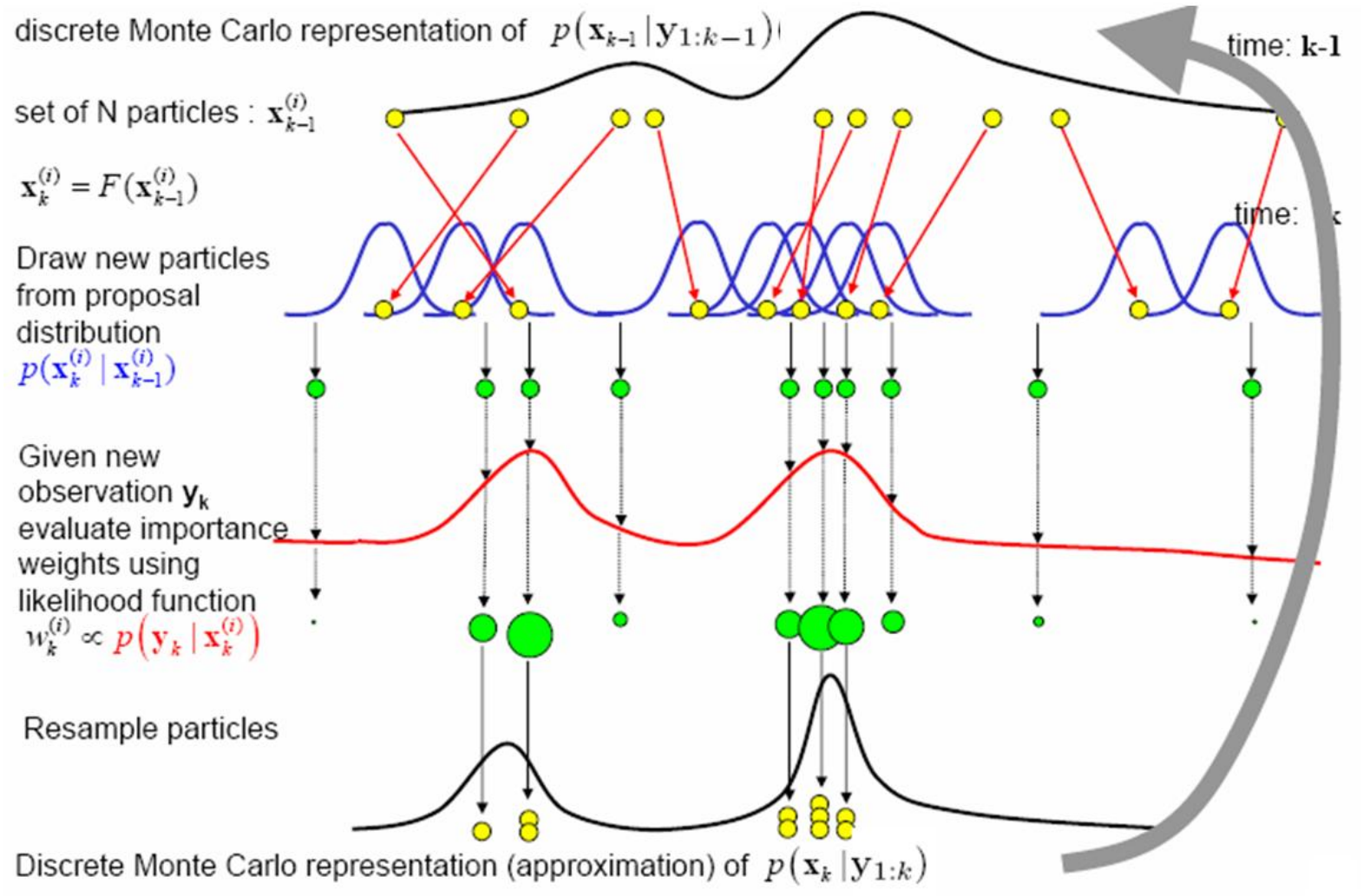
- So,

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{x}_k^{(i)})$$

- Particles degenerate over time (only few particles have significant weights)
- Sampling-importance Resampling (SIR): keep / multiply particles with high importance weights and map N unequally weighted particles into a new set of N equally weighted samples.

$$\{\mathbf{x}_k^{(i)}, w_k^{(i)}\} \longrightarrow \{\mathbf{x}_k^{(i)}, \frac{1}{N}\}$$

Sampling-importance Resampling: Bootstrap Filter [Gordon 1993]



[van der Merwe 01]

Theoretical Convergence

Theorem

If the importance weight

$$w_k \propto \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{y}_{1:k})}$$

is upper bounded for any $(\mathbf{x}_{k-1},\mathbf{y}_{1:k})$ *then, for all* $k \geq 0$*, there exists* c_k *independent of* N *such that for any* f_k

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} f_k(\mathbf{x}_{1:k}^{(i)}) - \int f_k(\mathbf{x}_{1:k})p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k})\right)^2\right] \leq c_k \frac{\|f_k\|^2}{N}$$

Problem with Bootstrap Filter

- Deterministic parameter and state evolution:
  $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k \qquad \mathbf{x}(t_{k+1}) = \mathbf{F}(\mathbf{x}(t_k), \mathbf{u}; \boldsymbol{\theta}_k) \rightarrow p(\mathbf{x}_{k+1}, \boldsymbol{\theta}_{k+1} | \mathbf{x}_k, \boldsymbol{\theta}_k)$ is a Dirac-Delta function.
- Parameter and state particles degenerate quickly after a few time steps

Problem with Bootstrap Filter

- Deterministic parameter and state evolution:
  $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k \qquad \mathbf{x}(t_{k+1}) = \mathbf{F}(\mathbf{x}(t_k), \mathbf{u}; \boldsymbol{\theta}_k) \rightarrow p(\mathbf{x}_{k+1}, \boldsymbol{\theta}_{k+1} | \mathbf{x}_k, \boldsymbol{\theta}_k)$ is a Dirac-Delta function.
- Parameter and state particles degenerate quickly after a few time steps $\rightarrow$ At each time step, a small amount of Gaussian noise is added to each resampled particle, which is equivalent to using a Gaussian kernel to smooth the filtering distribution [Liu and West 2001]
- $p(\boldsymbol{\theta}_k | \mathbf{y}_{1:k}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\boldsymbol{\theta}_k - \boldsymbol{\theta}_k^{(i)})$ with mean $\bar{\boldsymbol{\theta}}_k$ and variance V_k is replaced by:

$$p(\boldsymbol{\theta}_k | \mathbf{y}_{1:k}) \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}(\boldsymbol{\theta}_k | \mathbf{m}_k^{(i)}, h^2 V_k)$$

- Shrinkage of kernel locations to retain the mean $\bar{\boldsymbol{\theta}}_k$ and variance V_k (to prevent the lost of information caused by adding artificial Gaussian noise):

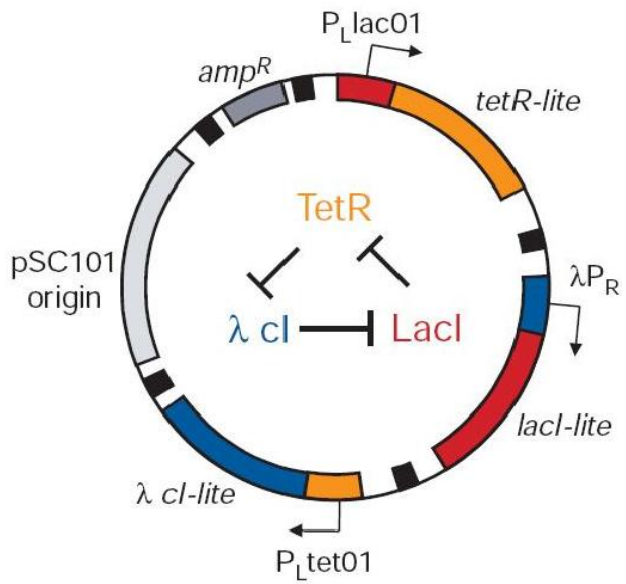
$$\mathbf{m}_k^{(i)} = a\boldsymbol{\theta}_k^{(i)} + (1 - a)\bar{\boldsymbol{\theta}}_k$$

where $a = \sqrt{1 - h^2}$

Outline

Repressilator

[Elowitz, Nature 2000]



$$\frac{dr_1}{dt} = v_1^{max} \frac{k_{12}^n}{k_{12}^n + p_2^n} - k_1^{mRNA} r_1$$

$$\frac{dr_2}{dt} = v_2^{max} \frac{k_{23}^n}{k_{23}^n + p_3^n} - k_2^{mRNA} r_2$$

$$\frac{dr_3}{dt} = v_3^{max} \frac{k_{31}^n}{k_{31}^n + p_1^n} - k_3^{mRNA} r_3$$

$$\frac{dp_1}{dt} = k_1 r_1 - k_1^{protein} p_1$$

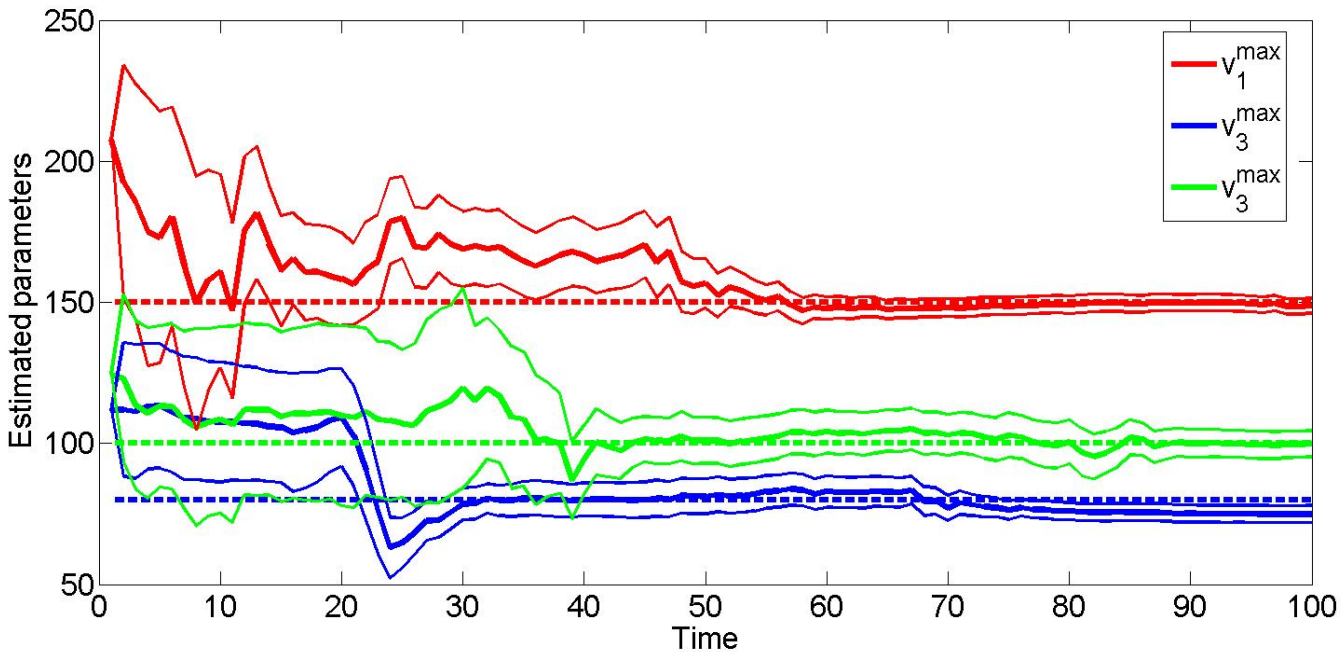
$$\frac{dp_2}{dt} = k_2 r_2 - k_2^{protein} p_2$$

$$\frac{dp_3}{dt} = k_3 r_3 - k_3^{protein} p_3$$

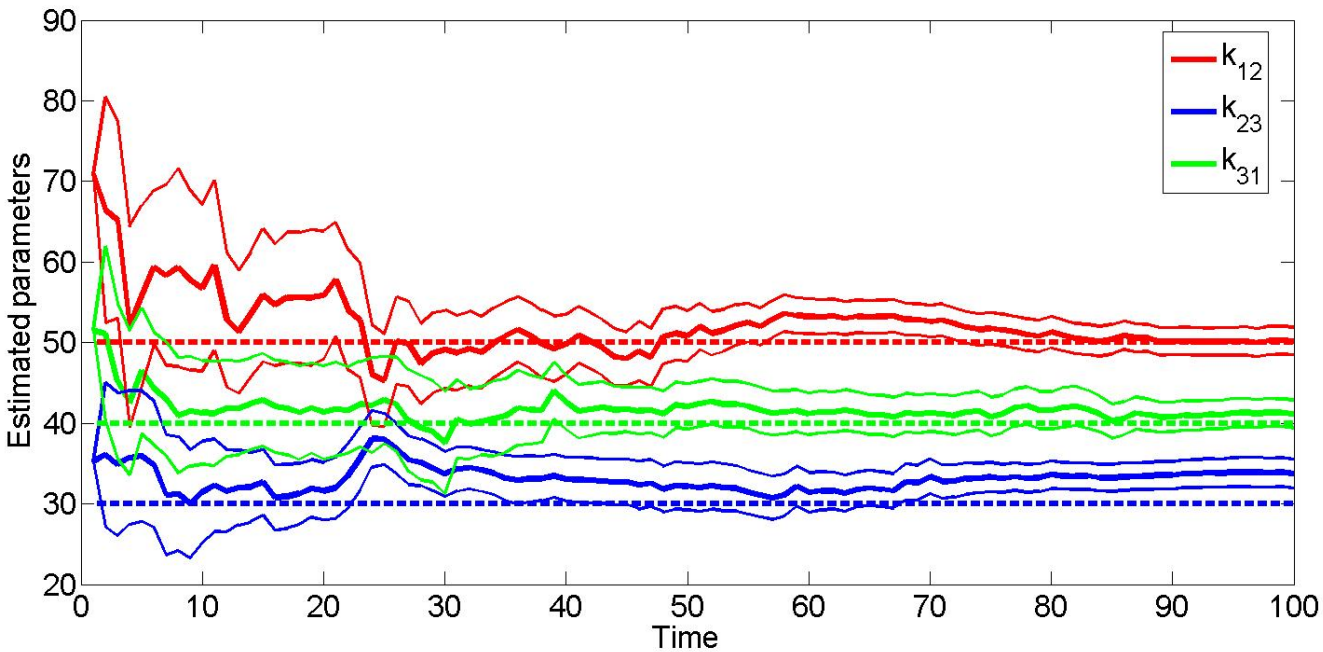
- mRNAs are observed, proteins are hidden
- mRNA and protein degradation rate constants are supposed to be known
- Estimate 9 parameters

Result for the Repressilator

20 % Gaussian noise, Gausian priors on parameters and initial states, 5000 particles.

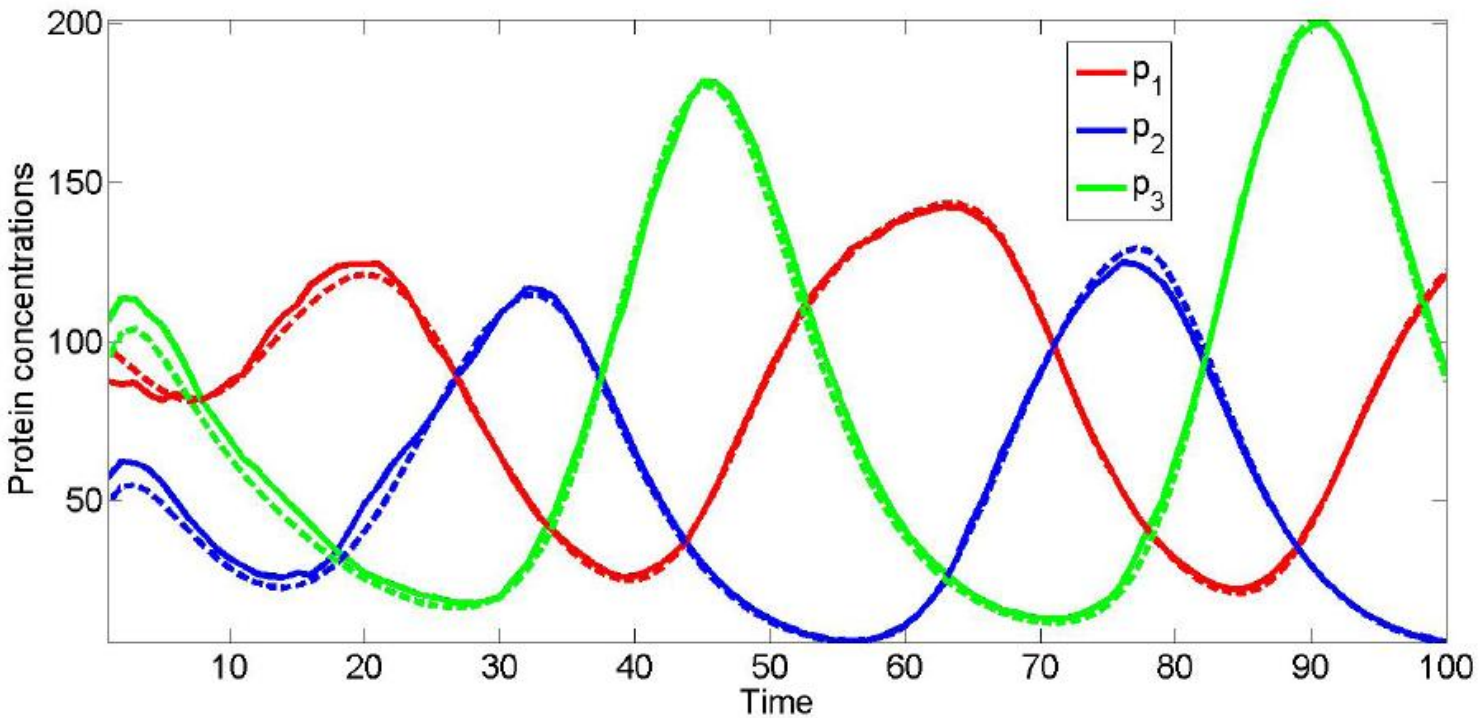


Repressilator



Repressilator

FIGURE 2. Estimation of protein concentrations with PF. The evolution of the true (dashed) and estimated (solid) protein concentrations.



Repressilator

TABLE 1. Root mean squared errors of UKF and PF

	RMSE: mean and variance	Execution time (seconds)
UKF	3.37 ± 0.29	4.25
PF	2.99 ± 0.29	2273.94

Repressilator

TABLE 2. Parameter estimation results

	True value	UKF	PF
v_1^{max}	150	150.06 ± 2.16	149.69 ± 2.65
v_2^{max}	80	79.73 ± 2.35	79.45 ± 2.98
v_3^{max}	100	98.96 ± 3.85	99.95 ± 4.60
k_{12}	50	49.75 ± 1.49	49.90 ± 1.71
k_{23}	30	30.50 ± 1.25	30.42 ± 1.79
k_{31}	40	40.39 ± 1.37	40.05 ± 1.71

Conclusion and Future work

- Choices of better proposal distributions (e.g. EKF, UKF proposals)
- Adaptive resampling schemes
- Larger networks