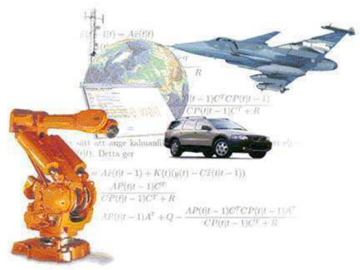
Part 4 - Nonlinear system identification



Thomas Schön

Division of Automatic Control
Linköping University
Sweden

The aim – Part 4

The **aim in part 4** is to show how EM together with SMC and MCMC together with SMC can be used to solve challenging nonlinear system identification problems.

In other words, we will here make use of most of the building blocks introduced throughout the course in order to show how they can be combined to solve nonlinear system identification problems.

Outline

1. Computing ML estimates using EM and PS.
   a) Derive the algorithm
   b) Example - parametric Wiener model
2. Computing Bayesian estimates using particle MCMC (PMCMC)
   a) Particle Metropolis Hastings (PMH)
   b) Example - semiparametric Wiener model

ML problem formulation

Task: Compute the ML estimate of the parameters θ in the SSM

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_\theta(x_{t+1} \mid x_t, u_t), \\ y_t \mid x_t &\sim h_\theta(y_t \mid x_t, u_t), \\ x_1 &\sim \mu_\theta(x_1), \end{aligned}$$

The ML estimate is obtained by solving the following optimisation problem,

$$\widehat{\theta}^{\text{ML}} = \arg\max_{\theta} L_\theta(y_{1:N}),$$

where the log-likelihood function is given by

$$L_\theta(y_{1:N}) = \log p_\theta(y_{1:N}) = \sum_{t=1}^{N} \log p_\theta(y_t \mid y_{1:t-1})$$

EM revisited

The expectation maximisation (EM) algorithm computes ML estimates of unknown parameters in probabilistic models involving latent variables.

Algorithm 1 Expectation Maximization (EM)

1. **Initialise:** Set $i = 1$ and choose an initial θ^1.
2. **While** not converged **do:**
   (a) **Expectation (E) step:** Compute

$$\mathcal{Q}(\theta, \theta^i) = \mathrm{E}_{\theta^i}\left[\log p_\theta(Z, Y) \mid Y\right] = \int \log p_\theta(Z, Y) p_{\theta^i}(Z \mid Y) \mathrm{d}Z$$

   (b) **Maximization (M) step:** Compute $\theta^{i+1} = \arg\max_{\theta \in \Theta} \mathcal{Q}(\theta, \theta^i)$

   (c) $i \leftarrow i + 1$

EM for nonlinear system identification

The **key property** rendering EM an appealing approach for computing maximum likelihood estimates in nonlinear SSM's is that the intermediate quantity $\mathcal{Q}(\theta, \theta^i)$ and its derivatives can be approximated arbitrarily well using particle smoothers.

EM provides a **strategy** for breaking down the problem into two manageable subproblems

1. A nonlinear state smoothing problem
2. A nonlinear optimisation problem

each of which can be handled using readily available algorithms.

Approximation of the $\mathcal{Q}$-function

The intermediate quantity $\mathcal{Q}(\theta, \theta^i)$ is approximated according to (using the particle smoother (FFBSi))

$$\widehat{\mathcal{Q}}(\theta, \theta^i) = \widehat{I}_1(\theta, \theta^i) + \widehat{I}_2(\theta, \theta^i) + \widehat{I}_3(\theta, \theta^i),$$

where

$$\widehat{I}_1(\theta, \theta^i) = \frac{1}{N}\sum_{i=1}^{N} \log \mu_\theta(x_1^i),$$

$$\widehat{I}_2(\theta, \theta^i) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T-1} \log f_\theta(x_{t+1}^i \mid x_t^i),$$

$$\widehat{I}_3(\theta, \theta^i) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \log h_\theta(y_t \mid x_t^i).$$

Maximisation (M) step

Use a numerical nonlinear optimisation algorithm, e.g., BFGS. The gradient is computed according to

$$\nabla_\theta \mathcal{Q}(\theta, \theta^i) = \nabla_\theta I_1(\theta, \theta^i) + \nabla_\theta I_2(\theta, \theta^i) + \nabla_\theta I_3(\theta, \theta^i),$$

and based on $\widehat{\mathcal{Q}}(\theta, \theta^i)$ it is straightforward to approximate these gradients according to,

$$\nabla_\theta I_1(\theta, \theta^i) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log \mu_\theta(x_1^i),$$

$$\nabla_\theta I_2(\theta, \theta^i) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T-1} \nabla_\theta \log f_\theta(x_{t+1}^i \mid x_t^i),$$

$$\nabla_\theta I_3(\theta, \theta^i) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \nabla_\theta \log h_\theta(y_t \mid x_t^i).$$

EM for nonlinear system identification

Algorithm 2 EM for nonlinear system identification

1. **Initialise:** Set $i = 1$ and choose an initial θ^1.
2. **While** not converged **do:**
   (a) **Expectation (E) step:** Run a PF and a FFBSi PS and compute

   $$\widehat{\mathcal{Q}}(\theta, \theta^i) = \widehat{I}_1(\theta, \theta^i) + \widehat{I}_2(\theta, \theta^i) + \widehat{I}_3(\theta, \theta^i)$$

   (b) **Maximization (M) step:** Compute $\theta^{i+1} = \underset{\theta \in \Theta}{\arg\max}\ \mathcal{Q}(\theta, \theta^i)$ using an off-the-shelf numerical optimisation algorithm.

   (c) $i \leftarrow i + 1$

A Specific Example – The Wiener Problem

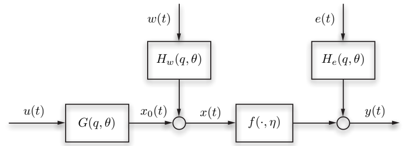


Figure: The general Wiener problem.

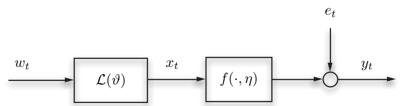


Figure: The blind Wiener problem.

Typical restrictions imposed are:

1. The nonlinearity f in invertible.
2. The measurement noise e is absent.
3. $H_w(q, \theta) = 1$ (white process noise) or $H_w(q, \theta) = 0$ (no process noise).

Using EM + PS we do not have to impose any of these assumptions.

Example – 2^{th} order system + 2 non-invertible NL

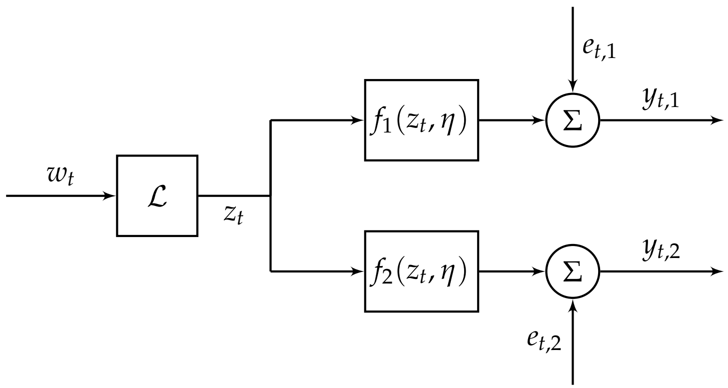


Figure: Block diagram of a blind Wiener model with two outputs.

Example – 2^{th} order system + 2 non-invertible NL

The linear system ($\mathcal{L}$) is given by

$$x_{t+1} = \begin{pmatrix} 1 & -0.9 \\ 1 & 0 \end{pmatrix} x_t + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u_t,$$
$$z_t = \begin{pmatrix} 1 & 0.3 \end{pmatrix} x_t.$$

Complex poles implies a resonant system. The nonlinearities are a saturation and a dead zone, respectively,

$$f_1(z_t, \eta) = \begin{cases} \eta_1 & : z_t < \eta_1 \\ z_t & : \eta_1 \le z_t \le \eta_2 \\ \eta_2 & : z_t > \eta_2 \end{cases} \quad f_2(z_t, \eta) = \begin{cases} z_t - \eta_3 & : z_t < \eta_3 \\ 0 & : \eta_3 \le z_t \le \eta_4 \\ z_t - \eta_4 & : z_t < \eta_4 \end{cases}$$

Example – 2^{th} order system + 2 non-invertible NL

The measurements are given by

$$y_t = \begin{pmatrix} y_{1,t} \\ y_{2,t} \end{pmatrix} = \begin{pmatrix} f_1(z_t, \eta) \\ f_2(z_t, \eta) \end{pmatrix} + e_t, \qquad e_t \sim \mathcal{N}\left(0, \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix}\right),$$

The task is to learn this model based on $T = 1000$ measurements of the output (“blind” case), $y_{1:1000}$.

The input is chosen as $u_t \sim \mathcal{N}(0,1)$. Initial values for the measurement variance are $\widehat{r}_1 = \widehat{r}_2 = 0.1$. The initial values for $\widehat{\eta}$ were chosen as $\widehat{\eta}_i = \frac{\eta_i^\star}{10}$, to reflect that they are unknown. The LGSS model is initialised via a subspace algorithm based on the measurements $\{y_{1,1}, \cdots, y_{1,T}\}$ from the dead zone nonlinearity.

Employ the EM alg. with $N = 100$ particles. The algorithm was terminated after just 100 iterations. Plots below are based on 100 realisations of data.

Example – 2^{th} order system + 2 non-invertible NL

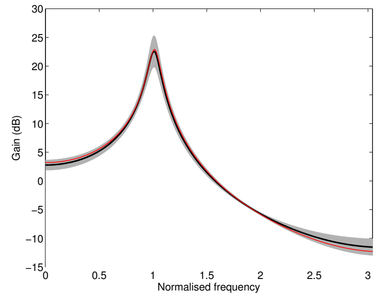


Figure: Bode plot of estimated mean (thick black), 95% confidence intervals (gray) and the true system (red).

Example – 2^{th} order system + 2 non-invertible NL

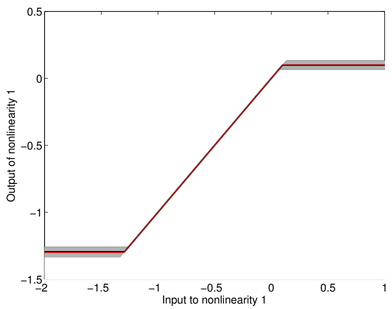


Figure: Estimated mean (thick black), 95% confidence intervals (gray) and the true static nonlinearities (red).

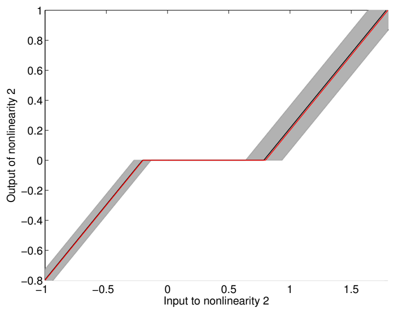


Figure: Estimated mean (thick black), 95% confidence intervals (gray) and the true static nonlinearities (red).

Particle Markov chain Monte Carlo (PMCMC)

PMCMC background and aim

PMCMC is a family of methods providing powerful solutions for joint state and parameter inference in SSM's.

The **aim** in PMCMC is to compute $p(\theta, x_{1:T} \mid y_{1:T})$ when the model is given by

$$x_{t+1} \mid x_t \sim f_\theta(x_{t+1} \mid x_t),$$
$$y_t \mid x_t \sim h_\theta(y_t \mid x_t).$$

The **fundamental idea** is to make use of a sequential Monte Carlo (SMC) sampler to construct a proposal for an MCMC sampler.

We will focus on the **particle Metropolis Hastings (PMH)** sampler in this course.

SMC again

Algorithm 3 Sequential Monte Carlo (SMC)

1. **Initialise:** Sample $x_1^i \sim Q_1(x_1)$ and set $w_1^i = W_1(x_1^i)$. Set $t = 1$.
2. **For** $t = 2 : T$ **do:**
   (a) Sample from the proposal kernel

$$M_t(i_t, x_t) = \frac{w_{t-1}^{i_t}}{\sum_{l=1}^N w_{t-1}^l} R_t(x_t \mid x_{t-1}^{i_t})$$

   (b) Weighting: $w_{t+1}^m = W_t(x_{t+1}^m, \tilde{x}_{t-1}^{i_t^m})$.

Note that this is exactly the same algorithm as before, but we have merged the resampling step and the proposal step into on step and introduced i_t^m to denote the **index** of the ancestor of particle $x_{t-1}^{i_t^m}$.

PB-BSi

We can also derive a Particle Gibbs (PG) sampler with backward simulation (PG-BSi). This is in fact what we used for the blind Wiener example mentioned in the introduction of Part 1.

References for PMCMC

General references for PMCMC:

- Christophe Andrieu, Arnaud Doucet and Roman Holenstein. **Particle Markov chain Monte Carlo methods**. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269-342, June 2010.
- Fredrik Lindsten and Thomas B. Schön. **On the use of backward simulation in the particle Gibbs sampler**. *Proceedings of the 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.

Using PMCMC for nonlinear system identification:

- Fredrik Lindsten, Thomas B. Schön and Michael I. Jordan, **A semiparametric Bayesian approach to Wiener system identification**. *Proceedings of the 16th IFAC Symposium on System Identification (SYSID)*, Brussels, Belgium, July, 2012.

The **aim of this course** has been to provide an introduction to the theory and application of (new) computational methods for inference in dynamical systems.

The **key computational methods** we refere to are,

- Sequential Monte Carlo (SMC) methods (particle filters and particle smoothers) for nonlinear state inference problems.
- Expectation maximisation (EM) and Markov chain Monte Carlo (MCMC) methods for nonlinear system identification.

Much interesting research remains to be done in solving nonlinear estimation problems using SMC and/or MCMC methods!!

We are organising an **invited session** for SYSID on this topic and many of the leading researchers in the area have accepted the invitation. Drop by if you are interested!

Thank you for listening!!