Sequential Monte Carlo Methods and Their Applications: An Overview and Recent Developments

Rong Chen

Department of Information and Decision Sciences

University of Illinois at Chicago

and

Department of Business Statistics and Econometrics

Peking University

Present to

Institute for Mathematical Sciences

National University of Singapore

March 16-17, 2004

Outline

- **SDS: A sequence of random distributions**

$$\pi_1(\boldsymbol{x}_1), \ldots, \pi_t(\boldsymbol{x}_t), \ldots, \pi_n(\boldsymbol{x}_n)$$

- **State variable** $\boldsymbol{x}_t = (\boldsymbol{x}_{t-1}, x_t)$
- **Slow moving:**

$$\int \pi_t(\boldsymbol{x}_{t-1}, x_t) dx_t \approx \pi_{t-1}(\boldsymbol{x}_{t-1})$$

- **Monte Carlo method: Generate samples** $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(m)}$ **from the target distribution** $\pi_t(\boldsymbol{x}_t)$**.**

1.1 State Space Models – General Form

state equation: $x_t = s_t(x_{t-1}, \varepsilon_t)$ **or** $x_t \sim q_t(\cdot \mid x_{t-1})$

observation equation: $y_t = h_t(x_t, e_t)$ **or** $y_t \sim f_t(\cdot \mid x_t)$

$$\begin{array}{ccccccc} \cdots & & y_t & & y_{t+1} & & \cdots \\ & & \uparrow & & \uparrow & & \\ \cdots & \longrightarrow & x_t & \rightarrow & x_{t+1} & \longrightarrow & \cdots \end{array}$$

$$p(x_1, \ldots, x_t \mid y_1, \ldots, y_t) \propto \prod_{s=1}^{t} f_s(y_s \mid x_s) q_s(x_s \mid x_{s-1})$$

Objective: On-Line in Real Time:

(1) Filtering: $p(x_t \mid y_1, \ldots y_t)$ **(concurrent estimation)**

(2) Prediction: $p(x_{t+1} \mid y_1, \ldots, y_t)$

(3) Smoothing: $p(x_1, \ldots, x_{t-1} \mid y_1, \ldots, y_t)$

(3.1) delayed estimation: $p(x_{t-d} \mid y_1, \ldots, y_t)$

1.1 State Space Models – Linear and Gaussian

Linear and Gaussian Systems:

$$\begin{aligned} x_t &= H_t x_{t-1} + W_t w_t \\ y_t &= G_t x_t + V_t v_t \end{aligned}$$

where $w_t \sim N(0, I)$ **and** $v_t \sim N(0, I)$**.**

$$p(x_t \mid y_1, \ldots y_t) \sim N(\mu_t, \Sigma_t)$$

Kalman Filter:

Recursive updating:

$$(\mu_t, \Sigma_t) \rightarrow (\mu_{t+1}, \Sigma_{t+1})$$

Very easy and fast!

1.1 State Space Models – Nonlinear and NonGaussian

Nonlinear and Non-Gaussian:

Easy to obtain: $p(x_1, \ldots, x_t \mid \boldsymbol{y}_t)$

Difficult:

$$E(x_t \mid \boldsymbol{y}_t) = \int \cdots \int x_t p(x_1, \ldots, x_t \mid \boldsymbol{y}_t) dx_1 \ldots dx_t$$

where $\boldsymbol{y}_t = (y_1, \ldots, y_t)$**.**

Our approach: Monte Carlo method

Generate samples $x_t^{(1)}, \ldots, x_t^{(m)}$ **from the target distribution** $p(x_t \mid \boldsymbol{y}_t)$**, then use approximation**

$$E[x_t \mid \boldsymbol{y}_t] \approx \frac{\sum_{i=1}^{m} x_t^{(i)}}{m}$$

1.2 The Growth Principle

The Growth Principle: [Rosenbluth & Rosenbluth, 1955]

Decompose a complex problem into a sequence of simpler problems.

- **Target distribution $\pi(\boldsymbol{x})$ where $\boldsymbol{x} = (x_1, \ldots, x_N)$**
- **Let $\boldsymbol{x}_t = (x_1, \ldots, x_t)$**
- **Define a sequence of intermediate distributions $\pi_t(\boldsymbol{x}_t)$.**
- **Moving from $\pi_{t-1}(\boldsymbol{x}_{t-1})$ to $\pi_t(\boldsymbol{x}_t)$ is simple.**
- $\pi_N(\boldsymbol{x}_N) = \pi(\boldsymbol{x})$

1.3 Examples: (I) Target Tracking

Tracking a target in clutter

- **A single target moving on a straight line**
- **with random (Gaussian) acceleration. Constant within a period $a_t = w_t/T$.**
- **in a clutter environment**
- **State vector: $x_t = (d_t, v_t)$.**
- **State Equation:**

$$\begin{aligned} d_t &= d_{t-1} + v_{t-1}T + w_t T/2 \\ v_t &= v_{t-1} + w_t \end{aligned}$$

- **True signal:**

$$z_t = d_t + e_t$$

where $w_t \sim N(0, q^2)$ and $e_t \sim N(0, r^2)$ and independent.

1.3 Examples: (I) Target Tracking

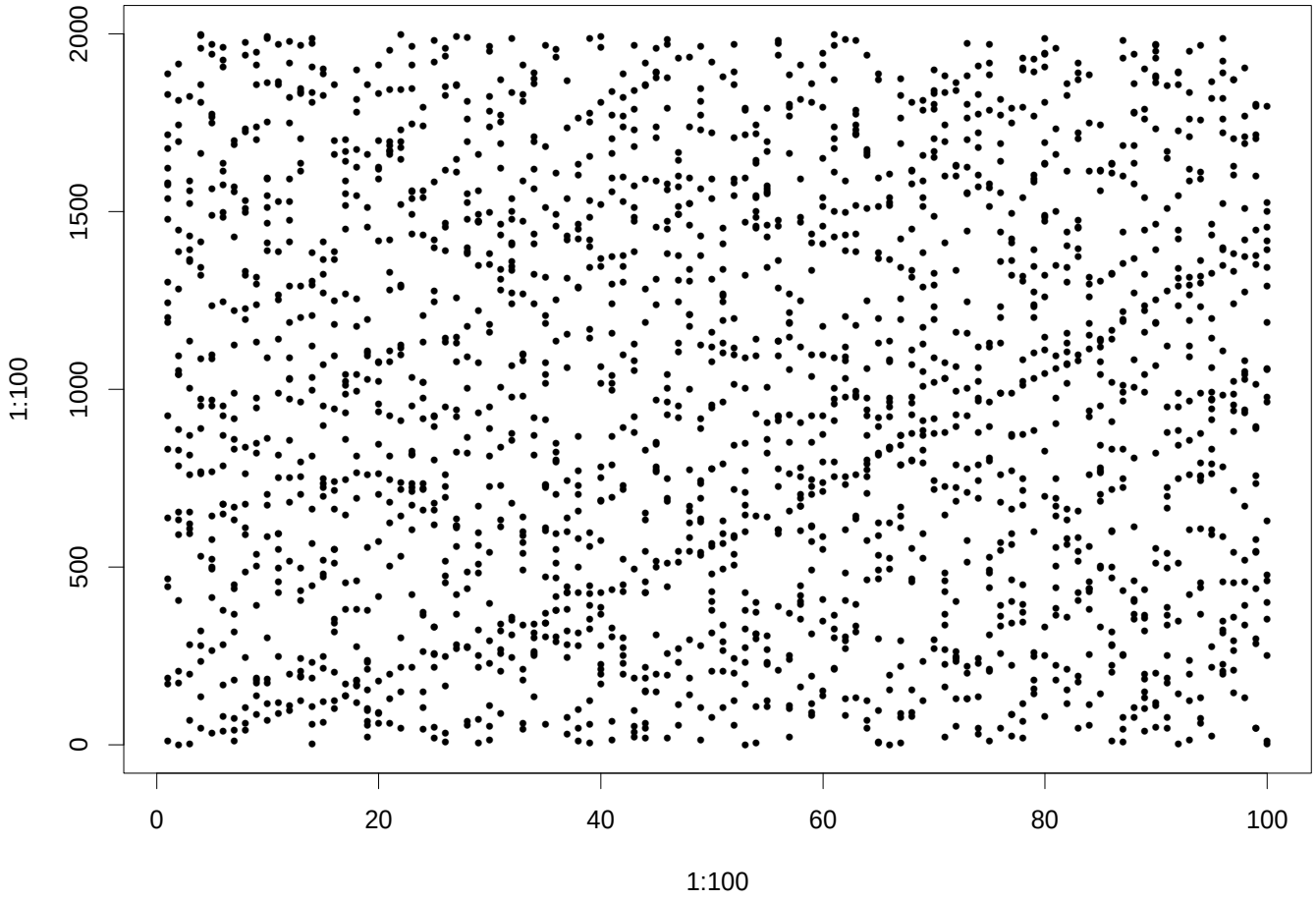
- **At time t, observe m_t signals, where**

$$m_t \sim \textbf{Bernoulli}(p_d) + \textbf{Poisson}(\lambda\Delta)$$

- **The true signal z_t has probability p_d to be observed.**
- **The false signals are uniformly distributed in the detection region Δ.**

$T = 1$, $p_d = 0.9$, $\lambda = 0.1$, $Var(w_t) = 0.1$, $Var(e_t) = 1$

1.3 Example: (I) Target Tracking



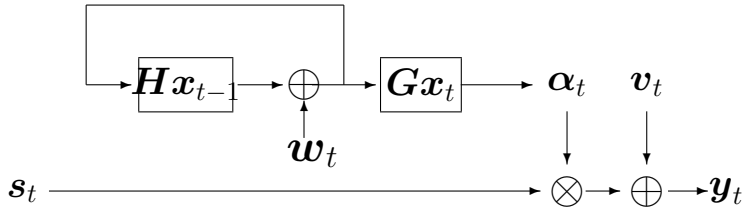
1.3 Examples: (I) Target Tracking

Other tracking examples:

- **Tracking maneuvering targets**
- **Tracking multiple targets**
- **Tracking and discriminating multiple targets**
- **Tracking targets with non-Gaussian innovation and noises**
- **2-d tracking (tanks, cars) with radar**
- **2-d tracking (cars, cellular phones) in a cellular network, w/o map**
- **2-d tracking with GPS**
- **2-d/3-d tracking with passive sonar**
- **Computer vision: tracking with a sequence of images**

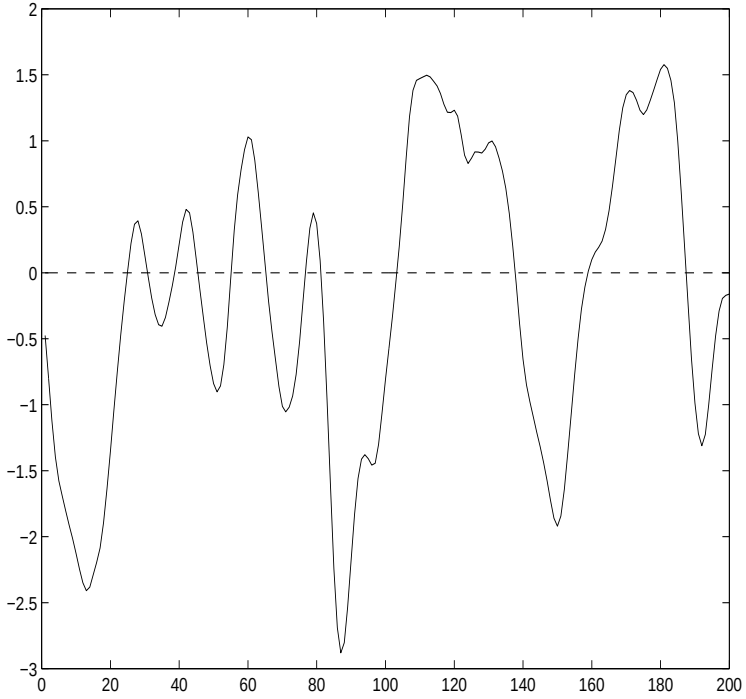
1.3 Examples: (II) Fading Channels

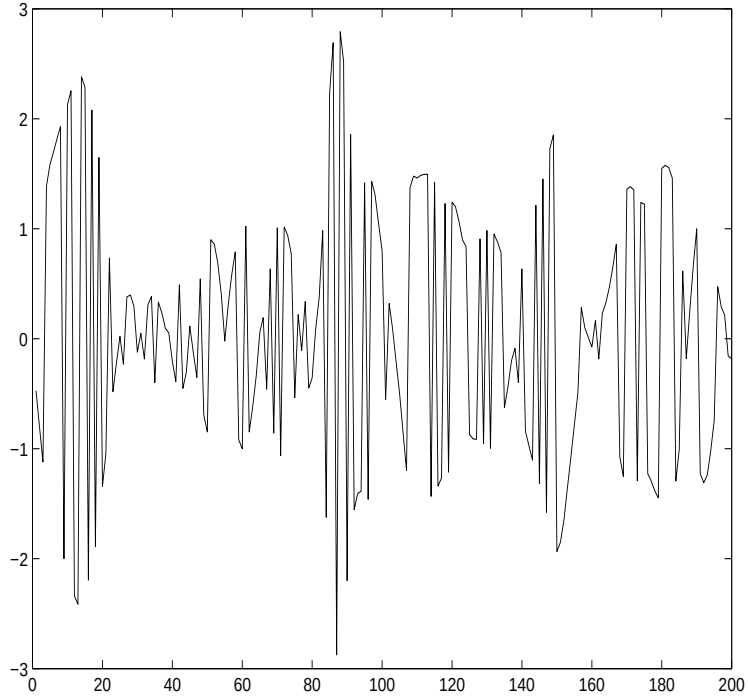
Digital Signal Extraction in Fading Channels



- **State Equations:**
$$\begin{cases} x_t &= Hx_{t-1} + w_t \\ \alpha_t &= Gx_t \\ s_t &\sim p(\cdot \mid s_{t-1}) \end{cases}$$
- **Observation equation:** $y_t = \alpha_t s_t + v_t$
- $\alpha_t = Gx_t$**: Butterworth filter of order** $r = 3$ **i.e. ARMA(3,3)**
  Cutoff frequency 0.1
- **Noise:** **(1)** $v_t \sim N(0, \sigma^2)$ **(2)** $v_t \sim (1-\alpha)N(0, \sigma_1^2) + \alpha N(0, \sigma_2^2)$

1.3 Examples: (II) Fading Channels





1.3 Examples: (II) Fading Channels

Phase Ambiguity: $p(\boldsymbol{\alpha}_t, \boldsymbol{s}_t \mid \boldsymbol{y}_t) = p(-\boldsymbol{\alpha}_t, -\boldsymbol{s}_t \mid \boldsymbol{y}_t)$

Differential coding:

Information sequence: $s_1, \ldots, s_t$**.**

Transmitted sequence: $s_1^*, \ldots, s_t^*$**, s.t** $s_{t-1}^* s_t^* = s_t$**,** $s_1^* = s_1$**.**

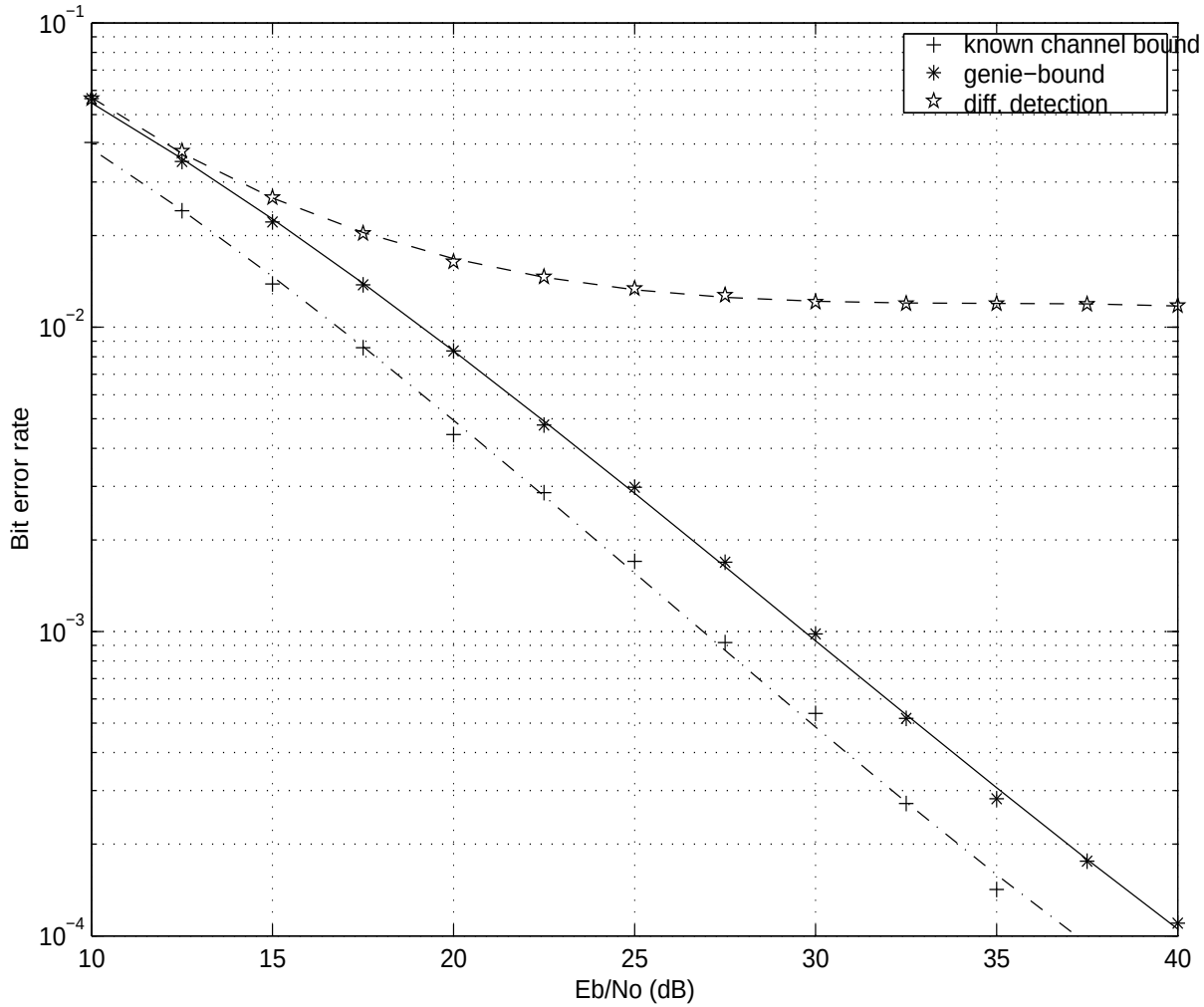
Differential detector:

$$\hat{s}_t = sign(y_t y_{t-1}) = sign(\alpha_t \alpha_{t-1} s_t + \alpha_t s_t^* e_{t-1} + \alpha_{t-1} s_{t-1}^* e_t + e_{t-1} e_t)$$

Assumption: α_t **changing slowly.**

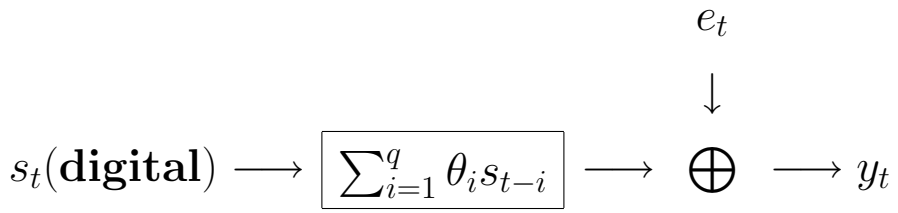
Error floor: the frequency that α_t **changes the sign.**

1.3 Examples: (II) Fading Channels



1.3 Examples: (III) Blind Equalization

Blind Equalization



Objective:

On-line estimation of s_t using the observed output y_t without knowing the system coefficients θ_i.

1.3 Examples: (III) Blind Equalization

$\boldsymbol{\theta}_t = (\theta_{t1}, \ldots, \theta_{tq})$ **and** $x_t = (s_t, s_{t-1}, \ldots, s_{t-q})'$

State Equation:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$$

$$x_t = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 1 \\ 0 & 0 & \cdots & 0 \end{pmatrix} x_{t-1} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} s_t$$

Observation equation:

$$y_t = \boldsymbol{\theta}_t x_t + \varepsilon_t$$

1.3 Examples: (IV) Switching AR Models

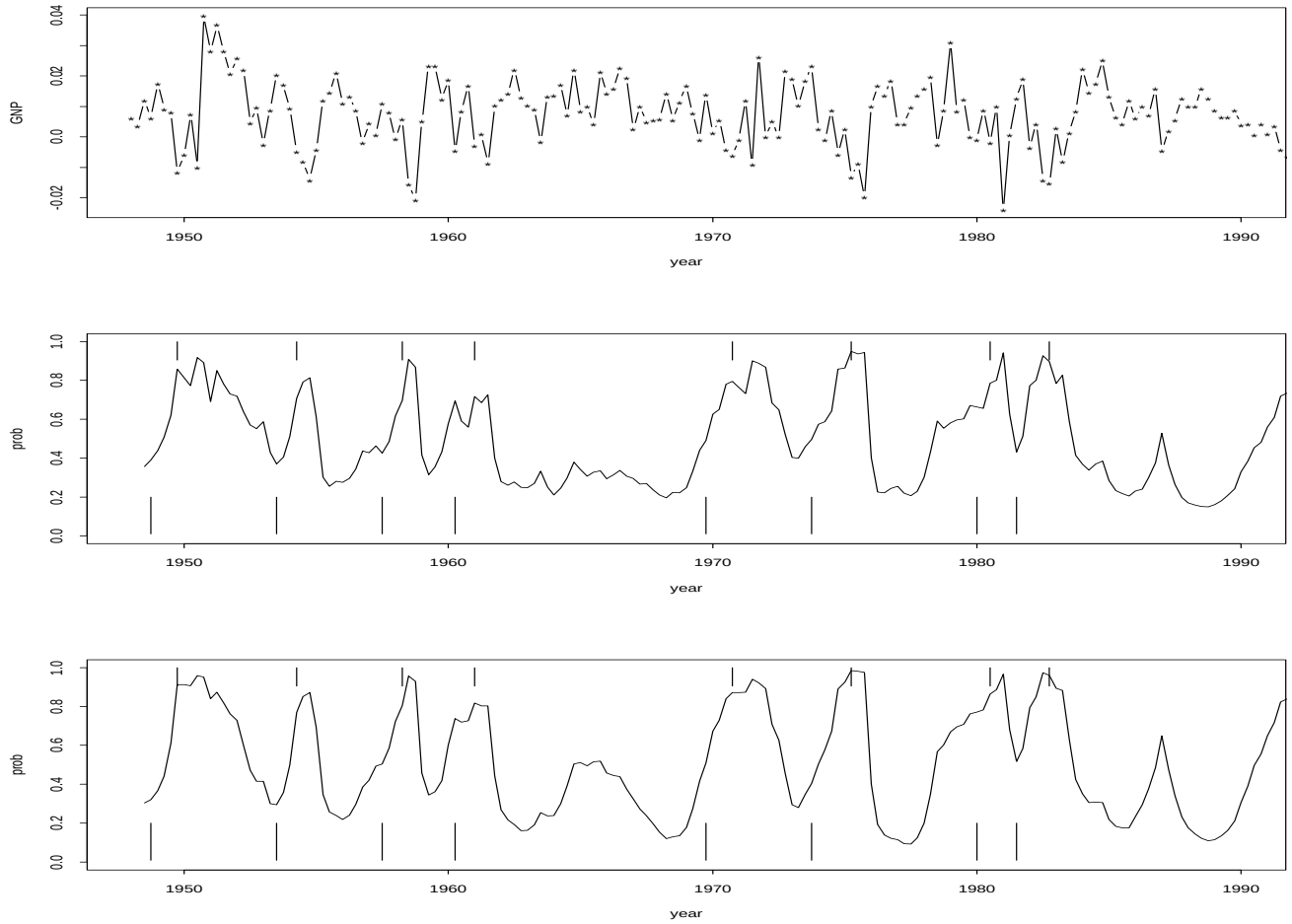
Switching Autoregressive Models:
(Hamilton 1989, McCulloch & Tsay 1992, Chen and Liu 1996)

Economic Status: 'Contraction' or 'Expansion'
Economic time series (Quarterly GNP): different AR model for different status.

An indicator I_t following a Markov chain.

$$\begin{cases} Y_t = \phi_1^{(1)} Y_{t-1} + \ldots + \phi_p^{(1)} Y_{t-p} + e_t & \textbf{if} \quad I_t = 1 \\ Y_t = \phi_1^{(2)} Y_{t-1} + \ldots + \phi_p^{(2)} Y_{t-p} + e_t & \textbf{if} \quad I_t = 2 \end{cases}$$

1.3 Examples: (IV) Switching AR Models



1.3 Examples: (V) Stochastic Volatility Models

Observe stock return: Y_t. Assume zero mean but non-constant variance.

State Equation: $\alpha_t = \phi\alpha_{t-1} + \eta_t$

Observation Equation: $Y_t \sim N(0, \beta \exp(\alpha_t))$

where $\eta_t \sim N(0, \sigma^2)$**, and** $\beta > 0$**.**

Or

State Equation: $\alpha_t = \phi\alpha_{t-1} + \eta_t$

Observation Equation: $log(Y_t^2) = \beta^* + \alpha_t + v_t$

where $v_t = log(e_t^2)$**,** $e_t \sim N(0, 1)$**.**

1.3 Examples: (VI) Self-Avoiding Walks on Lattice

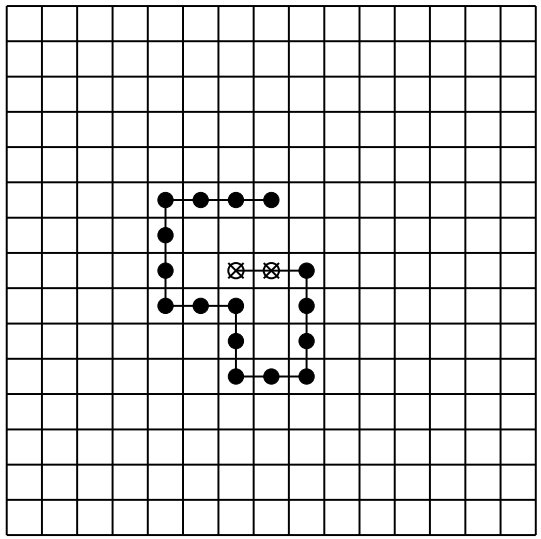
- **A simple discrete model.**

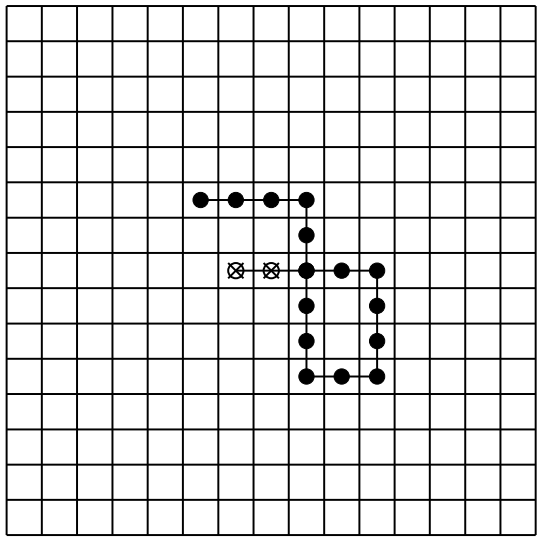
$$\boldsymbol{x}_N = (x_1 x_2 \dots x_N) \quad \textbf{where} \quad x_t = (i_t, j_t).$$

where $||x_t - x_{t-1}|| = 1$ **and** $x_t \neq x_s$ **for all** t, s**.**
Assume $x_1 = (0, 0)$ **and** $x_2 = (0, 1)$**.**

- **Captures basic properties of protein structure:**
  - **chain connectivity.**
  - **excluded volume.**
- **Easier for sampling.**
- **Test basic principles of protein folding.**

1.3 Examples: (VI) Self-Avoiding Walks on Lattice





,

1.3 Examples: (VI) Self-Avoiding Walks on Lattice

n	$\omega(n)$	$\omega_0(n)$	$\omega_1(n)$	$\omega_2(n)$	$\omega_3(n)$	$\omega_4(n)$
11	**5,513**	**5,393**	**120**	**0**	**0**	**0**
12	**15,037**	**14,508**	**529**	**0**	**0**	**0**
13	**40,617**	**39,078**	**1,536**	**3**	**0**	**0**
14	**110,188**	**104,566**	**5,602**	**20**	**0**	**0**
15	**296,806**	**280,599**	**16,088**	**119**	**0**	**0**
16	**802,075**	**748,335**	**53,149**	**591**	**0**	**0**
17	**2,155,667**	**2,002,262**	**151,052**	**2,353**	**0**	**0**
18	**5,808,335**	**5,327,888**	**470,386**	**10,051**	**10**	**0**
19	**15,582,342**	**14,222,389**	**1,325,590**	**34,287**	**76**	**0**
20	**41,889,578**	**37,784,447**	**3,973,361**	**131,298**	**472**	**0**
21	**112,212,146**	**100,673,771**	**11,119,456**	**416,239**	**2,680**	**0**
22	**301,100,754**	**267,136,710**	**32,479,871**	**1471,874**	**12,293**	**6**
23	**805,570,061**	**710,673,806**	**90,361,878**	**4479,355**	**54,998**	**24**
24	**2,158,326,727**	**1,883,960,171**	**259,195,774**	**14946,910**	**223,458**	**414**
25	**5,768,299,665**	**5,005,591,512**	**717,505,892**	**44337,381**	**862,748**	**2,132**

1.3 Examples: (VI) Self-Avoiding Walks on Lattice

Long Chains:

- **More interesting but impossible to enumerate.**
- **Target distribution:** $\pi_N(\boldsymbol{x}_N) = 1/Z_N$
  i.e. uniform distribution on the set of all possible self-avoiding walk (SAW) of length N.
- **Monte Carlo Approach: generate random samples to make inferences on key parameters.**
  - **Mean squared extension:** $E_{\pi_N}(||x_N - x_1||^2)$
  - **Mean packing density given certain compactness:**

$$E_{\pi_N}(p(\boldsymbol{x}_N) \mid \rho(\boldsymbol{x}_N) = \rho)$$

- **MCMC method: starting with a fully extended chain, engineer a sequence of random folding/unfolding (Markov) moves.**

1.3 Examples: (VI) Self-Avoiding Walks on Lattice

Alternative View – The Growth Principle:

The chain of length n is the result of growing the chain one monomer at a time – a stochastic dynamic system

- **The intermediate distribution: $\pi_t(\boldsymbol{x}_t)$: uniform distributed among all possible SAWs of length t.**
- **Moving from $\pi_t(\boldsymbol{x}_t)$ to $\pi_{t+1}(\boldsymbol{x}_{t+1})$ is simple**

1.3 Examples: Other Applications (partial list)

- **Target tracking** (Gordon et al 1993, Avitzour 1995, Bølviken et al 1997, McGinnity and Irwin, 2001, Irwin et al 2002. Salmond and Gordon, 2001, Arulampalam et al 2002, Orton and Fitzgerald, 2002, Hueet al, 2002, Gustafsson et al 2002)
- **Target Recognition** (Srivastava et al 2001).
- **Blind Equalization** (Liu and Chen 1995)
- **Speech recognition** (Rabiner 1989)
- **Computer vision** (Isard and Blake 1996, 1998, 2001, Torma and Szepesvari, 2001)
- **Mobile Robot Localization** (Dellaert et al 1999, Fox et al 1999, 2001)
- **Freeway traffic vision (for vehicle control)** (Huang et al 1994)
- **DNA sequence analysis** (Churchill 1989)
- **Stochastic volatility model** (Pitt and Shephard 1997, Barndorff-Nielsen and Shephard, 2002)
- **Expert systems** (Spiegelhalter et al 1990, Kong et al 1994, Berzuini et al. 1997)

1.3 Examples: Other Applications (partial list)

- **Switching (Auto)Regression Models** (Kaufmann, 2002)
- **Dynamic Bayesian Networks** (Koller and Lerner, 2001, Murphy and Russell, 2001)
- **On-line Control of Industrial Production** (Marrs, 2001)
- **Combinatorial optimizations** (Wong and Liang 1997)
- **Wireless Communications** (Chen et al 2000, Wang et al 2000)
- **Signal Processing** (Djuric, 2001, Wang et al 2002)
- **Audio Signal Enhancing** (Fong et al, 2002)
- **Data Network Analysis** (Coates and Nowak, 2002)
- **Chain Polymer** (Liang et al 2002, Liu et al 2002, Zhang et al 2003, Zhang et al 2004)
- **Counting 0-1 Tables** (Liu 2001)
- **Neural Networks** (Andriew et al 1999, de Freitas et al. 2000)

2. Sequential Monte Carlo – A Framework (Liu and Chen, 1998)

$$x_t = s_t(x_{t-1}, \varepsilon_t) \qquad y_t = h_t(x_t, e_t)$$
$$x_t \sim q_t(\cdot \mid x_{t-1}) \qquad y_t \sim f_t(\cdot \mid x_t)$$

Let $\boldsymbol{y}_t = (y_1, \ldots, y_t)$.

t	$t+1$
$p(x_t \mid \boldsymbol{y}_t)$	$p(x_{t+1} \mid \boldsymbol{y}_t, y_{t+1})$

$$\begin{array}{ccc} x_t^{(1)} & \longrightarrow & x_{t+1}^{(1)} \\ \vdots & & \vdots \\ x_t^{(m)} & \longrightarrow & x_{t+1}^{(m)} \end{array}$$

Note that

$$p(x_t, x_{t+1} \mid y_1, \ldots, y_t) \propto q_t(x_{t+1} \mid x_t) p(x_t \mid y_1, \ldots, y_t)$$

t	$t + \frac{1}{2}$	$t + 1$
$p(x_t \mid \boldsymbol{y}_t)$	$p(x_{t+1} \mid \boldsymbol{y}_t)$	$p(x_{t+1} \mid \boldsymbol{y}_t, y_{t+1})$
	$\searrow \quad \nearrow$	
	$q_t(\cdot \mid x_t)$	
.....	....	.....
$x_t^{(1)}$	$\longrightarrow \quad x_{t+1}^{(1)}$	$\longrightarrow$ **?**
$\vdots$	$\vdots$	$\vdots$
$x_t^{(m)}$	$\longrightarrow \quad x_{t+1}^{(m)}$	$\longrightarrow$ **?**

Importance Sampling:

Target distribution $\pi(\cdot)$. Available an iid sample $\{x_1, \ldots, x_m\}$ from a trial distribution $g(\cdot)$

$$E_\pi(f(X)) = \int f(x)\pi(x)dx = \int f(x)\frac{\pi(x)}{g(x)}g(x)dx = E_g(f(X)w(X))$$

where $w(x) = \pi(x)/g(x)$.

With an iid sample $\{x_1, \ldots, x_m\}$ from g and a set of weights $w_i \propto \pi(x_i)/g(x_i)$, we have

$$\frac{1}{\sum w_i}\sum_{i=1}^{m} w_i f(x_i) \approx E_\pi(f(x))$$

2. Sequential Monte Carlo – Importance Sampling

Definition: A sample $(x_j, w_j), j = 1, \ldots, m$ is said to be <u>properly weighted</u> with respect to distribution π if for all integrable function h, we have

$$\frac{1}{\sum w_j} \sum_{j=1}^{m} w_j h(x_j) \to E_\pi(h(x))$$

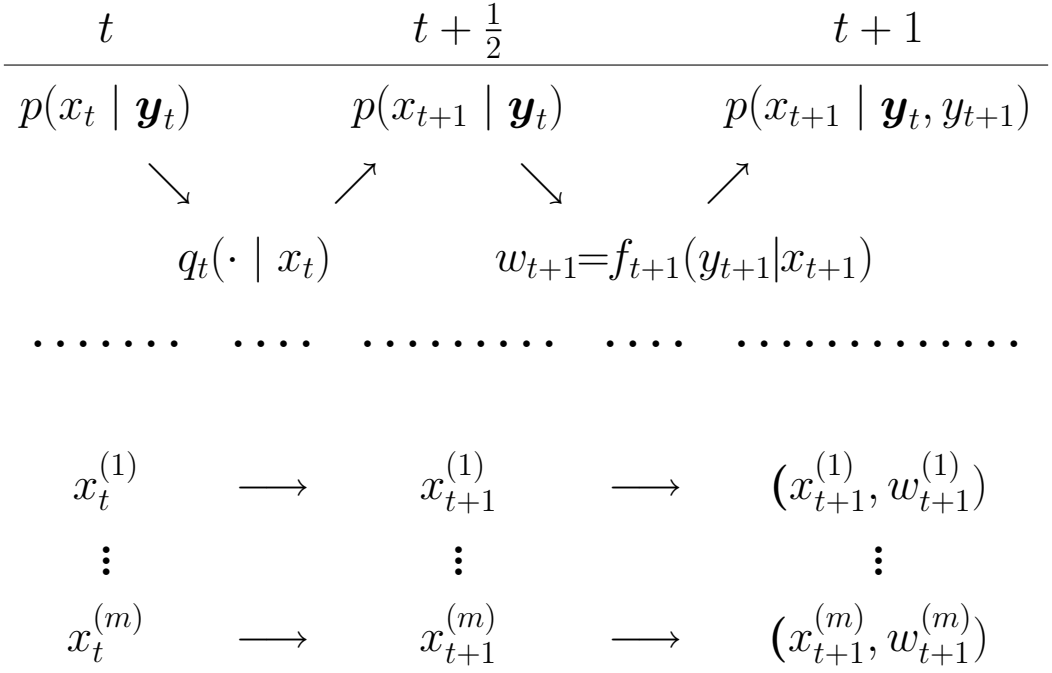
as $m \to \infty$

<u>Efficiency:</u>

$$\textbf{effective sample size} = \frac{m}{1 + cv^2(w)}$$

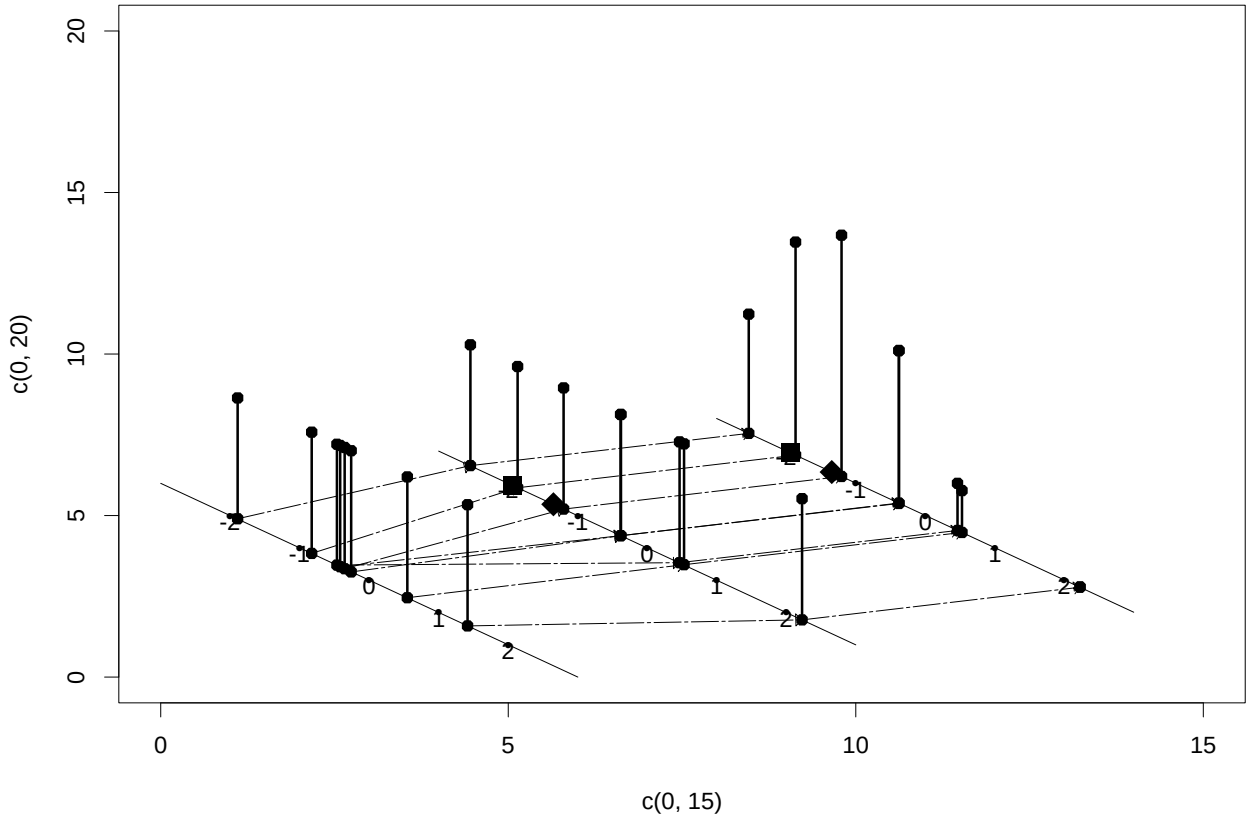
Note that

$$p(x_t, x_{t+1} \mid \boldsymbol{y}_t, y_{t+1}) \propto f_t(y_{t+1} \mid x_{t+1}) p(x_t, x_{t+1} \mid \boldsymbol{y}_t)$$

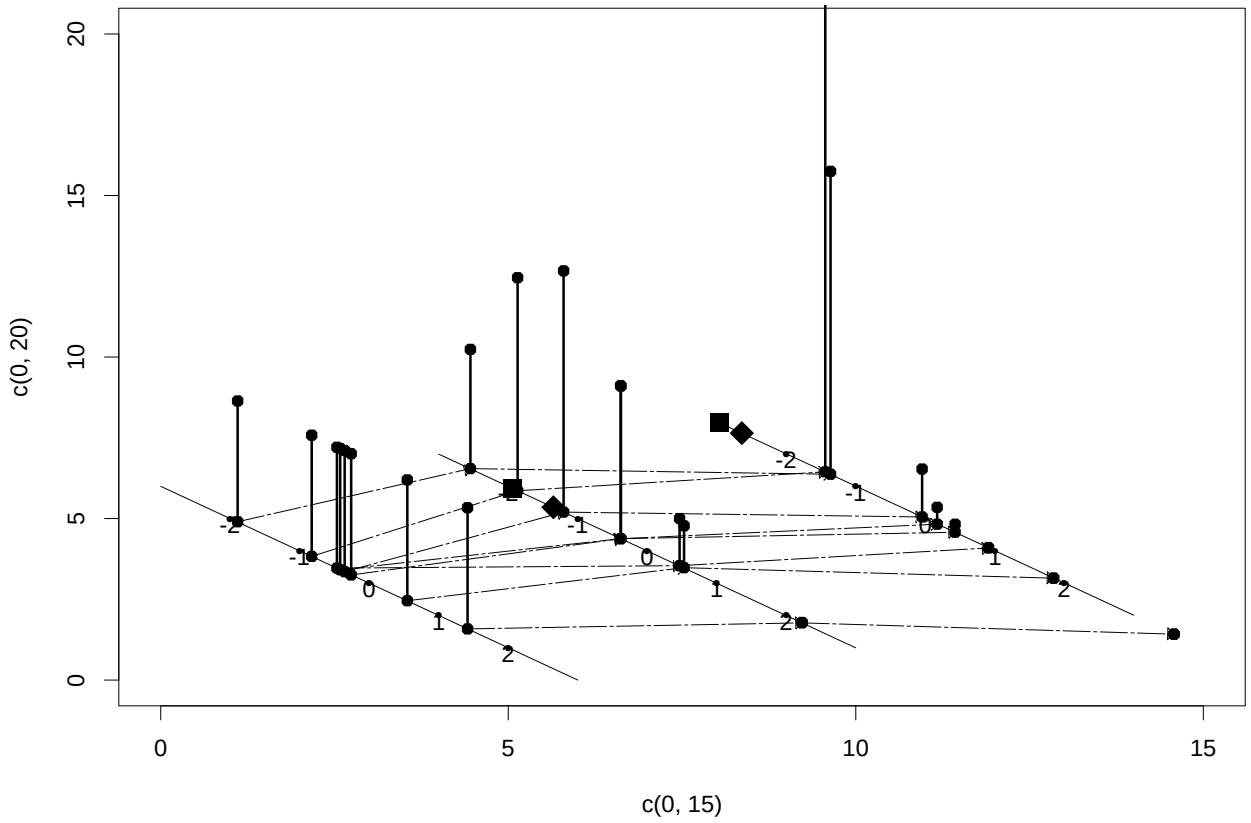


2. Sequential Monte Carlo – An Illustration

$$x_1 = x_0 + N(0,1), \quad y_1 = x_1 + N(0,1) \quad x_0 \sim N(0,1)$$

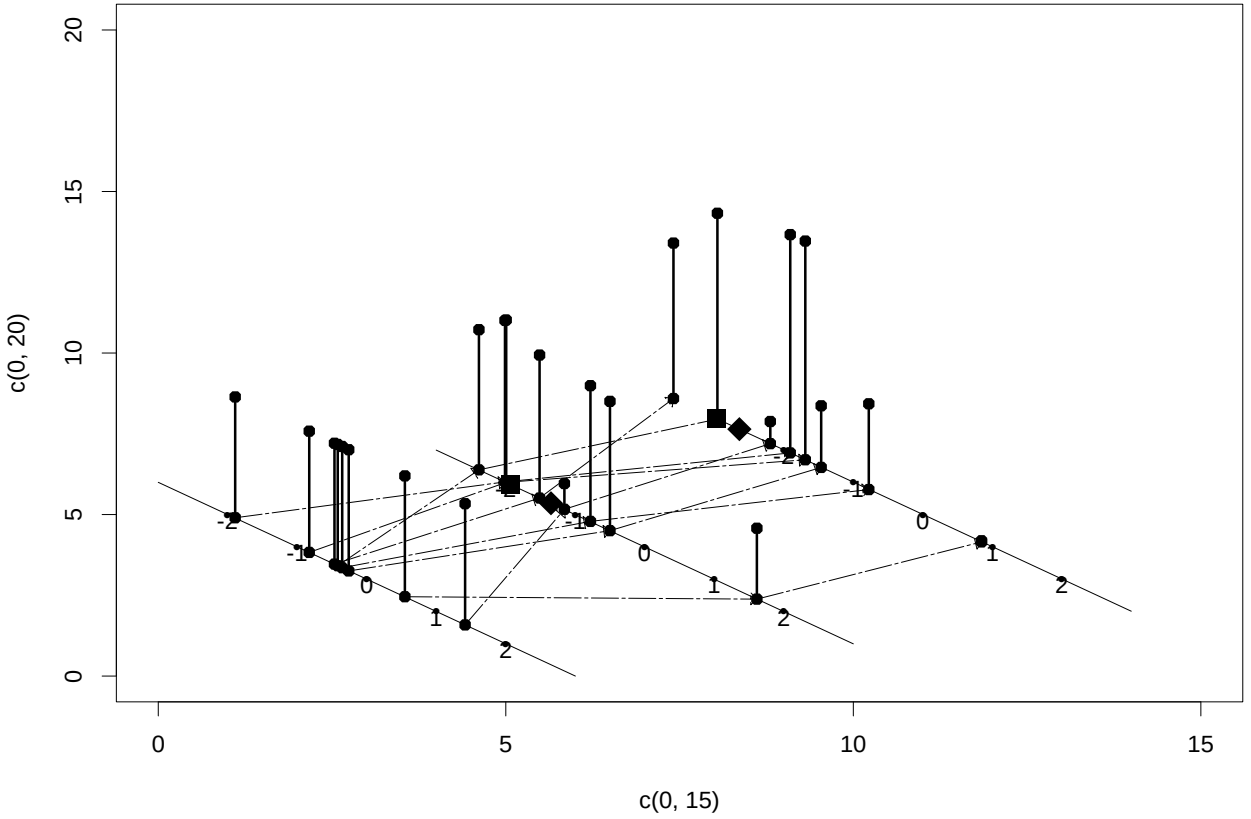


$$x_2 = x_1 + N(0,1), \quad y_2 = x_2 + N(0,1)$$



2. Sequential Monte Carlo – An Illustration

$$x_0 \sim N(0,1), \quad x_t = x_{t-1} + N(0,1) \quad y_t = x_t + N(0,1)$$



SMC Step: for $j = 1, \ldots, m$:

(A) Draw $x_{t+1}^{(j)}$ from $g(x_{t+1} \mid \boldsymbol{x}_t^{(j)})$.

Let $\boldsymbol{x}_{t+1}^{(j)} = (\boldsymbol{x}_t^{(j)}, x_{t+1}^{(j)})$.

(B) Compute the incremental weight

$$u_{t+1}^{(j)} \propto \frac{\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)})}{\pi_t(\boldsymbol{x}_t^{(j)}) g(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)})}$$

and the new weight

$$w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}$$

In State Space Model:

SMC Step: for $j = 1, \ldots, m$:

(A) Draw $x_{t+1}^{(j)}$ from a trial distribution $g_t(x_{t+1} \mid x_t^{(j)}, y_{t+1})$.

(B) Compute the incremental weight

$$u_{t+1}^{(j)} \propto \frac{q_t(x_{t+1}^{(j)} \mid x_t^{(j)}) f_t(y_{t+1} \mid x_{t+1}^{(j)})}{g(x_{t+1}^{(j)} \mid x_t^{(j)}, y_{t+1})}$$

and the new weight

$$w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}$$

2. Sequential Monte Carlo – Some references

- **Rosenbluth and Rosenbluth (1955, J. Chemical Physics)**
- **Gordon et. al (1993, IEE)**
- **Kong, Liu and Wong (1994, JASA)**
- **Kitagawa (1996, JCGS)**
- **Liu & Chen (1996, JASA)**
- **Liu & Chen (1998, JASA)**
- **Doucet et al (2000)**
- **Liu (2001)**
- **many others**

3. Design Issues

- **(optional) How to choose the intermediate distributions?**
- **How to propagate efficiently? — Use more information**
- **How to handle small weights? — Resampling**
- **How to make inference efficiently? — Rao-Blackwellization**

3.1 Propagation – Standard Particle Filter

$$\textbf{state equation:} \quad x_t = s_t(x_{t-1}, \varepsilon_t) \ \textbf{or} \ x_t \sim q_t(\cdot \mid x_{t-1})$$
$$\textbf{observation equation:} \quad y_t = h_t(x_t, e_t) \quad \textbf{or} \quad y_t \sim f_t(\cdot \mid x_t)$$

(1) Standard Particle Filters: (use the state equation only)

$$g_t(x_t \mid x_{t-1}, y_t) = q_t(x_t \mid x_{t-1})$$

with weight $w_t = w_{t-1} f_t(y_t \mid x_t)$**.**

- Algorithm: **For each** j**,**
  1. **Generate** $\varepsilon_t^{(j)}$ **and obtain** $x_t^{(j)} = s_t(x_{t-1}^{(j)}, \varepsilon_t^{(j)})$**.**
  2. **Calculate** $w_t^{(j)} = w_{t-1}^{(j)} f_t(y_t \mid x_t^{(j)})$**.**
- **Easy and fast**
- **when** $\varepsilon_t^{(j)}$ **is easy to generate,** s_t **and** f_t **is easy to evaluate.**
- **But uses only partial information.**

(2) Independent Particle Filters: (use the observation equation only) (Lin et al, 2004)

$$g_t(x_t \mid x_{t-1}, y_t) \propto f_t(y_t \mid x_t)$$

with weight $w_t = q_t(x_t \mid x_{t-1})$**.**

- **When information from** q_t **is weak and that from** f_t **is strong.**
- **Samples** $x_t^{(j)}$ **are independent to each other and to the past particles** $x_{t-1}^{(i)}$
- **Multiple matching**
- **Discharging**
- **Can be extended to any** g_t **that does not involve with** x_{t-1}**.**

3.1 Propagation – Use full information

(3) Full information (Liu and Chen 1998)

$$g_t(x_t \mid x_{t-1}, y_t) \propto q_t(x_t \mid x_{t-1}) f_t(y_t \mid x_t)$$

with weight

$$w_t = w_{t-1} \int q_t(x_t \mid x_{t-1}) f_t(y_t \mid x_t) dx_t$$

- **Weight does not depend on the new sample x_t**
- **Uses all information**
- **Very efficient**
- **Can be difficult to generate samples from**

(4) Auxiliary Particle Filters (Pitt & Shepherd, 1998):

$$g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t) \propto q_{t+1}(x_{t+1} \mid x_t)\hat{f}_{t+1}(y_{t+1} \mid x_{t+1})$$

with incremental weight

$$u_{t+1} \propto \frac{f_{t+1}(y_{t+1} \mid x_{t+1})}{\hat{f}_{t+1}(y_{t+1} \mid x_{t+1})} \int q_{t+1}(x_{t+1} \mid x_t)\hat{f}_{t+1}(y_{t+1} \mid x_{t+1})dx_{t+1}$$

where $\hat{f}_{t+1}$ is an approximation of f_{t+1}.

If q_{t+1} is normal, choose $\hat{f}_{t+1}$ as normal, or mixture normal.

(5) Extended Auxiliary Particle Filters

$$g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t) \propto \hat{q}_{t+1}(x_{t+1} \mid x_t)\hat{f}_{t+1}(y_{t+1} \mid x_{t+1})$$

with incremental weight

$$\begin{aligned} u_{t+1} \;\propto\; & \frac{q_{t+1}(x_{t+1} \mid x_t) f_{t+1}(y_{t+1} \mid x_{t+1})}{\hat{q}_{t+1}(x_{t+1} \mid x_t)\hat{f}_{t+1}(y_{t+1} \mid x_{t+1})} \\ & \times \int \hat{q}_{t+1}(x_{t+1} \mid x_t)\hat{f}_{t+1}(y_{t+1} \mid x_{t+1}) dx_{t+1} \end{aligned}$$

where $\hat{q}_{t+1}, \hat{f}_{t+1}$ are approximations of q_{t+1}, f_{t+1}.

3.1 Propagation – Delay Strategy (look ahead)

- **Dynamic systems often process strong 'memory'**
- **Future observations can reveal substantial information on the current state**
- **Slight delay is tolerable**
- **Make inference on the state x_t at time $t+d$, with information $y_1, \ldots, y_{t+d}$ available.**
- **The target distribution becomes**

$$\pi_t^*(\boldsymbol{x}_t) = \int \pi_{t+d}(\boldsymbol{x}_t, x_{t+1}, \ldots, x_{t+d}) dx_{t+1} \ldots x_{t+d})$$

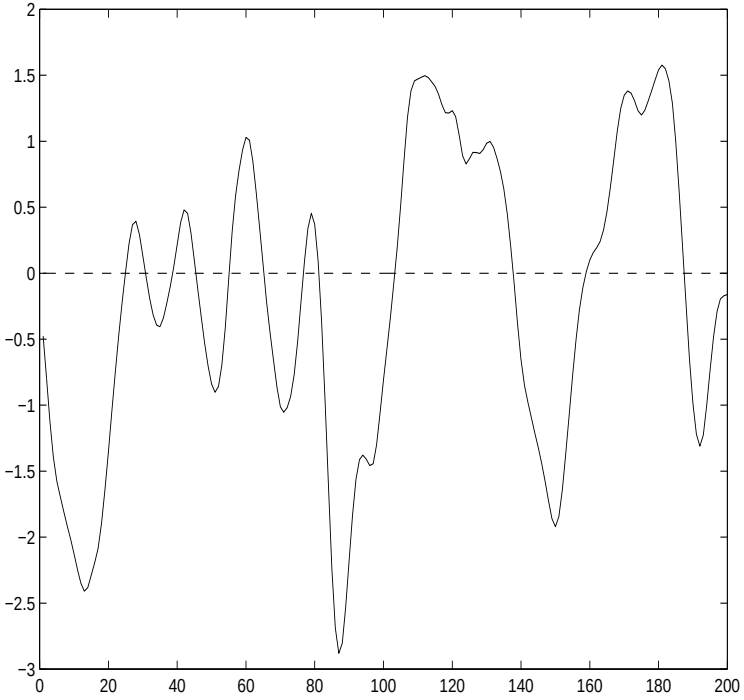
In state space model,

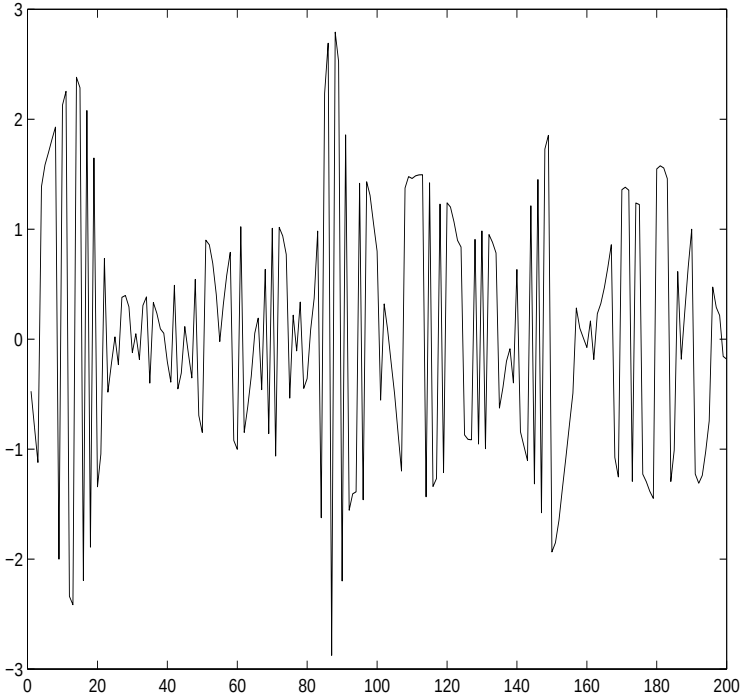
$$\pi_t(x_t) = p(x_t \mid y_1, \ldots, y_t, y_{t+1}, \ldots, y_{t+d})$$

- **Closer to the ultimate target distribution:** $p(x_t \mid y_1, \ldots, y_n)$

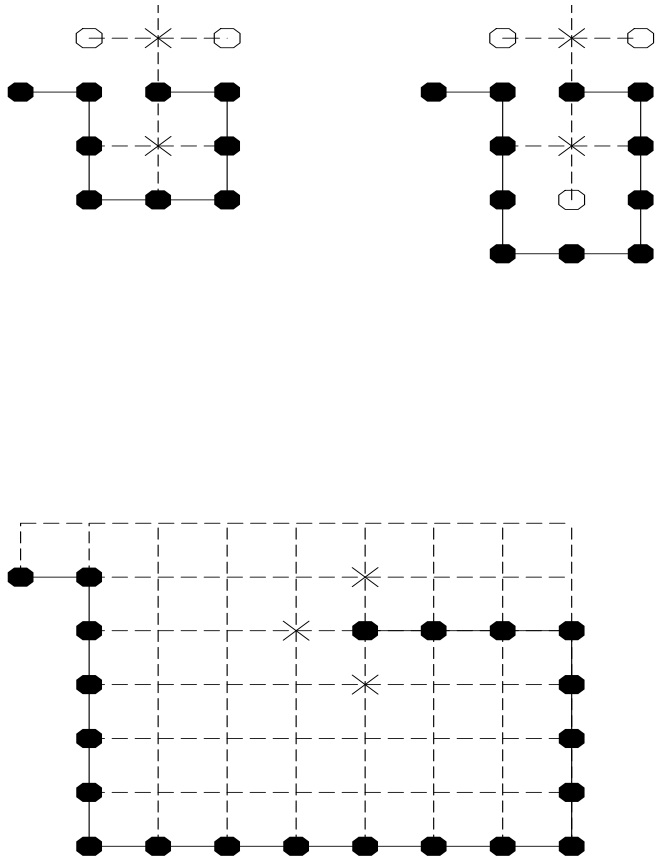
3.1 Propagation – Delay Strategy (look ahead)

Example (II) – Fading Channels





Example (VI): SAW



3.1 Propagation – Delay Strategy (look ahead)

(1) Delayed Weight Method:

If: $(\boldsymbol{x}_{t+d}^{(k)}, w_{t+d}^{(k)})$ **properly weighted w.r.t.** $p(\boldsymbol{x}_{t+d} \mid \boldsymbol{y}_{t+d})$

then: $(x_t^{(k)}, w_{t+d}^{(k)})$ **properly weighted w.r.t.** $p(x_t \mid \boldsymbol{y}_{t+d})$**.**

Hence,

- **Inference on** x_t **can be made using** $(x_t^{(k)}, w_{t+d}^{(k)})$**, with standard SMC.**
- **Sample of** x_t **is drawn at time** t**, based on** $\boldsymbol{y}_t$**, with weight** w_t
- **Inference on** x_t **is made at time** $t+d$**, with weight** w_{t+d}**;**
  —— w_{t+d} **is based on** $\boldsymbol{y}_{t+d}$ **and samples of** $x_{t+1}, \ldots, x_{t+d}$**.**

$$E(f(x_t) \mid \boldsymbol{y}_{t+d}) \approx \frac{\sum_{k=1}^{m} f(x_t^{(k)}) w_{t+d}^{(k)}}{\sum_{k=1}^{m} w_{t+d}^{(k)}}$$

3.1 Propagation – Delay Strategy (look ahead)

(2) Delayed Sample Method: (exact)

- **At time t, Target distribution:** $\pi_t(\boldsymbol{x}_t) = p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+d})$
- **Sample x_t based on the full information $\boldsymbol{y}_{t+d}$**

Suppose at time t, $(\boldsymbol{x}_t^{(j)}, w_t^{(j)})$ properly weighted w.r.t. $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+d})$.
For $t+1$

(A) Draw $x_{t+1}^{(j)}$ from $g(x_{t+1} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+d+1})$
(B) Compute the incremental weight

$$u_{t+1}^{(j)} = \frac{p(\boldsymbol{x}_{t+1}^{(j)} \mid \boldsymbol{y}_{t+d+1})}{p_t(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{y}_{t+d}) g(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+d+1})}$$

and the new weight

$$w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}$$

3.1 Propagation – Delay Strategy (look ahead)

For example, sampling distribution

$$p(x_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{y}_{t+d}, y_{t+d+1}) \propto \int p(\boldsymbol{x}_{t+d+1}, \boldsymbol{y}_{t+d+1}) dx_{t+2} \ldots dx_{t+d+1}$$

and weight is

$$w_{t+1} \propto w_t \frac{p(y_{t+d+1}, \boldsymbol{y}_{t+d} \mid \boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+d} \mid \boldsymbol{x}_t)}$$

Excessive computing cost

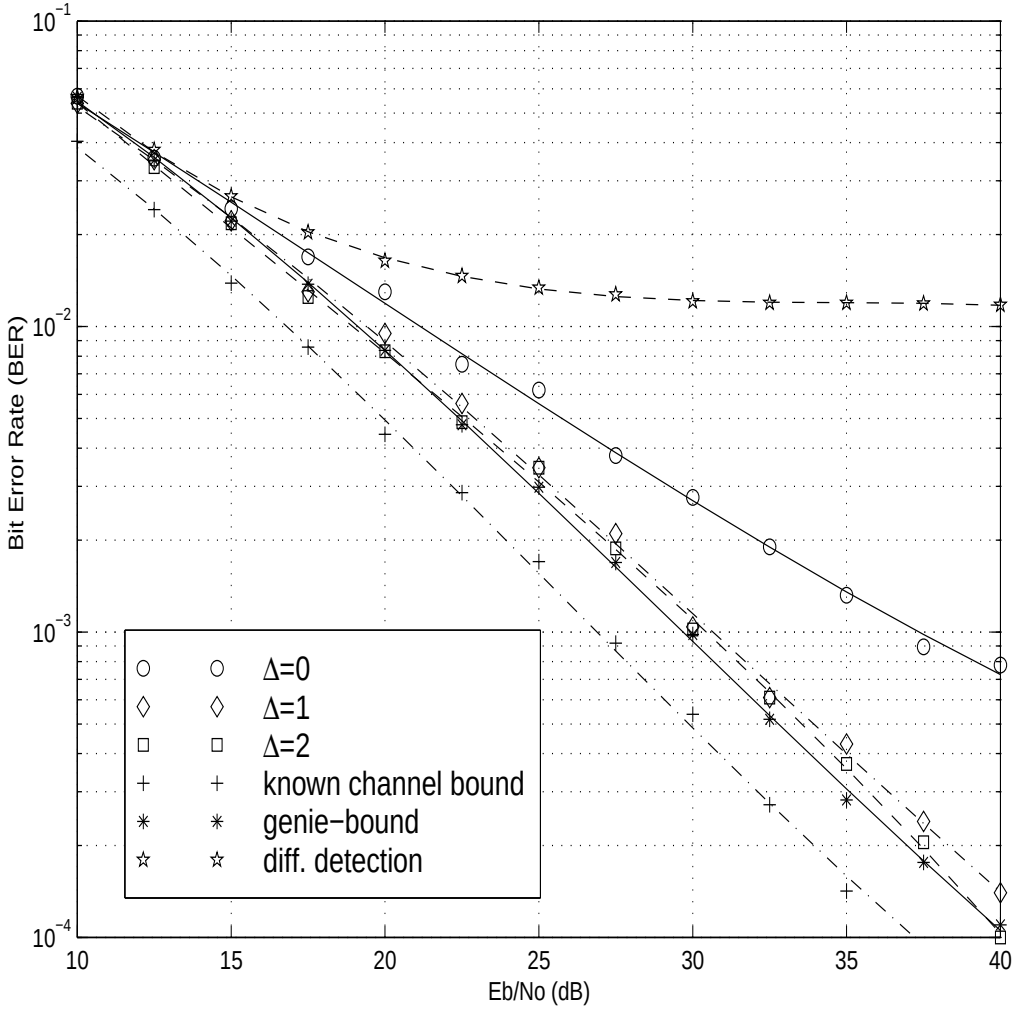
3. Design Issues – Propagation (choosing g_t)

More Efficient and Advance Delay Methods:

- **Hybrid method – combining of delay weight and delay sample**
- **Pilot sampling method – sending pilot to partially explore the future space**
- **Mutli-level method – coarsen (reconstruct) the state space and use delayed SMC on the simpler state space, refine results in the original state space**
- **Adaptive delay method – long delay when information is week and short/no delay when information is strong.**

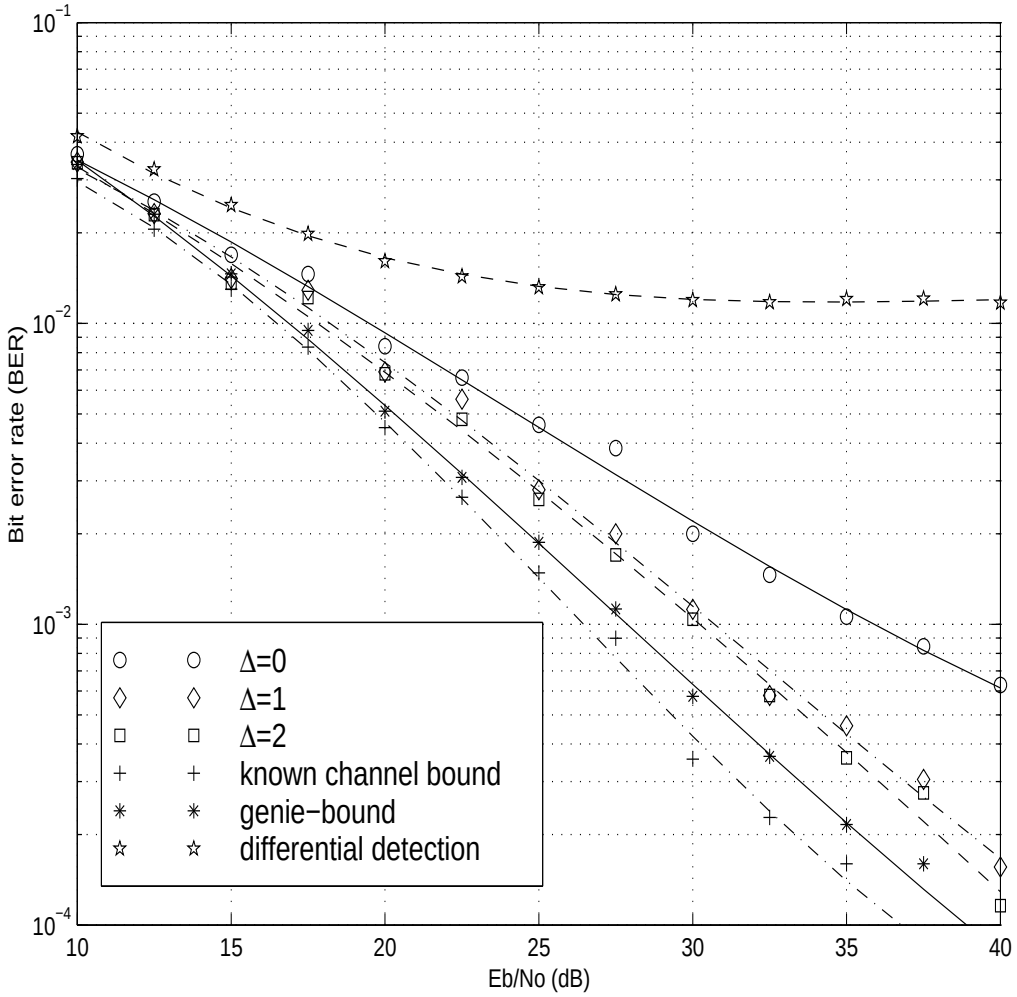
3. Design Issues – Propagation (choosing g_t)

Example (II) – Fading Channels – Uncoded – Gaussian Noise



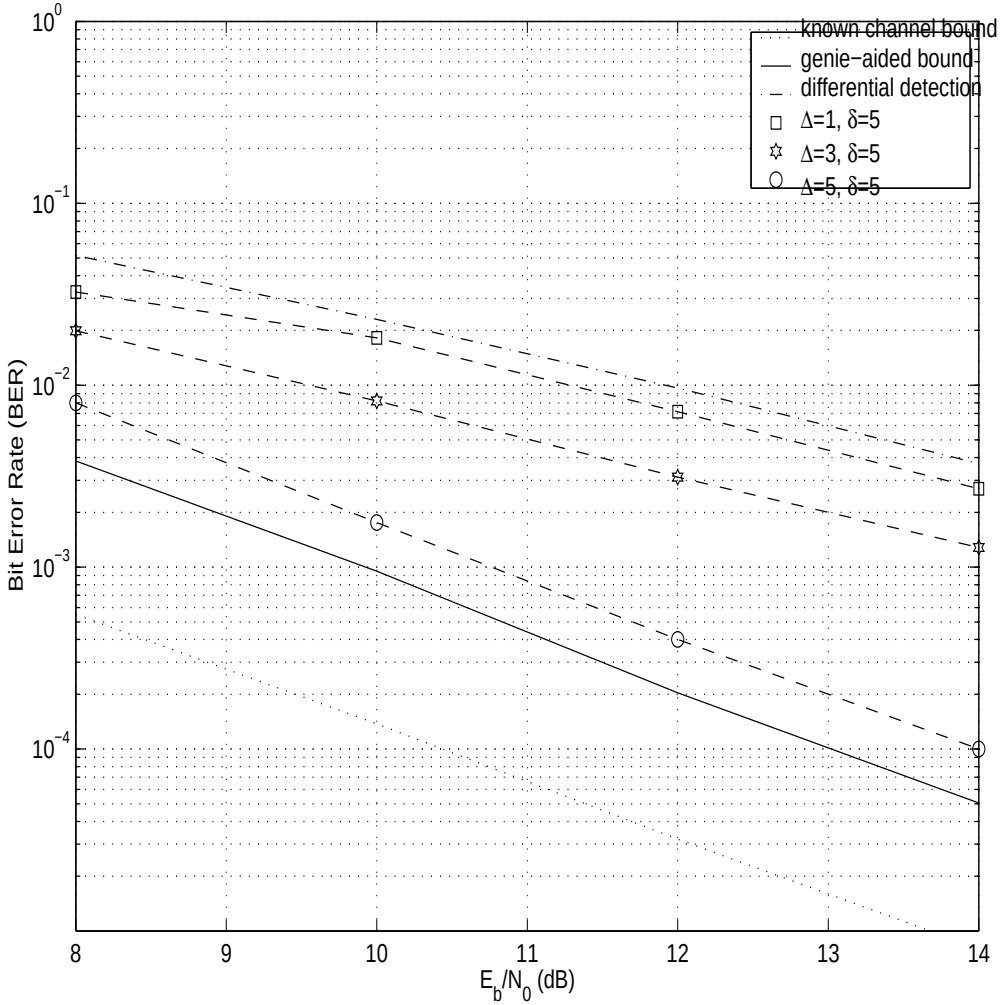
3. Design Issues – Propagation (choosing g_t)

Example (II) – Fading Channels – Uncoded – Mixture Gaussian Noise



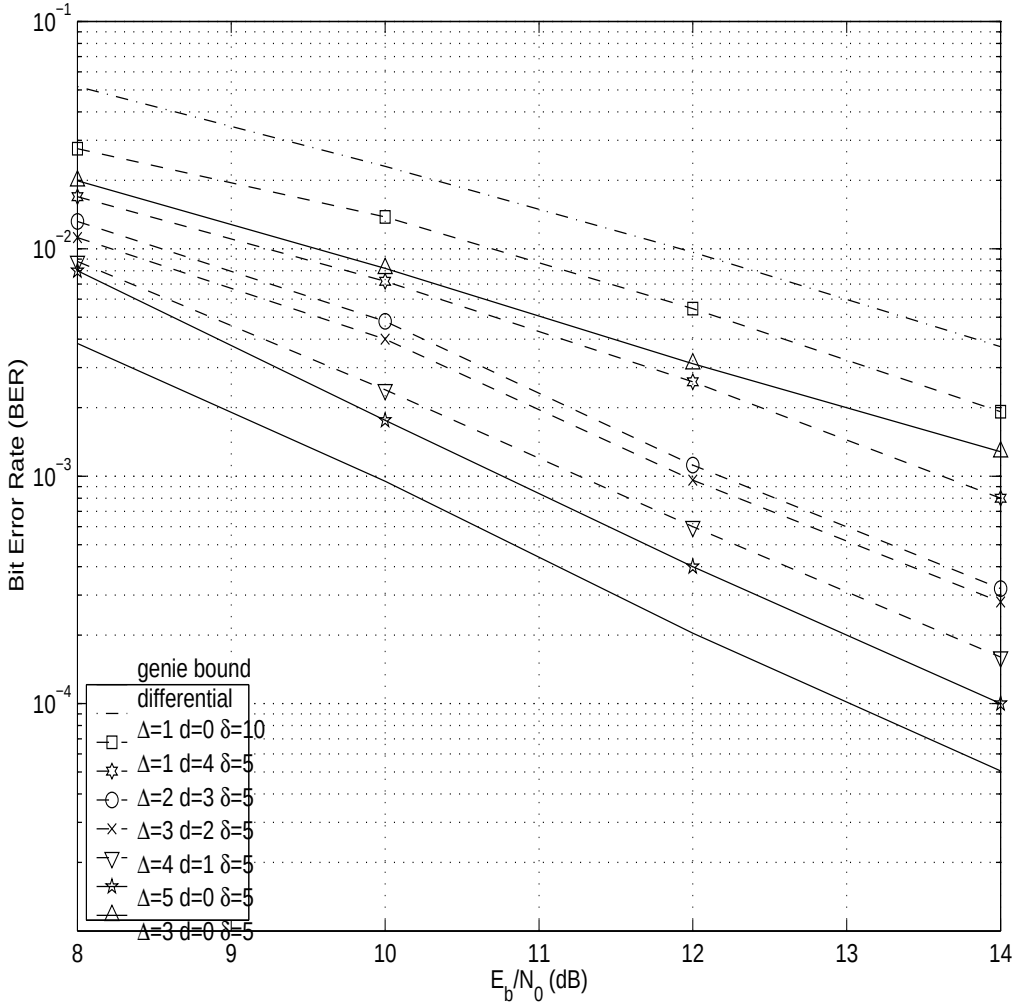
3. Design Issues – Propagation (choosing g_t)

Example (II) – Fading Channels – Coded – Hybrid Delay Sample and Delay Weight



3. Design Issues – Propagation (choosing g_t)

Example (II) – Fading Channels – Coded – Hybrid Delay Pilot, Sample and Weight



Fact: variance of w_t increases (stochastically) as t increases

Simple Resampling Step:
(A) Sample a new set of streams S_t' from S_t according to $\alpha_t^{(j)}$
(B) If stream $\boldsymbol{x}_t^{(j)}$ is sampled, assign it a new weight $w_t^{(j)}/\alpha_t^{(j)}$.

- **when $\alpha_t^{(j)} = w_t^{(j)}$, all new weight become one.**
- **$\alpha_t = w_t^c$, $0 < c < 1$ works better in many cases**

Remarks:

- **Resampling provides more efficient samples of future states**
- **Resampling increases sampling variation in the past states**
- **Resampling reduces the number of distinctive samples in the past states**
- **Frequent resampling can be** *shortsighted*

3.2 Resampling – Targeted resampling

Targeted resampling: choose $\alpha^{(j)}$ based on the objectives.

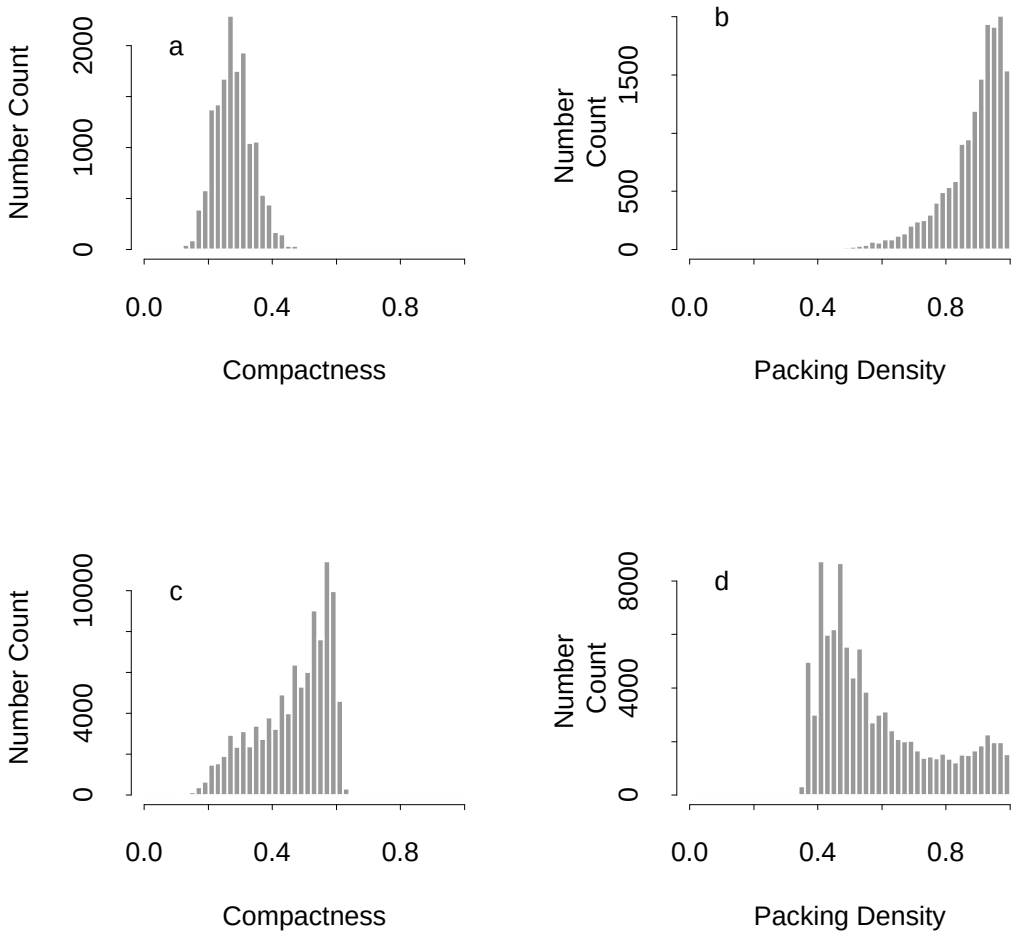
Specifically,

- **The final target distribution π_n is a truncated distribution on a smaller space C.**
- **It is difficult to grow SAWs so that they all in the space C.**

Our approach:

- **Grow properly weighted samples with respect to $\pi_t(\boldsymbol{x}_t)$, untruncated.**
- **Use rejection at the end to achieve truncation.**
- **Use targeted resampling to increase acceptance rate.**
- **Large α for the samples with better chance to grow into C.**

Example (IV): SAW – conditional inference



3.2 Resampling – Implementation

- **Resampling methods:**
  - **Simple random sampling**
  - **Residual sampling:** **Obtain** $[m\alpha^{(j)}]$ **copy of** $x_t^{(j)}$**.**
    Sample the remaining $m - \sum[m\alpha^{(j)}]$ **with prob.** $\propto m\alpha^{(j)} - [m\alpha^{(j)}]$**.**
  - **Local Monte Carlo method**
- **Resampling Schedule:**
  - **deterministic: resampling at time** $t_0, 2t_0, 3t_0, ...$
  - **dynamic: monitoring the weight variance**
- **Delayed resampling**
- **Pilot guided resampling**
- **Rejection control**
- **Adaptive resampling**

Example (IV): Blind Deconvolution

A simulated example:

$$y_t = x_t + 0.8x_{t-1} - 0.4x_{t-2} + \varepsilon_t$$

with x_t i.i.d from $\{0, 1, 3\}$ and SNR=15dB.

- **The coefficients ϕ are integrated out with a normal prior.**
- **200 simulated sequences. Sample size $T = 200$.**
- **Number of streams $m = 1000$.**
- **Delayed estimation: $\hat{x}_t = MAP(\pi_{t+3}(x_t))$**
- **simple random sampling (s) versus residual sampling (r)**
- **Deterministic schedule: $t_0, 2t_0, 3t_0, \ldots$**
  Dynamic schedule: when the effective sample size is less than 3.

	Deterministic Resampling Schedule t_0												dynamic schedule	
	1		5		20		50		100		200			
error	s	r	s	r	s	r	s	r	s	r	s	r	s	r
0-2	11	5	7	13	13	13	7	10	1	0	0	0	11	12
3-5	49	49	46	53	61	65	53	49	28	28	7	7	69	58
6-8	41	43	50	52	72	70	57	58	59	58	12	12	66	67
9-11	23	20	27	30	38	38	52	48	43	44	47	47	29	41
12-15	10	9	13	7	8	6	17	20	33	32	44	44	16	8
16-25	11	10	14	11	8	8	14	15	35	35	84	84	6	11
16-50	4	10	8	9	0	0	0	0	1	3	6	6	1	1
>50	51	54	35	25	0	0	0	0	0	0	0	0	2	2

3.3 Inference

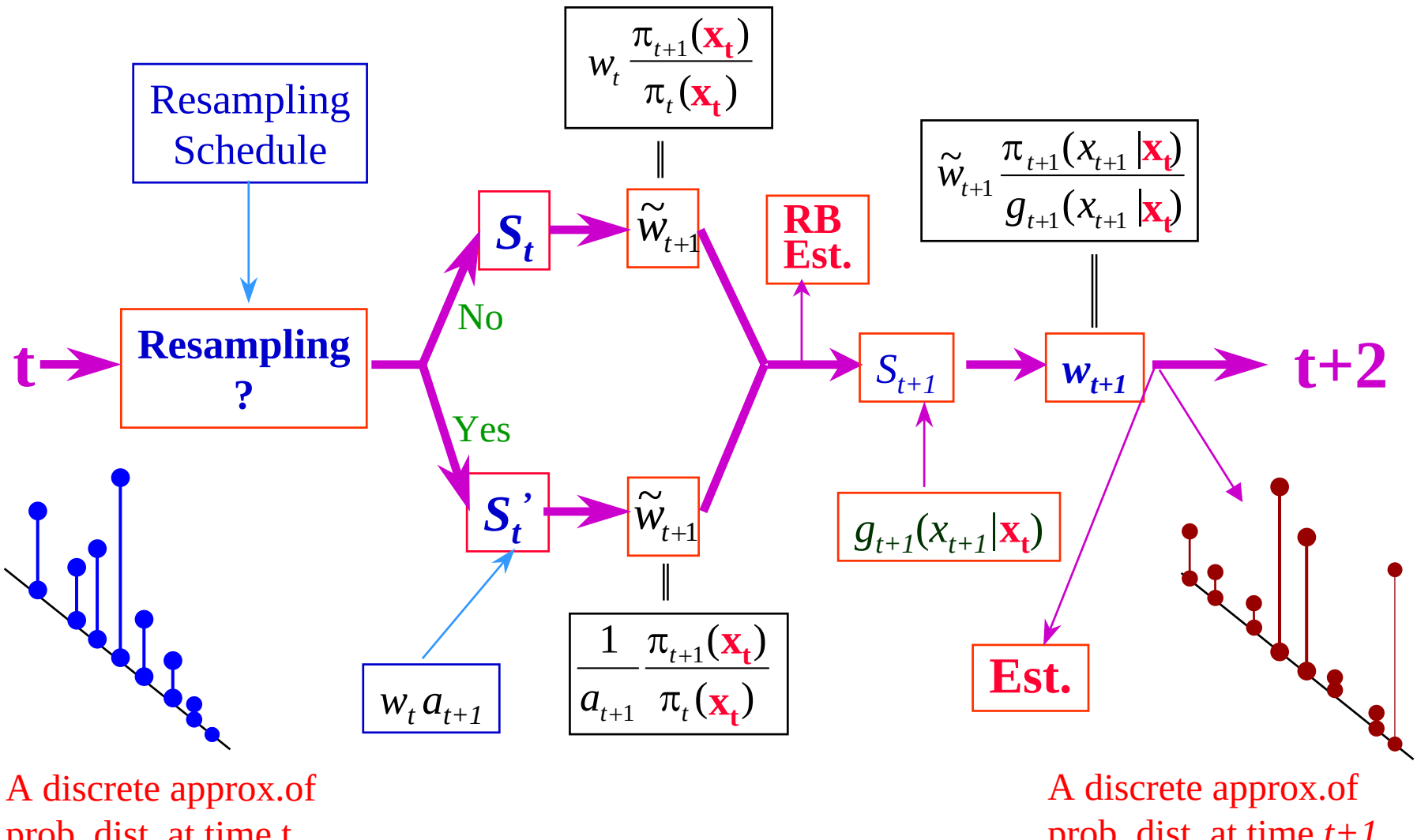
Inference:

$$\hat{E}_{\pi_t} h(\boldsymbol{x}_t) = \frac{\sum_{j=1}^m w_t^{(j)} h(\boldsymbol{x}_t^{(j)})}{\sum_{j=1}^m w_t^{(j)}}$$

- **Estimation should be done before a resampling step**
- **Rao-Blackwellization: For example, if w_{t+1} does not depend on x_{t+1}, then**

$$\hat{E}_{\pi_{t+1}} h(x_{t+1}) = \frac{\sum_{j=1}^m w_{t+1}^{(j)} E_{\pi_{t+1}}(h(x_{t+1}) \mid \boldsymbol{x}_t^{(j)})}{\sum_{j=1}^m w_{t+1}^{(j)}}$$

- **Delayed estimation (i.e. $E_{\pi_t} h(x_{t-k})$ at time t) is usually more accurate since the estimation is based on more information.**
- **Frequent resampling may have adverse effect.**



A discrete approx.of prob. dist. at time t.

A discrete approx.of prob. dist. at time *t+1*.

- **Indicator Λ_t: (unobserved)**
- **If $\Lambda_t = \lambda$ then**

$$\begin{aligned} x_t &= H_\lambda x_{t-1} + W_\lambda w_t \\ y_t &= G_\lambda x_t + V_\lambda v_t \end{aligned}$$

- **where $w_t \sim N(0, I)$ and $v_t \sim N(0, I)$ and independent.**
- **Given the trajectory of the indicator $\{\Lambda_1, \ldots \Lambda_t\}$, the system is linear and Gaussian.**

Example (I): Target tracking in clutter

Introducing an indicator I_t taking values in $\{0, 1, \ldots, m_t\}$.
$I_t = 0$ true signal missing. $I_t = i$, then $y_t^{(i)}$ is the true signal.

$$\begin{aligned} x_t &= Hx_{t-1} + Ww_t \\ y_t^{(i)} &= Gx_t + Vv_t \qquad \textbf{if} \quad I_t = i \\ y_t^{(j)} &\sim \textbf{Unif}(\Delta) \qquad \textbf{if} \quad I_t \neq j \end{aligned}$$

and

$$P(I_t = 0) = p_d \textbf{ and } \; P(I_t = i) = (1 - p_d)/m_t$$

Given the trajectory of the indicator $\{I_1, \ldots I_t\}$, the system is linear and Gaussian.

4. Mixture Kalman Filter

Let $\boldsymbol{y}_t = (y_1, \ldots, y_t)$ and $\boldsymbol{\Lambda}_t = (\Lambda_1, \ldots, \Lambda_t)$.

Note that

$$p(x_t \,|\, \boldsymbol{y}_t) = \int p(x_t \,|\, \boldsymbol{\Lambda}_t, \boldsymbol{y}_t) dF(\boldsymbol{\Lambda}_t \,|\, \boldsymbol{y}_t)$$

and

$$p(x_t \mid \boldsymbol{\Lambda}_t, \boldsymbol{y}_t) \sim N(\mu_t(\boldsymbol{\Lambda}_t), \sigma_t^2(\boldsymbol{\Lambda}_t))$$

where

$$KF_t(\boldsymbol{\Lambda}_t) \equiv (\mu_t(\boldsymbol{\Lambda}_t), \sigma_t^2(\boldsymbol{\Lambda}_t))$$

can be obtained from Kalman filter.

$p(x_t \mid y_1, \ldots y_t)$ **is a mixture Gaussian distribution.**

Monte Carlo Filter:
a discrete sample with weight

$$\{(x_t^{(1)}, w_t^{(1)}), \ldots, (x_t^{(m)}, w_t^{(m)})\} \Longrightarrow p(x_t \mid y_1, \ldots, y_t)$$

Mixture Kalman Filter:
a discrete sample with weight

$$\{(\boldsymbol{\lambda}_t^{(1)}, w_t^{(1)}), \ldots, (\boldsymbol{\lambda}_t^{(m)}, w_t^{(m)})\} \Longrightarrow p(\boldsymbol{\Lambda} \mid y_1, \ldots, y_t)$$

and a random mixture of Normal distributions

$$\sum_{i=1}^{m} w_t^{(i)} N(\mu_t(\boldsymbol{\lambda}_t^{(i)}), \sigma_t^2(\boldsymbol{\lambda}_t^{(i)})) \Longrightarrow p(x_t \mid y_1, \ldots, y_t).$$

Hence

$$E(f(x_t) \mid y_1, \ldots, y_t) \approx \sum_{i=1}^{m} w_t^{(i)} \int f(x)\phi(x; \mu_t(\boldsymbol{\lambda}_t^{(i)}), \sigma_t^2(\boldsymbol{\lambda}_t^{(i)}))dx$$

Benefit: improved efficiency

$$Var[f(x_t) \mid \boldsymbol{y}_t] \geq Var[E(f(x_t) \mid \boldsymbol{\Lambda}_t, \boldsymbol{y}_t) \mid \boldsymbol{y}_t]$$

Algorithm:

At time t, we have a sample $(\boldsymbol{\lambda}_t^{(i)}, KF_t^{(i)}, w_t^{(i)})$

For $t+1$,

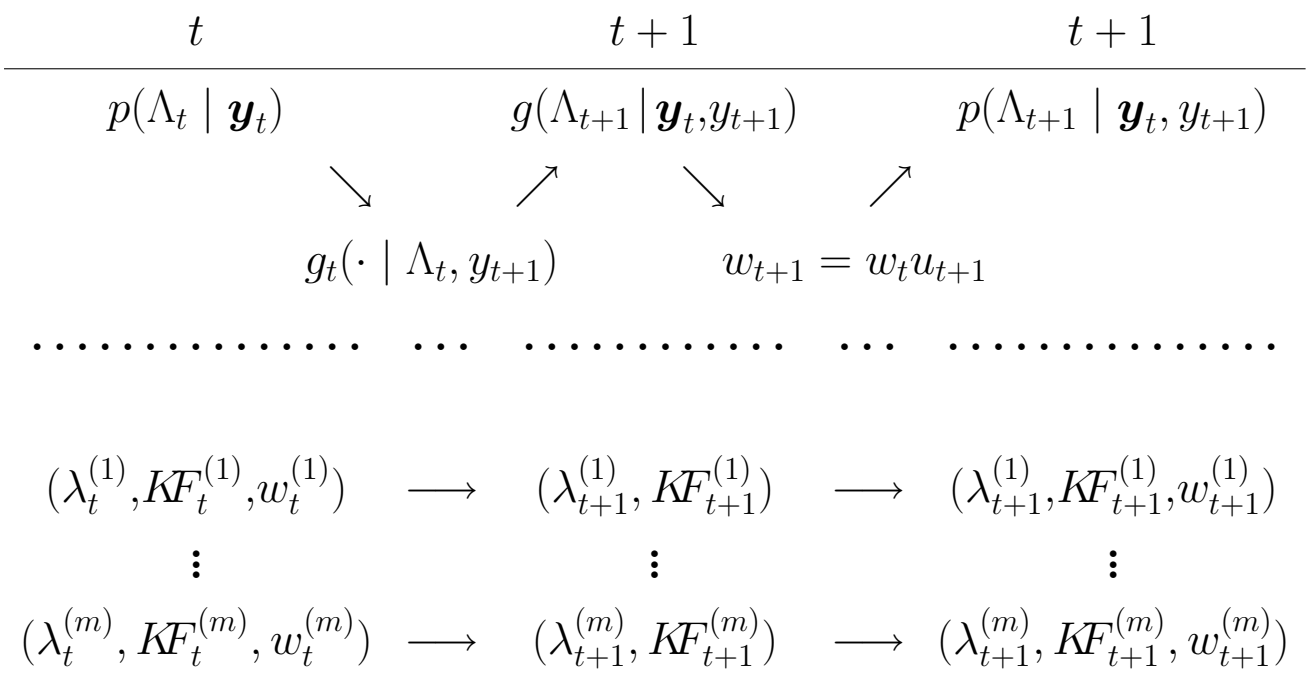
(1) : generate $\lambda_{t+1}^{(i)}$ from a trial distribution $g(\Lambda_{t+1} \mid \boldsymbol{\lambda}_t^{(i)}, KF_t^{(i)}, y_{t+1})$

(2) : run one step Kalman filter conditioning on $(\lambda_{t+1}^{(i)}, KF_t^{(i)}, y_{t+1})$ and obtain $KF_{t+1}^{(i)}$.

(3) : calculate the incremental weight

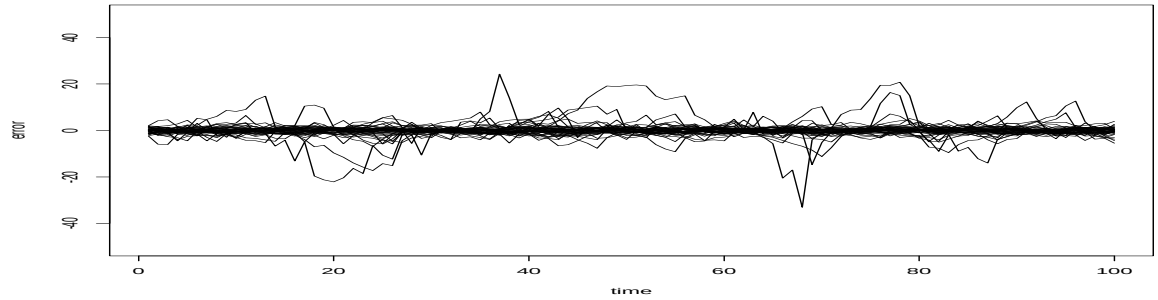
$$u_{t+1}^{(i)} = \frac{p(\boldsymbol{\lambda}_t^{(i)}, \lambda_{t+1}^{(i)} \mid \boldsymbol{y}_{t+1})}{p(\boldsymbol{\lambda}_t^{(i)} \mid \boldsymbol{y}_t) g(\lambda_{t+1} \mid \boldsymbol{\lambda}_t^{(i)}, KF_t^{(i)}, y_{t+1})}$$

and the new weight $w_{t+1}^{(i)} = w_t^{(i)} u_{t+1}^{(i)}$.

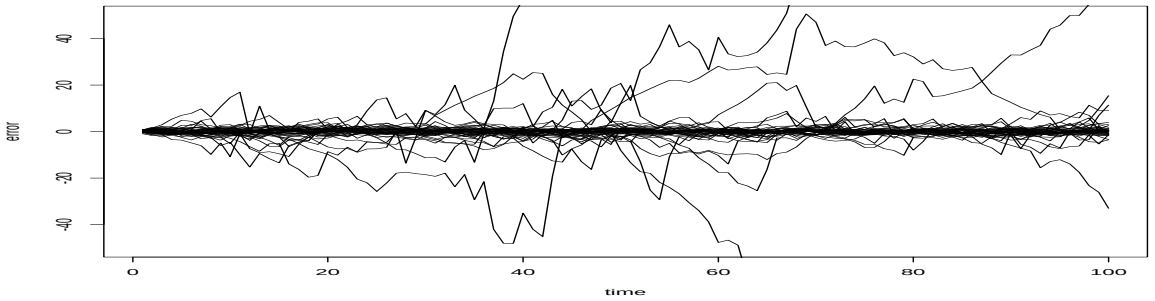

t
$t+1$
$t+1$
$p(\Lambda_t \mid \boldsymbol{y}_t)$
$g(\Lambda_{t+1} \mid \boldsymbol{y}_t, y_{t+1})$
$p(\Lambda_{t+1} \mid \boldsymbol{y}_t, y_{t+1})$
$g_t(\cdot \mid \Lambda_t, y_{t+1})$
$w_{t+1} = w_t u_{t+1}$
$(\lambda_t^{(1)}, KF_t^{(1)}, w_t^{(1)})$
$(\lambda_{t+1}^{(1)}, KF_{t+1}^{(1)})$
$(\lambda_{t+1}^{(1)}, KF_{t+1}^{(1)}, w_{t+1}^{(1)})$
$(\lambda_t^{(m)}, KF_t^{(m)}, w_t^{(m)})$
$(\lambda_{t+1}^{(m)}, KF_{t+1}^{(m)})$
$(\lambda_{t+1}^{(m)}, KF_{t+1}^{(m)}, w_{t+1}^{(m)})$

Example (I) – Target tracking

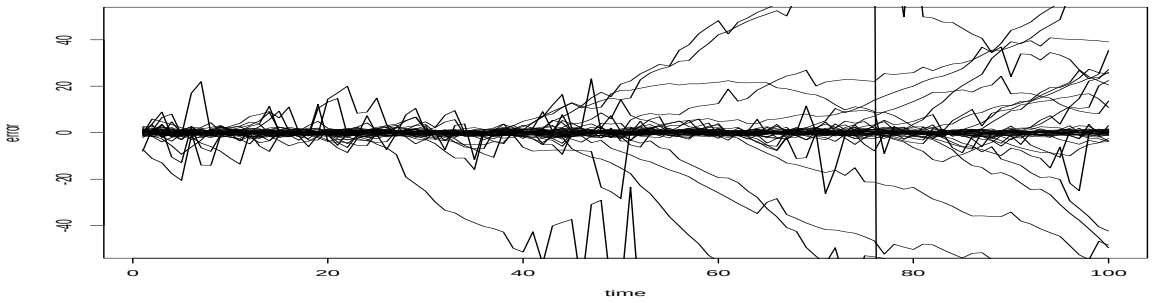
MKF:



PF1:



PF2:



THANK YOU!