Short Course

State Space Models, Generalized Dynamic Systems
and
Sequential Monte Carlo Methods,
and
their applications
in Engineering, Bioinformatics and Finance

Rong Chen
Rutgers University
Peking University

Part Three: Advanced Sequential Monte Carlo

3.1 Mixture Kalman Filter

3.1.1 Conditional Dynamic Linear Models

3.1.2 Mixture Kalman Filters

3.1.3 Partial Conditional Dynamic Linear Models

3.1.4 Extend Mixture Kalman Filters

3.1.5 Future Directions

3.2 Constrained SMC

3.3 Parametrer Estimation in SMC

3.1.1 Conditional Dynamic Linear Models

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: Tracking a target 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.

Example: Tracking a target with non-Gaussian innovations.

$$
\begin{aligned}
x_t &= Hx_{t-1} + Ww_t \\
y_t &= Gx_t + Vv_t
\end{aligned}
$$

where $w_t \sim t_{k_1}$**,** $v_t \sim t_{k_2}$**.**
Note that $t_k = N(0,1)/\sqrt{\chi_k^2/k}$

Introducing indicators $\Lambda_t = (\Lambda_{t1}, \Lambda_{t2})$**.**

$$
\begin{aligned}
x_t &= Hx_{t-1} + \frac{\sqrt{k_1}}{\sqrt{\lambda_1}} Ww_t \quad \textbf{if} \quad \Lambda_{t1} = \lambda_1 \\
y_t &= Gx_t + \frac{\sqrt{k_2}}{\sqrt{\lambda_2}} Vv_t \quad \textbf{if} \quad \Lambda_{t2} = \lambda_2
\end{aligned}
$$

with $v_t \sim N(0, I)$**,** $w_t \sim N(0, I)$ **and** $\Lambda_{t1} \sim \chi_{k_1}^2$**,** $\Lambda_{t2} \sim \chi_{k_2}^2$**.**
Given the trajectory of the indicator $\{\Lambda_1, \ldots \Lambda_t\}$**, the system is linear and Gaussian.**

Example: Tracking a target with random (Gaussian) acceleration plus maneuvering

$$
\begin{aligned}
x_t &= Hx_{t-1} + Fs_{I_t}u_t + Ww_t \\
y_t &= Gx_t + Vv_t
\end{aligned}
$$

where u_t, w_t **and** v_t **are all** $N(0, I)$ **independent.**

I_t **maneuvering status:**

$I_t = 0$**, no maneuvering,** $s_0 = 0$

$I_t = 1$**, slow maneuvering,** s_1 $I_t = 2$**, fast maneuvering,** s_2

With known transition matrix $P = P(I_{t+1} \mid I_t)$**.**

3.1.2 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.**

<u>(Sequential) Monte Carlo Filter:</u>
a discrete sample with weight

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

<u>Mixture Kalman Filter:</u>
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

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

Example: $X \sim N(\Lambda, \sigma_1^2)$ **and** $\Lambda \sim N(0, \sigma_2^2)$**. Estimate** $\mu = E(X)$

(1) directly sample from $X \sim N(0, \sigma_1^2 + \sigma_2^2)$**,**

$$Var(\hat{\mu}) = Var\left(\frac{\sum_{i=1}^m X_i}{m}\right) = \frac{\sigma_1^2 + \sigma_2^2}{m}$$

(2) sample $\Lambda \sim N(0, \sigma_2^2)$**.**

$$\hat{\mu} = \frac{\sum_{i=1}^m E(X \mid \Lambda_i)}{m} = \frac{\sum_{i=1}^m \Lambda_i}{m}$$

$$Var(\hat{\mu}) = Var\left(\frac{\sum_{i=1}^m \Lambda_i}{m}\right) = \frac{\sigma_2^2}{m}$$

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

$$
\begin{array}{ccc}
t & t+1 & t+1 \\
\hline
p(\Lambda_t \mid \boldsymbol{y}_t) & g(\Lambda_{t+1} \mid \boldsymbol{y}_t, y_{t+1}) & p(\Lambda_{t+1} \mid \boldsymbol{y}_t, y_{t+1})
\end{array}
$$

$$
\searrow \quad \nearrow \qquad \searrow \quad \nearrow
$$

$$
g_t(\cdot \mid \Lambda_t, y_{t+1}) \qquad w_{t+1} = w_t u_{t+1}
$$

$$
\cdots\cdots\cdots \quad \cdots \quad \cdots\cdots\cdots \quad \cdots \quad \cdots\cdots\cdots
$$

$$
\begin{array}{ccccc}
(\lambda_t^{(1)}, KF_t^{(1)}, w_t^{(1)}) & \longrightarrow & (\lambda_{t+1}^{(1)}, KF_{t+1}^{(1)}) & \longrightarrow & (\lambda_{t+1}^{(1)}, KF_{t+1}^{(1)}, w_{t+1}^{(1)}) \\
\vdots & & \vdots & & \vdots \\
(\lambda_t^{(m)}, KF_t^{(m)}, w_t^{(m)}) & \longrightarrow & (\lambda_{t+1}^{(m)}, KF_{t+1}^{(m)}) & \longrightarrow & (\lambda_{t+1}^{(m)}, KF_{t+1}^{(m)}, w_{t+1}^{(m)})
\end{array}
$$

When Λ_t is a discrete r.v. on a finite set, then

(0) : For each $j = 1, \ldots, J$, run a Kalman filter to obtain

$$u_j^{(i)} = p(y_{t+1} \mid \Lambda_{t+1} = j, KF_t^{(i)}) p(\Lambda_{t+1} = j \mid \boldsymbol{\lambda}_t^{(i)})$$

(1) : Sample a $\lambda_{t+1}^{(i)}$ from the set $\{1, \ldots, J\}$ with probability proportional to $u_j^{(i)}$.

i.e. sample a λ from $p(\Lambda_{t+1} \mid \boldsymbol{\lambda}_t, KF_t, y_{t+1})$

(2) : Let $KF_{t+1}^{(i)}$ be the one with $\lambda_{t+1}^{(i)}$.

(3) : The new weight is

$$w_{t+1}^{(i)} \propto w_t^{(i)} p(y_{t+1} \mid \boldsymbol{\lambda}_t^{(i)}, KF_t^{(i)}) \propto w_t^{(i)} \sum_{j=1}^{J} u_j^{(i)}$$

When Λ_t is a continuous r.v., a simple (but not optimum) algorithm is

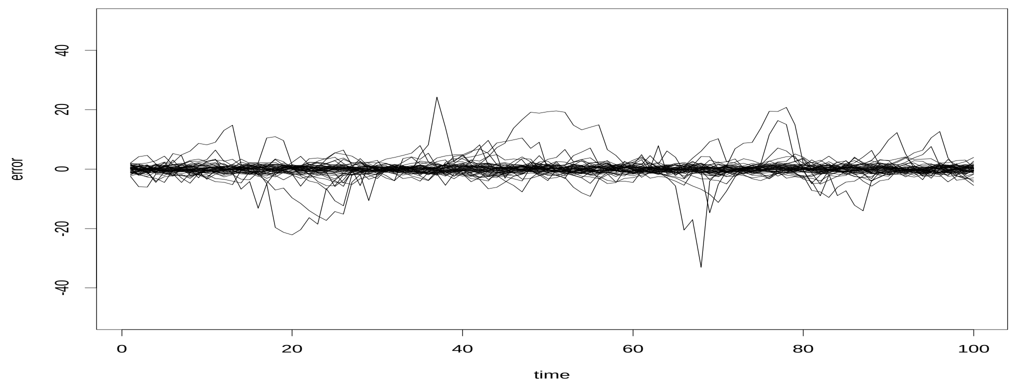
(1) : Sample a $\lambda_{t+1}^{(i)}$ from $p(\Lambda_{t+1} \mid \mathbf{\Lambda}_t = \boldsymbol{\lambda}_t^{(i)})$

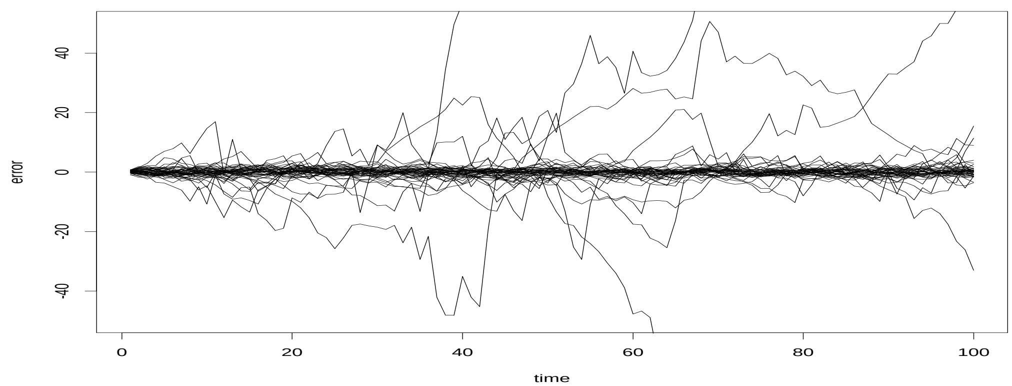
(2) : Run one step Kalman filter conditioning on $(\lambda_{t+1}^{(i)}, KF_t^{(i)}, y_{t+1})$ and obtain $KF_{t+1}^{(i)}$

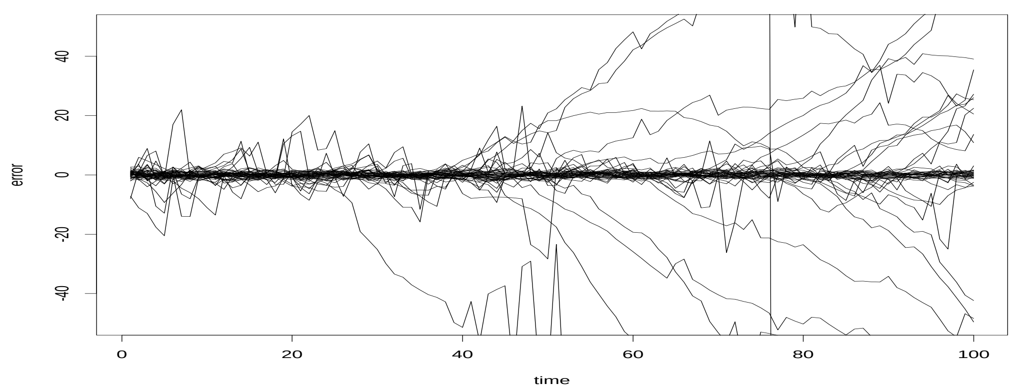
(3) : The new weight is

$$w_{t+1}^{(i)} = w_t^{(i)} p(y_{t+1} \mid \lambda_{t+1}^{(i)}, KF_t^{(i)})$$

Tracking in clutter:







Example: Tracking a target with non-Gaussian innovations

State Equation:

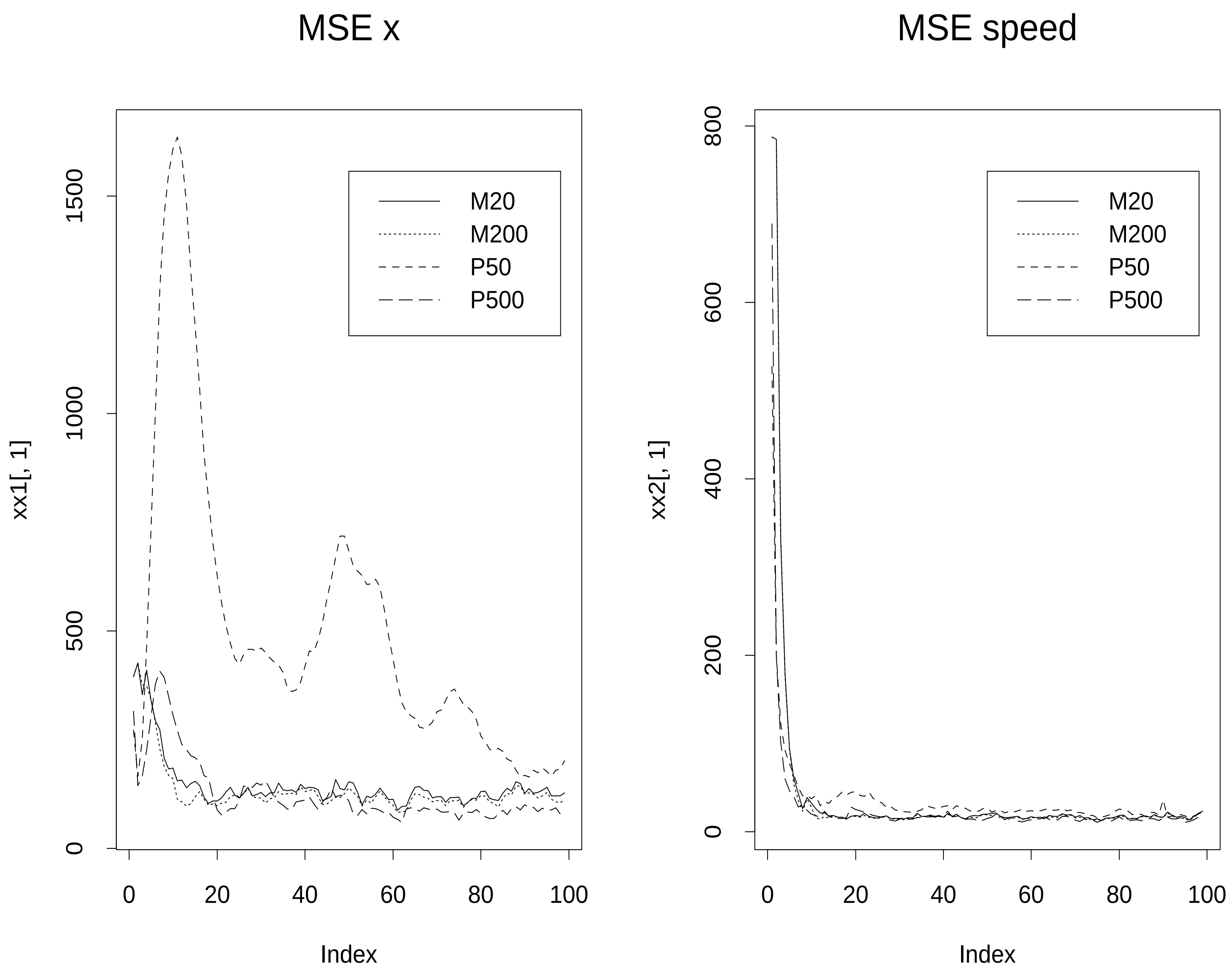
$$\begin{pmatrix} x_t^{(1)} \\ x_t^{(2)} \end{pmatrix} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1}^{(1)} \\ x_{t-1}^{(2)} \end{pmatrix} + q \begin{pmatrix} T/2 \\ 1 \end{pmatrix} w_t$$

true signal

$$y_t = x_t^{(1)} + r v_t$$

where $w_t \sim t_3$ **and** $v_t \sim t_3$**.**

$T = 1$**,** $q^2 = 400/3$**,** $r^2 = 1600/3$**.**


MSE x
M20
M200
P50
P500
xx1[, 1]
0
500
1000
1500
0
20
40
60
80
100
Index
MSE speed
M20
M200
P50
P500
xx2[, 1]
0
200
400
600
800
0
20
40
60
80
100
Index

noise var	**# chains**	**Particle**		**MKF**	
		cpu time	**# miss**	**cpu time**	**# miss**
	20	**9.49843**	**72**	**19.4277**	**1**
	50	**20.1622**	**20**	**51.6061**	**1**
$q^2 = \mathbf{16.0}$	**200**	**80.3340**	**7**	**181.751**	**1**
$r^2 = \mathbf{1600}$	**500**	**273.369**	**4**	**500.157**	**1**
	1500	**1063.36**	**3**	**2184.67**	**1**

3.1.3 Partial CDLM

$$\begin{aligned} \textbf{state equation:} \quad & x_t = g_t(x_{t-1}, \varepsilon_t) \\ \textbf{observation equation:} \quad & y_t = h_t(x_t, e_t) \end{aligned}$$

- **Extract the linear and Gaussian components out, and use Kalman Filter (integrating those components out)**
- **Nonlinear components are dealt with standard Monte Carlo Filters**
- **Non-Gaussian innovations are dealt with indicators and approximations**
- **Nonlinear functions are dealt with 'conditional linearization'?**

State: (x_t, x_t^*). **Observations:** (y_t, y_t^*)

$$
\begin{aligned}
x_t &= g_t(x_{t-1}, x_{t-1}^*, \varepsilon_t) && (1)\\
x_t^* &= H_{x_t} x_{t-1}^* + W_{x_t} w_t && (2)\\
y_t &= h_t(x_t, e_t) && (3)\\
y_t^* &= G_{x_t} x_t^* + V_{x_t} v_t && (4)
\end{aligned}
$$

- x_t, y_t: **nonlinear nonGaussian component**
- x_t^*, y_t^*: **conditional linear Gaussian component**
- $H_{x_t}, G_{x_t}, W_{x_t}, V_{x_t}$: **known matrices given** x_t
- $w_t \sim N(0, I)$ **and** $v_t \sim N(0, I)$ **and independent.**

Given the trajectory of the NLNG components $\{x_1, \ldots x_t\}$, **the system (2) (4) is linear and Gaussian.**

Example: Digital Signal Extraction in Fading Channels

$$\textbf{State Equations:}\begin{cases} x_t &= Hx_{t-1} + w_t \\ \alpha_t &= Gx_t \\ s_t &\sim p(\cdot \mid s_{t-1}) \end{cases}$$

$$\textbf{Observation equation: } y_t = \alpha_t s_t + v_t$$

Or

$$\textbf{State Equations:} \quad x_t = Hx_{t-1} + w_t$$

$$s_t \sim p(\cdot \mid s_{t-1})$$

$$\textbf{Observation equation:} \quad y_t = Gx_t s_t + v_t$$

Example: 2-d target with GPS and IMU sensor.

State:

- **position** p_{1t}, p_{2t}
- **speed** v_{1t}, v_{2t}
- **(total) acceleration** a_{1t}, a_{2t}
- **IMU facing** θ_t
- **IMU rotational speed** ψ_t
- **Two motion status:**
  - $M_t = 1$**: (roughly) zero acceleration (constant between observations)**
  - $M_t = 2$**: (roughly) constant acceleration**

δ: time gap between observations

State equations: ($M_t = 1$)

$$
\begin{aligned}
p_{it} &= p_{it-1} + v_{it-1}\delta + 0.5\delta\varepsilon_{it} \quad i = 1, 2 \\
v_{it} &= v_{it-1} + \varepsilon_{it} \quad i = 1, 2 \\
a_{it} &= \varepsilon_{it} \quad i = 1, 2 \\
\theta_t &= \theta_{t-1} + \psi_{t-1}\delta + 0.5\delta\varepsilon_t^* \\
\psi_t &= \psi_{t-1} + \varepsilon_t^* \\
P(M_t = i &\mid M_{t-1} = j) = p_{ij}
\end{aligned}
$$

Similar for $M_t = 2$

- **One can also impose constraints (maps) on the state equations.**
- **variance of ε_{it} depends on platform (walking or vehicle etc)**

Observations:

- p^*_{1t}, p^*_{2t}: **(post-processed) GPS signal**
- a^*_{1t}, a^*_{2t}: **acceleration in the direction of** θ_t
- η_t: **rotational acceleration**

Observational equations:

$$\begin{aligned} p^*_{it} &= p_{it} + e_{1t} \quad i = 1, 2 \\ a^*_{1t} &= cos(\theta_t)a_{1t} + sin(\theta_t)a_{2t} + w_{1t} \\ a^*_{2t} &= -sin(\theta_t)a_{1t} + cos(\theta_t)a_{2t} + w_{2t} \\ \eta_t &= \psi_t - \psi_{t-1} + w_{3t} \end{aligned}$$

Give θ_t, ψ_t, M_t, **the rest of the system is linear and Gaussian. Hence,**

- θ_t, ψ_t, M_t **are the NLNG stat components**
- η_t **is the NLNG observation component.**

Extended Mixture Kalman Filter:

Let $\boldsymbol{y}_t = (y_1, \ldots, y_t)$**,** $\boldsymbol{y}_t^* = (y_1^*, \ldots, y_t^*)$ **and** $\boldsymbol{x}_t = (x_1, \ldots, x_t)$**.**

Note that

$$\begin{aligned} p(x_t, x_t^* \,|\, \boldsymbol{y}_t, \boldsymbol{y}_t^*) &= \int p(x_t, x_t^*, \boldsymbol{x}_{t-1} \,|\, \boldsymbol{y}_t, \boldsymbol{y}_t^*) d\boldsymbol{x}_{t-1} \\ &= \int p(x_t^* \mid \boldsymbol{x}_t, \boldsymbol{y}_t^*) p(x_t \,|\, \boldsymbol{x}_{t-1}, y_t, y_t^*) dF(\boldsymbol{x}_{t-1} \mid \boldsymbol{y}_{t-1}, \boldsymbol{y}_{t-1}^*) \end{aligned}$$

where

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

where

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

can be obtained from Kalman filter.

$p(x_t^* \mid \boldsymbol{y}_t, \boldsymbol{y}_t^*)$ **is a mixture Gaussian distribution.**

Inference with EMKF

$$E(f_1(x_t) \mid \boldsymbol{y}_t, \boldsymbol{y}_t^*) \approx \sum_{i=1}^{m} w_t^{(i)} f_1(x_t^{(i)})$$

and

$$E(f_2(x_t^*) \mid \boldsymbol{y}_t, \boldsymbol{y}_t^*) \approx \sum_{i=1}^{m} w_t^{(i)} \int f_2(x^*)\phi(x^*; \mu_t(\boldsymbol{x}_t^{(i)}), \sigma_t^2(\boldsymbol{x}_t^{(i)}))dx^*$$

Specially,

$$E(x_t^* \mid \boldsymbol{y}_t, \boldsymbol{y}_t^*) \approx \sum_{i=1}^{m} w_t^{(i)} \mu_t(\boldsymbol{x}_t^{(i)})$$

Benefit: improved efficiency

$$Var[f_2(x_t^*) \mid \boldsymbol{y}_t, \boldsymbol{y}_t^*] \geq Var[E(f_2(x_t^*) \mid \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{y}_t^* \mid \boldsymbol{y}_t, \boldsymbol{y}_t^*)]$$

Algorithm:

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

For $t+1$,

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

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

(3) : calculate the incremental weight

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

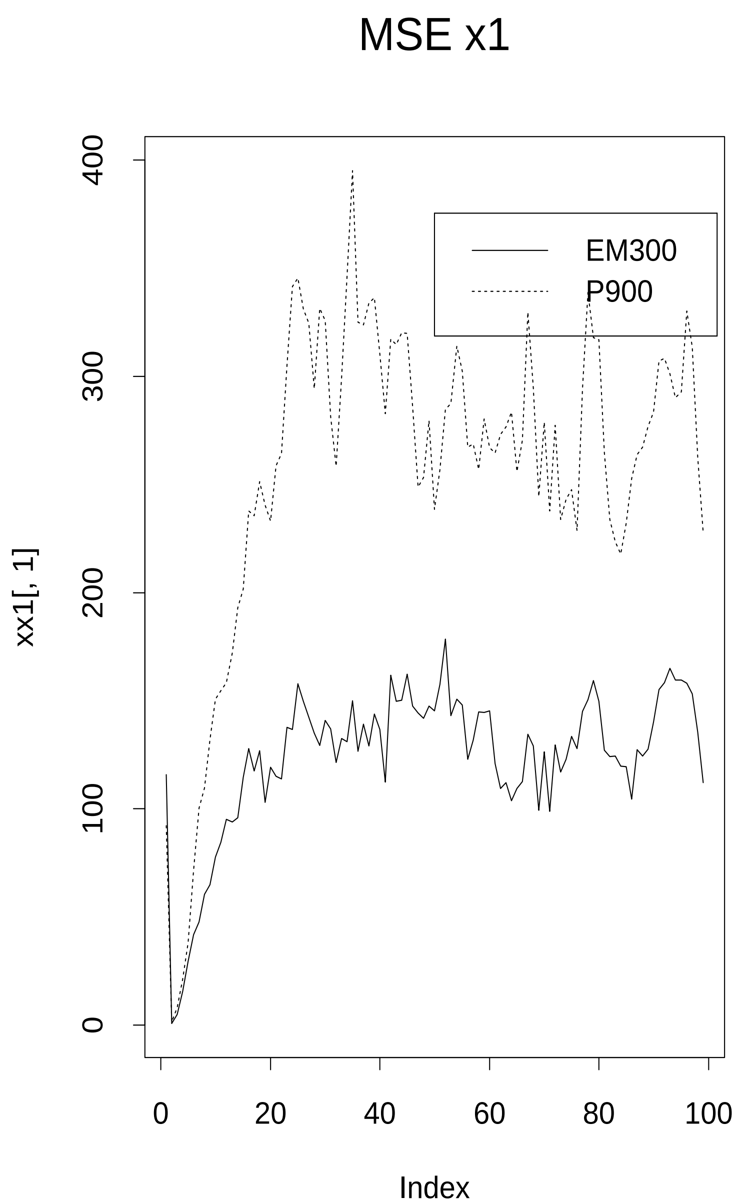
and the new weight $w_{t+1}^{(i)} = w_t^{(i)} u_{t+1}^{(i)}$.

t		$t+1$		$t+1$
$p(x_t \mid \boldsymbol{y}_t, \boldsymbol{y}^*_t)$		$g(x_{t+1} \mid \boldsymbol{y}_t, y_{t+1}, \boldsymbol{y}^*_t, y^*_{t+1})$		$p(x_{t+1} \mid \boldsymbol{y}_t, y_{t+1}, \boldsymbol{y}^*_t, y^*_{t+1})$
	$\searrow \quad \nearrow$		$\searrow \quad \nearrow$	
	$g_t(\cdot \mid x_t, y_{t+1}, y^*_{t+1})$		$w_{t+1} = w_t u_{t+1}$	
$\cdots\cdots\cdots\cdots\cdots$	$\cdots$	$\cdots\cdots\cdots\cdots\cdots\cdots$	$\cdots$	$\cdots\cdots\cdots\cdots\cdots\cdots$
$(x_t^{(1)}, KF_t^{(1)}, w_t^{(1)})$	$\longrightarrow$	$(x_{t+1}^{(1)}, KF_{t+1}^{(1)})$	$\longrightarrow$	$(x_{t+1}^{(1)}, KF_{t+1}^{(1)}, w_{t+1}^{(1)})$
$\vdots$		$\vdots$		$\vdots$
$(x_t^{(m)}, KF_t^{(m)}, w_t^{(m)})$	$\longrightarrow$	$(x_{t+1}^{(m)}, KF_{t+1}^{(m)})$	$\longrightarrow$	$(x_{t+1}^{(m)}, KF_{t+1}^{(m)}, w_{t+1}^{(m)})$

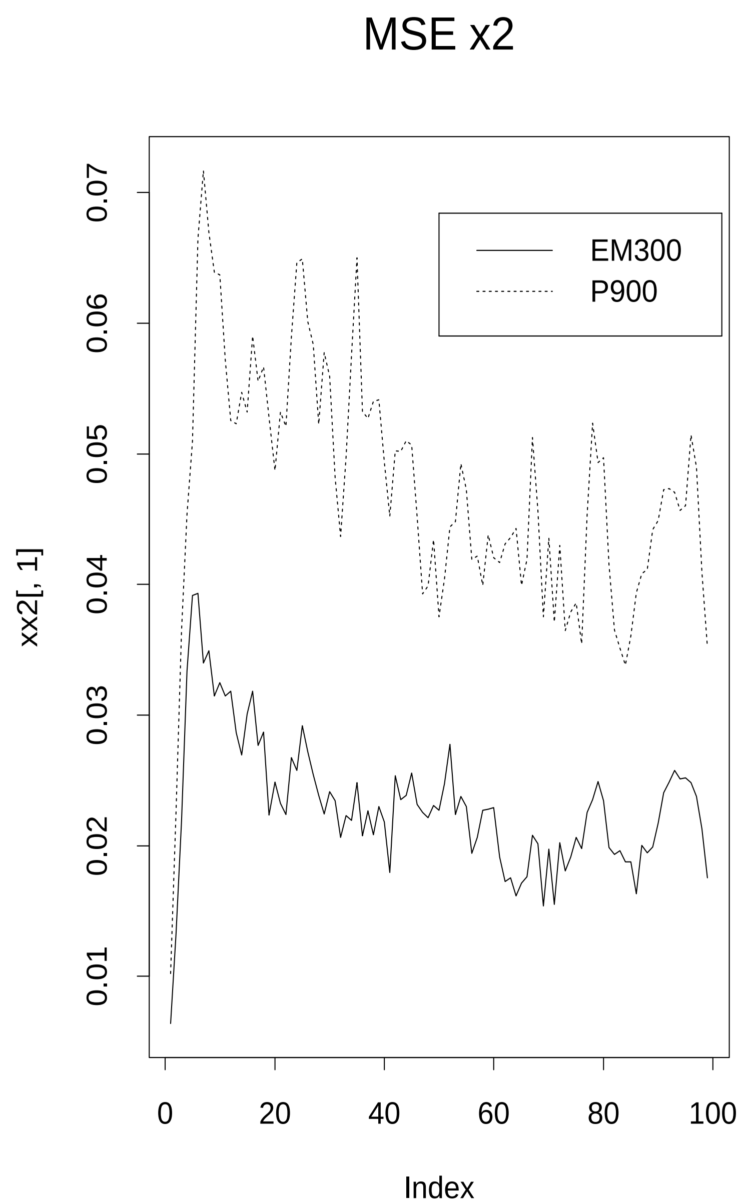
Simple example:

$$\begin{aligned} x_{t,1} &= x_{t-1,2} + 0.2x_{t-1,2}^2 + q_1 e_{t1} \\ x_{t,2} &= x_{t-1,2} + q_2 e_{t2} \\ y_t &= 0.5 x_{t,1} x_{t,2} + r v_t \end{aligned}$$

with e_{t1}**,** e_{t2}, v_t **all** $N(0,1)$



MSE x1
EM300
P900
xx1[, 1]
0
100
200
300
400
0
20
40
60
80
100
Index



MSE x2
EM300
P900
xx2[, 1]
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
20
40
60
80
100
Index

Example: Digital Signal Extraction in Fading Channels

α_t: Butterworth filter of order $r = 3$ i.e. ARMA(3,3)

Cutoff frequency 0.1

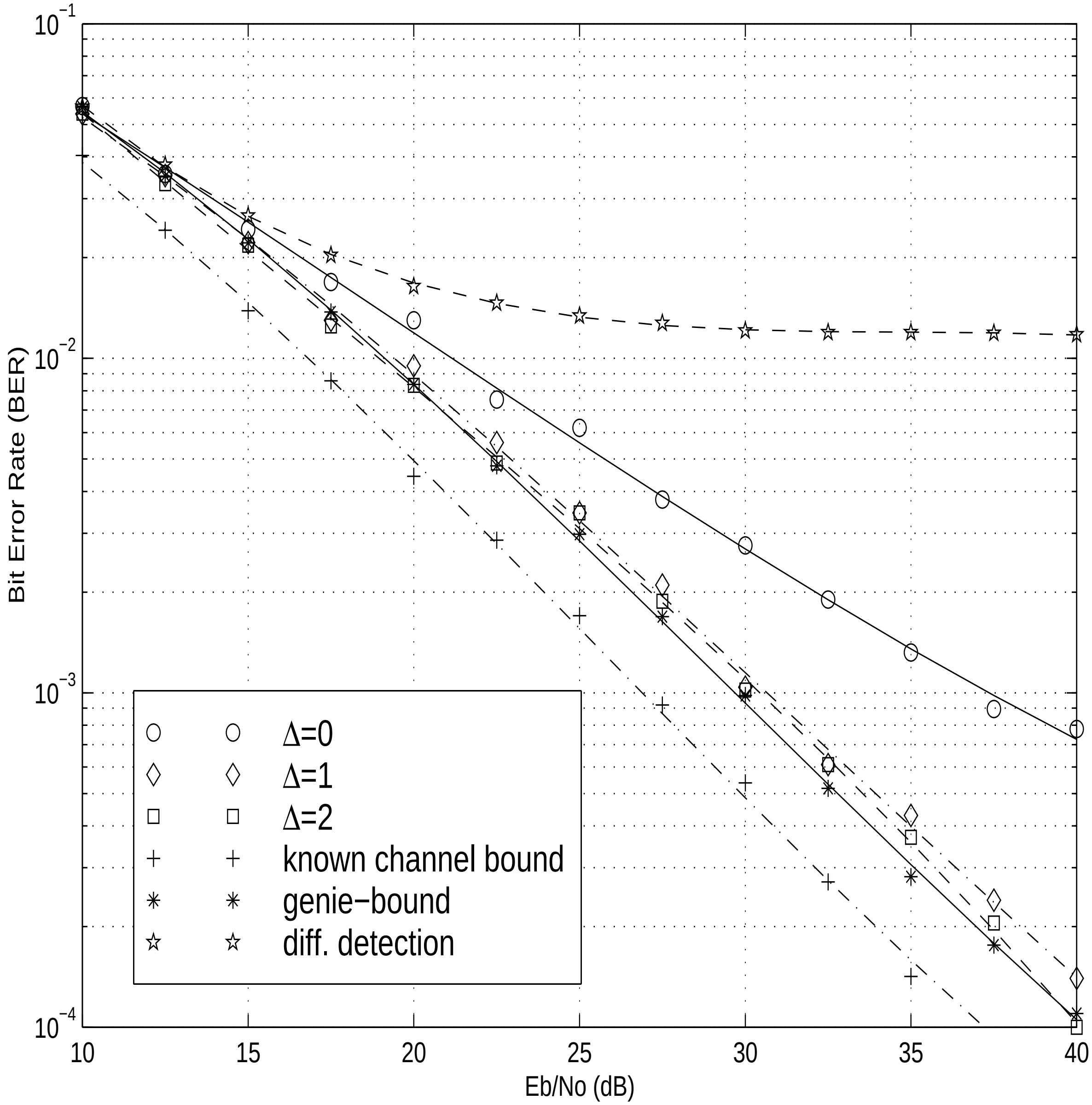
$s_t = \{-1, 1\}$.

Two cases:

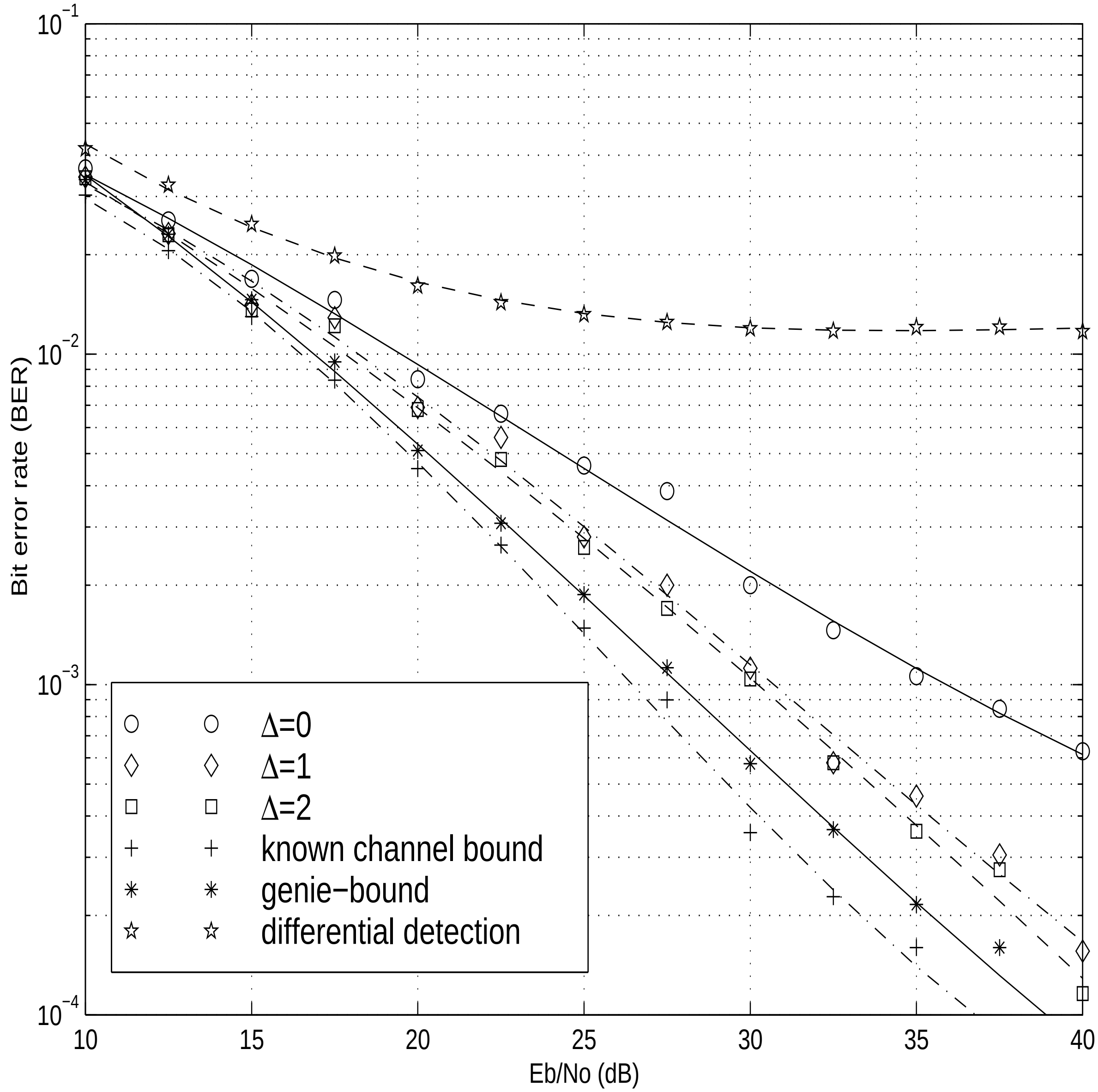
$v_t \sim N(0, \sigma^2)$

$v_t \sim (1 - \alpha)N(0, \sigma_1^2) + \alpha N(0, \sigma_2^2)$

Extra indicator I_t for noise.



Bit Error Rate (BER)
Eb/No (dB)
10^−1
10^−2
10^−3
10^−4
10
15
20
25
30
35
40
Δ=0
Δ=1
Δ=2
known channel bound
genie−bound
diff. detection



10^{-1}
10^{-2}
10^{-3}
10^{-4}
Bit error rate (BER)
10
15
20
25
30
35
40
Eb/No (dB)
Δ=0
Δ=1
Δ=2
known channel bound
genie−bound
differential detection

What we have done:

- **Separate NLNG and Conditional LG components**
- **Nonlinear components are dealt with standard Monte Carlo Filters**
- **Use Kalman Filter for the conditional linear and Gaussian component.**

Further Improvement – reduce the number of NLNG components with approximation

- **Linear approximation of the nonlinear functions, as EKF.**
- **Mixture Gaussian approximation of the Non-Gaussian innovations, by introducing indicators**
  - **conditional on the indicator, innovation is Gaussian**
  - t **distribution, double exponential, exponential power family, logistic, etc. and the mixture of them!**