Monte Carlo and Empirical Methods for Stochastic Inference (MASM11/FMSN50)

Magnus Wiktorsson
Centre for Mathematical Sciences
Lund University, Sweden

Lecture 7
Sequential Monte Carlo methods III
February 11, 2020

Plan of today's lecture

1. Last time: Sequential importance sampling (SIS)
    - SIS in a nutshell
    - Example: filtering in HMMs
2. Sequential importance sampling with resampling (SISR)
    - SIS + multinomial selection = SISR
    - Alternative selection strategies
    - A slide on convergence
3. Home assignment 2 (HA2)

We are here $\longrightarrow \bullet$

1. Last time: Sequential importance sampling (SIS)
   - SIS in a nutshell
   - Example: filtering in HMMs
2. Sequential importance sampling with resampling (SISR)
   - SIS + multinomial selection = SISR
   - Alternative selection strategies
   - A slide on convergence
3. Home assignment 2 (HA2)

We are here $\longrightarrow \bullet$

1. Last time: Sequential importance sampling (SIS)
   - SIS in a nutshell
   - Example: filtering in HMMs
2. Sequential importance sampling with resampling (SISR)
   - SIS + multinomial selection = SISR
   - Alternative selection strategies
   - A slide on convergence
3. Home assignment 2 (HA2)

Last time: Sequential MC problems

In the sequential MC framework, we aim at sequentially estimating sequences $(\tau_n)_{n\geq 0}$ of expectations

$$\tau_n = \mathbb{E}_{f_n}(\phi(X_{0:n})) = \int_{\mathsf{X}_n} \phi(x_{0:n}) f_n(x_{0:n}) \, \mathrm{d}x_{0:n}$$

over spaces X_n of increasing dimension, where the densities (f_n) are known up to normalizing constants only, i.e., for every $n \geq 0$,

$$f_n(x_{0:n}) = \frac{z_n(x_{0:n})}{c_n},$$

where c_n is an unknown constant.

Last time: Sequential importance sampling (SIS)

To derive the SIS algorithm we proceeded recursively. Assume that we have generated particles $(X_i^{0:n})$ from $g_n(x_{0:n})$ so that

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n}) \approx \mathbb{E}_{f_n}(\phi(X_{0:n})),$$

where, as usual, $\omega_n^i = \omega_n(X_i^{0:n}) = z_n(X_i^{0:n})/g_n(X_i^{0:n})$.

Key trick: Choose an instrumental distribution satisfying

$$g_{n+1}(x_{0:n+1}) = g_{n+1}(x_{n+1}|x_{0:n})g_n(x_{0:n}).$$

Last time: SIS

Consequently, $X_i^{0:n+1}$ and ω_{n+1}^i can be generated by

- keeping the previous $X_i^{0:n}$,
- simulating $X_i^{n+1} \sim g_{n+1}(x_{n+1} | X_i^{0:n})$,
- setting $X_i^{0:n+1} = (X_i^{0:n}, X_i^{n+1})$, and
- computing

$$\begin{aligned}
\omega_{n+1}^i &= \frac{z_{n+1}(X_i^{0:n+1})}{g_{n+1}(X_i^{0:n+1})} \\
&= \frac{z_{n+1}(X_i^{0:n+1})}{z_n(X_i^{0:n}) g_{n+1}(X_i^{n+1} | X_i^{0:n})} \times \frac{z_n(X_i^{0:n})}{g_n(X_i^{0:n})} \\
&= \frac{z_{n+1}(X_i^{0:n+1})}{z_n(X_i^{0:n}) g_{n+1}(X_i^{n+1} | X_i^{0:n})} \times \omega_n^i.
\end{aligned}$$

Last time: SIS

So, SIS updates the estimator

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n}) \approx \mathbb{E}_{f_n}(\phi(X_{0:n}))$$

to the estimator

$$\sum_{i=1}^{N} \frac{\omega_{n+1}^i}{\sum_{\ell=1}^{N} \omega_{n+1}^\ell} \phi(X_i^{0:n+1}) \approx \mathbb{E}_{f_{n+1}}(\phi(X_{0:n+1}))$$

by only adding a component X_i^{n+1} to $X_i^{0:n}$ and sequentially updating the weights. The algorithm is initialized by standard importance sampling of τ_0. We note that for each n, an unbiased estimate of c_n can, as usual, be obtained as

$$\frac{1}{N} \sum_{i=1}^{N} \omega_n^i \approx c_n.$$

We are here $\longrightarrow \bullet$

1. Last time: Sequential importance sampling (SIS)
   - SIS in a nutshell
   - Example: filtering in HMMs
2. Sequential importance sampling with resampling (SISR)
   - SIS + multinomial selection = SISR
   - Alternative selection strategies
   - A slide on convergence
3. Home assignment 2 (HA2)

HMMs

Graphically:

$$\begin{tikzcd} & Y_{k-1} & Y_k & Y_{k+1} & & \text{(Observations)} \\ \cdots \arrow[r] & X_{k-1} \arrow[u] \arrow[r] & X_k \arrow[u] \arrow[r] & X_{k+1} \arrow[u] \arrow[r] & \cdots & \text{(Markov chain)} \end{tikzcd}$$

$$Y_k | X_k = x_k \sim p(y_k | x_k) \qquad \text{(Observation density)}$$

$$X_{k+1} | X_k = x_k \sim q(x_{k+1} | x_k) \qquad \text{(Transition density)}$$

$$X_0 \sim \chi(x_0) \qquad \text{(Initial distribution)}$$

Linear/Gaussian HMM

Consider the linear HMM

$$\begin{aligned} Y_k &= X_k + S\varepsilon_k, & &\sim p(y_k|x_k) \\ X_{k+1} &= AX_k + R\eta_{k+1}, & &\sim q(x_{k+1}|x_k) \\ X_0 &= R/\sqrt{1-A^2}\eta_0, & &\sim \chi(x_0) \end{aligned}$$

where $|A| < 1$ and (η_k) and (ε_k) are independent standard Gaussian variables.

Given *fixed* observations (y_k), we want to estimate sequentially the filter means

$$\begin{aligned} \tau_n &= \mathbb{E}(X_n|Y_{0:n} = y_{0:n}) \\ &= \int \underbrace{x_n}_{\phi(x_{0:n})} \underbrace{\frac{\chi(x_0)p(y_0|x_0)\prod_{k=0}^{n-1} q(x_{k+1}|x_k)p(y_{k+1}|x_{k+1})}{L_n(y_{0:n})}}_{z_n(x_{0:n})/c_n} \, \mathrm{d}x_{0:n}. \end{aligned}$$

Linear/Gaussian HMM, SIS implementation

To obtain a SIS implementation, we set

$$g_{n+1}(x_{n+1}|x_{0:n}) = q(x_{n+1}|x_n) = \mathcal{N}(x_{k+1}; Ax_n, R^2),$$

implying

$$\begin{aligned}
&\omega_{n+1}^i = \\
&= \frac{z_{n+1}(X_i^{0:n+1})}{z_n(X_i^{0:n})g_{n+1}(X_i^{n+1}|X_i^{0:n})} \times \omega_n^i \\
&= \frac{\chi(X_i^0)p(y_0|X_i^0)\prod_{k=0}^n q(X_i^{k+1}|X_i^k)p(y_{k+1}|X_i^{k+1})}{\chi(X_i^0)p(y_0|X_i^0)\{\prod_{k=0}^{n-1} q(X_i^{k+1}|X_i^k)p(y_{k+1}|X_i^{k+1})\}q(X_i^{n+1}|X_i^n)} \times \omega_n^i \\
&= p(y_{n+1}|X_i^{n+1}) \times \omega_n^i \\
&= \mathcal{N}(y_{n+1}; X_i^{n+1}, S^2) \times \omega_n^i.
\end{aligned}$$

Linear/Gaussian HMM, SIS implementation

This gives the following scheme.

Assume that

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} X_i^n \approx \mathbb{E}(X_n | Y_{0:n} = y_{0:n});$$

then, for $i = 1, 2, \ldots, N$,

- draw $X_i^{n+1} \sim \mathcal{N}(AX_i^n, R^2)$,
- set $\omega_{n+1}^i = \mathcal{N}(y_{n+1}; X_i^{n+1}, S^2) \times \omega_n^i$,

yielding the approximation

$$\sum_{i=1}^{N} \frac{\omega_{n+1}^i}{\sum_{\ell=1}^{N} \omega_{n+1}^\ell} X_i^{n+1} \approx \mathbb{E}(X_{n+1} | Y_{0:n+1} = y_{0:n+1}).$$

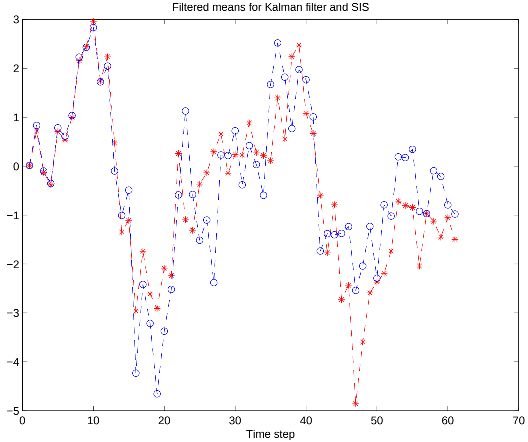
Linear/Gaussian HMM, SIS implementation

In Matlab:

```
N = 1000;
n = 60;
tau = zeros(1,n); % vector of estimates
p = @(x,y) normpdf(y,x,S); % observation density, for weights
part = R*sqrt(1/(1 - A^2))*randn(N,1); % initialization
w = p(part,Y(1));
tau(1) = sum(part.*w)/sum(w);
for k = 1:n, % main loop
    part = A*part + R*randn(N,1); % mutation
    w = w.*p(part,Y(k + 1)); % weighting
    tau(k + 1) = sum(part.*w)/sum(w); % estimation
end
```

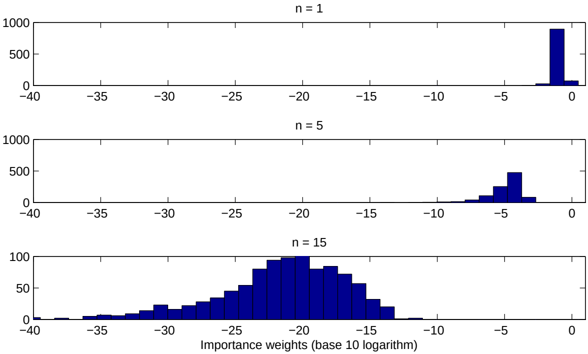
Linear/Gaussian HMM, SIS implementation

Comparison of SIS ($\circ$) with exact values ($*$) provided by the Kalman filter (possible only for linear/Gaussian models):



Linear/Gaussian HMM, SIS implementation

Distribution of importance weights: ☹



We are here $\longrightarrow \bullet$

We are here $\longrightarrow \bullet$

Multinomial resampling

A simple—but revolutionary!—idea: duplicate/kill particles with large/small weights! (Gordon *et al.*, 1993)

The most natural approach to such selection is to simply draw, with replacement, new particles $\tilde{X}_1^{0:n}, \tilde{X}_2^{0:n}, \ldots, \tilde{X}_N^{0:n}$ among the SIS-produced particles $X_1^{0:n}, X_2^{0:n}, \ldots, X_N^{0:n}$ with probabilities given by the normalized importance weights.

Formally, this amounts to setting, for $i = 1, 2, \ldots, N$,

$$\tilde{X}_i^{0:n} = X_j^{0:n} \text{ with probability } \frac{\omega_n^j}{\sum_{\ell=1}^N \omega_n^\ell}.$$

Criteria for good resampling strategies

Let N_n^i be the number of resampled copies of particle i.

- The total number of particles should remain constant.
- The weights should be set equal after resampling.
- It should hold that

$$\mathbb{E}[N_n^i | X^{0:n}] = N \frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell}, \; i = 1, 2, \cdots, N.$$

Assures that the resampling gives no additional bias.

Multinomial resampling (cont.)

After this, the resampled particles $(\tilde{X}_i^{0:n})$ are assigned equal weights $\tilde{\omega}_n^i = 1$, say, and we replace

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n}) \quad \text{by} \quad \frac{1}{N} \sum_{i=1}^{N} \phi(\tilde{X}_i^{0:n}).$$

Multinomial resampling does not add bias to the estimator:

Theorem

For all $N \geq 1$ and $n \geq 0$,

$$\mathbb{E}\left(\frac{1}{N} \sum_{i=1}^{N} \phi(\tilde{X}_i^{0:n})\right) = \mathbb{E}\left(\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n})\right).$$

The operation adds however some variance due to additional randomness.

Sequential importance sampling with resampling (SISR)

After selection, we proceed with standard SIS and move the selected particles $(\tilde{X}_i^{0:n})$ according to $g_{n+1}(x_{n+1}|x_{0:n})$.

The full scheme goes as follows. Given $(X_i^{0:n}, \omega_n^i)$,

- (*selection*) draw, with replacement, $(\tilde{X}_i^{0:n})$ among $(X_i^{0:n})$ according to probabilities $(\omega_n^i / \sum_{\ell=1}^N \omega_n^\ell)$
- (*mutation*) draw, for all i, $X_i^{n+1} \sim g_{n+1}(x_{n+1}|\tilde{X}_i^{0:n})$,
- set, for all i, $X_i^{0:n+1} = (\tilde{X}_i^{0:n}, X_i^{n+1})$, and
- set, for all i,

$$\omega_{n+1}^i = \frac{z_{n+1}(X_i^{0:n+1})}{z_n(\tilde{X}_i^{0:n}) g_{n+1}(X_i^{n+1}|\tilde{X}_i^{0:n})}.$$

SISR, estimation of τ_n and c_n

At every time step n, both

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n}) \quad \text{and} \quad \frac{1}{N} \sum_{i=1}^{N} \phi(\tilde{X}_i^{0:n})$$

are valid estimators of τ_n. The former has however somewhat lower variance. For estimation of normalizing constants using SISR, set

$$c_{N,n}^{\mathsf{SISR}} = \prod_{k=0}^{n} \left(\frac{1}{N} \sum_{i=1}^{N} \omega_k^i \right).$$

Theorem

For all $n \geq 0$ and $N \geq 1$,

$$\mathbb{E}\left(c_{N,n}^{\mathsf{SISR}}\right) = c_n.$$

Example: Linear/Gaussian HMM, SISR implementation

In Matlab:

```
N = 1000;
n = 60;
tau = zeros(1,n); % vector of filter means
w = zeros(N,1);
p = @(x,y) normpdf(y,x,S); % observation density, for weights
part = R*sqrt(1/(1 - A^2))*randn(N,1); % initialization
w = p(part,Y(1));
tau(1) = sum(part.*w)/sum(w);
for k = 1:n, % main loop
    part = A*part + R*randn(N,1); % mutation
    w = p(part,Y(k + 1)); % weighting
    tau(k + 1) = sum(part.*w)/sum(w); % estimation
    ind = randsample(N,N,true,w); % selection
    part = part(ind);
end
```

The Essential part of the randsample function

The Matlab (R2018b) function randsample has 177 lines of code but we actually could just use the essential part of the code

```
CW=cumsum([0 W]);
[~,ind] = histc(rand(1,N),CW/CW(end));
```

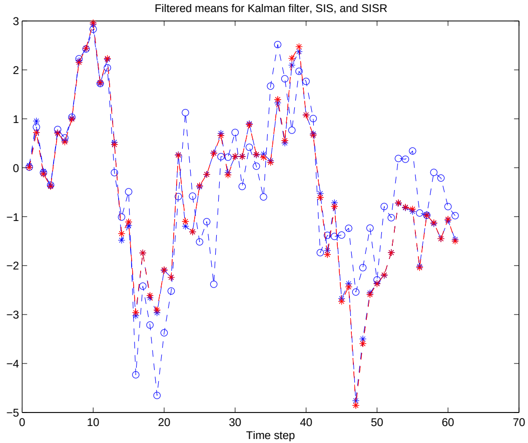
where histc is the builtin and very efficiently coded function for counting the number of points in each in bin and the indices of which bin we fall in for doing histograms.

We here use the normalised cumulative weightsum as bin boundaries and only look at the bin indices for points $\in \mathsf{U}(0,1)$. This gives us exactly the correct resampling distribution of the indicies ind.

It is also easy to modify the code to do a version of stratified resampling (see below).

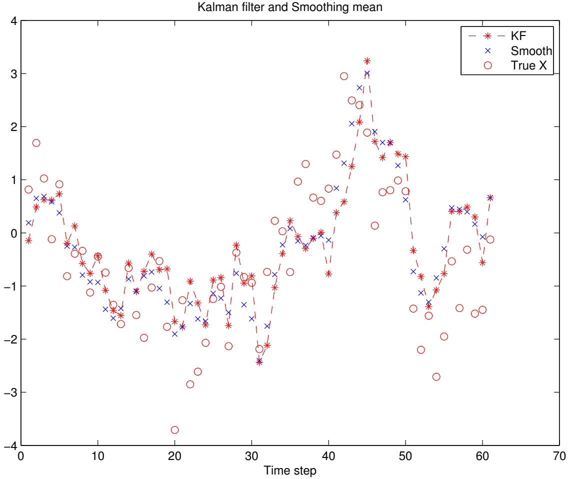
Example: Linear/Gaussian HMM, SIS implementation

Comparison of SIS (○) and SISR (∗, blue) with exact values (∗, red) provided by the Kalman filter:

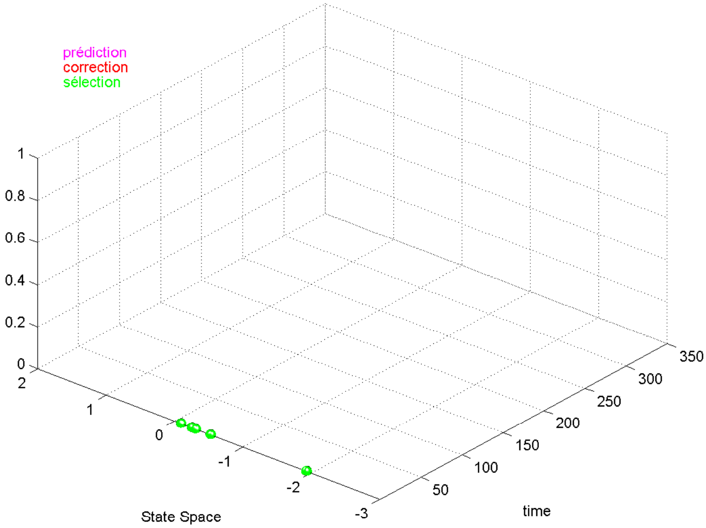


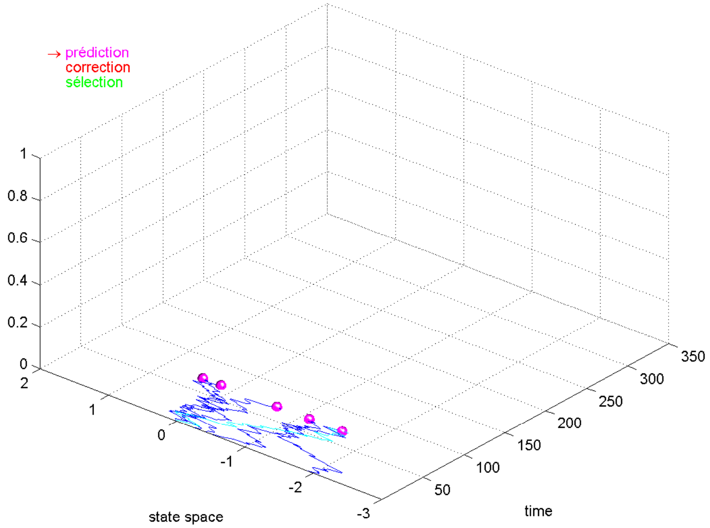
Example: Linear/Gaussian HMM, SIS implementation

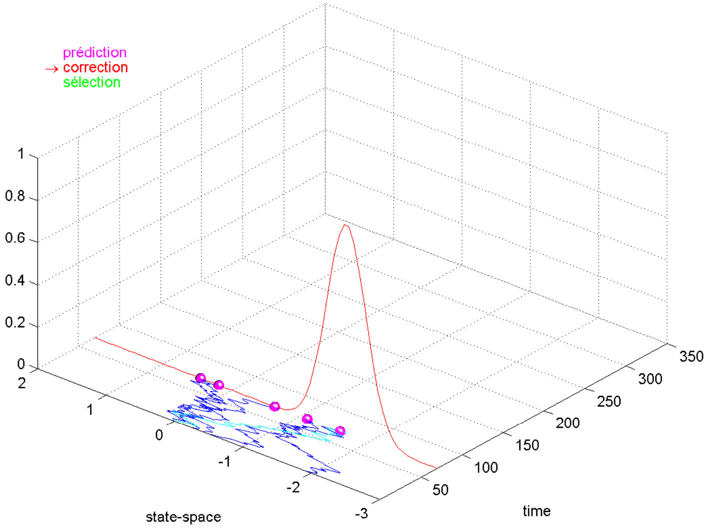
A comparison between Kalman filter $\mathbb{E}[X_k|Y_{0:k} = y_{0:k}]$ and Smoother $\mathbb{E}[X_k|Y_{0:n} = y_{0:n}]$ for $k = 0, 1, \cdots, n$:

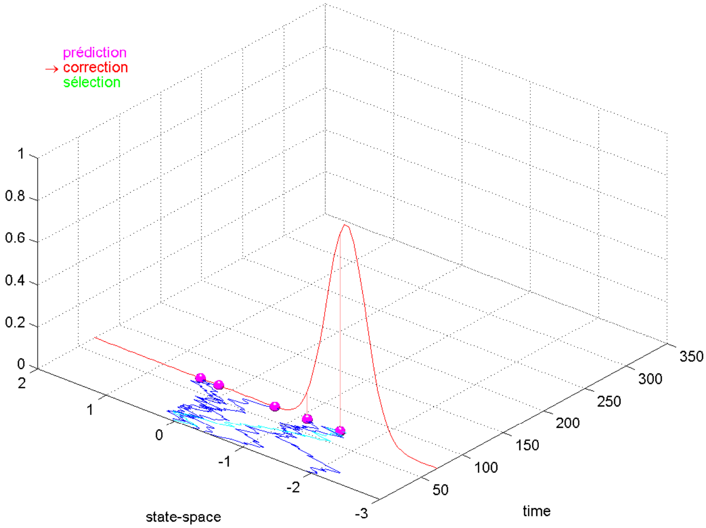


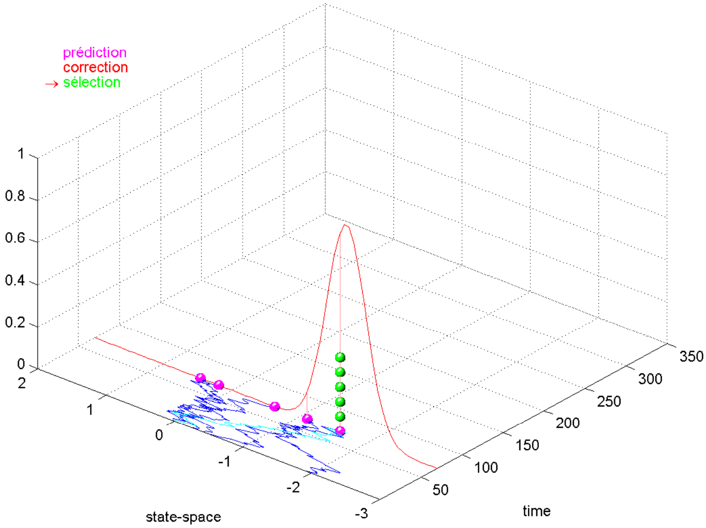
Film time! ☺

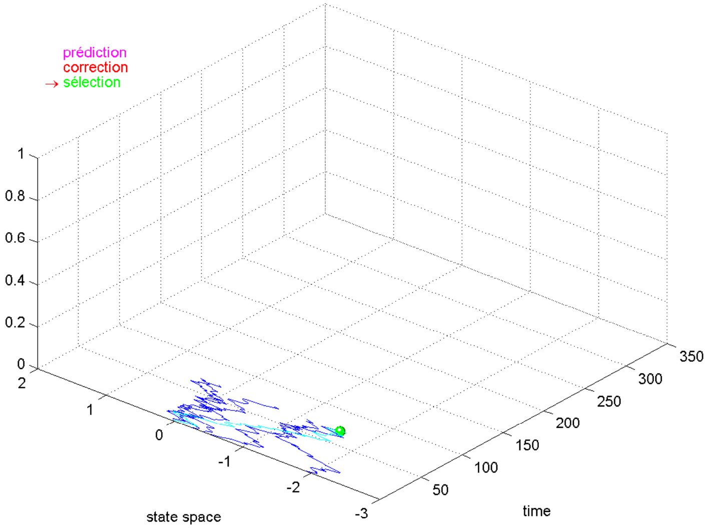

prédiction
correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
State Space
50
100
150
200
250
300
350
time

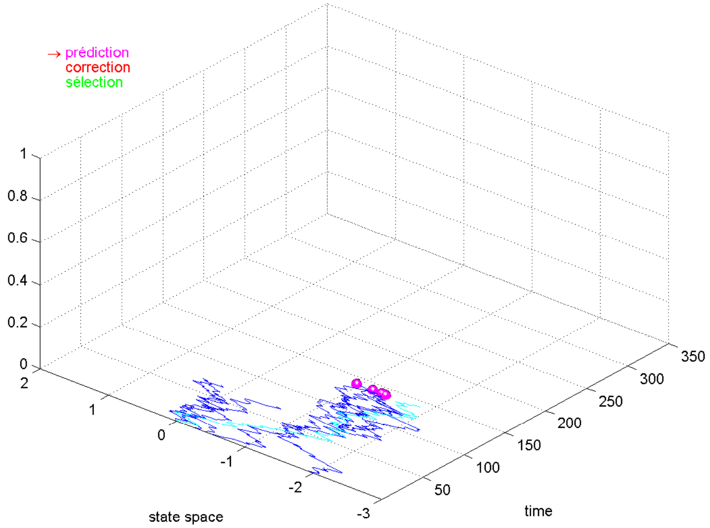

→ prédiction
correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

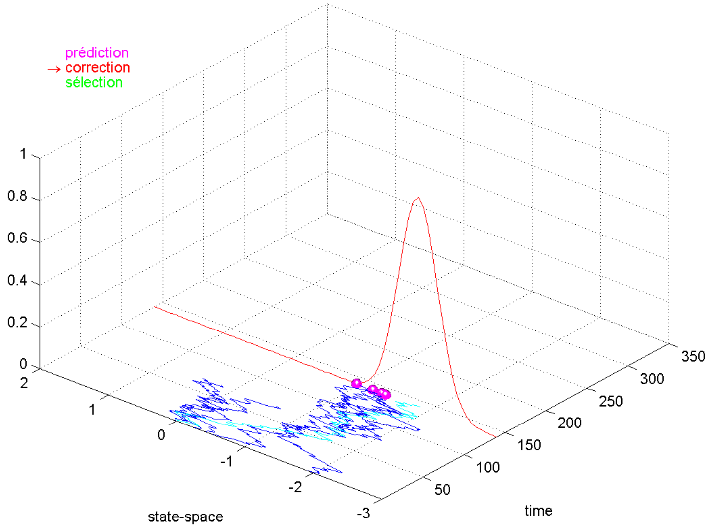

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

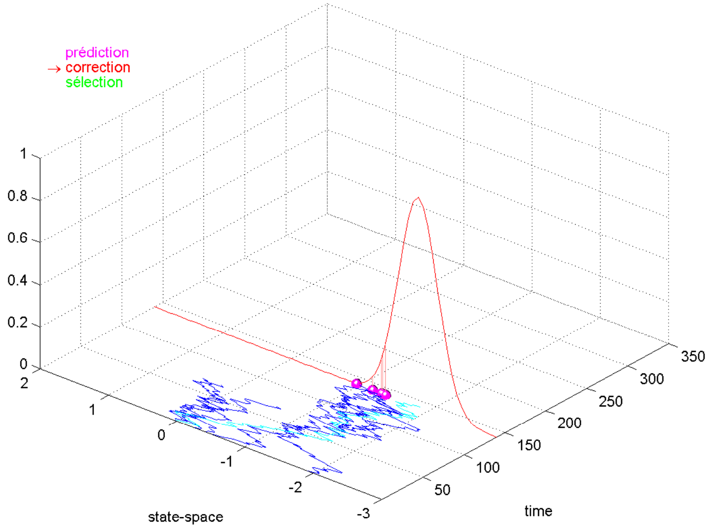

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

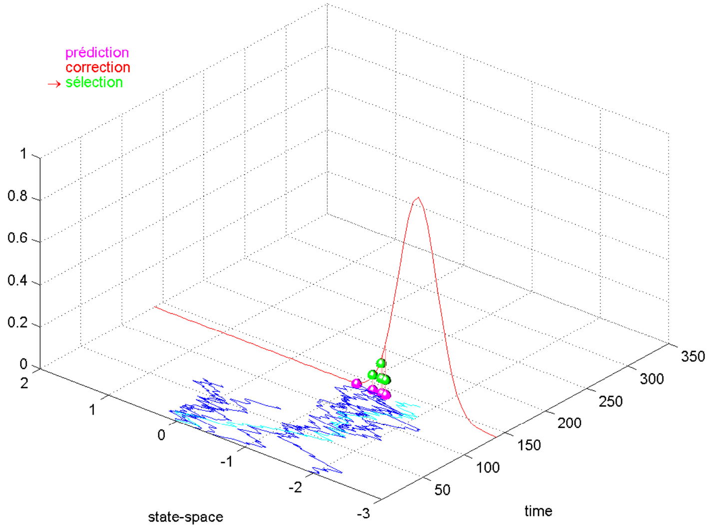

prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

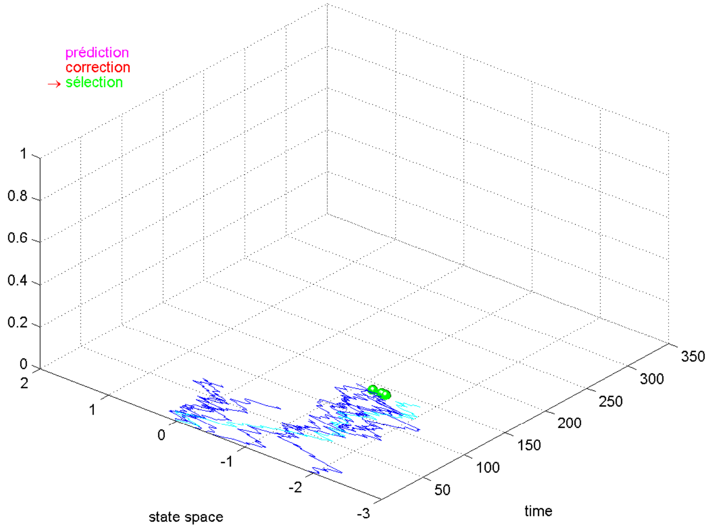

prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

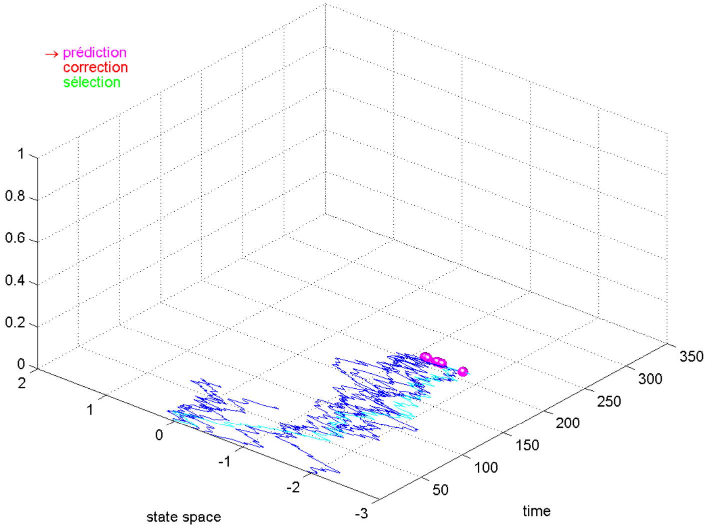

→ prédiction
correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

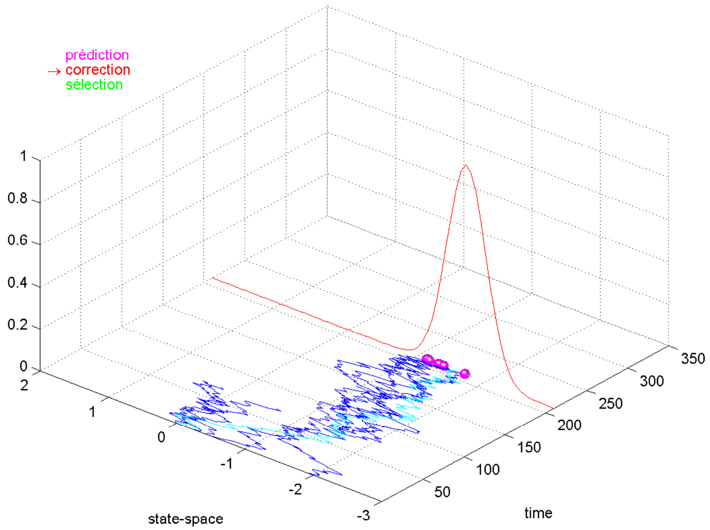

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

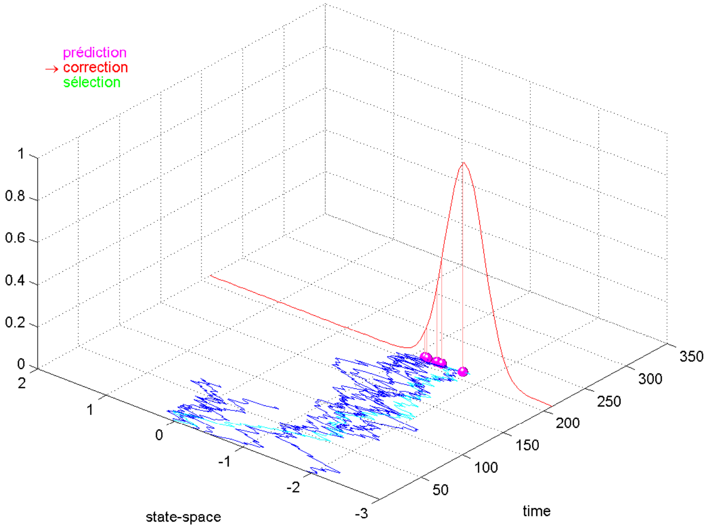

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

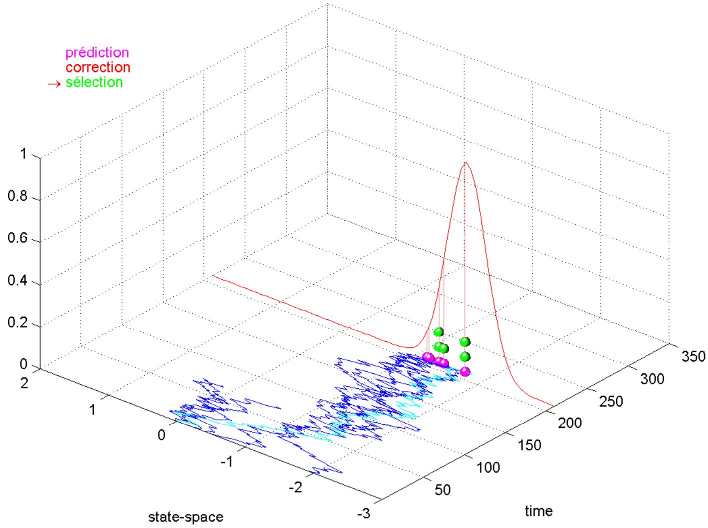

prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

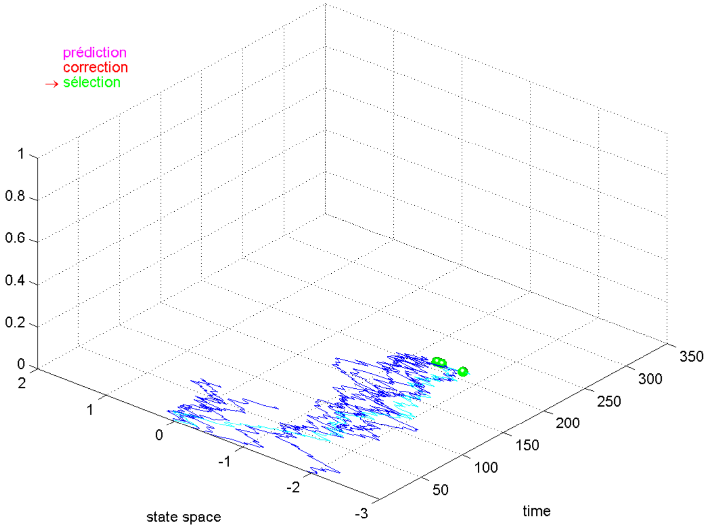

prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

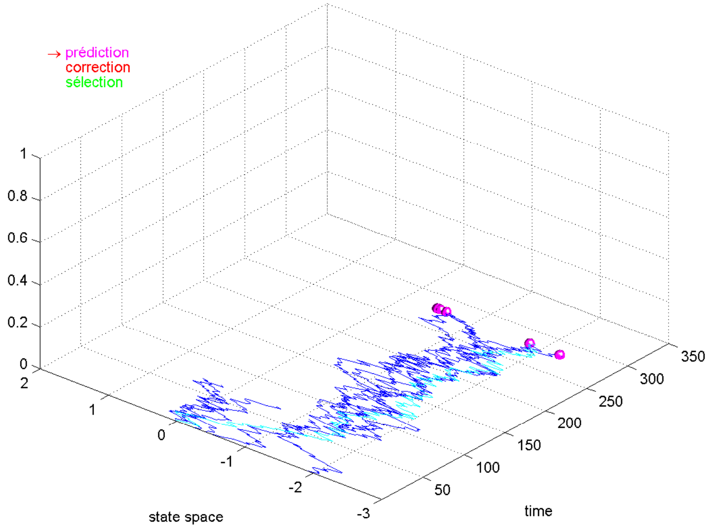

→ prédiction
correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

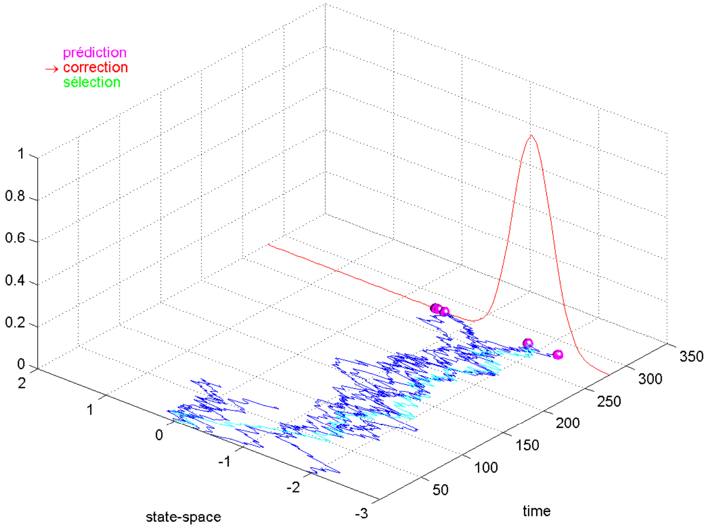

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

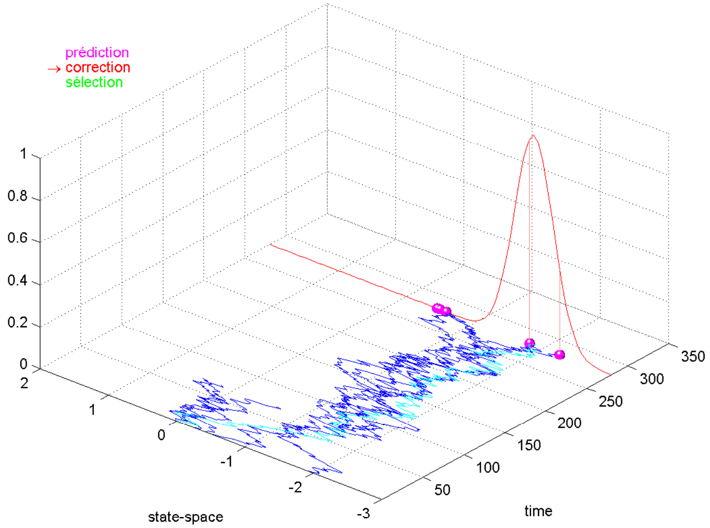

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

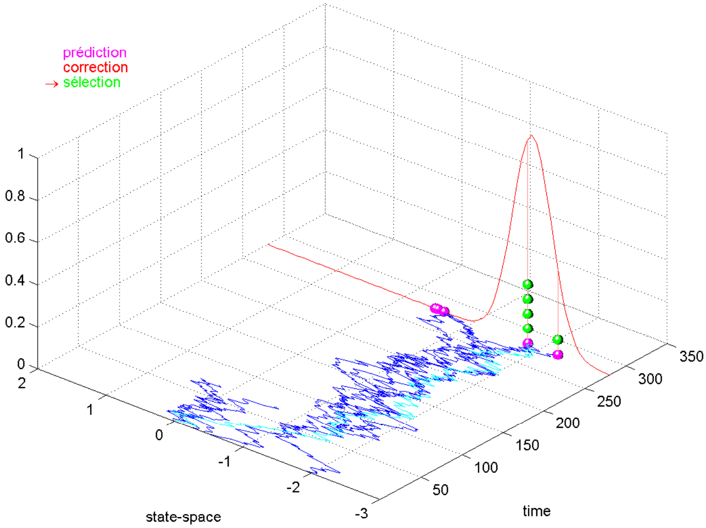

prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

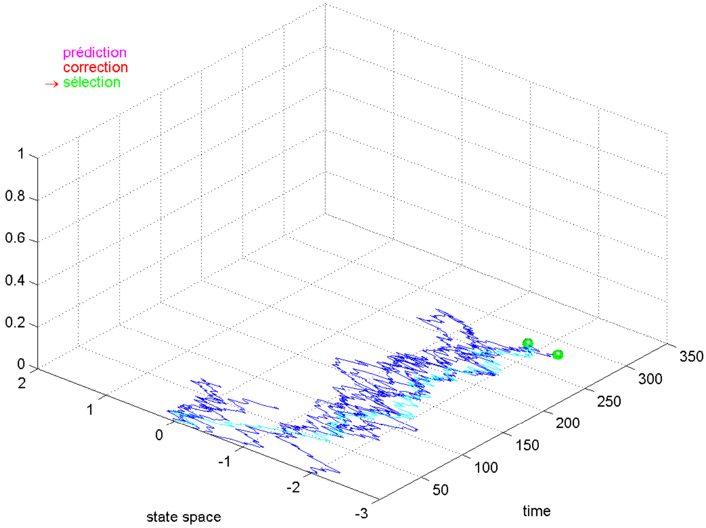

prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

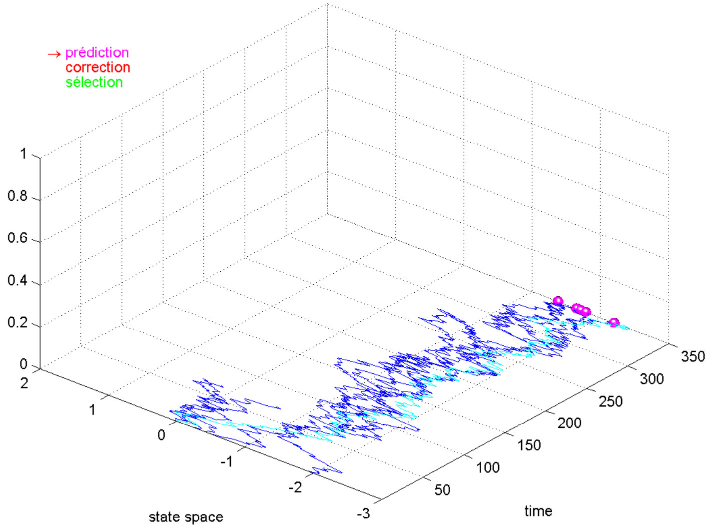

→ prédiction
correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time

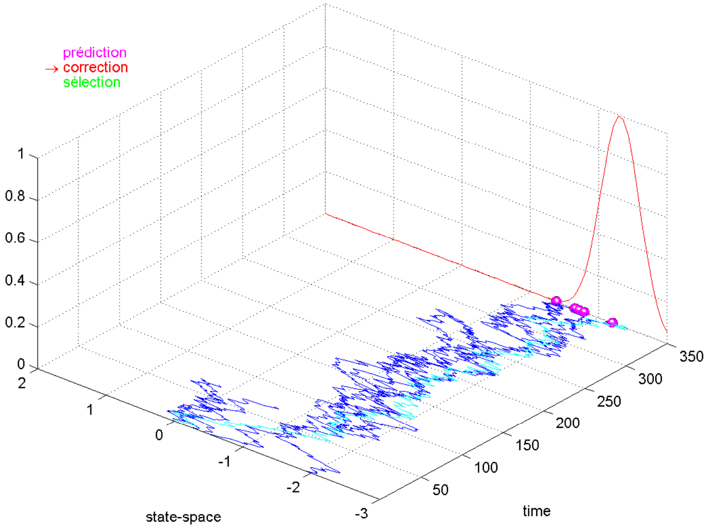

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

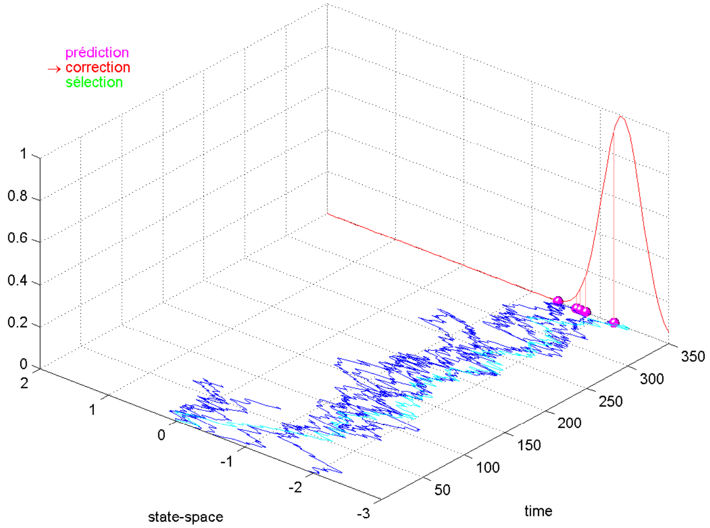

prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time


prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time


prédiction
correction
→ sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time


→ prédiction
correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state space
50
100
150
200
250
300
350
time


prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time


prédiction
→ correction
sélection
1
0.8
0.6
0.4
0.2
0
2
1
0
-1
-2
-3
state-space
50
100
150
200
250
300
350
time

We are here $\longrightarrow \bullet$

Residual resampling

There are several alternatives to multinomial selection. In the residual resampling scheme the number N_n^i of offspring of particle i is—"semi-deterministically"—set to

$$N_n^i = \left\lfloor N \frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell} \right\rfloor + \tilde{N}_n^i,$$

where the $\tilde{N}_n^i$'s are random integers obtained by randomly distributing the remaining $N - R$ offspring, with

$$R \stackrel{\text{def}}{=} \sum_{j=1}^N \left\lfloor N \frac{\omega_n^j}{\sum_{\ell=1}^N \omega_n^\ell} \right\rfloor,$$

among the ancestors as follows.

Residual resampling, pseudo-code

for $i = 1 \to N$ **do**

  set $\tilde{N}_n^i \leftarrow 0$

  set $\bar{\omega}_n^i \leftarrow \frac{1}{N-R}\left(N\frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell} - \left\lfloor N\frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell}\right\rfloor\right)$

end for

for $r = 1 \to N-R$ **do**

  set $I_r \leftarrow j$ with probability $\bar{\omega}_n^j$

  set $\tilde{N}_n^{I_r} \leftarrow \tilde{N}_n^{I_r} + 1$

end for

return $(\tilde{N}_n^i)$

Residual resampling, unbiasedness

Consequently, the residual resampling operation replaces the estimator

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n}) \quad \text{by} \quad \frac{1}{N} \sum_{i=1}^{N} N_n^i \phi(X_i^{0:n}).$$

Also residual resampling does not add to the bias (exercise!):

Theorem

For all $N \geq 1$ and $n \geq 0$,

$$\mathbb{E}\left(\frac{1}{N} \sum_{i=1}^{N} N_n^i \phi(X_i^{0:n})\right) = \mathbb{E}\left(\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_i^{0:n})\right).$$

One can also show that the variance of the estimator is smaller than the variance of the estimator obtained with multinomial selection.

Other selection strategies

Other selection strategies are

- Stratified resampling
- Bernoulli branching
- Poisson branching
- ...

Stratified resampling

A simple modification of the code gives stratified resampling instead!

```
CW=cumsum([0 W]);
[~,ind] = histc((rand(1,N)+(0:(N-1)))/N,CW/CW(end));
```

Compare with original multinomial resampling (as before)

```
CW=cumsum([0 W]);
[~,ind] = histc(rand(1,N),CW/CW(end));
```

We are here $\longrightarrow \bullet$

1. Last time: Sequential importance sampling (SIS)
   - SIS in a nutshell
   - Example: filtering in HMMs
2. Sequential importance sampling with resampling (SISR)
   - SIS + multinomial selection = SISR
   - Alternative selection strategies
   - A slide on convergence
3. Home assignment 2 (HA2)

SISR: Some theoretical results

Even though the theory of SISR is hard, there is a number of results establishing the convergence of the algorithm as N tends to infinity. For instance,

$$\sqrt{N}\left(\sum_{i=1}^{N}\frac{\omega_n^i}{\sum_{\ell=1}^{N}\omega_n^\ell}\phi(X_i^{0:n})-\tau_n\right)\stackrel{\text{d.}}{\longrightarrow}\mathcal{N}(0,\sigma_n^2),$$

where σ_n^2 in general lack a closed form expression. Thus, the convergence rate is still $\sqrt{N}$.

Open problem: find an online estimator of σ_n^2!

The dependence of σ_n^2 on n is crucial. However, for filtering in HMMs (particle filtering) one may prove, under weak assumptions,

$$\sigma_n^2 \leq c$$

for a constant $c<\infty$. Thus, the SISR estimates are stable in n.

A few references on SMC

- Cappé, O., Moulines, E., and Rydén, T. (2005) *Inference in Hidden Markov Models*. Springer.
- Douc, R., Cappe, O., Moulines, E. (2005) Comparison of Resampling Schemes for Particle Filtering. *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*
- Doucet, A., De Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*. Springer.
- Fearnhead, P. (1998) *Sequential Monte Carlo Methods in Filter Theory*. Ph.D. thesis, University of Oxford.

We are here $\longrightarrow \bullet$

HA2: Sequential Monte Carlo methods

HA2 deals exclusively with the self-avoiding walk (SAW) in $\mathbb{Z}^d$. Let $c_n(d)$ be the number of SAW:s of length n in dimension d. The home assignment has the following tasks:

- two minor theoretical questions on the asymptotics of the number $c_n(d)$ of SAW:s as the length n tends to infinity,
- two questions dealing with SIS-based approaches to estimation of $c_n(2)$,
- two questions dealing with an SISR-based approach to estimation of $c_n(2)$ and A_2, μ_2 and γ_2, and
- three additional questions dealing with SAW:s in $\mathbb{Z}^d$ for $d \geq 3$.

Submission of HA2

As for HA1, the following is to be submitted:

- A written report in PDF format (No MS Word-files).
- Upload in CANVAS HA2 before Tuesday 25 Feb, 13:00:00 (that is, 15 minutes before the beginning of the lecture).The uploaded files should include the report file as well as *all* your m-files with a file `proj2.m` that runs your analysis.
- Late submissions do not qualify for marks higher than 3.