Multi-Target Tracking
using
MCMC-Based Particle Algorithm

François Septier
Joint work with Simon Godsill, Sze Kim Pang & Julien Cornebise

SMC Transition Workshop
9th November 2009
**Aim**: Detect, track and identify the targets from a sequence of noisy observations.

⇒ Estimate sequentially a state $x_k$ → Targets’ position and velocity

**Bayesian Inference** ←→ Compute the filtering posterior distribution

$$p(x_k | z_{1:k}) = \frac{p(z_k | x_k)p(x_k | z_{1:k-1})}{p(z_k | z_{1:k-1})}$$

Bayes’ Rule

with $p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1})p(x_{k-1} | z_{1:k-1})dx_{k-1}$

This filtering posterior distribution is analytically intractable

⇒ Approximate methods ←→ Monte Carlo methods
Sequential Monte-Carlo methods (SMC) can be used. **But** difficult to design an efficient particle filter → due to the high dimensional system

**Alternatives to SMC methods** → MCMC :

~~ more effective in high-dimensional systems,

~~ more flexible : a lot of different sampling strategies can be used.

**Traditionally**, MCMC methods → Non-sequential setting

**But** several **Sequential** Markov Chain Monte-Carlo (MCMC) methods exist !

**Remark** :
These approaches are distinct from the Resample-Move algorithm [Gilks & Berzuini, 2001]

← Use neither resampling nor importance sampling.
**Sequential MCMC: Targeting the filtering posterior distribution**

**Aim:** Approximate, at time $t_k$, the filtering posterior distribution given by

$$p(x_k|z_{1:k}) \propto p(z_k|x_k) \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}$$

⇒ Use a MCMC procedure to produce samples from this distribution.

*Problem* → No closed form expression for \( \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \)

**Idea** [Khan & Blach & Dellaert, 2005]

← Use a set of unweighted particles to represent the posterior density at time $t_{k-1}$

$$p(x_{k-1}|z_{1:k-1}) \approx \frac{1}{N_P} \sum_{j=1}^{N_P} \delta(x_{k-1} - x^{(j)}_{k-1})$$

⇒ So, the filtering posterior becomes:

$$p(x_k|z_{1:k}) \propto p(z_k|x_k) \sum_{j=1}^{N_P} p(x_k|x^{(j)}_{k-1})$$

**A MCMC scheme can now be used with target distribution:**

\( p(z_k|x_k) \sum_{j=1}^{N_P} p(x_k|x^{(j)}_{k-1}) \)

The accepted samples from this MCMC approximates \( p(x_k|z_{1:k}) \approx \frac{1}{N_p} \sum_{j=1}^{N_P} \delta(x_k - x^{(j)}_{k}) \)

**Inconvenient:** Computational demand can become extensive as the number of particles increases

← Monte Carlo approximation of the predictive posterior distribution (\( \mathcal{O}(N_p) \) per MCMC iteration)
To avoid numerical integration of the predictive density at every MCMC iteration

$\Rightarrow$ An alternative algorithm has been developed in [Septier & Pang & Godsill & Carmi, 2009]

$\Rightarrow$ Generalization of the features of some existing sequential MCMC algorithms

[Berzuini & Best & Gilks & Larizza, 1997], [Golightly & Wilkinson, 2006]

**Idea:** Approximate, at time $t_k$, the joint (instead of the filtering) posterior distribution given by

$$p(x_k, x_{k-1} | z_{1:k}) \propto p(z_k | x_k) p(x_k | x_{k-1}) p(x_{k-1} | z_{1:k-1})$$

$\Rightarrow$ Use a MCMC procedure to produce samples from this distribution by using the following approximation

$$p(x_{k-1} | z_{1:k-1}) \approx \frac{1}{NP} \sum_{j=1}^{NP} \delta(x_{k-1} - x_{k-1}^{(j)})$$

$\Rightarrow$ The joint posterior distribution can now be written as:

$$p(x_k, x_{k-1} | z_{1:k}) \propto p(z_k | x_k) \sum_{j=1}^{NP} p(x_k | x_{k-1} = x_{k-1}^{(j)}) \delta(x_{k-1} - x_{k-1}^{(j)})$$

$\leftarrow$ Target distribution of the MCMC scheme
Sequential MCMC: Targeting the joint posterior distribution

To avoid numerical integration of the predictive density at every MCMC iteration

An alternative algorithm has been developed in [Septier & Pang & Godsill & Carmi, 2009]

→ Generalization of the features of some existing sequential MCMC algorithms
  [Berzuini & Best & Gilks & Larizza, 1997], [Golightly & Wilkinson, 2006]

Idea: Approximate, at time $t_k$, the joint (instead of the filtering) posterior distribution given by

$$p(x_k, x_{k-1}|z_{1:k}) \propto p(z_k|x_k) \sum_{j=1}^{NP} p(x_k|x_{k-1} = x_{k-1}^{(j)}) \delta(x_{k-1} - x_{k-1}^{(j)})$$

MCMC Procedure to obtain samples from this joint distribution

1 $\Rightarrow$ Make a joint draw for $\{x_k, x_{k-1}\}$ using a Metropolis-Hastings step,

2 $\Rightarrow$ Refine individually $x_k$ and $x_{k-1}$ using Metropolis-Hastings-within-Gibbs steps,

As a Comparison:

☆ [Berzuini & Best & Gilks & Larizza, 1997] made use of only the individual refinement step (step 2)
  ⇒ Poor mixing in high-dimensional problems due to the highly-disjoint predictive density of the
  particle representation

☆ [Golightly & Wilkinson, 2006] made use of only the joint draw (step 1)
  ⇒ Reduce the effectiveness of the MCMC as refinement moves are not employed to explore
  the structured probabilistic space

In [Septier & Pang & Carmi & Godsill & 2009] → Evolutionary extension of this algorithm
  (incorporates several attractive features of genetic algorithms and simulated annealing)
Targets can appear or disappear from the scene randomly over time

⇒ \( x_k \) is a variable dimension quantity!

In this work, we choose equivalently to model this birth and death process by a set of existence variables, \( e_k \)

\( \leftarrow \) each \( e_{k,i} \in \{0, 1\} \) models inactive or active target

⇒ \( x_k \) is regarded as a fixed dimensional quantity with \( N_{max} \) elements:

\[
\begin{bmatrix}
    x_{k,1} \\
    x_{k,2} \\
    \vdots \\
    x_{k,N_{max}}
\end{bmatrix} \leftrightarrow \begin{bmatrix}
    e_{k,1} \\
    e_{k,2} \\
    \vdots \\
    e_{k,N_{max}}
\end{bmatrix}
\]

with \( x_{k,i} = [x_{k,i} \ y_{k,i} \ \dot{x}_{k,i} \ \dot{y}_{k,i}]^T \): \( i^{th} \) target’s kinematics.

Aim:
Use the MCMC-based Particle Algorithm to approximate \( p(x_k, x_{k-1}, e_k, e_{k-1} | z_{1:k}) \)
Application to Multitarget Tracking: Problem Formulation

- Time-varying number of targets (max of 10) moving using a near-constant velocity model,
- Data association free observation model
  \[\rightarrow\] Poisson point process model [Gilhom & Godsill & Maskell & Salmond, 2005]

Illustrative Example

Expected \# of measurements per target: 1 - Expected \# of clutters: 100
At the $m^{th}$ MCMC iteration, the following procedure is adopted:

1. Make a joint draw for $\{x^m_k, x^m_{k-1}, e^m_k, e^m_{k-1}\}$ using a Metropolis-Hastings step,
2. Refine individually each target's component in $\{x^m_k, e^m_k\}$ using a series of Metropolis-Hastings-within-Gibbs steps.

**Joint Proposal**

1. Propose $\{x^*_{k-1}, e^*_{k-1}\} \sim q(x_{k-1}, e_{k-1}) = \hat{p}(x^*_{k-1}, e^*_{k-1} | z_{1:k-1}) = \frac{1}{Np} \sum_{j=1}^{Np} \delta(x_{k-1} - x(j)) \delta(e_{k-1} - e(j))$
2. Propose $e^*_k \sim q(e_k | e^*_{k-1}) = p(e_k | e^*_{k-1})$
3. Propose $x^*_k \sim q(x_k | x^*_{k-1}, e^*_k, e^*_{k-1})$
4. With $\{x^*_k, e^*_k, x^*_{k-1}, e^*_{k-1}\}$, calculate the acceptance ratio $\rho_1$:

\[
\rho_1 = \min \left( 1, \frac{p(z_k | x^*_k, e^*_k)p(x^*_k | x^*_{k-1}, e^*_{k-1})p(e^*_k | e^*_{k-1}) \hat{p}(x^*_{k-1}, e^*_{k-1} | z_{1:k-1})}{q(x^*_k | x^*_{k-1}, e^*_k, e^*_{k-1})q(e^*_k | e^*_{k-1})q(x^*_{k-1}, e^*_{k-1} | z_{1:k-1})} \right)
\]

5. Sample a uniform random variable $u$ from $\mathcal{U}(0, 1)$.
6. If $u \leq \rho_1$ then
7. \[
\{x^m_k, e^m_k, x^m_{k-1}, e^m_{k-1}\} = \{x^*_k, e^*_k, x^*_{k-1}, e^*_{k-1}\}
\]
8. Else
9. \[
\{x^m_k, e^m_k, x^m_{k-1}, e^m_{k-1}\} = \{x^{m-1}_k, e^{m-1}_k, x^{m-1}_{k-1}, e^{m-1}_{k-1}\}
\]
10. End if
At the \( m^{th} \) MCMC iteration, the following procedure is adopted:

\[ \sim \] Make a joint draw for \( \{x_k^m, x_{k-1}^m, e_k^m, e_{k-1}^m\} \) using a Metropolis-Hastings step,

\[ \sim \] Refine individually each target’s component in \( \{x_k^m, e_k^m\} \) using a series of Metropolis-Hastings-within-Gibbs steps.

**Joint Proposal**

1: Propose \( \{x_{k-1}^*, e_{k-1}^*\} \sim q(x_{k-1}, e_{k-1}) = \hat{\rho}(x_{k-1}^*, e_{k-1}^*|z_{1:k-1}) = \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(x_{k-1} - x_{k-1}^{(j)}) \delta(e_{k-1} - e_{k-1}^{(j)}) \)

2: Propose \( e_k^* \sim q(e_k|e_{k-1}^*) = p(e_k|e_{k-1}^*) \)

3: Propose \( x_k^* \sim q(x_k|x_{k-1}^*, e_k^*, e_{k-1}^*) \)

4: With \( \{x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^*\} \), calculate the acceptance ratio \( \rho_1 \):

\[
\rho_1 = \min \left( 1, \frac{p(z_k|x_k^*, e_k^*)p(x_k^*|x_{k-1}^*, e_{k-1}^*, e_k^*)p(e_k^*|e_{k-1}^*)}{p(z_k|x_k^{m-1}, e_k^{m-1})p(x_k^{m-1}|x_{k-1}^{m-1}, e_{k-1}^{m-1}, e_k^{m-1})p(e_k^{m-1}|e_{k-1}^{m-1})} \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(x_{k-1}^* - x_{k-1}^{(j)}) \delta(e_{k-1}^* - e_{k-1}^{(j)}) \right)
\]

5: Sample a uniform random variable \( u \) from \( \mathcal{U}(u|0, 1) \).

6: if \( u \leq \rho_1 \) then

7: \( \{x_k^m, e_k^m, x_{k-1}^m, e_{k-1}^m\} = \{x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^*\} \)

8: else

9: \( \{x_k^m, e_k^m, x_{k-1}^m, e_{k-1}^m\} = \{x_k^{m-1}, e_k^{m-1}, x_{k-1}^{m-1}, e_{k-1}^{m-1}\} \)

10: end if
At the $m^{th}$ MCMC iteration, the following procedure is adopted:

$\Rightarrow$ Make a joint draw for $\{x^m_k, x^m_{k-1}, e^m_k, e^m_{k-1}\}$ using a Metropolis-Hastings step,

$\Rightarrow$ Refine individually each target's component in $\{x^m_k, e^m_k\}$ using a series of Metropolis-Hastings-within-Gibbs steps.

**Joint Proposal**

1: Propose $\{x^*_k, e^*_k\} \sim q(x_k, e_k) = \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(x_k - x^{(j)}_k)\delta(e_k - e^{(j)}_k)$

2: Propose $e^*_k \sim q(e_k|e^*_k) = p(e_k|e^*_k)$

3: Propose $x^*_k \sim q(x_k|x^*_k, e^*_k, e^*_k)$

4: With $\{x^*_k, e^*_k, x^*_{k-1}, e^*_{k-1}\}$, calculate the acceptance ratio $\rho_1$:

$$\rho_1 = \min \left( 1, \frac{p(z_k|x^*_k, e^*_k)p(x^{*}_k|x^{*}_{k-1}, e^{*}_k, e^{*}_{k-1})}{q(x^{*}_k|x_{k-1}^{*}, e^{*}_k, e^{*}_{k-1})} \right)$$

5: Sample a uniform random variable $u$ from $\mathcal{U}(u|0, 1)$.
6: if $u \leq \rho_1$ then

7: $\{x^m_k, e^m_k, x^m_{k-1}, e^m_{k-1}\} = \{x^*_k, e^*_k, x^*_{k-1}, e^*_{k-1}\}$

8: else

9: $\{x^m_k, e^m_k, x^m_{k-1}, e^m_{k-1}\} = \{x^{m-1}_k, e^{m-1}_k, x^{m-1}_{k-1}, e^{m-1}_{k-1}\}$

10: end if
At the $m^{th}$ MCMC iteration, the following procedure is adopted:

$\rightsquigarrow$ Make a joint draw for $\{x^m_k, x^m_{k-1}, e^m_k, e^m_{k-1}\}$ using a Metropolis-Hastings step,

$\rightsquigarrow$ Refine individually each target's component in $\{x^m_k, e^m_k\}$ using a series of Metropolis-Hastings-within-Gibbs steps.

**Refinement Step**

1: **for** each target $n$ **do**
2: Propose $e^*_{k,n} \sim p(e_{k,n} | e^m_{k \setminus n}, e^m_{-k,n})$
3: Propose $x^*_{k,n} \sim q(x_{k,n} | x^m_{k-1,n}, e^*_{k,n}, e^m_{k-1,n})$
4: With $\{e^*_{k,n}, x^*_{k,n}\}$, calculate the acceptance ratio $\rho_2$

$$
\rho_2 = \min \left( \frac{p(z_k | x^*_{k,n}, x^m_{k \setminus n}, e^*_{k,n}, e^m_{k \setminus n}) p(x^*_{k,n} | x^m_{k-1,n}, e^*_{k,n}, e^m_{k-1,n})}{q(x^*_{k,n} | x^m_{k-1,n}, e^*_{k,n}, e^m_{k-1,n})} \frac{q(x^m_{k,n} | x^m_{k-1,n}, e^*_{k,n}, e^m_{k-1,n})}{p(z_k | x^m_{k,n}, e^m_{k,n}) p(x^m_{k,n} | x^m_{k-1,n}, e^m_{k-1,n})} \right)
$$

5: Sample a uniform random variable $u$ from $\mathcal{U}(0,1)$ and set $\{e^m_{k,n}, x^m_{k,n}\} = \{e^*_{k,n}, x^*_{k,n}\}$ if $u \leq \rho_2$, otherwise set $\{e^m_{k,n}, x^m_{k,n}\} = \{e^m_{k,n}, x^m_{k,n}\}$.

6: **end for**
Numerical Simulations

Results obtained with the MCMC-Based Particle Algorithm

6 000 Particles - Thinning = 2 - Burn-in = 1000 - $N_{max} = 10$

Expected ♯ of measurements per target : 2 - Expected ♯ of clutters : 100
Results obtained with the MCMC-Based Particle Algorithm

6 000 Particles - Thinning = 2 - Burn-in = 1000 - $N_{max} = 10$

Expected # of measurements per target : 2 - Expected # of clutters : 100
How can we compare algorithms’ performance?

Not trivial owing to the unknowledge of the number of targets.

Three hypothetical scenarios with ○ representing true targets and + representing estimated targets. Which estimate is closest to the truth?

Existing approaches are based on Miss-Distance using Wasserstein metric\(^1\)\(^2\)

Use point estimate and sensitive to parameters’ choice.

Ex. : Penalty given to the cardinality error

---


**Idea:** Use a more "probabilistic" method to compare particle algorithms

**Prop:** Evaluate the posterior value of the ground truth

\[
\mathcal{M}(N_k^{GT}, x_k^{GT}) = p(N_k^{GT}, x_k^{GT} | Z_{1:k}) = p(N_k^{GT} | Z_{1:k}) p(x_k | N_k^{GT}, Z_{1:k})
\]

\[
\approx \hat{p}(N_k^{GT} | Z_{1:k}) \prod_{i=1}^{N_k^{GT}} \hat{p}(x_{k,i}^{GT} | Z_{1:k})
\]

\[
\Rightarrow \text{Higher } \mathcal{M}(N_k^{GT}, x_k^{GT}) \text{ is, better is the algorithm}
\]

\(\hat{p}(N_k | Z_{1:k})\) and \(\hat{p}(x_{k,i} | Z_{1:k})\) are obtained using the particle approximation of the posterior distribution:

\[
\hat{p}(x_k, e_k | Z_{1:k}) = \frac{1}{N_P} \sum_{p=1}^{N_P} \delta(x_k - x_k^p) \delta(e_k - e_k^p)
\]

More precisely, \(\hat{p}(N_k | Z_{1:k})\) is obtained using the posterior of the existence variable:

\[
\hat{p}(N_k | Z_{1:k}) = \frac{1}{N_P} \sum_{p=1}^{N_P} \delta(N_k - \sum_{i=1}^{N_{max}} e_{k,i}^p) \quad \text{for } N_k = 1, ..., N_{max}
\]

(since each existence variable \(e_{k,i}\) is equal to 1 for active target and 0 otherwise)
Idea: Use a more "probabilistic" method to compare particle algorithms

Prop: Evaluate the posterior value of the ground truth

\[
\mathcal{M}(N_k^{GT}, x_k^{GT}) = p(N_k^{GT}, x_k^{GT} | Z_{1:k}) = p(N_k^{GT} | Z_{1:k})p(x_k | N_k^{GT}, Z_{1:k}) \\
\approx \hat{p}(N_k^{GT} | Z_{1:k}) \prod_{i=1}^{N_k^{GT}} \hat{p}(x_{k,i}^{GT} | Z_{1:k})
\]

⇒ Higher \( \mathcal{M}(N_k^{GT}, x_k^{GT}) \) is, better is the algorithm

\( \hat{p}(N_k | Z_{1:k}) \) and \( \hat{p}(x_{k,i} | Z_{1:k}) \) are obtained using the particle approximation of the posterior distribution:

\[
\hat{p}(x_k, e_k | Z_{1:k}) = \frac{1}{N_P} \sum_{p=1}^{N_P} \delta(x_k - x_k^p) \delta(e_k - e_k^p)
\]

More precisely, \( \hat{p}(x_{k,i} | Z_{1:k}) \) is obtained by using a Kernel on the particle output of the targets’ position (PHD surface):

\[
\hat{p}(x_{k,i} | Z_{1:k}) = Cte \sum_{p=1}^{N_P} \sum_{j=1}^{N_{max}} e_{k,i}^p K(x_{k,i}, x_{k,j}^p)
\]
Alternative evaluation method - First Thoughts

Particle 1 - 3 Targets

Particle 2 - 1 Target

Particle 3 - 2 Targets

Gaussian Kernel

\[ \hat{p}(x_{k,i} | Z_{1:k}) = C^t e^{\sum_{p=1}^{N_p} \sum_{j=1}^{N_{max}} e_{k,j} K(x_{k,i}, x_{k,j})} \]
Comparison of results obtained with 3 different parametrization of the MCMC-based particle algorithm

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of Burn-in Iterations $N_{burn}$</th>
<th>Number of Particles $N_p$</th>
<th>Chain Thinning $N_{thin}$</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 1</td>
<td>1000</td>
<td>4000</td>
<td>1</td>
<td>Joint Draw + Refinement Step</td>
</tr>
<tr>
<td>Algorithm 2</td>
<td>1000</td>
<td>4000</td>
<td>1</td>
<td>Joint Draw only</td>
</tr>
<tr>
<td>Algorithm 3</td>
<td>1000</td>
<td>4000</td>
<td>4</td>
<td>Joint Draw + Refinement Step</td>
</tr>
</tbody>
</table>

Representation of the CDF $p(\log \mathcal{M}(N_k^{GT}, x_k^{GT}) < \epsilon)$ for the three different configurations.

- Significative improvement when the refinement step is employed,
- Algorithm 3 better than Algorithm 1 owing to a higher thinning value.
This MCMC-based particle algorithm is able to efficiently:

- detect and track targets in hostile environments with heavy clutter,
- deal with variable number of targets.

Ongoing works:

- Further study of the proposed method to compare particle algorithms,
- Performance comparison with other algorithms:
  PHD filter, Particle filter, Resample-Move, SMC samplers
Thank you for your attention
**Aim:** Compute, at time $t_k$, the filtering posterior distribution given by

$$p(x_k, e_k | z_{1:k}) = \frac{p(z_k | x_k, e_k)p(x_k, e_k | z_{1:k-1})}{p(z_k | z_{1:k-1})}$$

with

$$p(x_k, e_k | z_{1:k-1}) = \int \int p(x_k, e_k | x_{k-1}, e_{k-1})p(x_{k-1}, e_{k-1} | z_{1:k-1}) dx_{k-1} de_{k-1}$$

We choose to expand the transition probability model as

$$p(x_k, e_k | x_{k-1}, e_{k-1}) = \underbrace{p(x_k | x_{k-1}, e_k, e_{k-1})}_{\text{Prior distribution of targets' kinematics}} \underbrace{p(e_k | e_{k-1})}_{\text{Prior distribution of existence variables}}$$
Each target’s existence variable is modeled as a discrete Markov chain, independent of one another:

$$p(e_k | e_{k-1}) = \prod_{n=1}^{N_{max}} p(e_{k,n} | e_{k-1,n})$$

with

$$p(e_{k,n} | e_{k-1,n}) = \begin{cases} P_B & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 0 \\ 1 - P_B & \text{if } e_{k,n} = 0 \text{ and } e_{k-1,n} = 0 \\ P_D & \text{if } e_{k,n} = 0 \text{ and } e_{k-1,n} = 1 \\ 1 - P_D & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 1 \end{cases}$$

⇒ Bernoulli process
Assumption: Independent targets

\[
\Rightarrow p(x_k | x_{k-1}, e_k, e_{k-1}) = \prod_{n=1}^{N_{max}} p(x_{k,n} | x_{k-1,n}, e_{k,n}, e_{k-1,n})
\]

Targets can be partitioned according to \(e_{k,n}\) and \(e_{k-1,n}\):

\[
p(x_{k,n} | x_{k-1,n}, e_{k-1:k,n}) = \begin{cases} 
  p_b(x_{k,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 0 \text{ (Target Birth)} \\
  p_d(x_{k,n}) & \text{if } e_{k,n} = 0 \text{ (Inactive Target)} \\
  p_u(x_{k,n} | x_{k-1,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 1 \text{ (Target Update)}
\end{cases}
\]
Assumption: Independent targets

\[ p(x_k | x_{k-1}, e_k, e_{k-1}) = \prod_{n=1}^{N_{\text{max}}} p(x_{k,n} | x_{k-1,n}, e_{k,n}, e_{k-1,n}) \]

Targets can be partitioned according to \( e_{k,n} \) and \( e_{k-1,n} \):

\[ p(x_{k,n} | x_{k-1,n}, e_{k-1:k,n}) = \begin{cases} 
  p_b(x_{k,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 0 \text{ (Target Birth)} \\
  p_d(x_{k,n}) & \text{if } e_{k,n} = 0 \text{ (Inactive Target)} \\
  p_u(x_{k,n} | x_{k-1,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 1 \text{ (Target Update)}
\end{cases} \]

**Target Birth**

\[ p_b(x_{k,n}) = \mathcal{U}(x_{k,n} | 0, L_x)\mathcal{U}(x_{k,n} - V_{\text{max}}, V_{\text{max}})\mathcal{U}(y_{k,n} | 0, L_y)\mathcal{U}(y_{k,n} - V_{\text{max}}, V_{\text{max}}) \]

A target can appear anywhere uniformly in the surveillance area.
**Transition probability of the targets**

**Assumption:** Independent targets

\[
\Rightarrow p(x_k|x_{k-1}, e_k, e_{k-1}) = \prod_{n=1}^{N_{max}} p(x_{k,n}|x_{k-1,n}, e_{k,n}, e_{k-1,n})
\]

Targets can be partitioned according to \(e_{k,n}\) and \(e_{k-1,n}\):

\[
p(x_{k,n}|x_{k-1,n}, e_{k-1:k,n}) = \begin{cases} 
p_b(x_{k,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 0 \text{ (Target Birth)} \\
p_d(x_{k,n}) & \text{if } e_{k,n} = 0 \text{ (Inactive Target)} \\
p_u(x_{k,n}|x_{k-1,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 1 \text{ (Target Update)}
\end{cases}
\]

**Inactive Target**

\[
p_d(x_{k,n}) = \delta(x_{\text{death}})
\]

*The target state is kept at some \(x_{\text{death}}\), which is the state where an inactive target is represented.*
**Assumption**: Independent targets

\[
\Rightarrow p(x_k|x_{k-1}, e_k, e_{k-1}) = \prod_{n=1}^{N_{\text{max}}} p(x_{k,n}|x_{k-1,n}, e_{k,n}, e_{k-1,n})
\]

Targets can be partitioned according to \(e_{k,n}\) and \(e_{k-1,n}\):

\[
p(x_{k,n}|x_{k-1,n}, e_{k-1:k,n}) = \begin{cases} 
    p_b(x_{k,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 0 \text{ (Target Birth)} \\
    p_d(x_{k,n}) & \text{if } e_{k,n} = 0 \text{ (Inactive Target)} \\
    p_u(x_{k,n}|x_{k-1,n}) & \text{if } e_{k,n} = 1 \text{ and } e_{k-1,n} = 1 \text{ (Target Update)}
\end{cases}
\]

**Update Target**

\[
p_u(x_{k,n}|x_{k-1,n}) = \mathcal{N}(x_{k,n}|A_{k,n}x_{k-1,n}, Q_{k,n})
\]

*Target motion follows a near constant velocity model*
Sequential MCMC in the literature


- **MCMC-Based Particle Algorithm**:
  - Z. Khan, T. Blach and F. Dellaert, “MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, Number 5, pp. 105-1819, Nov. 2005