MCMC-Based Particle filtering for Multi-Target Tracking

Alemany Nicolas, Calvet Rémi, Clayton Pierre

January 10, 2025

Article

- **On MCMC-Based Particle Methods for Bayesian Filtering : Application to Multitarget Tracking**
- François Septier (University of Southern Brittany), Sze Kim Pang (DSO National Laboratories), Avishy Carmi, (Ben-Gurion University of the Negev), S.J. Godsill(University of Cambridge)

Objective

Objective: track multiple targets moving within a surveillance area, where the number of targets can change over time due to births and deaths in a highly cluttered environment, using MCMC based particle filtering

Reminder

Bayesian Filtering: At each time step k, compute the posterior distribution of the hidden state x_k given all observations up to that time $z_{0:k}$:

$$p(x_k|z_{0:k}) = \frac{p(z_k|x_k)\, p(x_k|z_{0:k-1})}{p(z_k|z_{0:k-1})}$$

where:

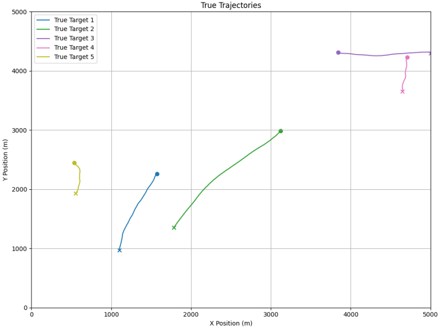
- $p(z_k|x_k)$ is the **likelihood function**.
- $p(x_k|z_{0:k-1})$ is the **predictive distribution**, obtained from the prior and the state transition model.
- $p(z_k|z_{0:k-1})$ is the **evidence** or **marginal likelihood**, serving as a normalization constant.

Reminder

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from a probability distribution by constructing a Markov chain that has the desired distribution as its equilibrium distribution.
Key Idea: Generate a sequence of samples $\{x^{(0)}, x^{(1)}, x^{(2)}, \dots\}$ such that, after a sufficient number of steps, the samples are from the target distribution $p(x)$.

Problem Formulation

- 5000 x 5000 surveillance area
- Up to 5 targets (varying to birth and deaths) starting at random positions (uniformly distributed)
- Near constant velocity motion



State Transition Equations for Active Targets

State Transition Equation for Active Targets

For an active target ($e_{k,n} = 1$), the state transition is given by:

$$x_{k,n} = Ax_{k-1,n} + w_{k,n}$$

where:

- $x_{k,n}$: State vector of the n-th target at time k.
- A: State transition matrix.
- $w_{k,n}$: Process noise, modeled as zero-mean Gaussian noise.

State Transition Equations for Active Targets

State Transition Matrix A

The state transition matrix A captures the near-constant velocity assumption:

$$A = \begin{bmatrix} I_2 & \Delta t \, I_2 \\ 0_{2\times 2} & I_2 \end{bmatrix}$$

- I_2: 2×2 identity matrix.
- Δt: Time interval between $k-1$ and k.
- $0_{2\times 2}$: 2×2 zero matrix.

State Transition Equations for Active Targets

Process Noise $w_{k,n}$

The process noise accounts for random accelerations and maneuvers:

$$w_{k,n} \sim \mathcal{N}(0, Q)$$

where the covariance matrix Q is:

$$Q = \sigma^2 \begin{bmatrix} \frac{\Delta t^3}{3} I_2 & \frac{\Delta t^2}{2} I_2 \\ \frac{\Delta t^2}{2} I_2 & \Delta t \, I_2 \end{bmatrix}$$

- σ^2: Process noise variance, representing the uncertainty in the target's acceleration.
- Q: Captures correlation between position and velocity due to the integration of acceleration over time.

State Representation

State Vector x_k

At each time step k, the overall state vector x_k encapsulates the kinematic states of all potential targets up to a predefined maximum number N_{max}. Each target's kinematic state includes its position and velocity in two dimensions:

$$x_k = [x_{k,1}, \ldots, x_{k,N_{\text{max}}},\ y_{k,1}, \ldots, y_{k,N_{\text{max}}},\ \dot{x}_{k,1}, \ldots, \dot{x}_{k,N_{\text{max}}},\ \dot{y}_{k,1}, \ldots, \dot{y}_{k,N_{\text{max}}}]^T$$

- $x_{k,n}$, $y_{k,n}$: Position coordinates of the n-th target.
- $\dot{x}_{k,n}$, $\dot{y}_{k,n}$: Velocity components of the n-th target.

State Representation

Existence Vector e_k

Each potential target is associated with an existence indicator $e_{k,n}$, which is a binary variable indicating whether the target is active (1) or inactive (0) at time k:

$$e_k = [e_{k,1}, e_{k,2}, \ldots, e_{k,N_{\max}}]^T, \quad e_{k,n} \in \{0, 1\}$$

- Active Target: $e_{k,n} = 1$, the n-th target is present and moving according to the motion model.
- Inactive Target: $e_{k,n} = 0$, the n-th target is not present; its state variables are set to a predefined value x_{death} representing inactivity.

Observation Model

Measurements z_k

At time k, the sensor produces M_k measurements:

$$z_k = \{z_k^1, z_k^2, \ldots, z_k^{M_k}\}$$

Each measurement z_k^m is a two-dimensional vector representing a position in the surveillance area.

Clutters

Clutter measurements are false alarms that do not correspond to any target. They are modeled as a Poisson process over the surveillance area.

Probability Density of Clutter:

$p_C(z_k^m) = \frac{1}{A}$

A: Area of the surveillance region.

Observation Model

Measurement Model for Targets

For an active target ($e_{k,n} = 1$), the measurement is modeled as:

$$z_k^m = Hx_{k,n} + v_k^m$$

where:

- H: Measurement matrix, typically extracting the position components:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- v_k^m: Measurement noise, modeled as zero-mean Gaussian noise with covariance R:

$$v_k^m \sim \mathcal{N}(0, R)$$

Observation Model

Likelihood of a Measurement from a Target

$$p(z_k^m | x_{k,n}) = \mathcal{N}(z_k^m; Hx_{k,n}, R)$$

Reminder

Objective

The objective of the MCMC steps is to approximate the distribution of $p(x_k, e_k | z_{0:k})$ in order to be able to sample from it to make predictions of $(x_k, e_k | z_{0:k})$.

MCMC based Particle Filtering

Here is an overview of the MCMC-based particle filtering algorithm:

1. Initialization

Generate an initial set of particles

$$\{x_{-1}^{(j)}, e_{-1}^{(j)}\}_{j=1}^{N_p}$$

representing the prior distribution at time $k = -1$ (or $k = 0$).

MCMC based Particle Filtering

2. For each time step k:

- **MCMC Iterations & Refinement Steps:**
  - Jointly propose new particles $(x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^*)$ and compute the Metropolis-Hastings acceptance probability, using the set of particles $(\{x_{k-1}^{(j)}, e_{k-1}^{(j)}\}_{j=1}^{N_p})$ given by the previous time step iteration.
  - Refinement: Perform individual updates of $x_{k,n}$ and $e_{k,n}$ for each target n with their acceptance probabilities.
- **Burn-in and Thinning:**
  - Discard burn-in iterations and keep every N_{thin}-th sample.
- **Approximating the Filtering Distribution:** Use the accepted particles

$$\{x_k^{(j)}, e_k^{(j)}\}_{j=1}^{N_p}$$

to approximate $p(x_k, e_k \mid z_{0:k})$ using the empirical distribution.

Target Distribution and Acceptance Probabilities

Target Distribution $p(x_k, e_k, x_{k-1}, e_{k-1}|z_{0:k})$:

- The joint posterior distribution is proportional to the product of the likelihood and the prior:

$$p(x_k, e_k, x_{k-1}, e_{k-1}|z_{0:k}) \propto p(z_k|x_k, e_k)\, p(x_k, e_k|x_{k-1}, e_{k-1})\, p(x_{k-1}, e_{k-1}|z_{0:k-1})$$

  Since $p(x_{k-1}, e_{k-1}|z_{0:k-1})$ is untractable, use the approximation calculated at the step before :

$$\hat{p}(x_k, e_k, x_{k-1}, e_{k-1}|z_{0:k}) = p(z_k|x_k, e_k) * p(x_k, e_k|x_{k-1}, e_{k-1}) * \hat{p}(x_{k-1}, e_{k-1}|z_{0:k-1})$$

  where

$$\hat{p}(x_{k-1}, e_{k-1}|z_{0:k-1}) = \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(x_{k-1} - x_{k-1}^{(j)})\delta(e_{k-1} - e_{k-1}^{(j)})$$

- The "real" target distribution we use is $\hat{p}(x_k, e_k, x_{k-1}, e_{k-1}|z_{0:k})$.

Target Distribution and Acceptance Probabilities

Computing Necessary Densities:

- *Transition Probabilities*:

  $$p(x_k, e_k | x_{k-1}, e_{k-1}) \quad \text{is given by the motion model.}$$

- *Existence Probabilities*:

  $$p(e_k | e_{k-1}) \quad \text{is defined by the birth and death probabilities.}$$

- *Observation Likelihood*:

  $$p(z_k | x_k, e_k) \quad \text{is computed using the measurement model and accounts for clutter.}$$

- *Prior Distributions*:

  $$p(x_{k-1}, e_{k-1} | z_{0:k-1}) \quad \text{is approximated using particles from the previous time step.}$$

Target Distribution and Acceptance Probabilities

Acceptance Probability for Joint Proposal:

- Formula:
$$\alpha = \min\left\{1, \frac{\hat{p}(x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^* | z_{0:k})}{q_1(x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^* | x_k^{(m-1)}, e_k^{(m-1)}, x_{k-1}^{(m-1)}, e_{k-1}^{(m-1)})} \quad \times \frac{q_1(x_k^{(m-1)}, e_k^{(m-1)}, x_{k-1}^{(m-1)}, e_{k-1}^{(m-1)} | x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^*)}{\hat{p}(x_k^{(m-1)}, e_k^{(m-1)}, x_{k-1}^{(m-1)}, e_{k-1}^{(m-1)} | z_{0:k})}\right\}$$
- Accept $\{x_k^{(m)}, e_k^{(m)}, x_{k-1}^{(m)}, e_{k-1}^{(m)}\} = \{x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^*\}$ with probability α, else retain previous state.

Algorithm 1 Updating Individual Targets

for each target n **do**
  Propose New Existence Variable $e^*_{k,n}$:
  Sample $e^*_{k,n}$ from the transition probability:

$$e^*_{k,n} \sim p(e_{k,n} | e^{(m)}_{k-1,n})$$

  Propose New State $x^*_{k,n}$:
  if $e^*_{k,n} = 1$ **then**
    Propose $x^*_{k,n}$ from the state transition model:

$$x^*_{k,n} \sim p(x_{k,n} | x^{(m)}_{k-1,n}, e^*_{k,n}, e^{(m)}_{k-1,n})$$

  else
    Set $x^*_{k,n} = x_{\text{death}}$.
  end if
  Compute Acceptance Probability α_n:
  Compute the acceptance probability:

$$\alpha_n = \min\left\{1, \frac{p(x^*_{k,n}, e^*_{k,n} | x^{(m)}_{k-1,n}, e^{(m)}_{k-1,n}, z_{0:k})}{q_3(x^*_{k,n}, e^*_{k,n} | x^{(m)}_{k,n}, e^{(m)}_{k,n}, x^{(m)}_{k-1,n}, e^{(m)}_{k-1,n})} \times \frac{q_3(x^{(m)}_{k,n}, e^{(m)}_{k,n} | x^*_{k,n}, e^*_{k,n}, x^{(m)}_{k-1,n}, e^{(m)}_{k-1,n})}{p(x^{(m)}_{k,n}, e^{(m)}_{k,n} | x^{(m)}_{k-1,n}, e^{(m)}_{k-1,n}, z_{0:k})}\right\}$$

  Accept or Reject:
  if $U(0,1) \leq \alpha_n$ **then**
    Accept $(x^*_{k,n}, e^*_{k,n})$.
  else
    Retain previous values $(x^{(m)}_{k,n}, e^{(m)}_{k,n})$.
  end if
end for

Reminder

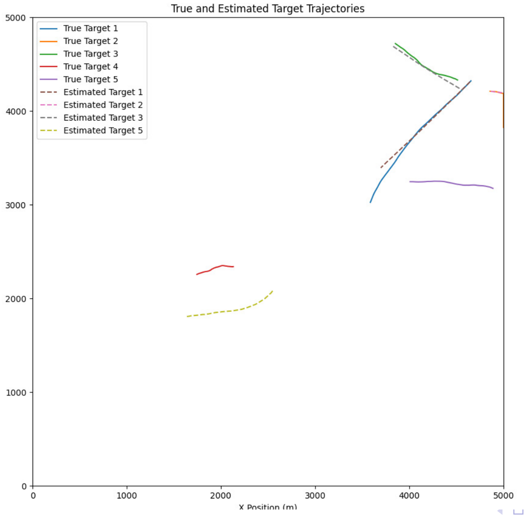
Filtering algorithm:

- **MCMC Iterations & Refinement Steps:**
  - Jointly propose new particles $(x_k^*, e_k^*, x_{k-1}^*, e_{k-1}^*)$ and compute the Metropolis-Hastings acceptance probability, using the set of particles $(\{x_{k-1}^{(j)}, e_{k-1}^{(j)}\}_{j=1}^{N_p})$ given by the previous time step iteration.
  - Refinement: Perform individual updates of $x_{k,n}$ and $e_{k,n}$ for each target n with their acceptance probabilities.
- **Burn-in and Thinning:**
  - Discard burn-in iterations and keep every N_{thin}-th sample.
- **Approximating the Filtering Distribution:** Use the accepted particles

$$\{x_k^{(j)}, e_k^{(j)}\}_{j=1}^{N_p}$$

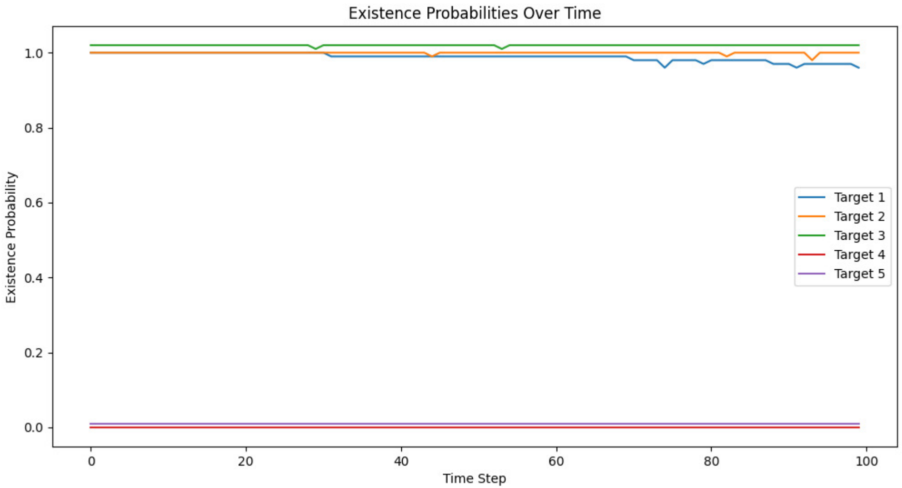
to approximate $p(x_k, e_k \mid z_{0:k})$ using the empirical distribution.

Trajectories



Results

Cardinality



Git Hub

https://github.com/Pierre-Clayton/Multitarget-Tracking

Algorithm 1 2 MCMC-Based Particle Algorithm adapted

Require:
- Number of particles N_p, burn-in period N_{burn}, thinning interval N_{thin}
- The number of MCMC iterations per time step N_{MCMC} is given by $N_{MCMC} = N_p * N_{thin} + N_{burn}$
- Model parameters: P_B, P_D, $\sigma^2_{x,n}$, Λ_C, Λ^n_x, etc.
- Observations $\{z_{0:k}\}$ for each time step k

Ensure: Particle-based approximation of the filtering posterior $p(\boldsymbol{x}_k, \boldsymbol{e}_k \mid \boldsymbol{z}_{0:k})$ that will be noted $\hat{p}(x_k, e_k|z_{0:k})$. It also makes predictions using those approximations.

```
Initialization:
Initialize particle set {(x_{-1}^{(j)}, e_{-1}^{(j)})}_{j=1}^{N_p}
for k = 1 to T do
```

(We start MCMC with target distribution $\hat{p}(x_k, e_k, x_{k-1}, e_{k-1}|z_{0:k}) = p(z_k|x_k, e_k) * p(x_k, e_k|x_{k-1}, e_{k-1}) * \hat{p}(x_{k-1}, e_{k-1}|z_{0:k-1})$ where $\hat{p}(x_{k-1}, e_{k-1}|z_{0:k-1}) = \frac{1}{N_p}\sum_{j=1}^{N_p} \delta(x_{k-1} - x_{k-1}^{(j)})\delta(e_{k-1} - e_{k-1}^{(j)})$.

for $m = 1$ **to** N_{MCMC} **do**

(a) Joint Draw

Propose $\{\boldsymbol{x}_k^*, \boldsymbol{e}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{e}_{k-1}^*\} \sim q_1(\boldsymbol{x}_k, \boldsymbol{e}_k, \boldsymbol{x}_{k-1}, \boldsymbol{e}_{k-1} \mid \boldsymbol{x}_k^{(m-1)}, \boldsymbol{e}_k^{(m-1)}, \boldsymbol{x}_{k-1}^{(m-1)}, \boldsymbol{e}_{k-1}^{(m-1)}) \propto \sum_{j=1}^{N_p} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \mathbf{e}_k) p(\mathbf{e}_k \mid \mathbf{e}_{k-1}^{(j)}) \delta(\mathbf{x}_{k-1}^{(j)}) \delta(\mathbf{e}_{k-1}^{(j)})$ (equation 14 of the article)

Compute MH acceptance probability ρ_1:

$$\rho_1 = \min\left\{1, \frac{\hat{p}(\boldsymbol{x}_k^*, \boldsymbol{e}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{e}_{k-1}^* \mid \boldsymbol{z}_{0:k})}{q_1(\boldsymbol{x}_k^*, \boldsymbol{e}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{e}_{k-1}^* \mid \boldsymbol{x}_k^{(m-1)}, \boldsymbol{e}_k^{(m-1)}, \boldsymbol{x}_{k-1}^{(m-1)}, \boldsymbol{e}_{k-1}^{(m-1)})} \times \frac{q_1(\boldsymbol{x}_k^{(m-1)}, \boldsymbol{e}_k^{(m-1)}, \boldsymbol{x}_{k-1}^{(m-1)}, \boldsymbol{e}_{k-1}^{(m-1)} \mid \boldsymbol{x}_k^*, \boldsymbol{e}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{e}_{k-1}^*)}{\hat{p}(\boldsymbol{x}_k^{(m-1)}, \boldsymbol{e}_k^{(m-1)}, \boldsymbol{x}_{k-1}^{(m-1)}, \boldsymbol{e}_{k-1}^{(m-1)} \mid \boldsymbol{z}_{0:k})}\right\}$$

Accept $\{\boldsymbol{x}_k^{(m)}, \boldsymbol{e}_k^{(m)}, \boldsymbol{x}_{k-1}^{(m)}, \boldsymbol{e}_{k-1}^{(m)}\} = \{\boldsymbol{x}_k^*, \boldsymbol{e}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{e}_{k-1}^*\}$ with probability ρ_1, else retain previous state.

(b) Refinement

for each target n **do**

Propose $\{\boldsymbol{x}_{k,n}^*, \boldsymbol{e}_{k,n}^*\} \sim q_3(\boldsymbol{x}_{k,n}, \boldsymbol{e}_{k,n}|\boldsymbol{x}_{k,n}^{(m)}, \boldsymbol{e}_{k,n}^{(m)}, \boldsymbol{x}_{k-1,n}^{(m)}, \boldsymbol{e}_{k-1,n}^{(m)}) = p(\boldsymbol{x}_{k,n}|\boldsymbol{x}_{k-1,n}^{(m)}, \boldsymbol{e}_{k,n}) \times p(\boldsymbol{e}_{k,n}|\boldsymbol{e}_{k-1,n}^{(m)})$

Compute MH acceptance probability ρ_3:

$$\rho_3 = \min\left\{1, \frac{p(\boldsymbol{x}_{k,n}^*, \boldsymbol{e}_{k,n}^* \mid \boldsymbol{x}_{k-1,n}^{(m)}, \boldsymbol{e}_{k-1,n}^{(m)}, \boldsymbol{z}_{0:k})}{q_3(\boldsymbol{x}_{k,n}^*, \boldsymbol{e}_{k,n}^* \mid \boldsymbol{x}_{k,n}^{(m)}, \boldsymbol{e}_{k,n}^{(m)}, \boldsymbol{x}_{k-1,n}^{(m)}, \boldsymbol{e}_{k-1,n}^{(m)})} \times \frac{q_3(\boldsymbol{x}_{k,n}^{(m)}, \boldsymbol{e}_{k,n}^{(m)} \mid \boldsymbol{x}_{k,n}^*, \boldsymbol{e}_{k,n}^*, \boldsymbol{x}_{k-1,n}^{(m)}, \boldsymbol{e}_{k-1,n}^{(m)})}{p(\boldsymbol{x}_{k,n}^{(m)}, \boldsymbol{e}_{k,n}^{(m)} \mid \boldsymbol{x}_{k-1,n}^{(m)}, \boldsymbol{e}_{k-1,n}^{(m)}, \boldsymbol{z}_{0:k})}\right\}$$

Accept $\{\boldsymbol{x}_{k,n}^{(m)}, \boldsymbol{e}_{k,n}^{(m)}\} = \{\boldsymbol{x}_{k,n}^*, \boldsymbol{e}_{k,n}^*\}$ with probability ρ_3, else retain previous state.

end for

(c) Thinning and Burn-in

if $m > N_{\text{burn}}$ **and** $m \equiv 0 \pmod{N_{\text{thin}}}$ **then**

Store $\{\boldsymbol{x}_k^{(j)}, \boldsymbol{e}_k^{(j)}\} \leftarrow \{\boldsymbol{x}_k^{(m)}, \boldsymbol{e}_k^{(m)}\}$

end if

end for

(d) Approximation $\hat{p}(x_k, e_k|z_{0:k})$ with the new particle set

Store $\hat{p}(x_k, e_k|z_{0:k}) = \frac{1}{N_p}\sum_{j=1}^{N_p} \delta(x_k - x_k^{(j)}) * \delta(e_k - e_k^{(j)})$.

(e) Sampling and make predictions using the approximation $\hat{p}(x_k, e_k|z_{0:k})$

Sample multiple times from $\hat{p}(x_k, e_k|z_{0:k})$ and take the mean for example. (could consider other procedures) (Note that the sampling is easy since it is the density of a discrete random variable)

end for