The Particle Filtering Methodology in Signal Processing

Petar M. Djurić

Department of Electrical & Computer Engineering
Stony Brook University

September 25, 2009

Preface

- Many problems in adaptive filtering are nonlinear and non-Gaussian
- Of the many methods that have been proposed in the literature for solving such problems, particle filtering (PF) has become one of the most popular

Preface

- Many problems in adaptive filtering are nonlinear and non-Gaussian
- Of the many methods that have been proposed in the literature for solving such problems, particle filtering (PF) has become one of the most popular
- In this talk the basics of PF are revisited and a review of some of its most important implementations is discussed

Preface

- Many problems in adaptive filtering are nonlinear and non-Gaussian
- Of the many methods that have been proposed in the literature for solving such problems, particle filtering (PF) has become one of the most popular
- In this talk the basics of PF are revisited and a review of some of its most important implementations is discussed

Outline

Outline

1. Introduction
2. Motivation for use of particle filtering
3. The basic idea

Outline

1. Introduction
2. Motivation for use of particle filtering
3. The basic idea
4. Some particle filtering methods

Outline

Outline

Outline

Outline

1. Introduction
2. Motivation for use of particle filtering
3. The basic idea
4. Some particle filtering methods
5. Handling constant parameters
6. Rao-Blackwellization
7. Prediction and smoothing
8. Multiple particle filters

Outline

Outline

Outline

Outline

Introducing PF

- Some of the most important problems in signal processing require sequential processing of data

- The data describe a system that is mathematically represented by equations, which model the system's "random" evolution with time

Introducing PF

- Some of the most important problems in signal processing require sequential processing of data
- The data describe a system that is mathematically represented by equations, which model the system's "random" evolution with time
- The system is defined by its state variables of which some are dynamic and some are constant

Introducing PF

- Some of the most important problems in signal processing require sequential processing of data
- The data describe a system that is mathematically represented by equations, which model the system's "random" evolution with time
- The system is defined by its state variables of which some are dynamic and some are constant
- A typical assumption about the state is that it is Markovian

Introducing PF

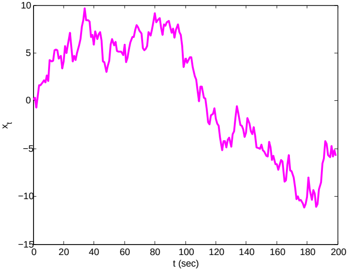
- Some of the most important problems in signal processing require sequential processing of data
- The data describe a system that is mathematically represented by equations, which model the system's "random" evolution with time
- The system is defined by its state variables of which some are dynamic and some are constant
- A typical assumption about the state is that it is Markovian
- The state is not directly observable and we acquire measurements which are degraded by noise and obtained sequentially in time

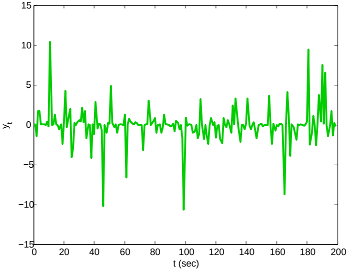
Introducing PF

- Some of the most important problems in signal processing require sequential processing of data
- The data describe a system that is mathematically represented by equations, which model the system's "random" evolution with time
- The system is defined by its state variables of which some are dynamic and some are constant
- A typical assumption about the state is that it is Markovian
- The state is not directly observable and we acquire measurements which are degraded by noise and obtained sequentially in time

Goal

- The main objective of sequential signal processing is the recursive estimation of the state of the system based on the available measurements

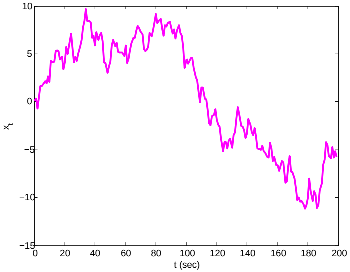


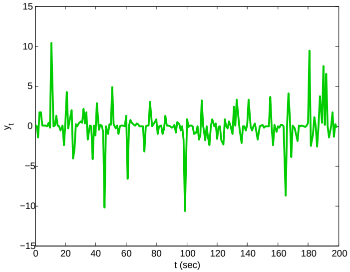


- The methods that are developed for estimation of the state are called *filters*

Goal

- The main objective of sequential signal processing is the recursive estimation of the state of the system based on the available measurements



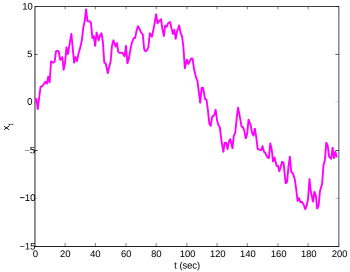


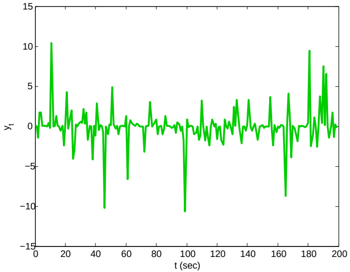
- The methods that are developed for estimation of the state are called *filters*

- Estimates of the state in the past or prediction of its values in the future may also be of interest

Goal

- The main objective of sequential signal processing is the recursive estimation of the state of the system based on the available measurements





- The methods that are developed for estimation of the state are called *filters*

- Estimates of the state in the past or prediction of its values in the future may also be of interest

Optimal filtering methods

- Optimal filtering methods for sequential signal processing are based on analytical updates of the densities of interest

$$\downarrow$$

Integration is a key operation

- Closed form solutions are possible in a very small number of situations

Optimal filtering methods

- Optimal filtering methods for sequential signal processing are based on analytical updates of the densities of interest

$$\downarrow$$

Integration is a key operation

- Closed form solutions are possible in a very small number of situations

$$\left.\begin{array}{c}\text{Gaussian noise}\\ \text{and}\\ \text{linear functions}\end{array}\right\} \longrightarrow \text{Optimal estimation with } \textit{Kalman filtering}$$

Optimal filtering methods

- Optimal filtering methods for sequential signal processing are based on analytical updates of the densities of interest

$$\downarrow$$

Integration is a key operation

- Closed form solutions are possible in a very small number of situations

$$\left.\begin{array}{c}\text{Gaussian noise}\\ \text{and}\\ \text{linear functions}\end{array}\right\} \longrightarrow \text{Optimal estimation with } \textit{Kalman filtering}$$

Suboptimal filtering methods

$$\left.\begin{array}{c}\text{Non-Gaussian noise}\\ \text{or}\\ \text{nonlinear functions}\end{array}\right\} \rightarrow \text{Numerical techniques}$$

- Extended Kalman filter *[Anderson & Moore, 1979]*

Suboptimal filtering methods

$$\left.\begin{array}{c}\text{Non-Gaussian noise}\\ \text{or}\\ \text{nonlinear functions}\end{array}\right\} \rightarrow \text{Numerical techniques}$$

- Extended Kalman filter *[Anderson & Moore, 1979]*
- Gaussian sum filters *[Alspach & Sorenson, 1972]*

Suboptimal filtering methods

$$\left.\begin{array}{c}\text{Non-Gaussian noise}\\ \text{or}\\ \text{nonlinear functions}\end{array}\right\} \rightarrow \text{Numerical techniques}$$

- Extended Kalman filter *[Anderson & Moore, 1979]*
- Gaussian sum filters *[Alspach & Sorenson, 1972]*
- Methods based on approximations of the first two moments of the densities *[Masreliez, 1975; West, 1985]*

Suboptimal filtering methods

$$\left.\begin{array}{c}\text{Non-Gaussian noise}\\ \text{or}\\ \text{nonlinear functions}\end{array}\right\} \rightarrow \text{Numerical techniques}$$

- Extended Kalman filter *[Anderson & Moore, 1979]*
- Gaussian sum filters *[Alspach & Sorenson, 1972]*
- Methods based on approximations of the first two moments of the densities *[Masreliez, 1975; West, 1985]*
- Evaluation of densities over grids *[Kitagawa, 1987; Sorenson, 1988]*

Suboptimal filtering methods

$$\left.\begin{array}{c}\text{Non-Gaussian noise}\\ \text{or}\\ \text{nonlinear functions}\end{array}\right\} \rightarrow \text{Numerical techniques}$$

- Extended Kalman filter *[Anderson & Moore, 1979]*
- Gaussian sum filters *[Alspach & Sorenson, 1972]*
- Methods based on approximations of the first two moments of the densities *[Masreliez, 1975; West, 1985]*
- Evaluation of densities over grids *[Kitagawa, 1987; Sorenson, 1988]*
- The unscented Kalman filter *[Julier et al., 2000]*

Suboptimal filtering methods

$$\left.\begin{array}{c}\text{Non-Gaussian noise}\\ \text{or}\\ \text{nonlinear functions}\end{array}\right\}\rightarrow \text{Numerical techniques}$$

- Extended Kalman filter *[Anderson & Moore, 1979]*
- Gaussian sum filters *[Alspach & Sorenson, 1972]*
- Methods based on approximations of the first two moments of the densities *[Masreliez, 1975; West, 1985]*
- Evaluation of densities over grids *[Kitagawa, 1987; Sorenson, 1988]*
- The unscented Kalman filter *[Julier et al., 2000]*

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way
- At one time instant the grid is composed of one set of nodes, and at the next time instant, this set of nodes is completely different

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way
- At one time instant the grid is composed of one set of nodes, and at the next time instant, this set of nodes is completely different
- The nodes are called *particles*, and they have assigned *weights*, which can be interpreted as probability masses

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way
- At one time instant the grid is composed of one set of nodes, and at the next time instant, this set of nodes is completely different
- The nodes are called *particles*, and they have assigned *weights*, which can be interpreted as probability masses
- The particles and the weights form a *discrete random measure*, which is used to approximate the densities of interest

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way
- At one time instant the grid is composed of one set of nodes, and at the next time instant, this set of nodes is completely different
- The nodes are called *particles*, and they have assigned *weights*, which can be interpreted as probability masses
- The particles and the weights form a *discrete random measure*, which is used to approximate the densities of interest
- In the generation of particles, each particle has a "parent", and the parent its own parent and so on

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way
- At one time instant the grid is composed of one set of nodes, and at the next time instant, this set of nodes is completely different
- The nodes are called *particles*, and they have assigned *weights*, which can be interpreted as probability masses
- The particles and the weights form a *discrete random measure*, which is used to approximate the densities of interest
- In the generation of particles, each particle has a "parent", and the parent its own parent and so on
- Such sequence of particles is called a *particle stream*

PF methodology

- PF is very similar in spirit as the one that exploits deterministic grids
- However, PF uses *random* grids, which means that the location of the nodes vary with time in a random way
- At one time instant the grid is composed of one set of nodes, and at the next time instant, this set of nodes is completely different
- The nodes are called *particles*, and they have assigned *weights*, which can be interpreted as probability masses
- The particles and the weights form a *discrete random measure*, which is used to approximate the densities of interest
- In the generation of particles, each particle has a “parent”, and the parent its own parent and so on
- Such sequence of particles is called a *particle stream*

PF philosophy

- A challenge in implementing PF is the placement of the nodes of the grids
- The generation of particles should be in regions of the state space over which the densities carry significant probability masses, avoiding parts of the state space with negligible probability masses

PF philosophy

- A challenge in implementing PF is the placement of the nodes of the grids
- The generation of particles should be in regions of the state space over which the densities carry significant probability masses, avoiding parts of the state space with negligible probability masses
- To that end, one exploits concepts from statistics known as *importance sampling* (IS) and *sampling-importance-resampling* (SIR)

PF philosophy

- A challenge in implementing PF is the placement of the nodes of the grids
- The generation of particles should be in regions of the state space over which the densities carry significant probability masses, avoiding parts of the state space with negligible probability masses
- To that end, one exploits concepts from statistics known as *importance sampling* (IS) and *sampling-importance-resampling* (SIR)
- Bayesian theory is applied to the computation of the particle weights

PF philosophy

- A challenge in implementing PF is the placement of the nodes of the grids
- The generation of particles should be in regions of the state space over which the densities carry significant probability masses, avoiding parts of the state space with negligible probability masses
- To that end, one exploits concepts from statistics known as *importance sampling* (IS) and *sampling-importance-resampling* (SIR)
- Bayesian theory is applied to the computation of the particle weights
- The sequential processing amounts to recursive updating of the discrete random measure with the arrival of new observations, where the updates correspond to generation of new particles and computation of their weights

PF philosophy

- A challenge in implementing PF is the placement of the nodes of the grids
- The generation of particles should be in regions of the state space over which the densities carry significant probability masses, avoiding parts of the state space with negligible probability masses
- To that end, one exploits concepts from statistics known as *importance sampling* (IS) and *sampling-importance-resampling* (SIR)
- Bayesian theory is applied to the computation of the particle weights
- The sequential processing amounts to recursive updating of the discrete random measure with the arrival of new observations, where the updates correspond to generation of new particles and computation of their weights

Some history

- The roots of PF were established about fifty years ago with methods for estimating the mean squared extension of a self-avoiding random walk on lattice spaces
- One of the first applications of the methodology was on the simulation of chain polymers

Some history

- The roots of PF were established about fifty years ago with methods for estimating the mean squared extension of a self-avoiding random walk on lattice spaces
- One of the first applications of the methodology was on the simulation of chain polymers
- The control community produced interesting work on sequential Monte Carlo integration methods in the sixties and seventies

Some history

- The roots of PF were established about fifty years ago with methods for estimating the mean squared extension of a self-avoiding random walk on lattice spaces
- One of the first applications of the methodology was on the simulation of chain polymers
- The control community produced interesting work on sequential Monte Carlo integration methods in the sixties and seventies
- The popularity of PF in the last fifteen years was triggered by the application of the SIR filter to tracking problems

Some history

- The roots of PF were established about fifty years ago with methods for estimating the mean squared extension of a self-avoiding random walk on lattice spaces
- One of the first applications of the methodology was on the simulation of chain polymers
- The control community produced interesting work on sequential Monte Carlo integration methods in the sixties and seventies
- The popularity of PF in the last fifteen years was triggered by the application of the SIR filter to tracking problems
- The timing (1993) was perfect because it came during a period when computing power started to become widely available

Some history

- The roots of PF were established about fifty years ago with methods for estimating the mean squared extension of a self-avoiding random walk on lattice spaces
- One of the first applications of the methodology was on the simulation of chain polymers
- The control community produced interesting work on sequential Monte Carlo integration methods in the sixties and seventies
- The popularity of PF in the last fifteen years was triggered by the application of the SIR filter to tracking problems
- The timing (1993) was perfect because it came during a period when computing power started to become widely available
- Ever since, the amount of work on particle filtering has proliferated and many important advances have been made

Some history

- The roots of PF were established about fifty years ago with methods for estimating the mean squared extension of a self-avoiding random walk on lattice spaces
- One of the first applications of the methodology was on the simulation of chain polymers
- The control community produced interesting work on sequential Monte Carlo integration methods in the sixties and seventies
- The popularity of PF in the last fifteen years was triggered by the application of the SIR filter to tracking problems
- The timing (1993) was perfect because it came during a period when computing power started to become widely available
- Ever since, the amount of work on particle filtering has proliferated and many important advances have been made

Why PF?

- One driving force for the advancement of PF is the ever increasing range of its applications

- PF can be applied to *any* state space model where the likelihood and the prior are computable up to proportionality constants

Why PF?

- One driving force for the advancement of PF is the ever increasing range of its applications
- PF can be applied to *any* state space model where the likelihood and the prior are computable up to proportionality constants
- Its accuracy depends on how well we generate the particles and how many particles we use to represent the random measure

Why PF?

- One driving force for the advancement of PF is the ever increasing range of its applications
- PF can be applied to *any* state space model where the likelihood and the prior are computable up to proportionality constants
- Its accuracy depends on how well we generate the particles and how many particles we use to represent the random measure
- It is well known that PF is computationally intensive which is an issue because in sequential signal processing the required operations must be completed before a deadline

Why PF?

- One driving force for the advancement of PF is the ever increasing range of its applications
- PF can be applied to *any* state space model where the likelihood and the prior are computable up to proportionality constants
- Its accuracy depends on how well we generate the particles and how many particles we use to represent the random measure
- It is well known that PF is computationally intensive which is an issue because in sequential signal processing the required operations must be completed before a deadline
- However, PF allows for significant parallelization of its operations

Why PF?

- One driving force for the advancement of PF is the ever increasing range of its applications
- PF can be applied to *any* state space model where the likelihood and the prior are computable up to proportionality constants
- Its accuracy depends on how well we generate the particles and how many particles we use to represent the random measure
- It is well known that PF is computationally intensive which is an issue because in sequential signal processing the required operations must be completed before a deadline
- However, PF allows for significant parallelization of its operations

Applications of PF

- Target tracking, positioning and navigation
- Robotics

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)
- Speech processing for speech enhancement

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)
- Speech processing for speech enhancement
- Seismic signal processing for blind deconvolution

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)
- Speech processing for speech enhancement
- Seismic signal processing for blind deconvolution
- System engineering for fault detection

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)
- Speech processing for speech enhancement
- Seismic signal processing for blind deconvolution
- System engineering for fault detection
- Computer vision

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)
- Speech processing for speech enhancement
- Seismic signal processing for blind deconvolution
- System engineering for fault detection
- Computer vision
- Econometrics

Applications of PF

- Target tracking, positioning and navigation
- Robotics
- Communications (detection in flat fading, equalization and synchronization)
- Speech processing for speech enhancement
- Seismic signal processing for blind deconvolution
- System engineering for fault detection
- Computer vision
- Econometrics

Outline

Representation of dynamic systems

Many problems can be stated in terms of *dynamic systems*

1. *State equation*: hidden random signal to be estimated

$$\mathbf{x}_t = f\left(\mathbf{x}_{t-1}, \mathbf{u}_t\right)$$

$\mathbf{x}_t$	signal (state) of interest
f	state transition function (possibly nonlinear)
$\mathbf{u}_t$	state perturbation noise

2. *Observation equation*: available information

$$\mathbf{y}_t = g\left(\mathbf{x}_t, \mathbf{v}_t\right)$$

$\mathbf{y}_t$	observed signal
g	measurement function (possibly nonlinear)
$\mathbf{v}_t$	observation noise

Representation of dynamic systems

Many problems can be stated in terms of *dynamic systems*

1. *State equation*: hidden random signal to be estimated

$$\mathbf{x}_t = f\left(\mathbf{x}_{t-1}, \mathbf{u}_t\right)$$

$\mathbf{x}_t$	signal (state) of interest
f	state transition function (possibly nonlinear)
$\mathbf{u}_t$	state perturbation noise

2. *Observation equation*: available information

$$\mathbf{y}_t = g\left(\mathbf{x}_t, \mathbf{v}_t\right)$$

$\mathbf{y}_t$	observed signal
g	measurement function (possibly nonlinear)
$\mathbf{v}_t$	observation noise

Alternative representation of dynamic systems

One can also represent the dynamic system in terms of *densities*

1. *State equation*: $p\left(\mathbf{x}_t|\mathbf{x}_{t-1},\boldsymbol{\theta}\right)$
2. *Observation equation*: $p\left(\mathbf{y}_t|\mathbf{x}_t,\boldsymbol{\psi}\right)$

The form of the density functions depends on:

- the functions $f(\cdot)$ and $g(\cdot)$
- the densities of $\mathbf{u}_t$ and $\mathbf{v}_t$

$$\left.\begin{array}{c}\boldsymbol{\theta}\in\Re^{L_\theta}\\ \boldsymbol{\psi}\in\Re^{L_\psi}\end{array}\right\}\longrightarrow\begin{array}{l}\text{Unknown fixed parameter vectors}\\ \text{They can be included in } \mathbf{x}_t \text{ as static parameters}\end{array}$$

Important densities

Three densities play a critical role in *sequential signal processing*

1. *Filtering density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$, where $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t\}$
2. *Predictive density*: $p\left(\mathbf{x}_{t+l}|\mathbf{y}_{1:t}\right)$, where $l \geq 1$

Important densities

Three densities play a critical role in *sequential signal processing*

1. *Filtering density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$, where $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t\}$
2. *Predictive density*: $p\left(\mathbf{x}_{t+l}|\mathbf{y}_{1:t}\right)$, where $l \geq 1$
3. *Smoothing density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:T}\right)$, where $T > t$ and $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T\}$

Important densities

Three densities play a critical role in *sequential signal processing*

1. *Filtering density*: $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, where $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t\}$
2. *Predictive density*: $p(\mathbf{x}_{t+l}|\mathbf{y}_{1:t})$, where $l \geq 1$
3. *Smoothing density*: $p(\mathbf{x}_t|\mathbf{y}_{1:T})$, where $T > t$ and $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T\}$

Tracking the important densities

Objective: Track the densities by exploiting *recursive relationships*

1. *Filtering density*: $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ from $p\left(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}\right)$
2. *Predictive density*: $p(\mathbf{x}_{t+l}|\mathbf{y}_{1:t})$ from $p(\mathbf{x}_{t+l-1}|\mathbf{y}_{1:t})$
3. *Smoothing density*: $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ from $p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})$

- Complete information about $\mathbf{x}_t$ is given by these densities

Tracking the important densities

Objective: Track the densities by exploiting *recursive relationships*

1. *Filtering density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$ from $p\left(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}\right)$
2. *Predictive density*: $p\left(\mathbf{x}_{t+l}|\mathbf{y}_{1:t}\right)$ from $p\left(\mathbf{x}_{t+l-1}|\mathbf{y}_{1:t}\right)$
3. *Smoothing density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:T}\right)$ from $p\left(\mathbf{x}_{t+1}|\mathbf{y}_{1:T}\right)$

- Complete information about $\mathbf{x}_t$ is given by these densities
- An algorithm that can track these densities exactly is called an *optimal algorithm*

Tracking the important densities

Objective: Track the densities by exploiting *recursive relationships*

1. *Filtering density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$ from $p\left(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}\right)$
2. *Predictive density*: $p\left(\mathbf{x}_{t+l}|\mathbf{y}_{1:t}\right)$ from $p\left(\mathbf{x}_{t+l-1}|\mathbf{y}_{1:t}\right)$
3. *Smoothing density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:T}\right)$ from $p\left(\mathbf{x}_{t+1}|\mathbf{y}_{1:T}\right)$

- Complete information about $\mathbf{x}_t$ is given by these densities
- An algorithm that can track these densities exactly is called an *optimal algorithm*
- In many practical situations, optimal algorithms are impossible to implement

Tracking the important densities

Objective: Track the densities by exploiting *recursive relationships*

1. *Filtering density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$ from $p\left(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}\right)$
2. *Predictive density*: $p\left(\mathbf{x}_{t+l}|\mathbf{y}_{1:t}\right)$ from $p\left(\mathbf{x}_{t+l-1}|\mathbf{y}_{1:t}\right)$
3. *Smoothing density*: $p\left(\mathbf{x}_t|\mathbf{y}_{1:T}\right)$ from $p\left(\mathbf{x}_{t+1}|\mathbf{y}_{1:T}\right)$

- Complete information about $\mathbf{x}_t$ is given by these densities
- An algorithm that can track these densities exactly is called an *optimal algorithm*
- In many practical situations, optimal algorithms are impossible to implement

Example: Bearings-only tracking

Consider the tracking of an object based on bearings-only measurements

$$\mathbf{x}_t = \mathbf{G}_x \mathbf{x}_{t-1} + \mathbf{G}_u \mathbf{u}_t$$

where $\mathbf{x}_t = [x_{1,t} \;\; x_{2,t} \;\; \dot{x}_{1,t} \;\; \dot{x}_{2,t}]^\top$

Bearings-only range measurements are obtained by J sensors placed at known locations in the sensor field

$$y_{j,t} = \arctan\left(\frac{x_{2,t} - l_{2,j}}{x_{1,t} - l_{1,j}}\right) + v_t(n)$$

- Given the measurements of J sensors, $\mathbf{y}_t = [y_{1,t} \; y_{2,t} \;\; \cdots \;\; y_{J,t}]^\top$, and the movement model of the object, the goal is to track the object in time, that is, estimate its position and velocity

Example: Bearings-only tracking

Consider the tracking of an object based on bearings-only measurements

$$\mathbf{x}_t = \mathbf{G}_x \mathbf{x}_{t-1} + \mathbf{G}_u \mathbf{u}_t$$

where $\mathbf{x}_t = [x_{1,t} \;\; x_{2,t} \;\; \dot{x}_{1,t} \;\; \dot{x}_{2,t}]^\top$

Bearings-only range measurements are obtained by J sensors placed at known locations in the sensor field

$$y_{j,t} = \arctan\left(\frac{x_{2,t} - l_{2,j}}{x_{1,t} - l_{1,j}}\right) + v_t(n)$$

- Given the measurements of J sensors, $\mathbf{y}_t = [y_{1,t} \; y_{2,t} \;\; \cdots \;\; y_{J,t}]^\top$, and the movement model of the object, the goal is to track the object in time, that is, estimate its position and velocity
- In the literature this problem is known as *target tracking*

Example: Bearings-only tracking

Consider the tracking of an object based on bearings-only measurements

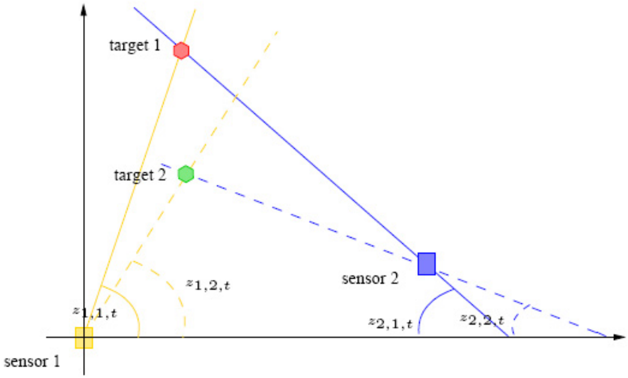
$$\mathbf{x}_t = \mathbf{G}_x \mathbf{x}_{t-1} + \mathbf{G}_u \mathbf{u}_t$$

where $\mathbf{x}_t = [x_{1,t} \;\; x_{2,t} \;\; \dot{x}_{1,t} \;\; \dot{x}_{2,t}]^\top$

Bearings-only range measurements are obtained by J sensors placed at known locations in the sensor field

$$y_{j,t} = \arctan\left(\frac{x_{2,t} - l_{2,j}}{x_{1,t} - l_{1,j}}\right) + v_t(n)$$

- Given the measurements of J sensors, $\mathbf{y}_t = [y_{1,t} \; y_{2,t} \;\; \cdots \;\; y_{J,t}]^\top$, and the movement model of the object, the goal is to track the object in time, that is, estimate its position and velocity
- In the literature this problem is known as *target tracking*



A target in a two-dimensional space and two BOT sensors

Recursion for filtering

- Suppose that at time $t-1$, we know the observations $\mathbf{y}_{1:t-1}$ and the a posteriori PDF $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
- Once $\mathbf{y}_t$ becomes available, we update the PDF

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \quad \propto \quad p(\mathbf{y}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

Recursion for filtering

- Suppose that at time $t-1$, we know the observations $\mathbf{y}_{1:t-1}$ and the a posteriori PDF $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
- Once $\mathbf{y}_t$ becomes available, we update the PDF

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \quad \propto \quad p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

- The *predictive density* can be obtained as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\; p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$$

Recursion for filtering

- Suppose that at time $t-1$, we know the observations $\mathbf{y}_{1:t-1}$ and the a posteriori PDF $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
- Once $\mathbf{y}_t$ becomes available, we update the PDF

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

- The *predictive density* can be obtained as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\; p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$$

- The recursive equation for updating the filtering density becomes

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t}) &\propto p(\mathbf{y}_t|\mathbf{x}_t) \\ &\times \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\; p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1} \end{aligned}$$

Recursion for filtering

- Suppose that at time $t-1$, we know the observations $\mathbf{y}_{1:t-1}$ and the a posteriori PDF $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
- Once $\mathbf{y}_t$ becomes available, we update the PDF

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \quad \propto \quad p(\mathbf{y}_t|\mathbf{x}_t)\,p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

- The *predictive density* can be obtained as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\; p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\,d\mathbf{x}_{t-1}$$

- The recursive equation for updating the filtering density becomes

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t}) \quad &\propto \quad p(\mathbf{y}_t|\mathbf{x}_t) \\ &\times \quad \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\; p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\,d\mathbf{x}_{t-1} \end{aligned}$$

Carrying out the filtering recursion

There are at least two problems in carrying out the above recursion

1. Solving the integral in order to obtain the predictive density

Carrying out the filtering recursion

There are at least two problems in carrying out the above recursion

1. Solving the integral in order to obtain the predictive density
2. Combining the likelihood and the predictive density in order to get the updated filtering density

Carrying out the filtering recursion

There are at least two problems in carrying out the above recursion

1. Solving the integral in order to obtain the predictive density
2. Combining the likelihood and the predictive density in order to get the updated filtering density

- In many problems the recursive evaluation of the densities of the state space model cannot be done analytically, which entails that we have to resort to *numerical methods*

Carrying out the filtering recursion

There are at least two problems in carrying out the above recursion

1. Solving the integral in order to obtain the predictive density
2. Combining the likelihood and the predictive density in order to get the updated filtering density

- In many problems the recursive evaluation of the densities of the state space model cannot be done analytically, which entails that we have to resort to *numerical methods*
- PF can be employed with elegance and with performance characterized by high accuracy

Carrying out the filtering recursion

There are at least two problems in carrying out the above recursion

1. Solving the integral in order to obtain the predictive density
2. Combining the likelihood and the predictive density in order to get the updated filtering density

- In many problems the recursive evaluation of the densities of the state space model cannot be done analytically, which entails that we have to resort to *numerical methods*

- PF can be employed with elegance and with performance characterized by high accuracy

Outline

Approximation of densities by discrete random measures

Densities can be approximated by *discrete random measures*

$$\chi = \left\{ \mathbf{x}_t^{(m)}, w_t^{(m)} \right\}_{m=1}^{M}$$

- The discrete random measure approximates the density by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(m)}\right)$$

Approximation of densities by discrete random measures

Densities can be approximated by *discrete random measures*

$$\chi = \left\{\mathbf{x}_t^{(m)}, w_t^{(m)}\right\}_{m=1}^{M}$$

- The discrete random measure approximates the density by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(m)}\right)$$

- Computations of expectations simplify to summations

$$E\left(h(\mathbf{X}_t)\right) = \int h(\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t$$

$$\Downarrow$$

$$E\left(h(\mathbf{X}_t)\right) \approx \sum_{m=1}^{M} w_t^{(m)} h\left(\mathbf{x}_t^{(m)}\right)$$

Approximation of densities by discrete random measures

Densities can be approximated by *discrete random measures*

$$\chi = \left\{ \mathbf{x}_t^{(m)}, w_t^{(m)} \right\}_{m=1}^{M}$$

- The discrete random measure approximates the density by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(m)}\right)$$

- Computations of expectations simplify to summations

$$\begin{aligned} E\left(h(\mathbf{X}_t)\right) &= \int h(\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t \\ &\Downarrow \\ E\left(h(\mathbf{X}_t)\right) &\approx \sum_{m=1}^{M} w_t^{(m)} h\left(\mathbf{x}_t^{(m)}\right) \end{aligned}$$

Example of approximation of densities

Assume that independent particles, $\mathbf{x}_t^{(m)}$, can be drawn from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

- All the particles have the same weight

$$E\left(h(\mathbf{X}_t)\right) = \int h(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t \rightarrow \widehat{E}\left(h(\mathbf{X}_t)\right) = \frac{1}{M}\sum_{m=1}^{M} h\left(\mathbf{x}_t^{(m)}\right)$$

Example of approximation of densities

Assume that independent particles, $\mathbf{x}_t^{(m)}$, can be drawn from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

- All the particles have the same weight

$$E\left(h(\mathbf{X}_t)\right) = \int h(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t \rightarrow \widehat{E}\left(h(\mathbf{X}_t)\right) = \frac{1}{M}\sum_{m=1}^{M} h\left(\mathbf{x}_t^{(m)}\right)$$

- $\widehat{E}(\cdot)$ is an unbiased estimator of the conditional expectation $E(\cdot)$

Example of approximation of densities

Assume that independent particles, $\mathbf{x}_t^{(m)}$, can be drawn from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

- All the particles have the same weight

$$E\left(h(\mathbf{X}_t)\right) = \int h(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t \rightarrow \widehat{E}\left(h(\mathbf{X}_t)\right) = \frac{1}{M}\sum_{m=1}^{M} h\left(\mathbf{x}_t^{(m)}\right)$$

- $\widehat{E}(\cdot)$ is an unbiased estimator of the conditional expectation $E(\cdot)$
- If the variance $\sigma_h^2 < \infty \rightarrow$ the variance of $\widehat{E}(\cdot)$ is $\sigma^2_{\widehat{E}(h(\cdot))} = \frac{\sigma_h^2}{M}$

Example of approximation of densities

Assume that independent particles, $\mathbf{x}_t^{(m)}$, can be drawn from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

- All the particles have the same weight

$$E\left(h(\mathbf{X}_t)\right) = \int h(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t \rightarrow \widehat{E}\left(h(\mathbf{X}_t)\right) = \frac{1}{M}\sum_{m=1}^{M} h\left(\mathbf{x}_t^{(m)}\right)$$

- $\widehat{E}(\cdot)$ is an unbiased estimator of the conditional expectation $E(\cdot)$
- If the variance $\sigma_h^2 < \infty \rightarrow$ the variance of $\widehat{E}(\cdot)$ is $\sigma^2_{\widehat{E}(h(\cdot))} = \frac{\sigma_h^2}{M}$
- As $M \longrightarrow \infty$

Strong law of large numbers	*Central limit theorem*
$\widehat{E}(h(\mathbf{X}_t)) \xrightarrow{a.s.} E(h(\mathbf{X}_t))$	$\widehat{E}(h(\mathbf{X}_t)) \xrightarrow{d} \mathcal{N}\left(E(h(\mathbf{X}_t)), \frac{\sigma_h^2}{M}\right)$

Example of approximation of densities

Assume that independent particles, $\mathbf{x}_t^{(m)}$, can be drawn from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

- All the particles have the same weight

$$E\left(h(\mathbf{X}_t)\right) = \int h(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t \rightarrow \widehat{E}\left(h(\mathbf{X}_t)\right) = \frac{1}{M}\sum_{m=1}^{M} h\left(\mathbf{x}_t^{(m)}\right)$$

- $\widehat{E}(\cdot)$ is an unbiased estimator of the conditional expectation $E(\cdot)$
- If the variance $\sigma_h^2 < \infty \rightarrow$ the variance of $\widehat{E}(\cdot)$ is $\sigma^2_{\widehat{E}(h(\cdot))} = \frac{\sigma_h^2}{M}$
- As $M \longrightarrow \infty$

Strong law of large numbers	*Central limit theorem*
$\widehat{E}(h(\mathbf{X}_t)) \xrightarrow{a.s.} E(h(\mathbf{X}_t))$	$\widehat{E}(h(\mathbf{X}_t)) \xrightarrow{d} \mathcal{N}\left(E(h(\mathbf{X}_t)), \frac{\sigma_h^2}{M}\right)$

The concept of importance sampling (IS)

- In practice, we often cannot draw samples directly from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$
- Particles are drawn from $\pi(\mathbf{x}_t)$ and the estimate of $E\left(h(\mathbf{X}_t)\right)$ becomes

$$E\left(h(\mathbf{X}_t)\right) \approx \frac{1}{M}\sum_{m=1}^{M} w_t^{*(m)} h\left(\mathbf{x}_t^{(m)}\right)$$

or

$$E\left(h(\mathbf{X}_t)\right) \approx \sum_{m=1}^{M} w_t^{(m)} h\left(\mathbf{x}_t^{(m)}\right)$$

where $w_t^{*(m)} = \frac{p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_t^{(m)})}$ and $w_t^{(m)} = \frac{w_t^{*(m)}}{\sum_{i=1}^{M} w_t^{*(i)}}$

The concept of importance sampling (IS)

- In practice, we often cannot draw samples directly from $p(\mathbf{x}_t|\mathbf{y}_{1:t})$
- Particles are drawn from $\pi(\mathbf{x}_t)$ and the estimate of $E\left(h(\mathbf{X}_t)\right)$ becomes

$$E\left(h(\mathbf{X}_t)\right) \approx \frac{1}{M}\sum_{m=1}^{M} w_t^{*(m)} h\left(\mathbf{x}_t^{(m)}\right)$$

or

$$E\left(h(\mathbf{X}_t)\right) \approx \sum_{m=1}^{M} w_t^{(m)} h\left(\mathbf{x}_t^{(m)}\right)$$

where $w_t^{*(m)} = \frac{p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_t^{(m)})}$ and $w_t^{(m)} = \frac{w_t^{*(m)}}{\sum_{i=1}^{M} w_t^{*(i)}}$

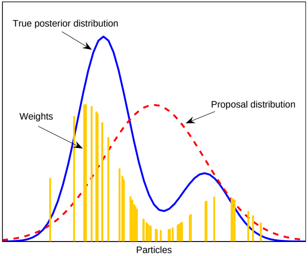
Summary

We use a random measure to approximate $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ in two steps:

1. Drawing samples from a proposal function $\pi(\mathbf{x}_t)$, which needs to be known only up to a multiplicative constant, i.e.,

$$\mathbf{x}_t^{(m)} \sim \pi(\mathbf{x}_t), \quad m = 1, 2, \cdots, M$$

2. Computing the weights of the particles $w_t^{(m)}$



How do we obtain χ_t from χ_{t-1}?

We assume $\chi_{t-1} = \left\{ \mathbf{x}_{t-1}^{(m)}, w_{t-1}^{(m)} \right\}_{m=1}^{M}$ is given.

- *Step one: Generation of new particles from a proposal function, i.e.,*

$$\mathbf{x}_t^{(m)} \sim \pi(\mathbf{x}_t), \quad m = 1, \cdots, M$$

  and augmentation of the particle stream $\mathbf{x}_{0:t-1}^{(m)}$ with $\mathbf{x}_t^{(m)}$

- *Step two: Computation of the particle weights of* $\mathbf{x}_t^{(m)}$, $m = 1, \cdots, M$, *i.e.*

$$w_t^{(m)} \propto w_{t-1}^{(m)} \times \text{update factor}, \quad m = 1, \cdots, M$$

  and normalization of the weights

How do we obtain χ_t from χ_{t-1}?

We assume $\chi_{t-1} = \left\{ \mathbf{x}_{t-1}^{(m)}, w_{t-1}^{(m)} \right\}_{m=1}^{M}$ is given.

- *Step one: Generation of new particles from a proposal function, i.e.,*

$$\mathbf{x}_t^{(m)} \sim \pi\left(\mathbf{x}_t\right), \quad m = 1, \cdots, M$$

  and augmentation of the particle stream $\mathbf{x}_{0:t-1}^{(m)}$ with $\mathbf{x}_t^{(m)}$

- *Step two: Computation of the particle weights of* $\mathbf{x}_t^{(m)}$, $m = 1, \cdots, M$, *i.e.*

$$w_t^{(m)} \propto w_{t-1}^{(m)} \times \text{update factor}, \quad m = 1, \cdots, M$$

  and normalization of the weights

Mathematical formulation of the algorithm

Particle generation

Basis

$$\pi\left(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}\right) = \pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t}\right)\pi\left(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}\right)$$

$$\mathbf{x}_{0:t-1}^{(m)} \sim \pi\left(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}\right)$$

$$w_{t-1}^{(m)} \propto \frac{p(\mathbf{x}_{0:t-1}^{(m)}|\mathbf{y}_{t-1})}{\pi(\mathbf{x}_{0:t-1}^{(m)}|\mathbf{y}_{1:t-1})}$$

Augmentation of the trajectory $\mathbf{x}_{0:t-1}^{(m)}$ with $\mathbf{x}_t^{(m)}$

$$\mathbf{x}_t^{(m)} \sim \pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)},\mathbf{y}_{1:t}\right), \quad m = 1,\cdots,M$$

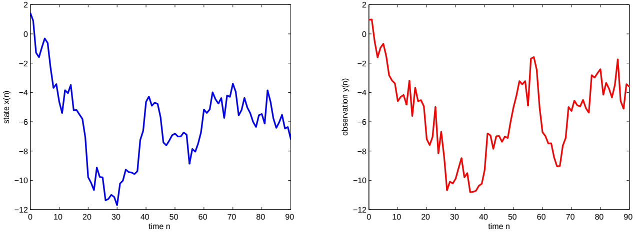
Weight update

$$w_t^{(m)} \propto \frac{p\left(\mathbf{y}_t|\mathbf{x}_t^{(m)}\right)p\left(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)}\right)}{\pi\left(\mathbf{x}_t^{(m)}|\mathbf{x}_{0:t-1}^{(m)},\mathbf{y}_{1:t}\right)}w_{t-1}^{(m)}, \quad m = 1,\cdots,M$$

Example: random walk

Consider the state-space model

$$
\begin{aligned}
x_t &= x_{t-1} + u_t \\
y_t &= x_t + v_t
\end{aligned}
$$



	$t = 0$	$t = 1$	$t = 2$
x_t	1.44	0.89	-1.28
y_t	0.96	0.98	-0.51

Example: SIR for random walk

- We choose as a proposal function

$$\pi(x_t) = p(x_t|x_{t-1})$$

- Therefore, we generate the particles according to

$$x_t^{(m)} \sim p(x_t|x_{t-1}^{(m)}), \quad m = 1, 2, \cdots, M$$

Example: SIR for random walk

- We choose as a proposal function

$$\pi(x_t) = p(x_t|x_{t-1})$$

- Therefore, we generate the particles according to

$$x_t^{(m)} \sim p(x_t|x_{t-1}^{(m)}), \quad m = 1, 2, \cdots, M$$

- The update of the weights is done according to

$$w_t^{(m)} \propto p(y_t|x_t^{(m)})w_{t-1}^{(m)}$$

Example: SIR for random walk

- We choose as a proposal function

$$\pi(x_t) = p(x_t|x_{t-1})$$

- Therefore, we generate the particles according to

$$x_t^{(m)} \sim p(x_t|x_{t-1}^{(m)}), \quad m = 1, 2, \cdots, M$$

- The update of the weights is done according to

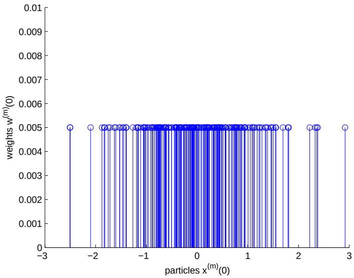
$$w_t^{(m)} \propto p(y_t|x_t^{(m)})w_{t-1}^{(m)}$$

Example: Step by step SIR for random walk

Initialization: $t = 0$

$$x_0^{(m)} \sim \mathcal{N}(0,1) \qquad m = 1, 2, \ldots, M.$$

Note that all the weights are equal



Example: Step by step SIR for random walk (cont.)

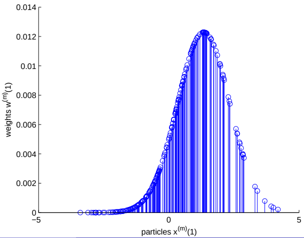
Time instant: $t = 1$

- Generation of particles using the proposal function:

$$x_1^{(m)} \sim p\left(x_1 | x_0^{(m)}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_1 - x_0^{(m)})^2}{2}\right)$$

- Weight update:

$$w_1^{(m)} \propto \exp\left(-\frac{(y_1 - x_1^{(m)})^2}{2}\right)$$



Example: Step by step SIR for random walk (cont.)

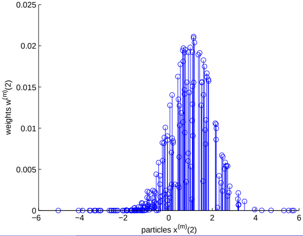
Time instant: $t = 2$

- Generation of particles using the proposal function:

$$x_2^{(m)} \sim p\left(x_2 | x_1^{(m)}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_2 - x_1^{(m)})^2}{2}\right)$$

- Weight update:

$$w_2^{(m)} \propto w_1^{(m)} \exp\left(-\frac{(y_2 - x_2^{(m)})^2}{2}\right)$$



Choice of proposal function

- The proposal (importance) function plays a crucial role in the performance of PF
- It is desirable to use easy-to-sample proposal functions that produce particles with a large enough variance in order to avoid exploration of the state space in too narrow regions and thereby contributing to losing the tracks of the state, but not too large to alleviate generation of too dispersed particles

Choice of proposal function

- The proposal (importance) function plays a crucial role in the performance of PF
- It is desirable to use easy-to-sample proposal functions that produce particles with a large enough variance in order to avoid exploration of the state space in too narrow regions and thereby contributing to losing the tracks of the state, but not too large to alleviate generation of too dispersed particles
- The support of the proposal functions has to be the same as that of the targeted distribution

Choice of proposal function

- The proposal (importance) function plays a crucial role in the performance of PF
- It is desirable to use easy-to-sample proposal functions that produce particles with a large enough variance in order to avoid exploration of the state space in too narrow regions and thereby contributing to losing the tracks of the state, but not too large to alleviate generation of too dispersed particles
- The support of the proposal functions has to be the same as that of the targeted distribution

The optimal importance function

The optimal choice for the importance function is the true posterior distribution, i.e.,

$$\pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)}, \mathbf{y}_{1:t}\right) = p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \mathbf{y}_t\right)$$

This choice corresponds to the weight calculation

$$w_t^{(m)} \propto w_{t-1}^{(m)} p\left(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}\right)$$

- Advantage: this importance function minimizes the variance of the weights

The optimal importance function

The optimal choice for the importance function is the true posterior distribution, i.e.,

$$\pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)},\mathbf{y}_{1:t}\right) = p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)},\mathbf{y}_t\right)$$

This choice corresponds to the weight calculation

$$w_t^{(m)} \propto w_{t-1}^{(m)} p\left(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}\right)$$

- Advantage: this importance function minimizes the variance of the weights
- Disadvantage: it is difficult to sample from $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)},\mathbf{y}_t\right)$ and the update of the weights requires the computation of the integral

$$p\left(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(m)}\right) = \int p(\mathbf{y}_t \mid \mathbf{x}_t)p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(m)}\right) d\mathbf{x}_t$$

The optimal importance function

The optimal choice for the importance function is the true posterior distribution, i.e.,

$$\pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)},\mathbf{y}_{1:t}\right)=p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)},\mathbf{y}_t\right)$$

This choice corresponds to the weight calculation

$$w_t^{(m)}\propto w_{t-1}^{(m)}p\left(\mathbf{y}_t|\mathbf{x}_{t-1}^{(m)}\right)$$

- Advantage: this importance function minimizes the variance of the weights
- Disadvantage: it is difficult to sample from $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)},\mathbf{y}_t\right)$ and the update of the weights requires the computation of the integral

$$p\left(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(m)}\right)=\int p(\mathbf{y}_t \mid \mathbf{x}_t)p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(m)}\right)d\mathbf{x}_t$$

The prior importance function

A popular choice for the importance function is the prior

$$\pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)}, \mathbf{y}_{1:t}\right) = p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}\right)$$

This choice corresponds to the weight calculation

$$w_t^{(m)} \propto w_{t-1}^{(m)}\, p\left(\mathbf{y}_t|\mathbf{x}_t^{(m)}\right)$$

- Advantage: the computation of the weights is easy

The prior importance function

A popular choice for the importance function is the prior

$$\pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)}, \mathbf{y}_{1:t}\right) = p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}\right)$$

This choice corresponds to the weight calculation

$$w_t^{(m)} \propto w_{t-1}^{(m)}\, p\left(\mathbf{y}_t|\mathbf{x}_t^{(m)}\right)$$

- Advantage: the computation of the weights is easy
- Disadvantage: the generation of the particles is implemented without the use of observations

The prior importance function

A popular choice for the importance function is the prior

$$\pi\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(m)}, \mathbf{y}_{1:t}\right) = p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}\right)$$

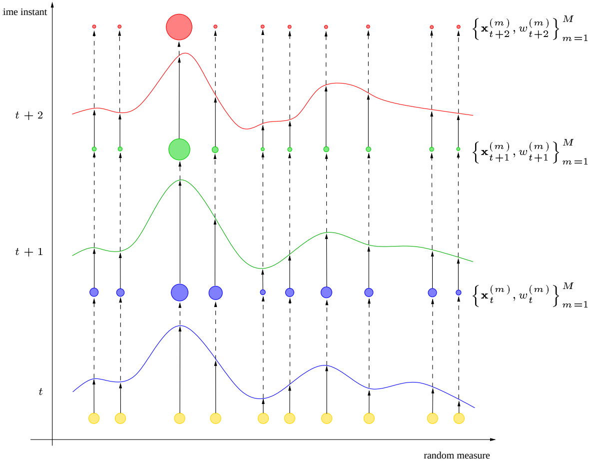
This choice corresponds to the weight calculation

$$w_t^{(m)} \propto w_{t-1}^{(m)}\, p\left(\mathbf{y}_t|\mathbf{x}_t^{(m)}\right)$$

- Advantage: the computation of the weights is easy
- Disadvantage: the generation of the particles is implemented without the use of observations

Degeneracy of the random measure: Resampling

In particle filtering the discrete random measure degenerates quickly and only few particles are assigned meaningful weights

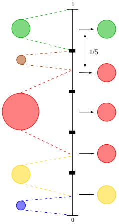


Resampling

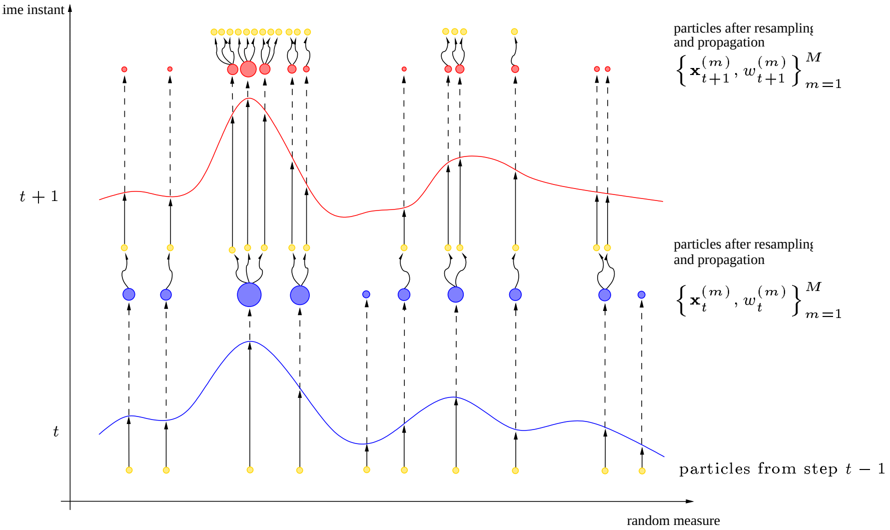
- A measure of this degeneracy is the *effective particle size*

$$M_{\text{eff}} = \frac{M}{1 + \mathrm{Var}\left(w_t^{*(m)}\right)} \quad \rightarrow \quad \widehat{M}_{\text{eff}} = \frac{1}{\sum_{m=1}^{M}\left(w_t^{(m)}\right)^2}$$

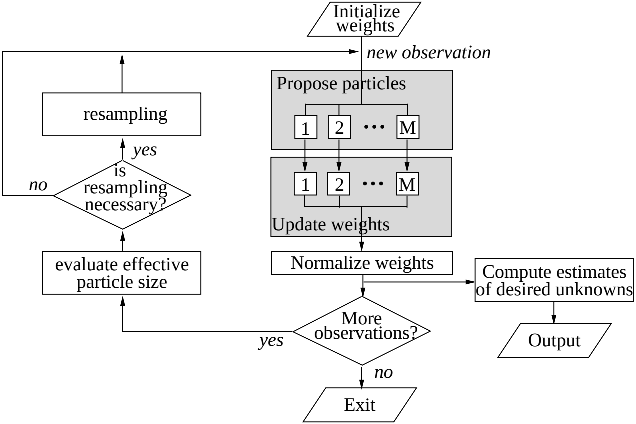
- Degeneracy is reduced by using good importance sampling functions and resampling
- Resampling eliminates particles with small weights and replicates the ones with large weights



Pictorial description of resampling



Flowchart of the SIR algorithm



Outline

SIR PF

- The SIR method is the simplest of all the particle filtering methods
- It was named bootstrap filter

SIR PF

- The SIR method is the simplest of all the particle filtering methods
- It was named bootstrap filter
- It employs the prior density for drawing particles, which implies that the weights are only proportional to the likelihood of the drawn particles

SIR PF

- The SIR method is the simplest of all the particle filtering methods
- It was named bootstrap filter
- It employs the prior density for drawing particles, which implies that the weights are only proportional to the likelihood of the drawn particles

Auxiliary particle filtering (APF)

It explores the state space by using the latest measurements.

- Particles are propagated using the prior
- Weights of the newly generated particles are evaluated by the latest measurement

Auxiliary particle filtering (APF)

It explores the state space by using the latest measurements.

- Particles are propagated using the prior
- Weights of the newly generated particles are evaluated by the latest measurement
- Resampling is applied using the so computed weights

Auxiliary particle filtering (APF)

It explores the state space by using the latest measurements.

- Particles are propagated using the prior
- Weights of the newly generated particles are evaluated by the latest measurement
- Resampling is applied using the so computed weights
- The resampled streams are propagated by generating new particles

Auxiliary particle filtering (APF)

It explores the state space by using the latest measurements.

- Particles are propagated using the prior
- Weights of the newly generated particles are evaluated by the latest measurement
- Resampling is applied using the so computed weights
- The resampled streams are propagated by generating new particles
- The weights of the new particles are computed

Auxiliary particle filtering (APF)

It explores the state space by using the latest measurements.

- Particles are propagated using the prior
- Weights of the newly generated particles are evaluated by the latest measurement
- Resampling is applied using the so computed weights
- The resampled streams are propagated by generating new particles
- The weights of the new particles are computed

Unscented particle filtering (UPF)

- The method is based on the unscented transform
- Rather than linearizing the model, the UKF uses a small deterministic grid of points (called sigma points) which when propagated through the system capture accurately both the mean and the covariance

Unscented particle filtering (UPF)

- The method is based on the unscented transform
- Rather than linearizing the model, the UKF uses a small deterministic grid of points (called sigma points) which when propagated through the system capture accurately both the mean and the covariance
- In the context of particle filtering, the UKF computes Gaussians which are then used as importance functions

Unscented particle filtering (UPF)

- The method is based on the unscented transform
- Rather than linearizing the model, the UKF uses a small deterministic grid of points (called sigma points) which when propagated through the system capture accurately both the mean and the covariance
- In the context of particle filtering, the UKF computes Gaussians which are then used as importance functions
- Each particle stream has its own UKF

Unscented particle filtering (UPF)

- The method is based on the unscented transform
- Rather than linearizing the model, the UKF uses a small deterministic grid of points (called sigma points) which when propagated through the system capture accurately both the mean and the covariance
- In the context of particle filtering, the UKF computes Gaussians which are then used as importance functions
- Each particle stream has its own UKF

Gaussian particle filtering (GPF)

- The posterior and the predictive densities are approximated with Gaussians as is done by the extended Kalman filter (EKF)
- However, unlike with the EKF, we do not linearize any of the nonlinearities in the system

Gaussian particle filtering (GPF)

- The posterior and the predictive densities are approximated with Gaussians as is done by the extended Kalman filter (EKF)
- However, unlike with the EKF, we do not linearize any of the nonlinearities in the system
- Its advantage over other PF methods is that it does not require resampling

Gaussian particle filtering (GPF)

- The posterior and the predictive densities are approximated with Gaussians as is done by the extended Kalman filter (EKF)
- However, unlike with the EKF, we do not linearize any of the nonlinearities in the system
- Its advantage over other PF methods is that it does not require resampling
- Another advantage is that the estimation of constant parameters is not a problem as is the case with other PF methods

Gaussian particle filtering (GPF)

- The posterior and the predictive densities are approximated with Gaussians as is done by the extended Kalman filter (EKF)
- However, unlike with the EKF, we do not linearize any of the nonlinearities in the system
- Its advantage over other PF methods is that it does not require resampling
- Another advantage is that the estimation of constant parameters is not a problem as is the case with other PF methods
- Its disadvantage is that if the approximated densities are not Gaussians, the estimates may be inaccurate and the filter may diverge

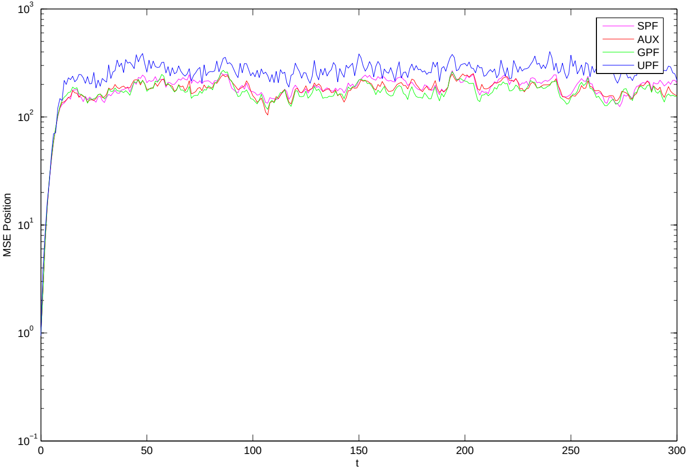
Gaussian particle filtering (GPF)

- The posterior and the predictive densities are approximated with Gaussians as is done by the extended Kalman filter (EKF)
- However, unlike with the EKF, we do not linearize any of the nonlinearities in the system
- Its advantage over other PF methods is that it does not require resampling
- Another advantage is that the estimation of constant parameters is not a problem as is the case with other PF methods
- Its disadvantage is that if the approximated densities are not Gaussians, the estimates may be inaccurate and the filter may diverge
- An alternative could be the Gaussian sum particle filtering where the densities in the system are approximated by mixture Gaussians

Gaussian particle filtering (GPF)

- The posterior and the predictive densities are approximated with Gaussians as is done by the extended Kalman filter (EKF)
- However, unlike with the EKF, we do not linearize any of the nonlinearities in the system
- Its advantage over other PF methods is that it does not require resampling
- Another advantage is that the estimation of constant parameters is not a problem as is the case with other PF methods
- Its disadvantage is that if the approximated densities are not Gaussians, the estimates may be inaccurate and the filter may diverge
- An alternative could be the Gaussian sum particle filtering where the densities in the system are approximated by mixture Gaussians

Comparison of methods – Bearings only tracking



Outline

Handling constant parameters

- The PF methodology was originally devised for the estimation of dynamic signals rather than static parameters
- The most efficient way to address the problem is to integrate out the unknown parameters when possible, either analytically or by Monte Carlo procedures

Handling constant parameters

- The PF methodology was originally devised for the estimation of dynamic signals rather than static parameters
- The most efficient way to address the problem is to integrate out the unknown parameters when possible, either analytically or by Monte Carlo procedures
- A common feature of these approaches is that they introduce artificial evolution of the fixed parameters

Handling constant parameters

- The PF methodology was originally devised for the estimation of dynamic signals rather than static parameters
- The most efficient way to address the problem is to integrate out the unknown parameters when possible, either analytically or by Monte Carlo procedures
- A common feature of these approaches is that they introduce artificial evolution of the fixed parameters
- A recent work introduced a special class of PF called density assisted particle filters that approximate the filtering density with a predefined parametric density by generalizing the concepts of Gaussian particle filters and Gaussian sum particle filters

Handling constant parameters

- The PF methodology was originally devised for the estimation of dynamic signals rather than static parameters
- The most efficient way to address the problem is to integrate out the unknown parameters when possible, either analytically or by Monte Carlo procedures
- A common feature of these approaches is that they introduce artificial evolution of the fixed parameters
- A recent work introduced a special class of PF called density assisted particle filters that approximate the filtering density with a predefined parametric density by generalizing the concepts of Gaussian particle filters and Gaussian sum particle filters
- These new filters can cope with constant parameters more naturally than previously proposed methods

Handling constant parameters

- The PF methodology was originally devised for the estimation of dynamic signals rather than static parameters
- The most efficient way to address the problem is to integrate out the unknown parameters when possible, either analytically or by Monte Carlo procedures
- A common feature of these approaches is that they introduce artificial evolution of the fixed parameters
- A recent work introduced a special class of PF called density assisted particle filters that approximate the filtering density with a predefined parametric density by generalizing the concepts of Gaussian particle filters and Gaussian sum particle filters
- These new filters can cope with constant parameters more naturally than previously proposed methods

Kernel-based auxiliary particle filter

- The inclusion of fixed parameters in the model implies extending the random measure to the form

$$\chi_t = \left\{ \mathbf{x}_t^{(m)}, \boldsymbol{\theta}_t^{(m)}, w_t^{(m)} \right\}_{m=1}^{M}$$

- The random measure approximates the density of interest $p(\mathbf{x}_t, \theta \mid \mathbf{y}_t)$, which can be decomposed as

$$p(\mathbf{x}_t, \theta \mid \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t \mid \mathbf{x}_t, \theta) p(\mathbf{x}_t \mid \theta, \mathbf{y}_{1:t}) p(\theta \mid \mathbf{y}_{1:t})$$

Kernel-based auxiliary particle filter

- The inclusion of fixed parameters in the model implies extending the random measure to the form

$$\chi_t = \left\{ \mathbf{x}_t^{(m)}, \boldsymbol{\theta}_t^{(m)}, w_t^{(m)} \right\}_{m=1}^{M}$$

- The random measure approximates the density of interest $p(\mathbf{x}_t, \boldsymbol{\theta} \mid \mathbf{y}_t)$, which can be decomposed as

$$p(\mathbf{x}_t, \boldsymbol{\theta} \mid \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t \mid \mathbf{x}_t, \boldsymbol{\theta}) p(\mathbf{x}_t \mid \boldsymbol{\theta}, \mathbf{y}_{1:t}) p(\boldsymbol{\theta} \mid \mathbf{y}_{1:t})$$

- It is clear that there is a need for approximation of the density, $p(\boldsymbol{\theta} \mid \mathbf{y}_{1:t})$, e.g., by

$$p(\boldsymbol{\theta} \mid \mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \mathcal{N}\left(\boldsymbol{\theta} \mid \bar{\boldsymbol{\theta}}_t^{(m)}, h^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta},t}\right)$$

Kernel-based auxiliary particle filter

- The inclusion of fixed parameters in the model implies extending the random measure to the form

$$\chi_t = \left\{ \mathbf{x}_t^{(m)}, \boldsymbol{\theta}_t^{(m)}, w_t^{(m)} \right\}_{m=1}^{M}$$

- The random measure approximates the density of interest $p(\mathbf{x}_t, \boldsymbol{\theta} \mid \mathbf{y}_t)$, which can be decomposed as

$$p(\mathbf{x}_t, \boldsymbol{\theta} \mid \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t \mid \mathbf{x}_t, \boldsymbol{\theta}) p(\mathbf{x}_t \mid \boldsymbol{\theta}, \mathbf{y}_{1:t}) p(\boldsymbol{\theta} \mid \mathbf{y}_{1:t})$$

- It is clear that there is a need for approximation of the density, $p(\boldsymbol{\theta} \mid \mathbf{y}_{1:t})$, e.g., by

$$p(\boldsymbol{\theta} \mid \mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \mathcal{N}\left(\boldsymbol{\theta} \mid \bar{\boldsymbol{\theta}}_t^{(m)}, h^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta},t} \right)$$

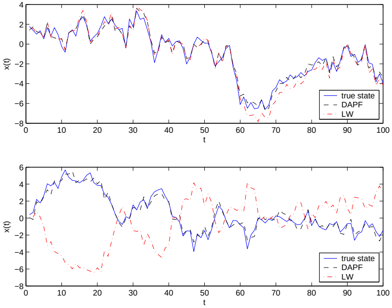
Density assisted particle filter

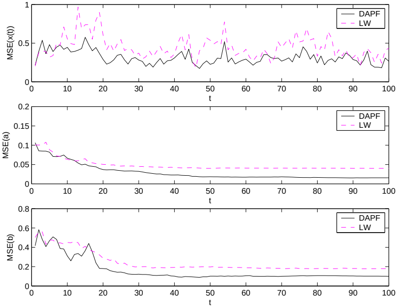
The approximating densities can be other than Gaussians or mixtures of Gaussians

1. Draw particles of $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\mathbf{y}_{0:t-1})$, i.e.,
$$\boldsymbol{\theta}_{t-1}^{(m)} \sim p(\boldsymbol{\theta}|\mathbf{y}_{0:t-1})$$
2. Draw particles according to
$$\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}, \boldsymbol{\theta}_{t-1}^{(m)})$$
3. Set $\boldsymbol{\theta}_t^{(m)} = \boldsymbol{\theta}_{t-1}^{(m)}$
4. Update and normalize the weights
$$w_t^{(m)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)}, \boldsymbol{\theta}_t^{(m)})$$
5. Estimate the parameters of the density $p(\boldsymbol{\theta}|\mathbf{y}_{0:t})$ from
$$\chi_t = \left\{\mathbf{x}_t^{(m)}, \boldsymbol{\theta}_t^{(m)}, w_t^{(m)}\right\}_{m=1}^{M}$$

Handling constant parameters – example

$$
\begin{aligned}
x_t &= a x_{t-1} + u_t \\
y_t &= b x_t + v_t
\end{aligned}
$$





Outline

An efficient approach: Rao-Blackwellized PF

- Rao-Blackwellization allows for marginalization of part of the states which are conditionally linear on the remaining states
- This approach leads to a more accurate performance of SPF since a smaller state space is explored

An efficient approach: Rao-Blackwellized PF

- Rao-Blackwellization allows for marginalization of part of the states which are conditionally linear on the remaining states
- This approach leads to a more accurate performance of SPF since a smaller state space is explored
- Standard implementation of Rao-Blackwellization requires running as many Kalman filters as particle streams

An efficient approach: Rao-Blackwellized PF

- Rao-Blackwellization allows for marginalization of part of the states which are conditionally linear on the remaining states
- This approach leads to a more accurate performance of SPF since a smaller state space is explored
- Standard implementation of Rao-Blackwellization requires running as many Kalman filters as particle streams
- An efficient realization of the Rao-Blackwellized PF with only one Kalman filter has also been proposed

An efficient approach: Rao-Blackwellized PF

- Rao-Blackwellization allows for marginalization of part of the states which are conditionally linear on the remaining states
- This approach leads to a more accurate performance of SPF since a smaller state space is explored
- Standard implementation of Rao-Blackwellization requires running as many Kalman filters as particle streams
- An efficient realization of the Rao-Blackwellized PF with only one Kalman filter has also been proposed
- The new approach shows considerable computational savings without sacrificing tracking performance

An efficient approach: Rao-Blackwellized PF

- Rao-Blackwellization allows for marginalization of part of the states which are conditionally linear on the remaining states
- This approach leads to a more accurate performance of SPF since a smaller state space is explored
- Standard implementation of Rao-Blackwellization requires running as many Kalman filters as particle streams
- An efficient realization of the Rao-Blackwellized PF with only one Kalman filter has also been proposed
- The new approach shows considerable computational savings without sacrificing tracking performance

Example: Bearings-only tracking with biased measurements

Consider K targets moving along a 2D sensor field of N sensors

$$\mathbf{x}_t = \mathbf{G}_x \mathbf{x}_{t-1} + \mathbf{G}_u \mathbf{u}_t$$

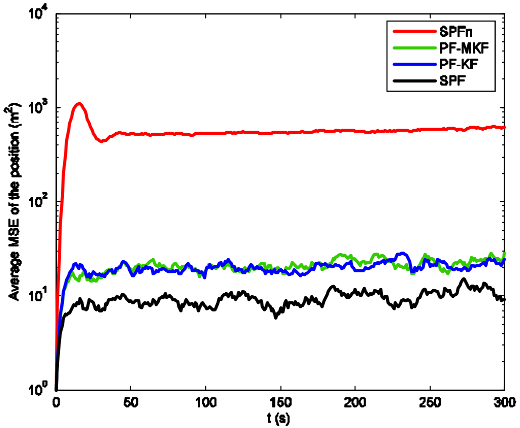
The $n-$th sensor at time instant t measures the bearing information of the targets sensed in the field

$$\begin{aligned} \mathbf{y}_{n,t} &= \mathbf{g}_n(\mathbf{x}_t) + \mathbf{b}_n + \mathbf{v}_{n,t} \qquad\qquad n = 1, \cdots, N \\ \mathbf{g}_n(\mathbf{x}_t) &= \left[\arctan\left(\frac{x_{2,1,t} - x_{2,n}}{x_{1,1,t} - x_{1,n}} \right), \cdots, \arctan\left(\frac{x_{2,K,t} - x_{2,n}}{x_{1,K,t} - x_{1,n}} \right) \right]^\top \end{aligned}$$

where $\mathbf{b}_n$ represents the unknown bias of the $n-$th sensor

Comparison of algorithms

Averaged MSE in m^2 of two targets obtained by different methods using $M = 500$ particles and $J = 50$ independent runs



Outline

Prediction

The prediction problem revolves around the estimation of the predictive density $p(\mathbf{x}_{t+l}|\mathbf{y}_{1:t})$, where $l > 1$

The approximation of the predictive density $p(\mathbf{x}_{t+l}|\mathbf{y}_{1:t})$ is obtained by

1. drawing particles $\mathbf{x}_{t+1}^{(m)}$ from $p(\mathbf{x}_{t+1}|\mathbf{x}_{t}^{(m)})$
2. drawing particles $\mathbf{x}_{t+2}^{(m)}$ from $p(\mathbf{x}_{t+2}|\mathbf{x}_{t+1}^{(m)})$
3. $\cdots$
4. drawing particles $\mathbf{x}_{t+l}^{(m)}$ from $p(\mathbf{x}_{t+l}|\mathbf{x}_{t+l-1}^{(m)})$
5. and associating with the samples $\mathbf{x}_{t+l}^{(m)}$ the weights $w_t^{(m)}$ and thereby forming the random measure $\{\mathbf{x}_{t+l}^{(m)}, w_t^{(m)}\}_{m=1}^{M}$.

Smoothing

All the information about $\mathbf{x}_t$ in this case is in the PDF $p(\mathbf{x}_t|\mathbf{y}_{1:T})$

Forward PF

Run PF in the forward direction and store all the random measures

$\chi_t = \{\mathbf{x}_t^{(m)}, w_t^{(m)}\}_{m=1}^M, \quad n = 1, 2, \cdots, N$

Backward recursions

Set the smoothing weights

$w_{s,T}^{(m)} = w_T^{(m)}$ and $\chi_{s,T} = \{\mathbf{x}_T^{(m)}, w_{s,T}^{(m)}\}_{m=1}^M$

For $n = N-1, \cdots, 1, 0$

Computation of the smoothing weights $w_{s,t}$

Compute the smoothing weights of $\mathbf{x}_t^{(m)}$ by

$$w_{s,t}^{(m)} = \sum_{j=1}^M w_{s,t+1}^{(j)} \frac{w_t^{(m)} p(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(m)})}{\sum_{l=1}^M w_t^{(l)} p(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(l)})}$$

Construction of the smoothing random measure $\chi_{s,t}$

Set $\chi_{s,t} = \{\mathbf{x}_t^{(m)}, w_{s,t}^{(m)}\}_{m=1}^M$

Outline

Multiple particle filters

We are interested in particle filtering methods for complex systems that can be represented by the following state-space model:

$$
\begin{aligned}
\mathbf{x}_t &= f_x(\mathbf{x}_{t-1}, \mathbf{u}_t) \\
\mathbf{y}_t &= f_y(\mathbf{x}_t, \mathbf{v}_t)
\end{aligned}
$$

We assume that the dimension of $\mathbf{x}_t$ is high

From practice we know that high-dimensional states would, in general, require a very large number of particles for accurate tracking of the posterior pdf of $\mathbf{x}_t$

Multiple particle filters

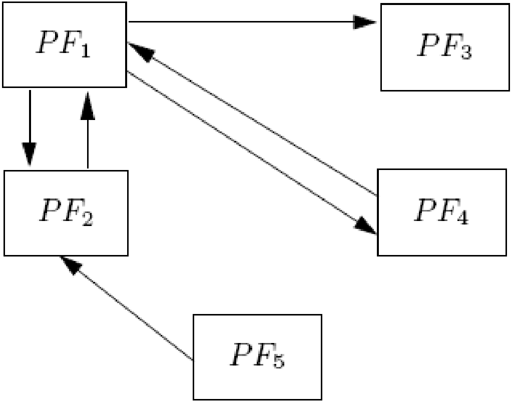
The underlying idea is to use multiple particle filters that communicate information about their posterior pdfs

The particle filters operate on partitioned subspaces of the complete state space

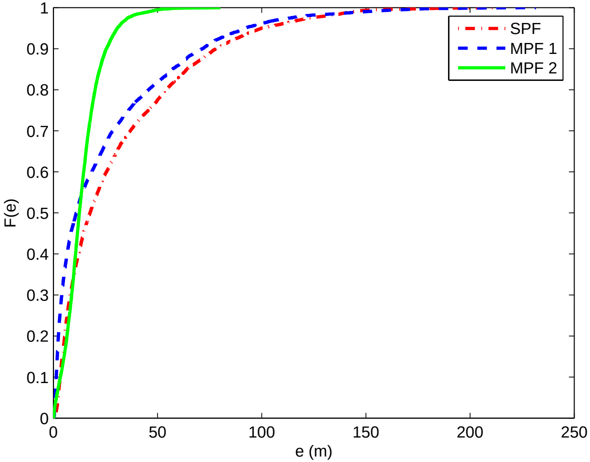
The state space is decomposed into separate subspaces of lower dimensionality which form a partition of the original space

We assume that the interest is in finding the marginal posterior densities of the state vectors that span these subspaces

Multiple particle filters



A system of multiple particle filters.



CDFs of RMSEs ($M = 1000,\ M_k = 500$)

Outline

Convergence issues

- PF is based on approximations of PDFs by discrete random measures
- We expect that the approximations deteriorate as the dimension of the state space increases but hope that as the number of particles $M \rightarrow \infty$, the approximation of the density will improve

Convergence issues

- PF is based on approximations of PDFs by discrete random measures
- We expect that the approximations deteriorate as the dimension of the state space increases but hope that as the number of particles $M \to \infty$, the approximation of the density will improve
- The approximations are based on particle streams, and they are *dependent* because of the necessity for resampling

Convergence issues

- PF is based on approximations of PDFs by discrete random measures
- We expect that the approximations deteriorate as the dimension of the state space increases but hope that as the number of particles $M \to \infty$, the approximation of the density will improve
- The approximations are based on particle streams, and they are *dependent* because of the necessity for resampling
- Therefore classical limit theorems for studying convergence cannot be applied, which makes the analysis more difficult

Convergence issues

- PF is based on approximations of PDFs by discrete random measures
- We expect that the approximations deteriorate as the dimension of the state space increases but hope that as the number of particles $M \to \infty$, the approximation of the density will improve
- The approximations are based on particle streams, and they are *dependent* because of the necessity for resampling
- Therefore classical limit theorems for studying convergence cannot be applied, which makes the analysis more difficult

Convergence theorems

Theorem 1: If $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ satisfies some mild conditions and the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ is bounded, continuous and strictly positive, then

$$\lim_{M\to\infty} \chi_t^M = p(\mathbf{x}_t|\mathbf{y}_{1:t})$$

almost surely

Theorem 2: If the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ is bounded, for all $t \geq 1$ there exists a constant c_t independent of M such that for any bounded function

$$E(e_t^2) \leq c_t \frac{\|h(\mathbf{x}_t)\|^2}{M}$$

where $E(\cdot)$ is an expectation operator, $h(\mathbf{x}_t)$ is a function of $\mathbf{x}_t$, and $\|h(\mathbf{x}_t)\|$ denotes the supremum norm, i.e.,

$$\|h(\mathbf{x}_t\| = \sup_{\mathbf{x}_t} |h(\mathbf{x}_t)|$$

Convergence theorems

Theorem 1: If $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ satisfies some mild conditions and the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ is bounded, continuous and strictly positive, then

$$\lim_{M\to\infty} \chi_t^M = p(\mathbf{x}_t|\mathbf{y}_{1:t})$$

almost surely

Theorem 2: If the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$ is bounded, for all $t \geq 1$ there exists a constant c_t independent of M such that for any bounded function

$$E(e_t^2) \leq c_t \frac{\|h(\mathbf{x}_t)\|^2}{M}$$

where $E(\cdot)$ is an expectation operator, $h(\mathbf{x}_t)$ is a function of $\mathbf{x}_t$, and $\|h(\mathbf{x}_t)\|$ denotes the supremum norm, i.e.,

$$\|h(\mathbf{x}_t\| = \sup_{\mathbf{x}_t} |h(\mathbf{x}_t)|$$

Outline

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed
- PF can be improved if some of the states are tracked by optimal (Kalman) filters

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed
- PF can be improved if some of the states are tracked by optimal (Kalman) filters
- Convergence results exist

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed
- PF can be improved if some of the states are tracked by optimal (Kalman) filters
- Convergence results exist
- Efforts of generalizing PF include application of the philosophy when the noise distributions in the state and observation equations are unknown

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed
- PF can be improved if some of the states are tracked by optimal (Kalman) filters
- Convergence results exist
- Efforts of generalizing PF include application of the philosophy when the noise distributions in the state and observation equations are unknown
- PF is computationally intensive but special hardware can be built that exploits the parallelizability of PF

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed
- PF can be improved if some of the states are tracked by optimal (Kalman) filters
- Convergence results exist
- Efforts of generalizing PF include application of the philosophy when the noise distributions in the state and observation equations are unknown
- PF is computationally intensive but special hardware can be built that exploits the parallelizability of PF
- Challenges of PF include tracking in high-dimensional state spaces

Conclusions

- PF is a Monte Carlo-based methodology for sequential estimation of dynamic (and static) signals (states)
- The performance of the method depends on the number of particles that are used and the way how new particles are proposed
- PF can be improved if some of the states are tracked by optimal (Kalman) filters
- Convergence results exist
- Efforts of generalizing PF include application of the philosophy when the noise distributions in the state and observation equations are unknown
- PF is computationally intensive but special hardware can be built that exploits the parallelizability of PF
- Challenges of PF include tracking in high-dimensional state spaces

The Particle Filtering Methodology in Signal Processing

Petar M. Djurić

Department of Electrical & Computer Engineering
Stony Brook University

September 25, 2009