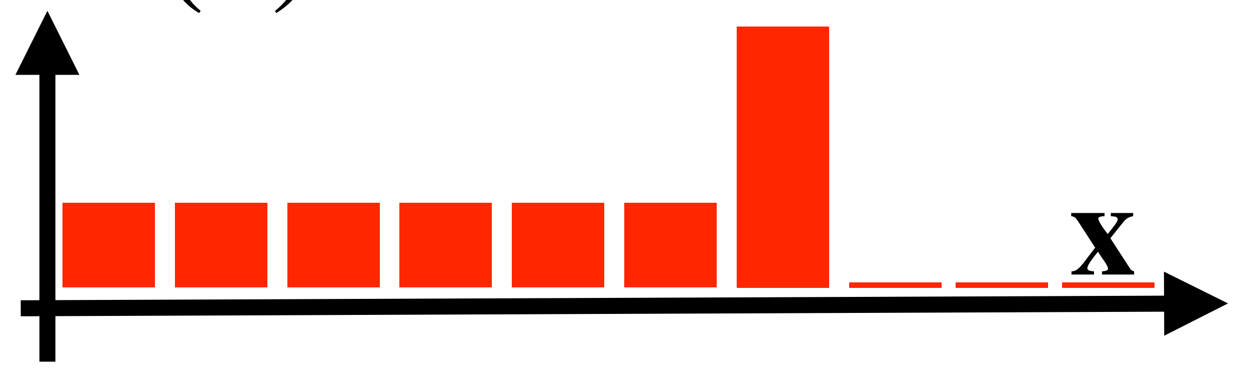
Particle filter

Karel Zimmermann

	Drawbacks	Advantages
Bayes filter	○ course of dimensionality ○ spatial discretization	○ represents arbitrary prob. distribution
Kalman filter	○ represent only gaussians ○ suffers from linearization	○ nicely scales with higher dimensions
Partical filter		

Bayes filter

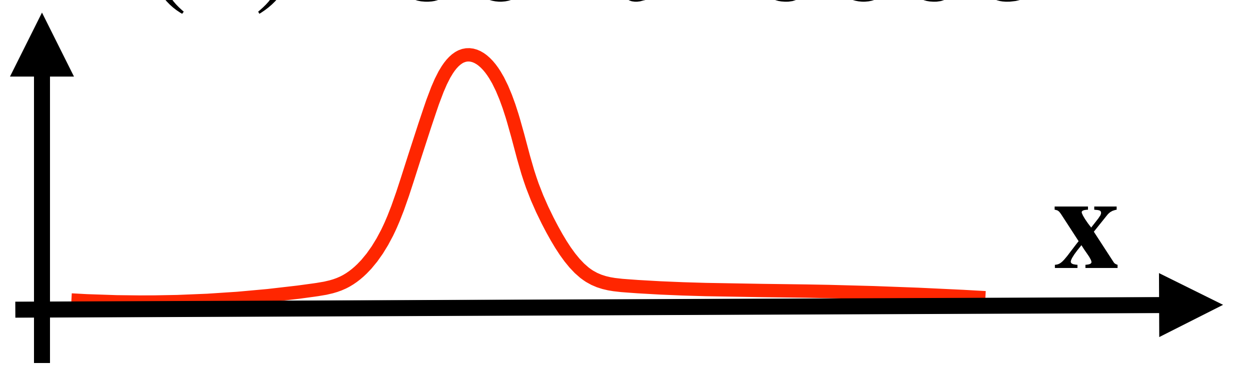
$\mathrm{bel}(\mathbf{x})$ discrete



$$\mathrm{bel}(\mathbf{x}) = [p_1, p_2, \ldots, p_m]$$

Kalman filter

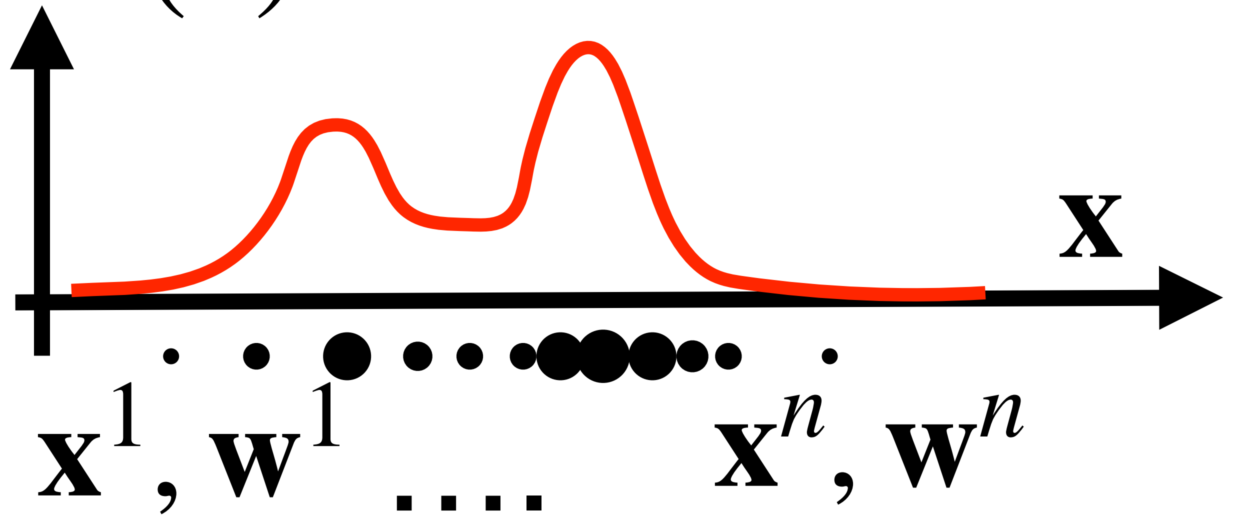
$\mathrm{bel}(\mathbf{x})$ continuous



$$\mathrm{bel}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma})}}$$

Partical filter

$\mathrm{bel}(\mathbf{x})$ continuous



$$\mathrm{bel}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{w}^i \cdot \delta_{\mathbf{x}^i}(\mathbf{x})$$

Dirac impulse function

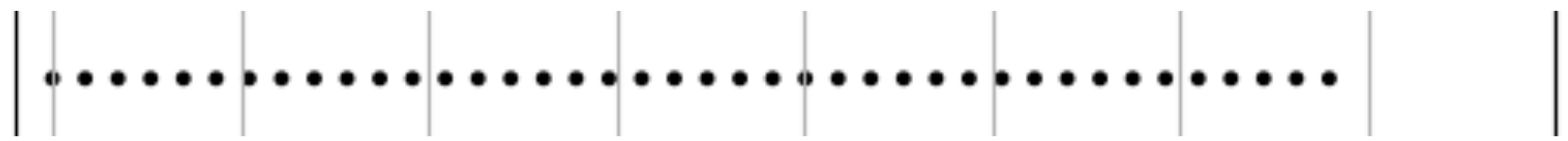
Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$

Kidnapped robot problem

Particles = hypothesis about the current state

$\mathbf{x}_0^i$



Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

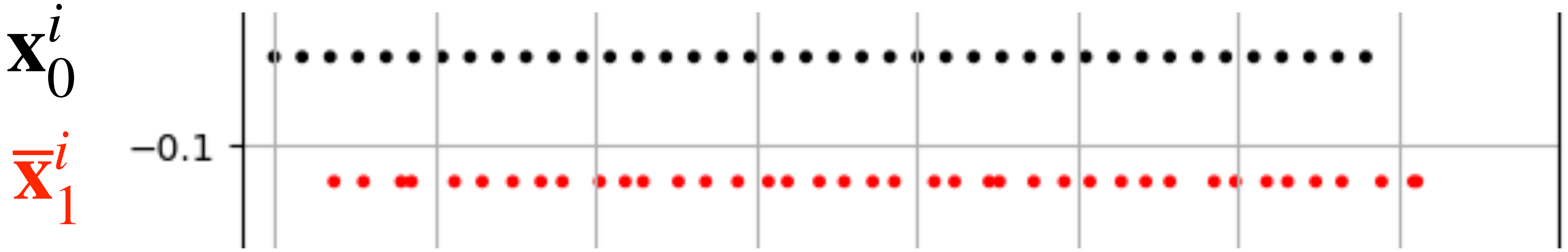
   $$\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

   $$\mathbf{w}_t^i = p(\mathbf{z}_t \mid \overline{\mathbf{x}}_t^i)$$

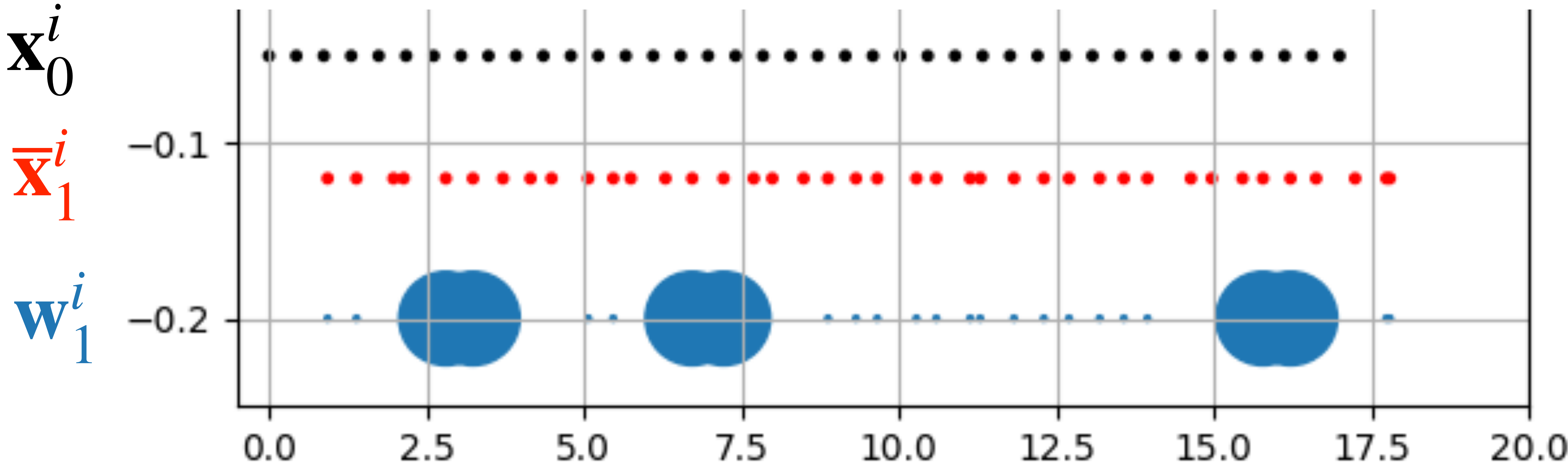
$p(\mathbf{z}_t = 1 \mid \mathbf{x}_t)$



Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):
   For all $\mathbf{x}_{t-1}^i$
   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
3. Measurement update (new $\mathbf{z}_t$ received):
   For all $\overline{\mathbf{x}}_t^i$
   $\mathbf{w}_t^i = p(\mathbf{z}_t \mid \overline{\mathbf{x}}_t^i)$

$p(\mathbf{z}_t = 1 \mid \mathbf{x}_t)$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

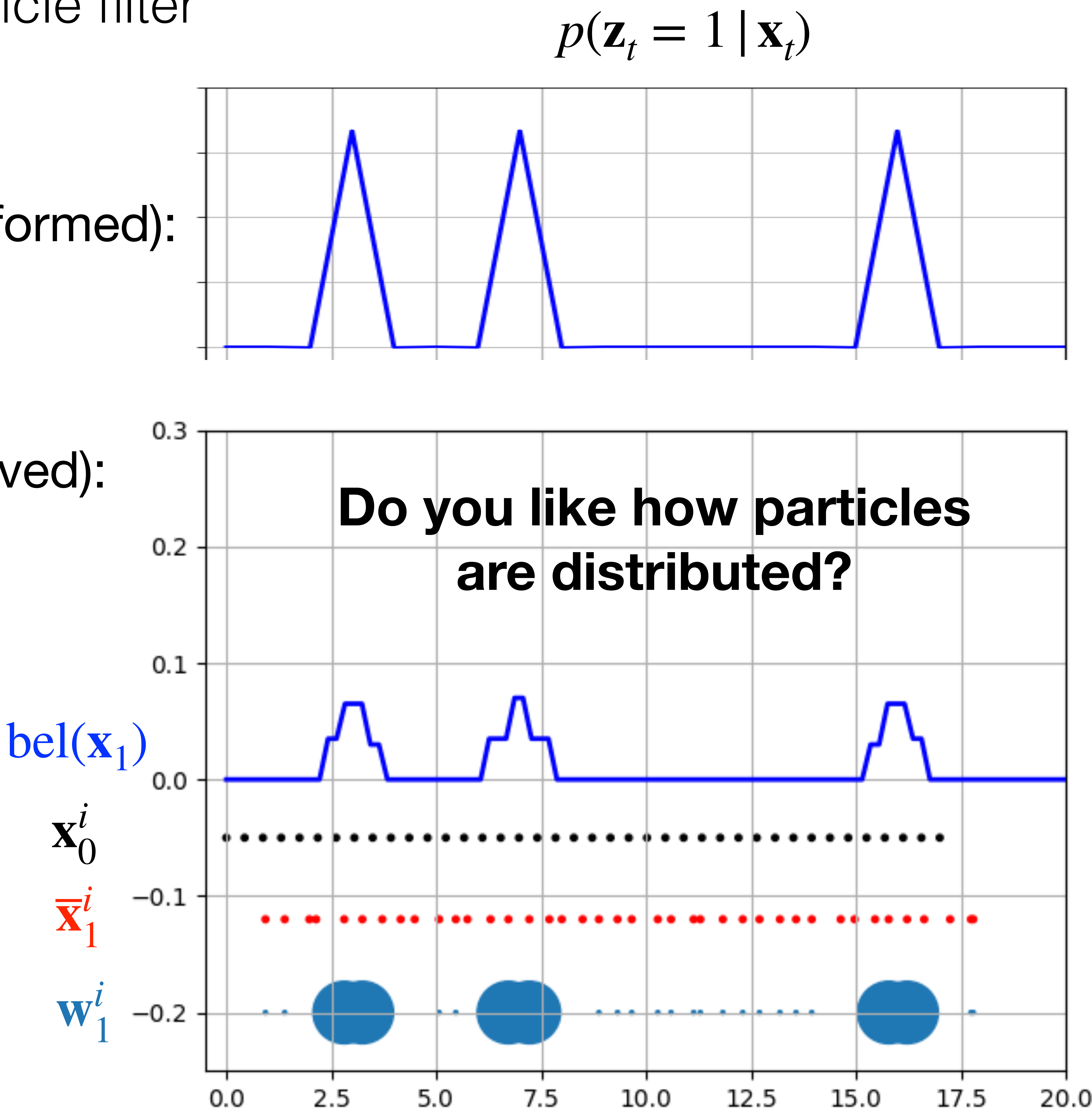
   For all $\mathbf{x}_{t-1}^i$

   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \overline{\mathbf{x}}_t^i)$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

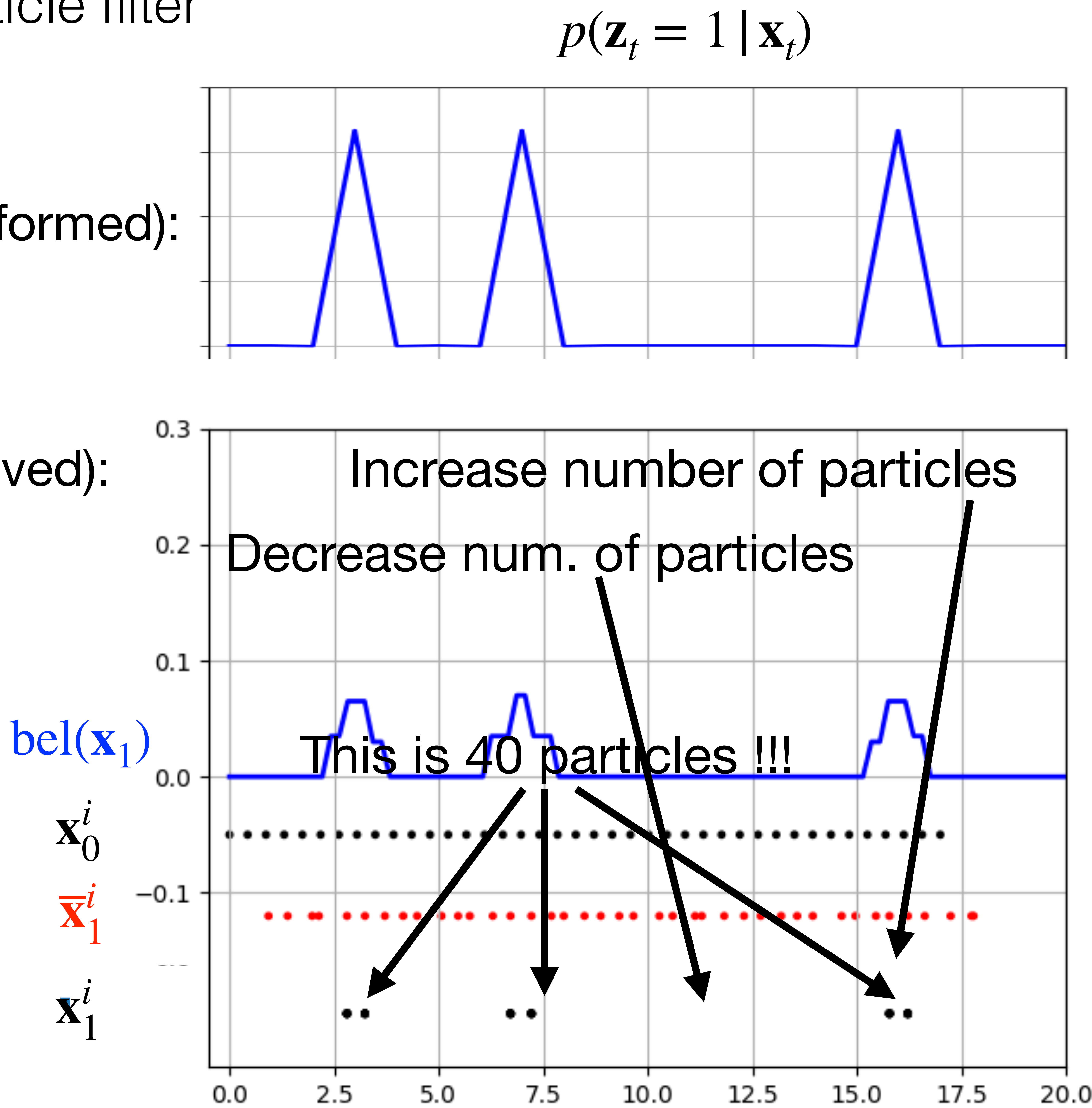
   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \overline{\mathbf{x}}_t^i)$

4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
3. Measurement update (new $\mathbf{z}_t$ received):

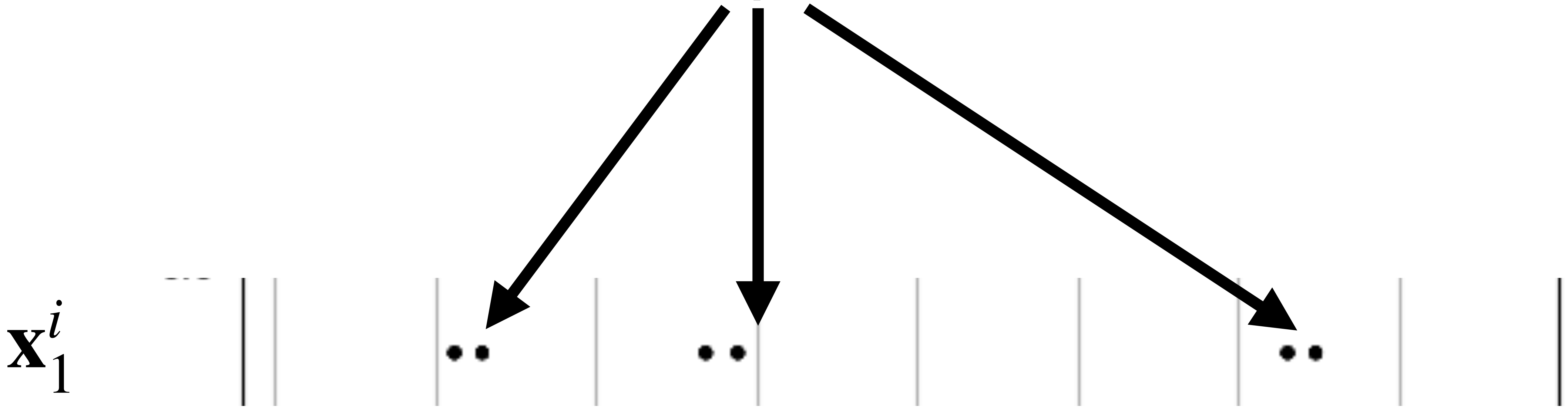
   For all $\overline{\mathbf{x}}_t^i$

   $\mathbf{w}_t^i = p(\mathbf{z}_t | \overline{\mathbf{x}}_t^i)$
4. Resample

   Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:

   $t = t + 1$

This is 40 particles !!!



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

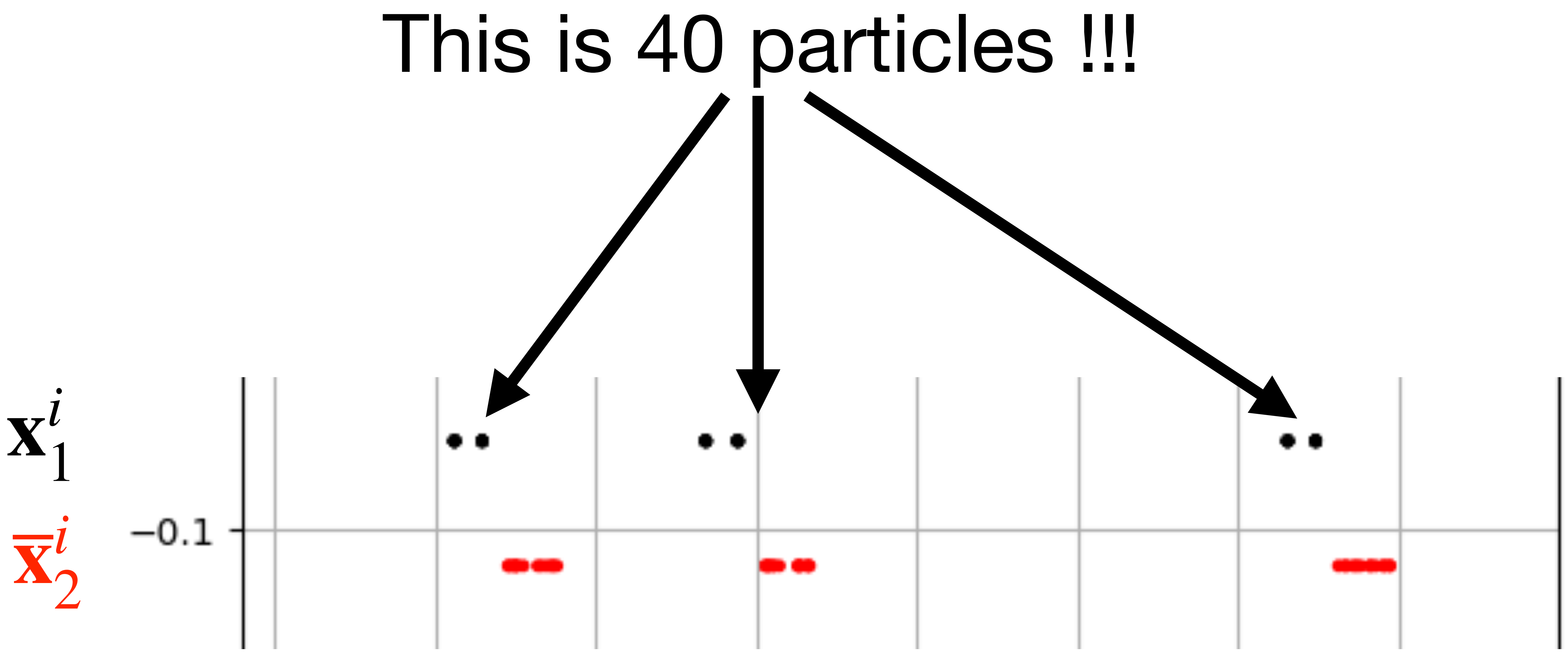
   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \overline{\mathbf{x}}_t^i)$
4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $$\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\bar{\mathbf{x}}_t^i$

   $$\mathbf{w}_t^i = p(\mathbf{z}_t \mid \bar{\mathbf{x}}_t^i)$$

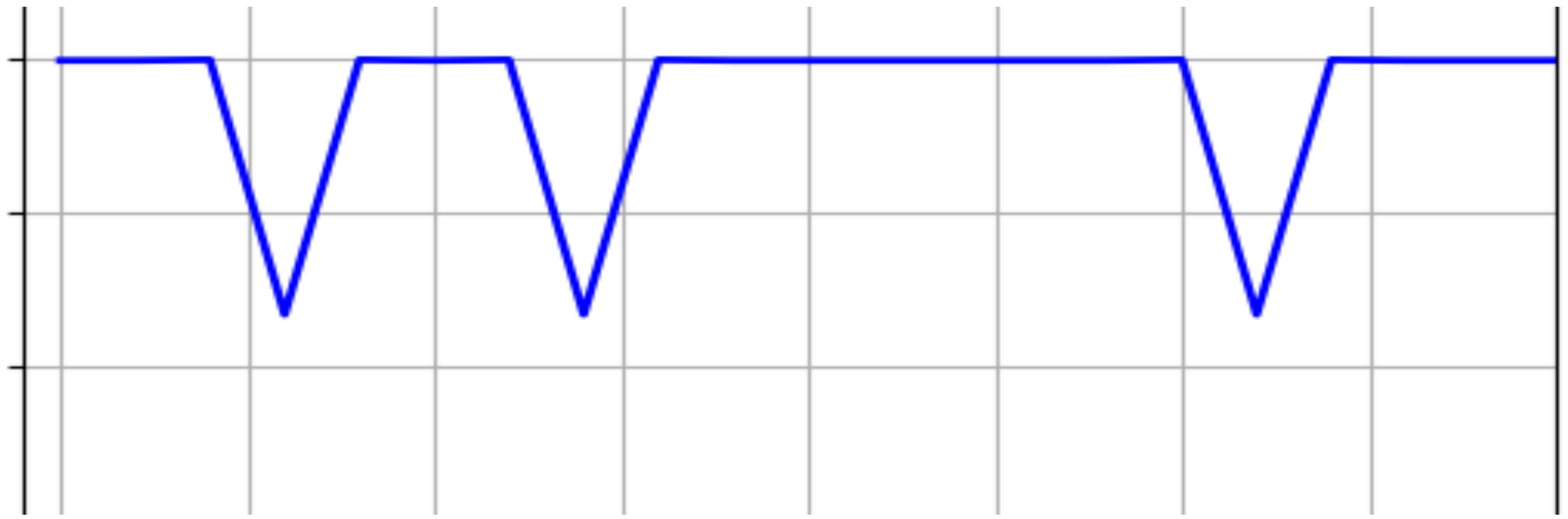
4. Resample

   Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

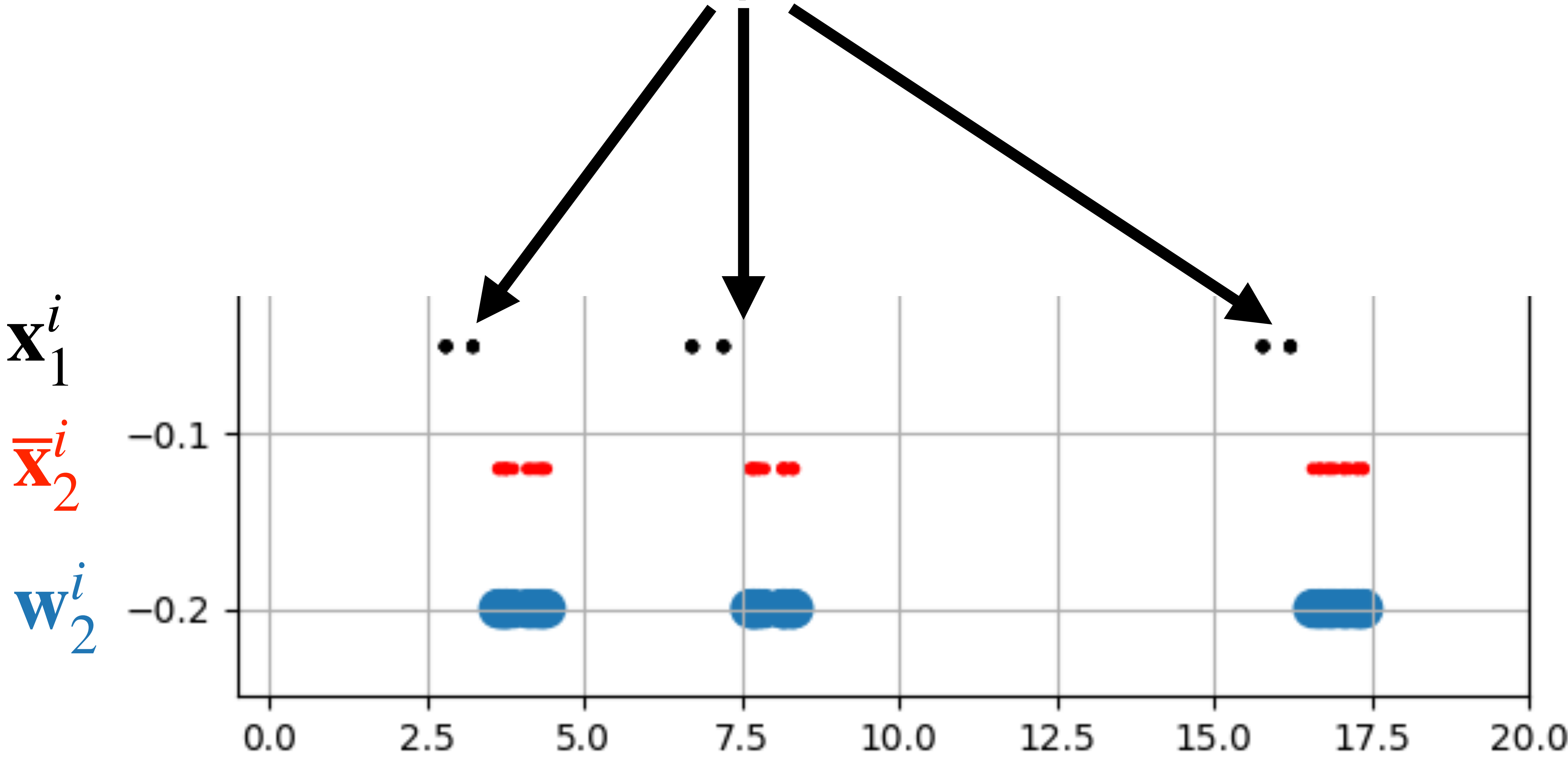
5. Repeat from 2:

   $t = t + 1$

$p(\mathbf{z}_t = 0 \mid \mathbf{x}_t)$



This is 40 particles !!!



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $$\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

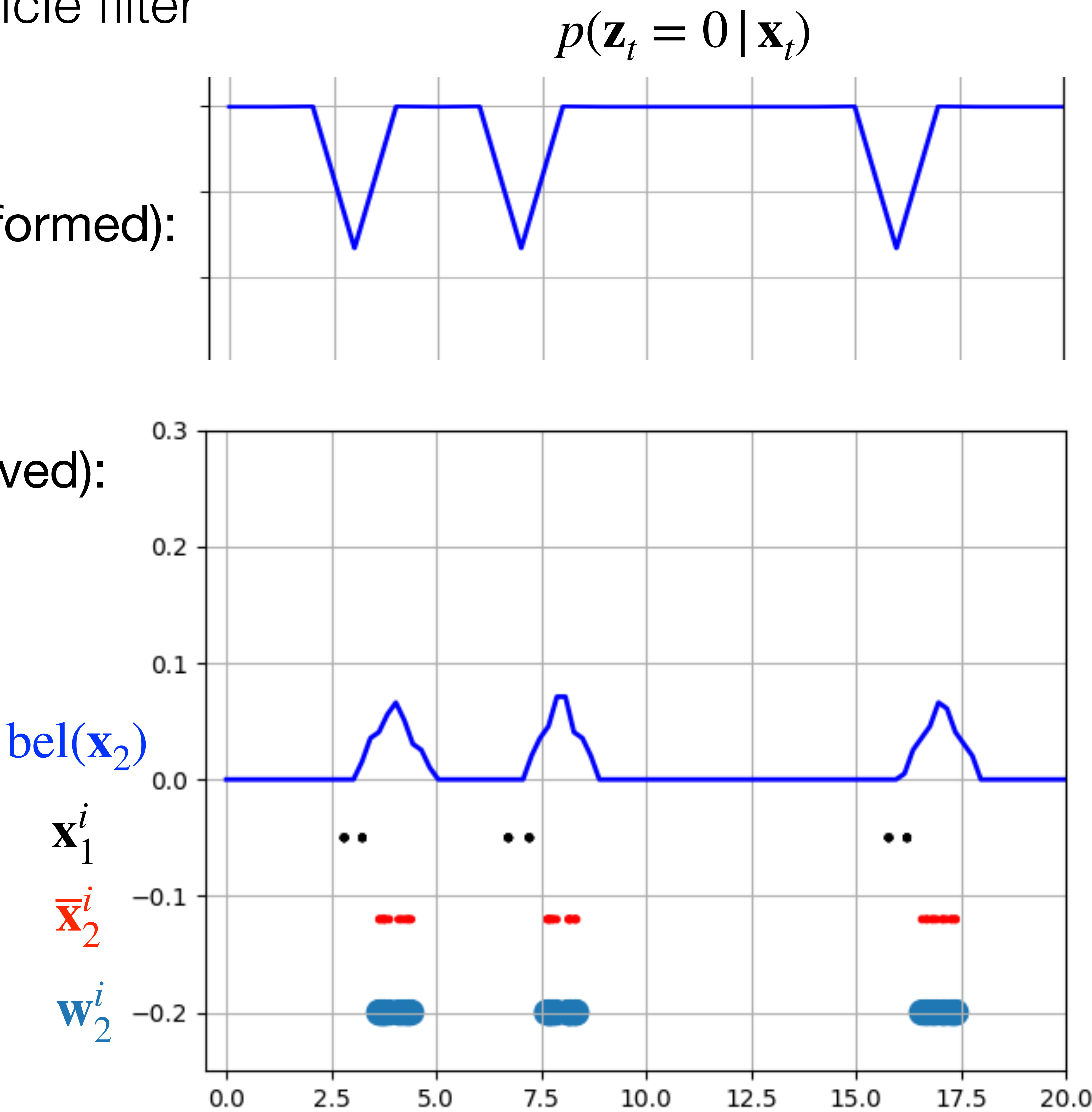
   $$\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \overline{\mathbf{x}}_t^i)$$

4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

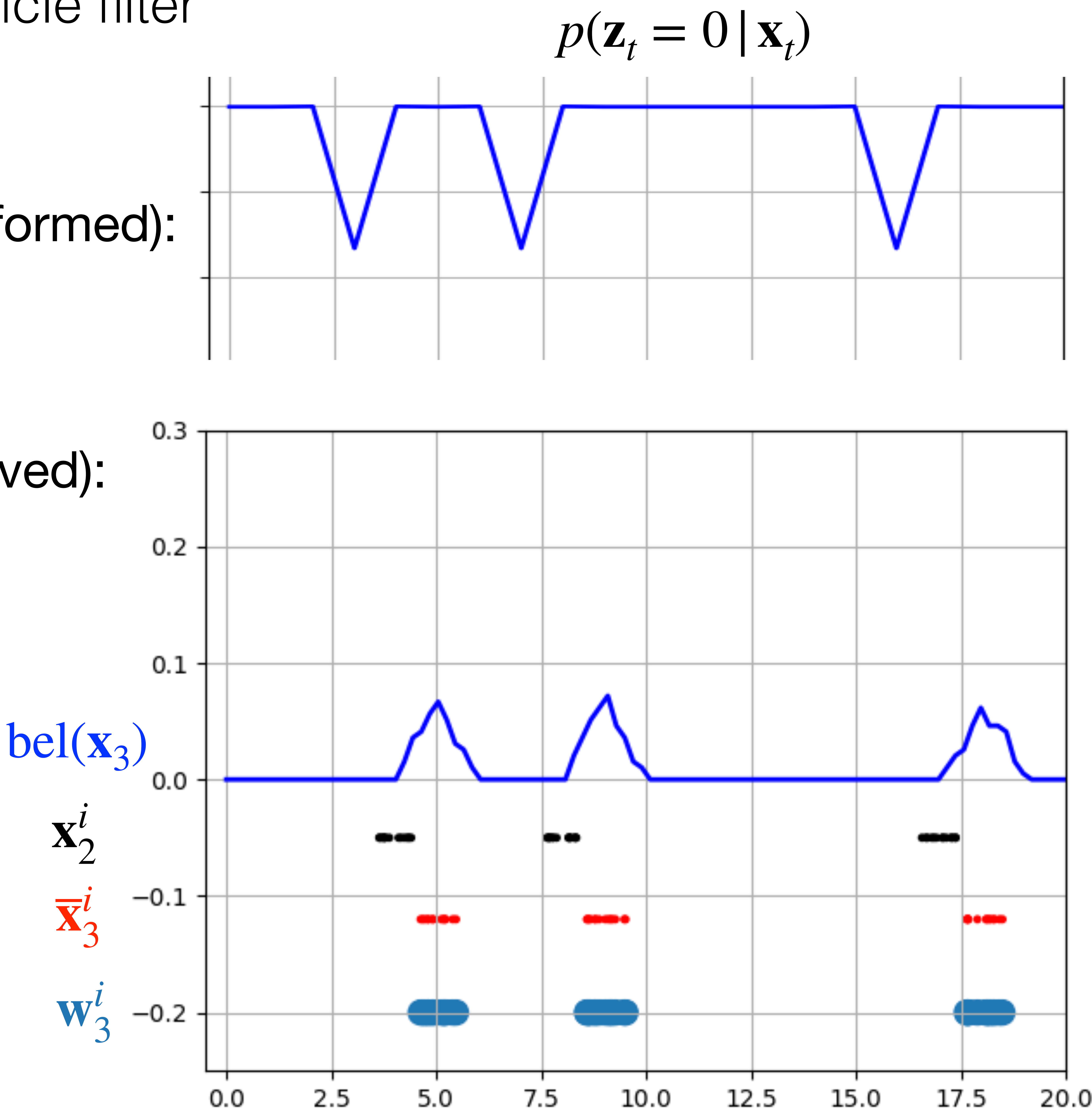
   $\mathbf{w}_t^i = p(\mathbf{z}_t \mid \overline{\mathbf{x}}_t^i)$

4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

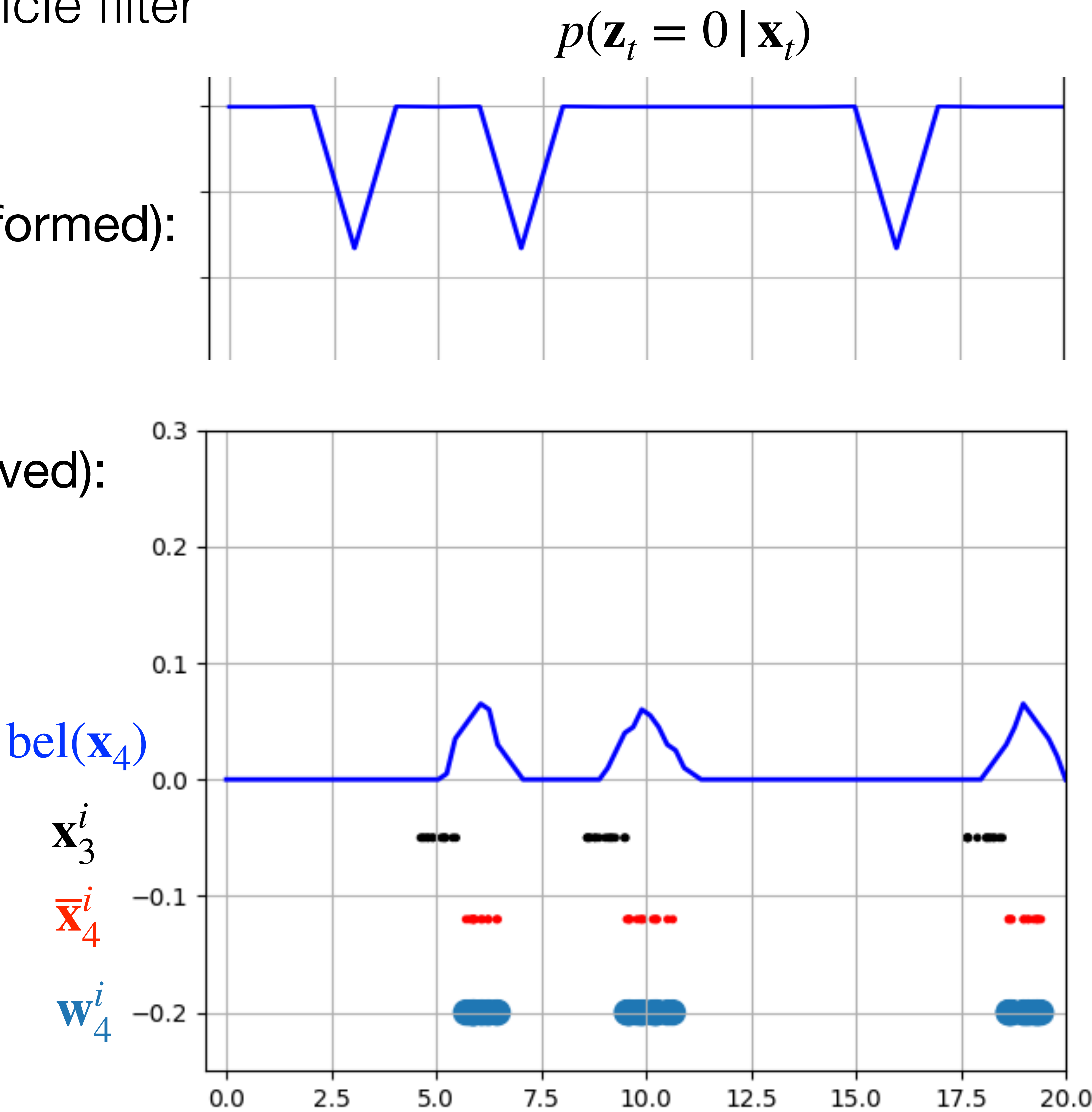
   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \overline{\mathbf{x}}_t^i)$

4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$



Particle filter

$p(\mathbf{z}_t = 1 \,|\, \mathbf{x}_t)$

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

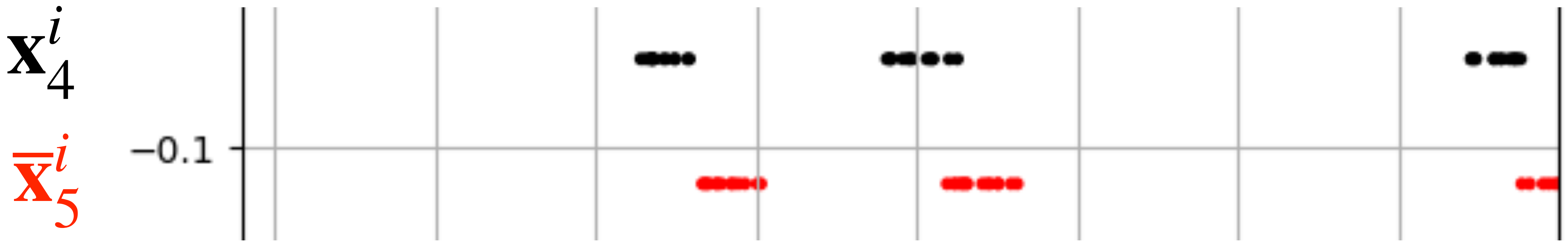
   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \overline{\mathbf{x}}_t^i)$
4. Resample

   Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\bar{\mathbf{x}}_t^i$

   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \bar{\mathbf{x}}_t^i)$

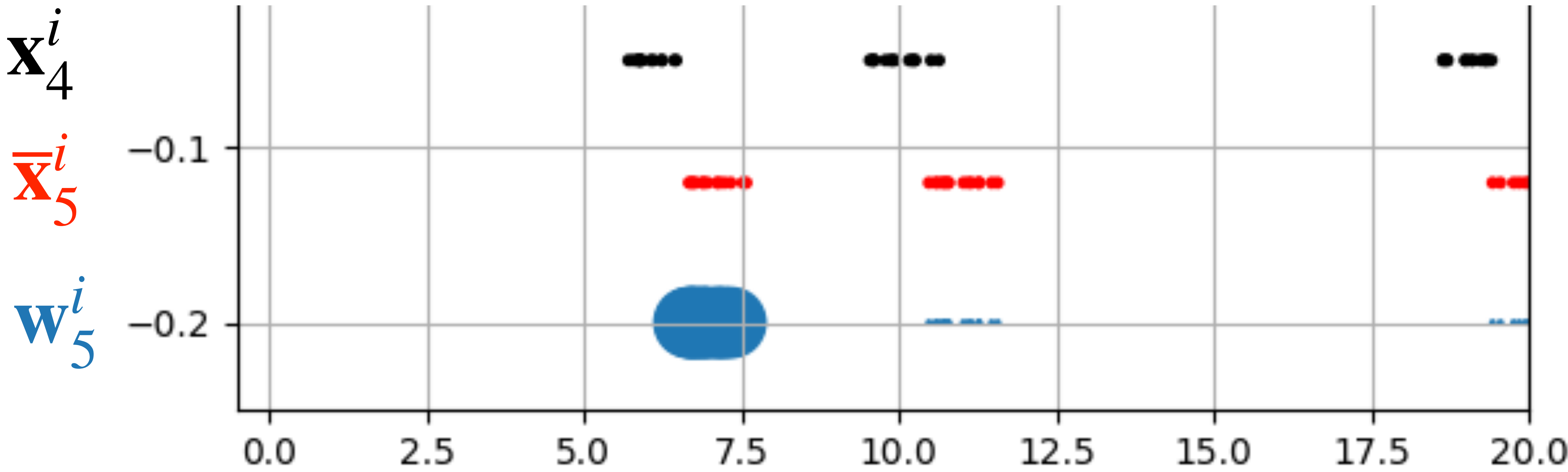
4. Resample

   Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$

$p(\mathbf{z}_t = 1 \,|\, \mathbf{x}_t)$



Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\bar{\mathbf{x}}_t^i$

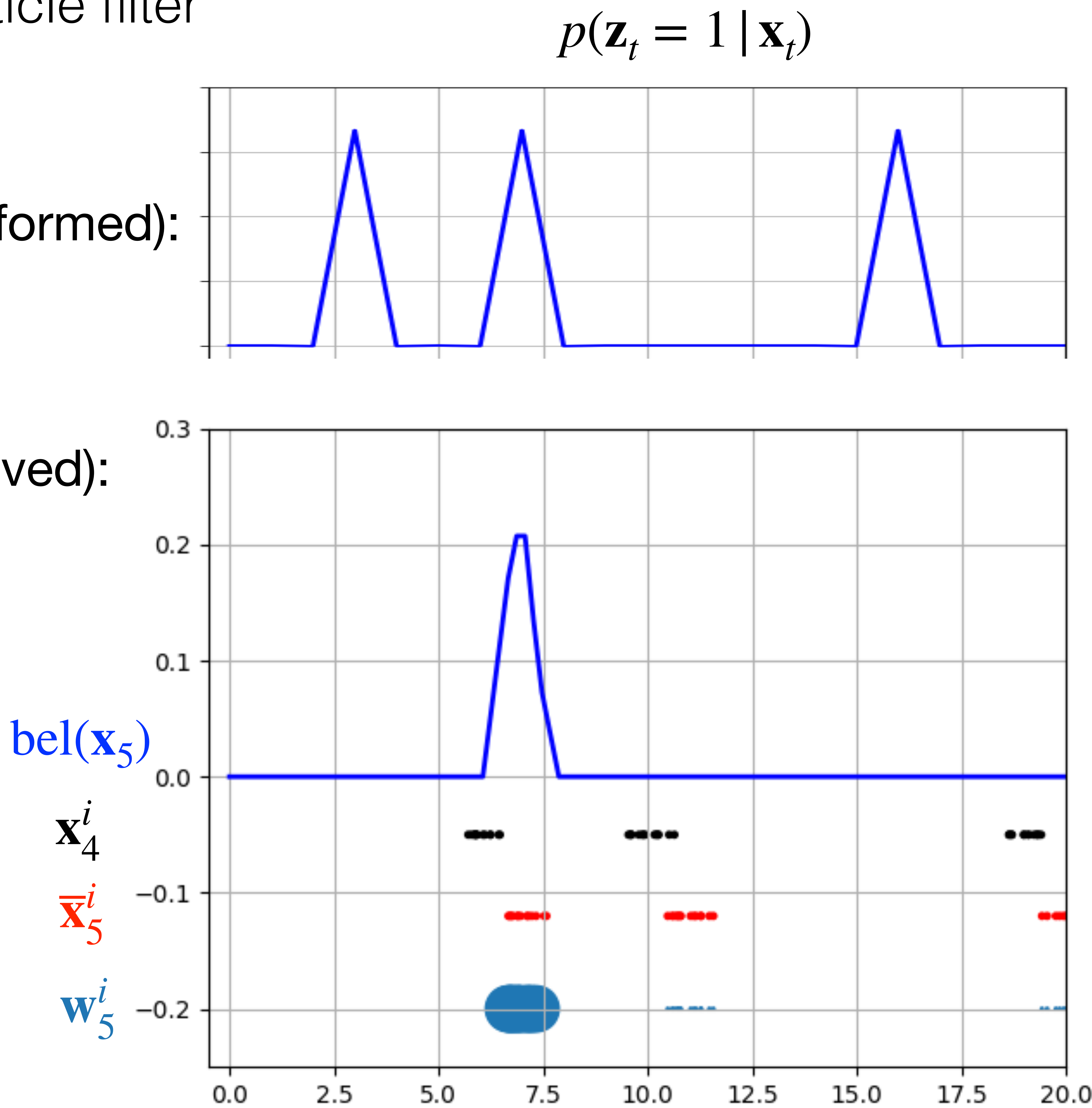
   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \bar{\mathbf{x}}_t^i)$

4. Resample

   Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\bar{\mathbf{x}}_t^i$

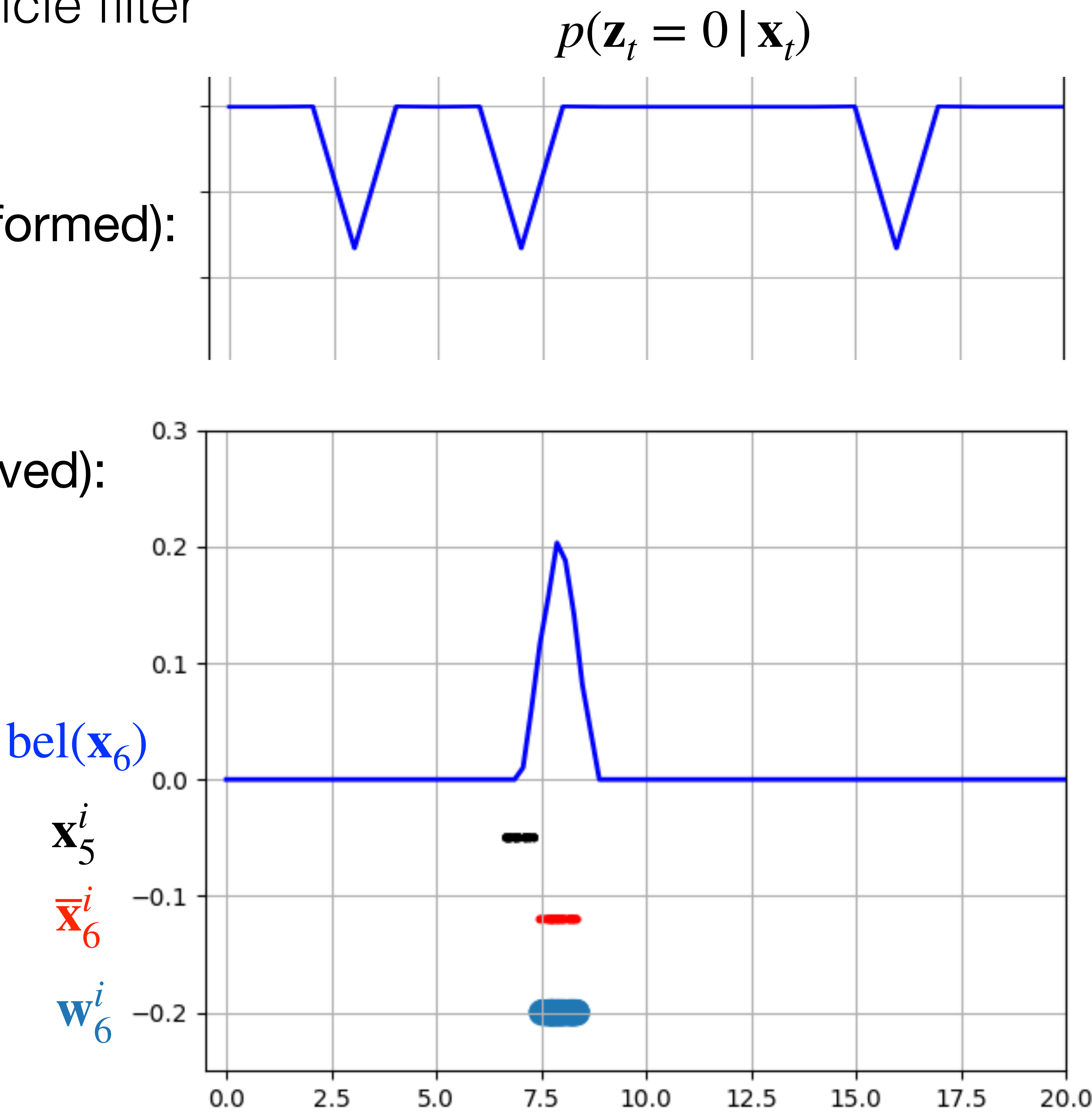
   $\mathbf{w}_t^i = p(\mathbf{z}_t \mid \bar{\mathbf{x}}_t^i)$

4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

   $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}^i, \mathbf{u}_t)$

3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\bar{\mathbf{x}}_t^i$

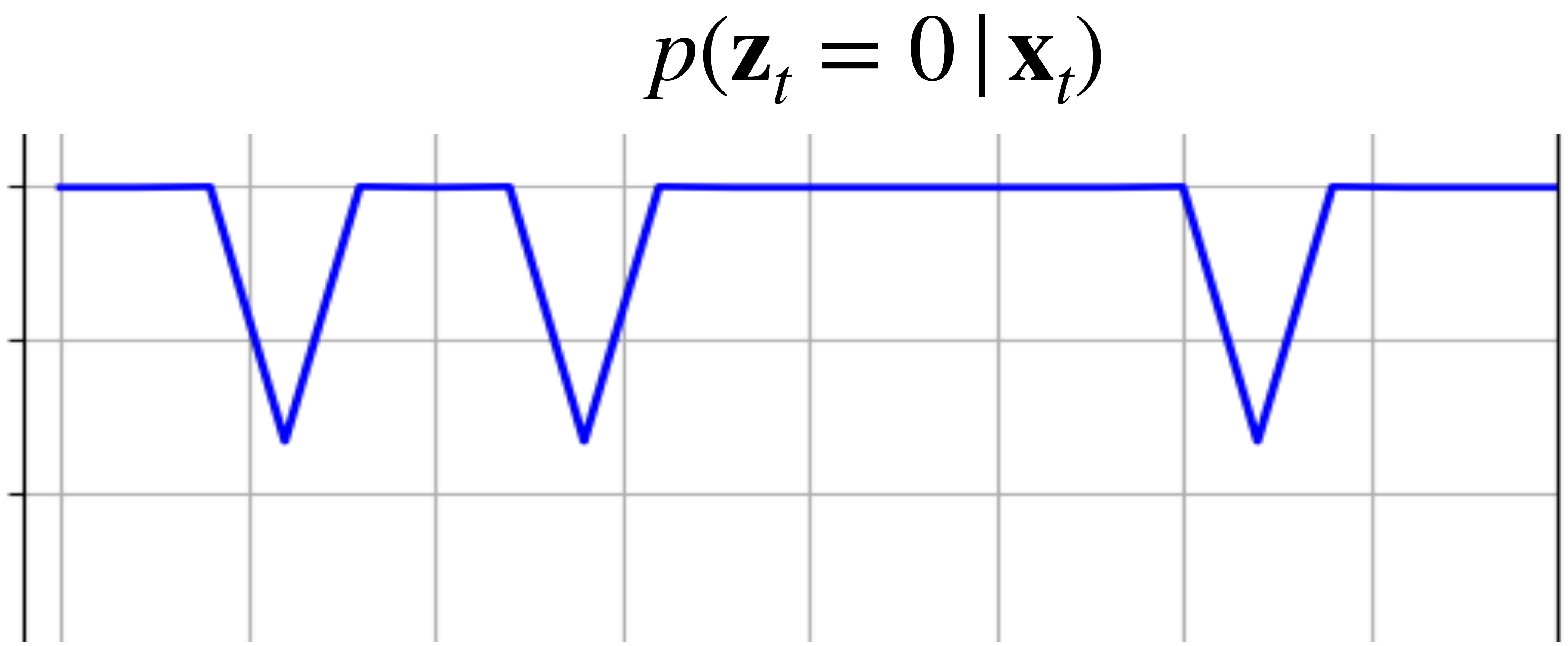
   $\mathbf{w}_t^i = p(\mathbf{z}_t \,|\, \bar{\mathbf{x}}_t^i)$

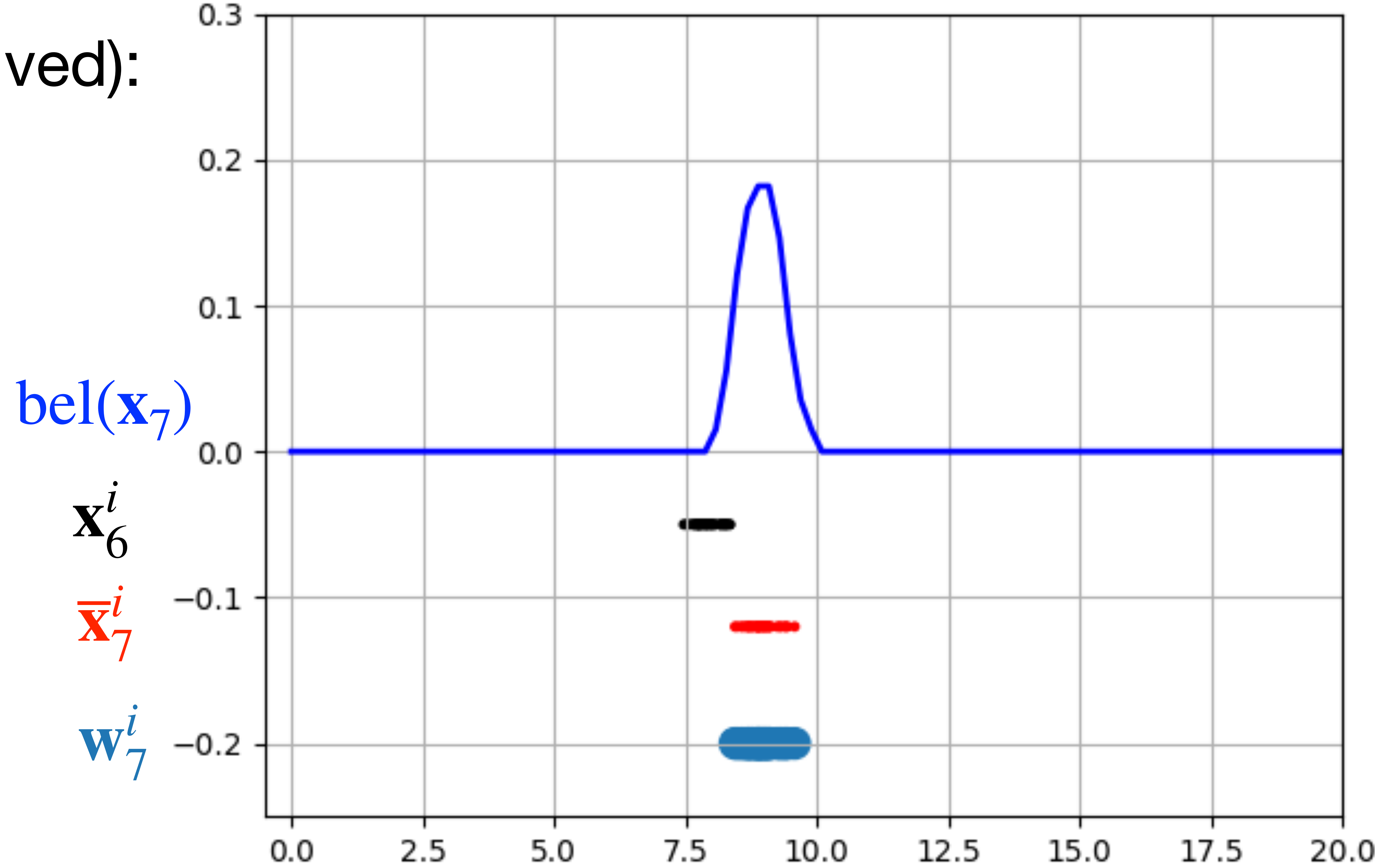
4. Resample

   Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

5. Repeat from 2:

   $t = t + 1$





Particle filter

1. Initialize particles: $\mathscr{X}_0 = \{\mathbf{x}_0^1, \ldots, \mathbf{x}_0^n\}$
2. Prediction step (new action $\mathbf{u}_t$ performed):

   For all $\mathbf{x}_{t-1}^i$

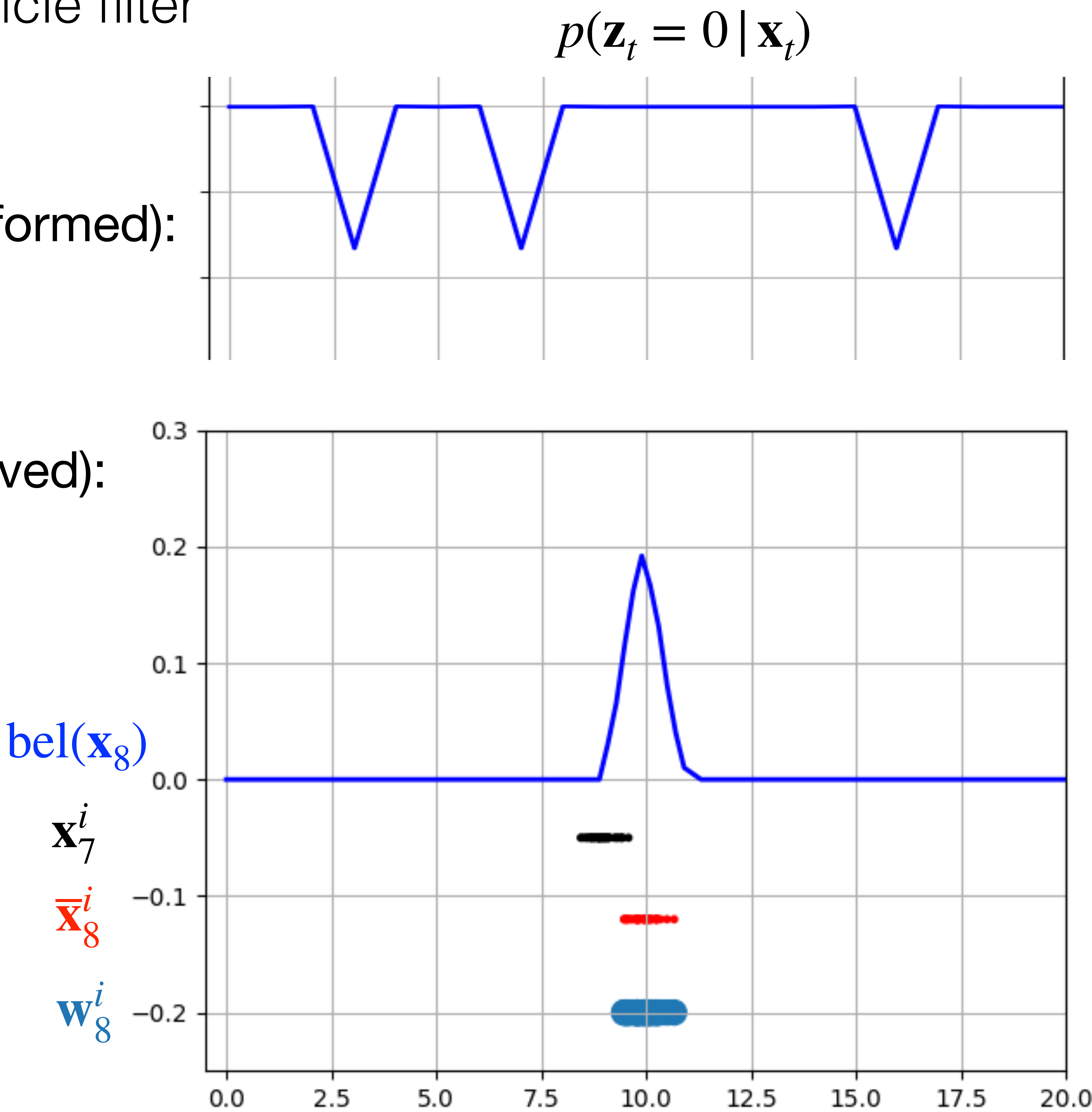
   $\overline{\mathbf{x}}_t^i \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
3. Measurement update (new $\mathbf{z}_t$ received):

   For all $\overline{\mathbf{x}}_t^i$

   $\mathbf{w}_t^i = p(\mathbf{z}_t \mid \overline{\mathbf{x}}_t^i)$
4. Resample

   Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:

   $t = t + 1$



Running the same example multiple times

Running the same example multiple times

Result heavily depends on initialization

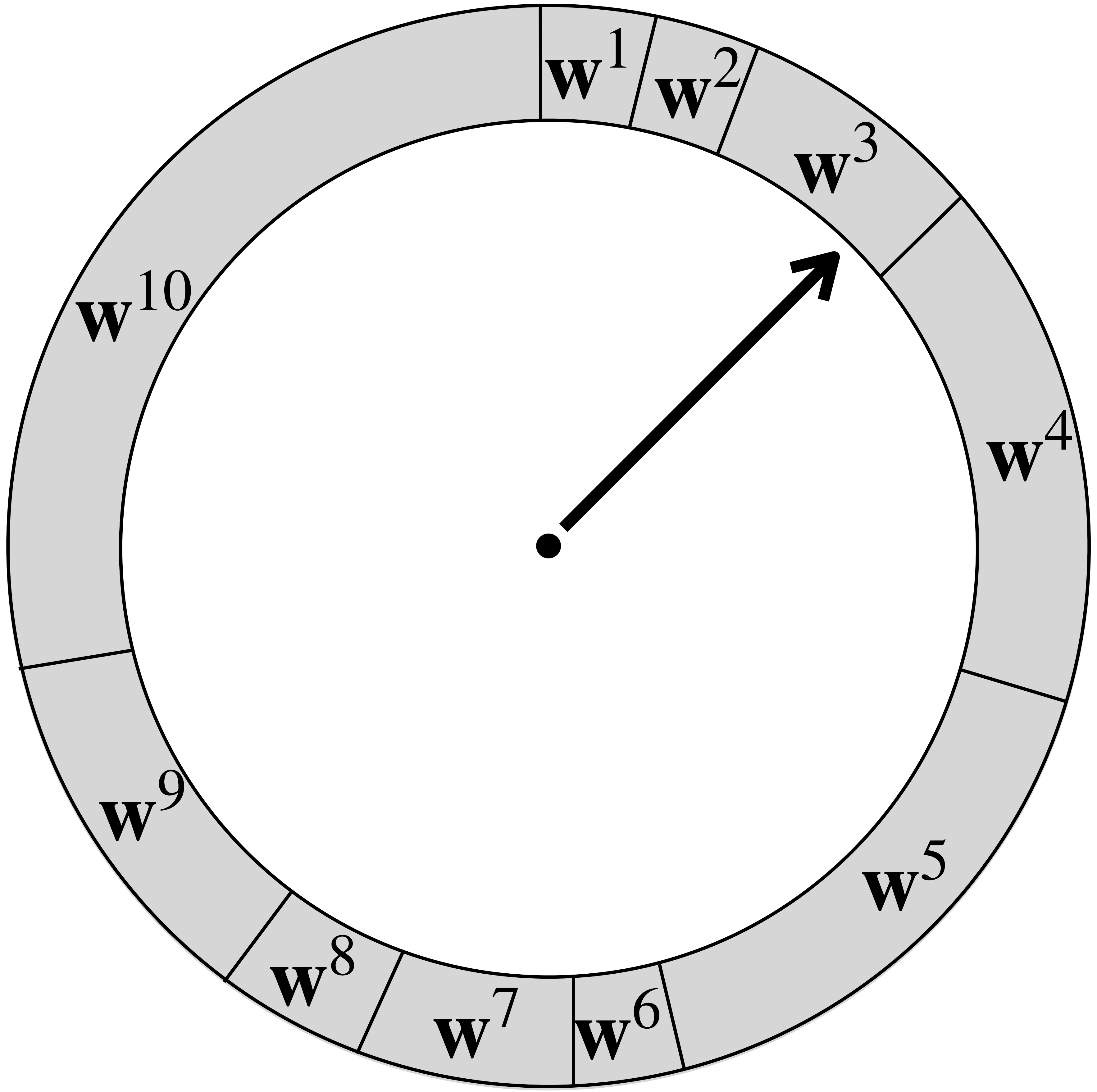
Online demo: https://tommycohn.com/Occupancy-Grid-SLAM-JS/

4. Resample

Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

4. Resample

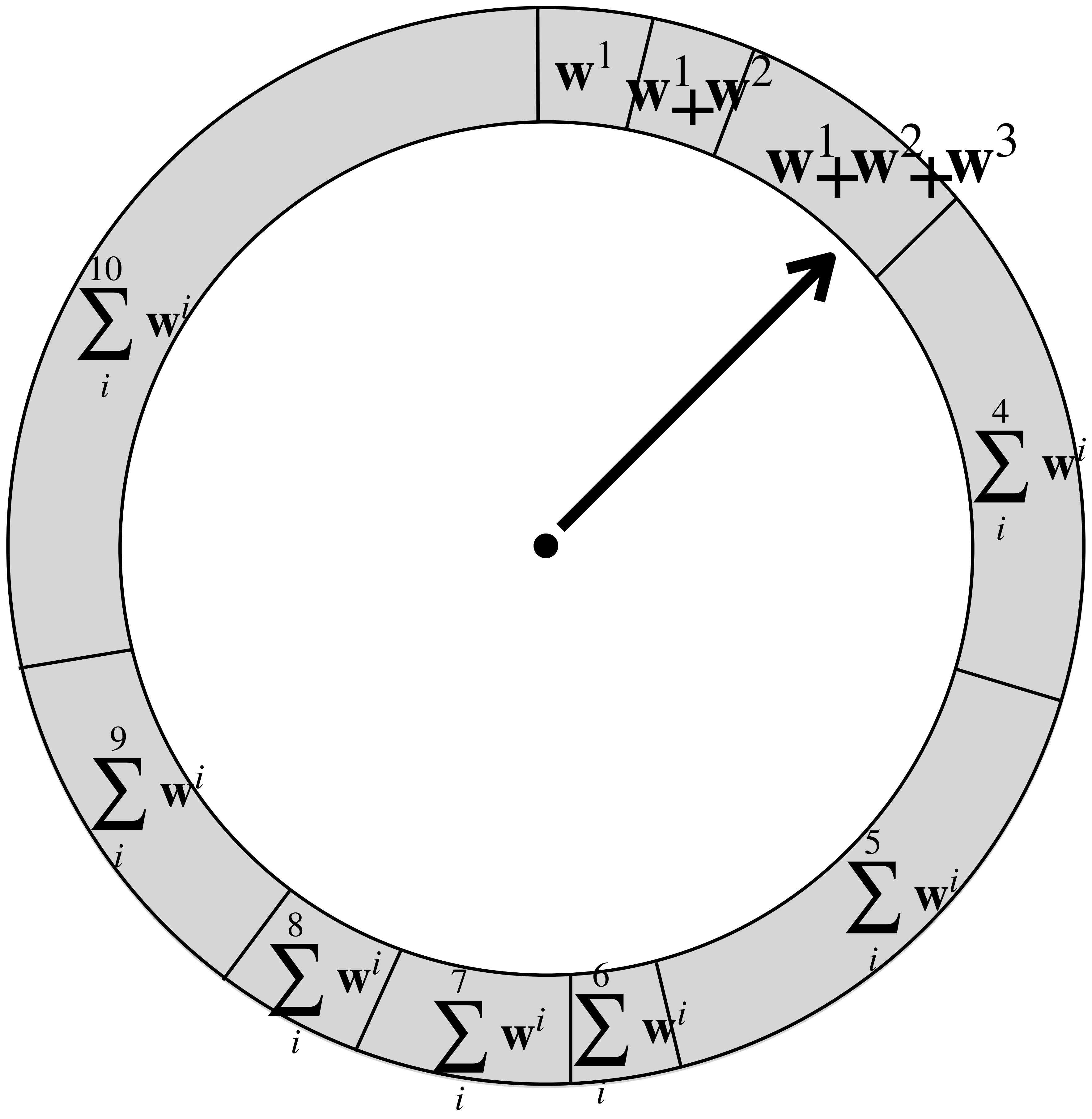
Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Roulette wheel

4. Resample

Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

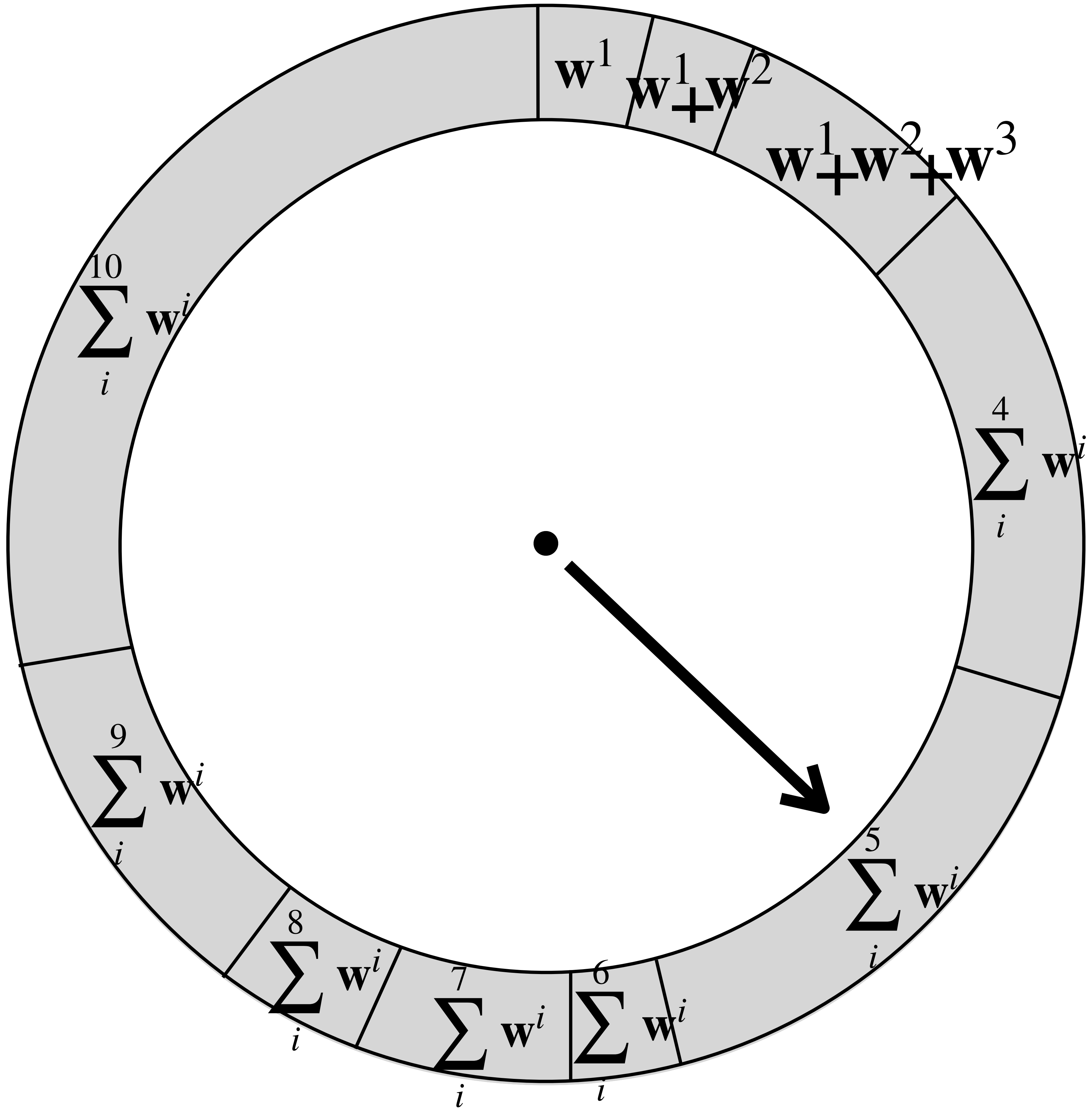


Roulette wheel

Replace values by cumsum

4. Resample

Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Generate random number

Find corresponding slot by binary search $\mathcal{O}\big(\log(N)\big)$

Repeat is N times

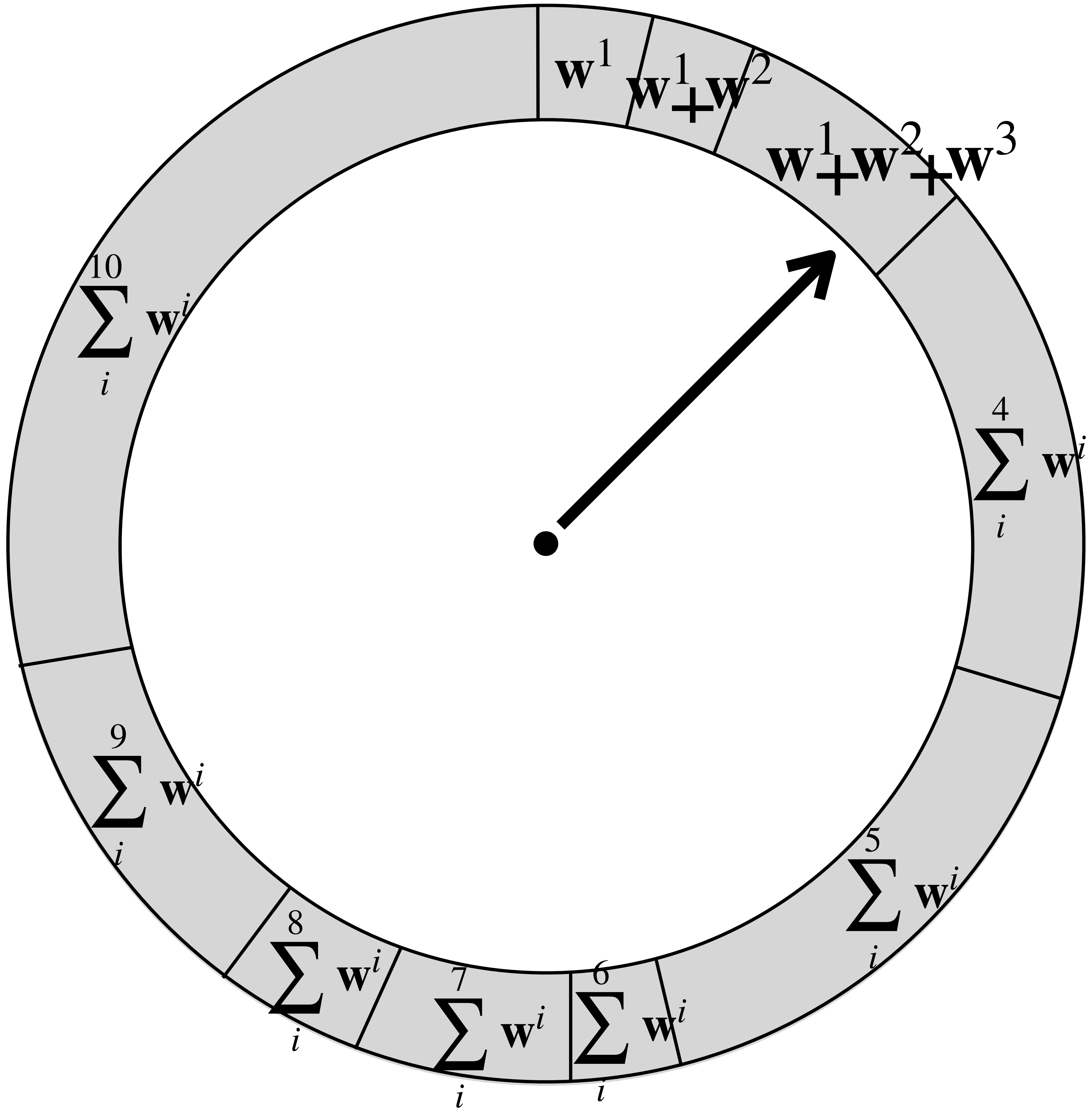
Roulette wheel

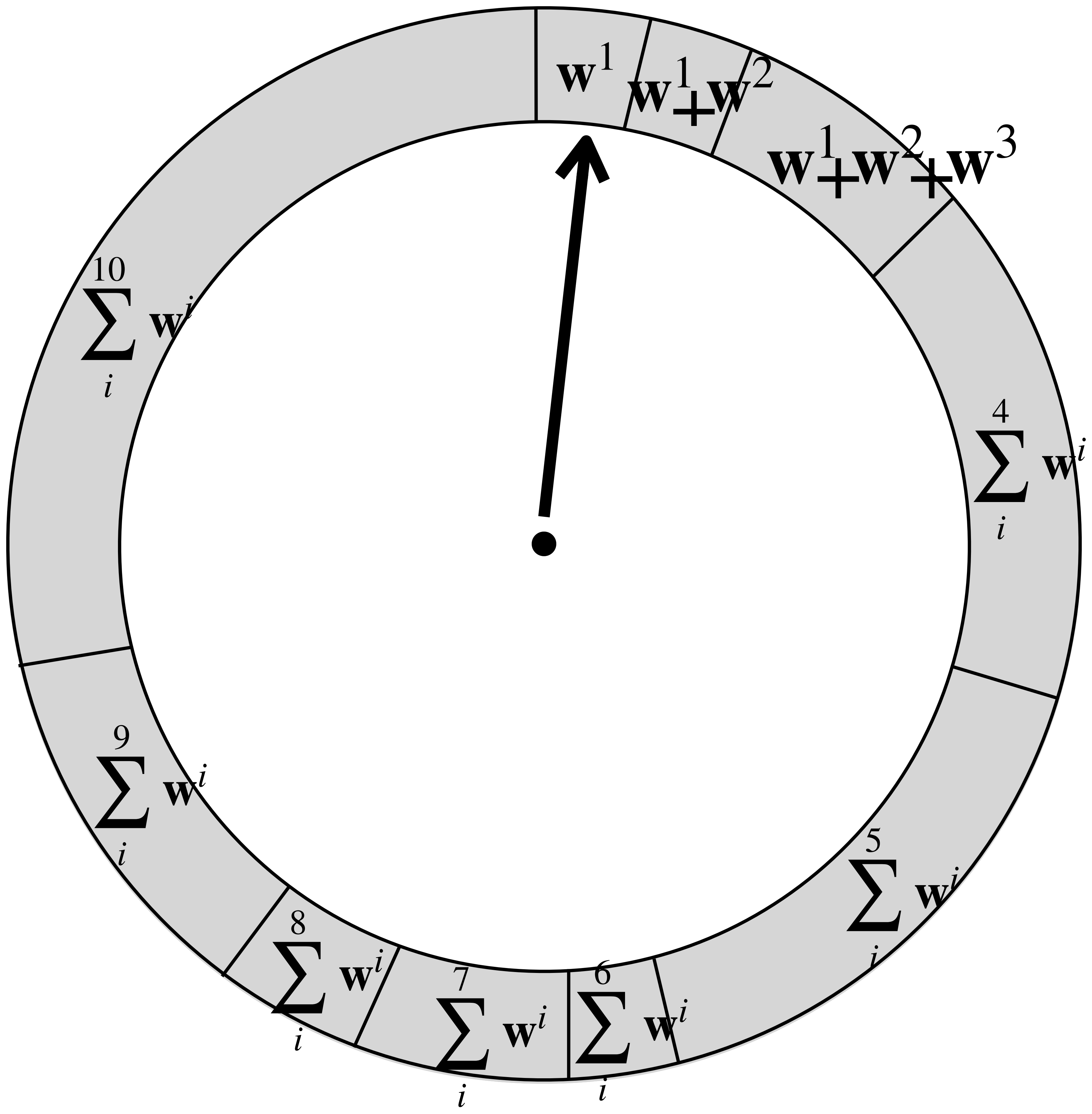
- N particles $\mathcal{O}\big(N\log(N)\big)$
- easy to understand

4. Resample

Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$

Generate only one small random number





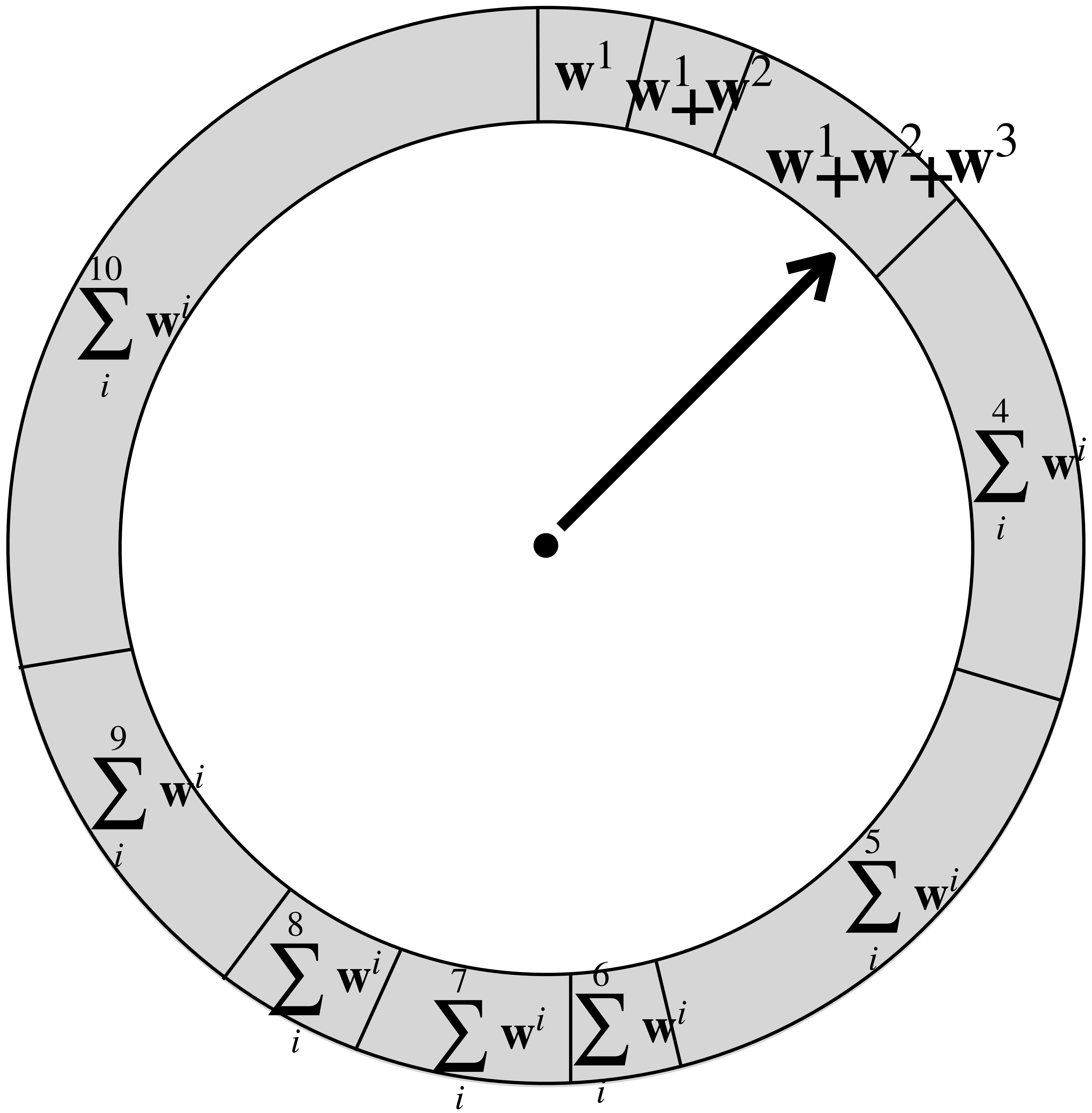
Roulette wheel

- N particles $\mathcal{O}\left(N \log(N)\right)$
- easy to understand

Stochastic universal resampling

4. Resample

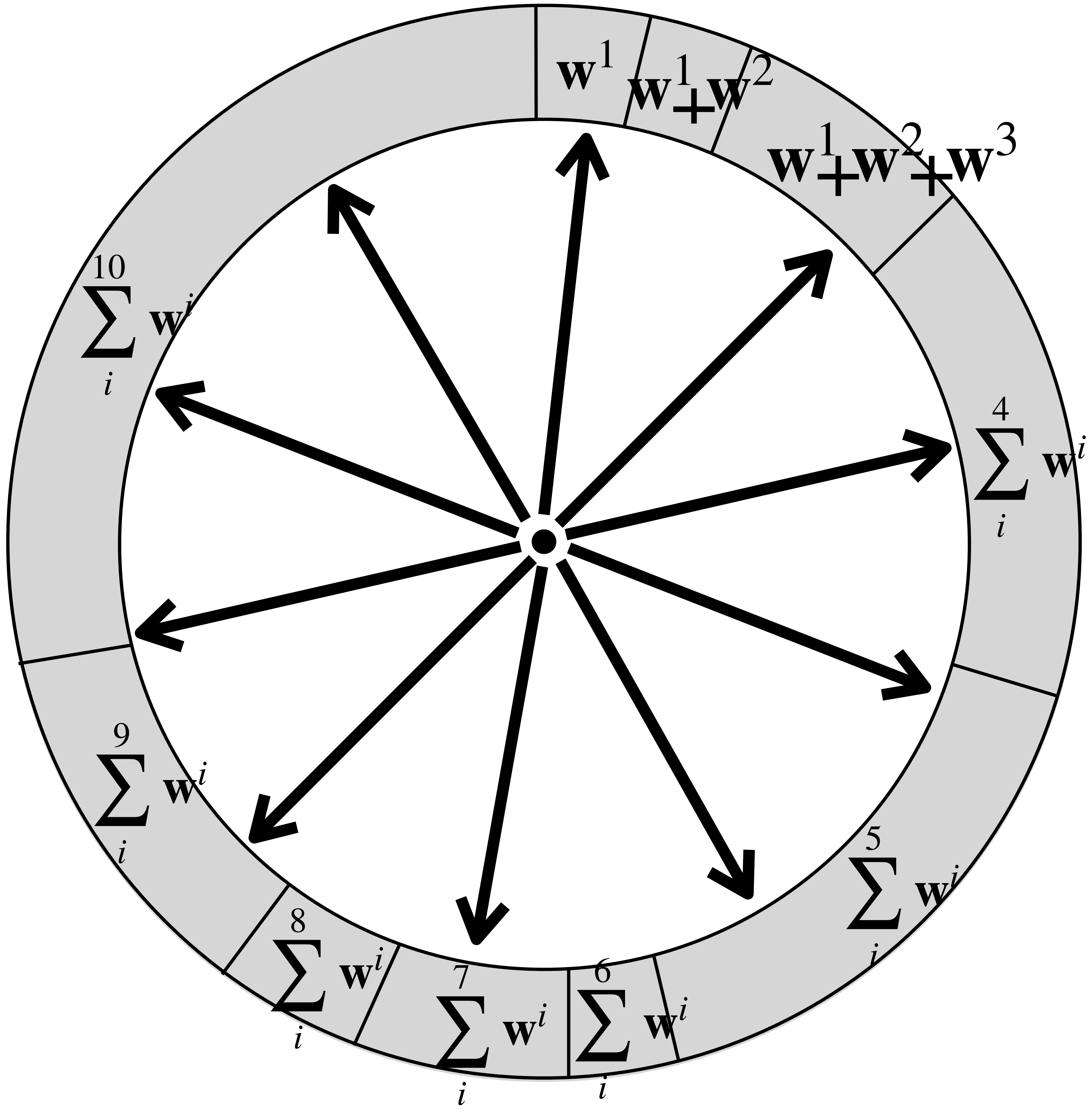
Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Roulette wheel

- N particles $\mathcal{O}\big(N \log(N)\big)$
- easy to understand

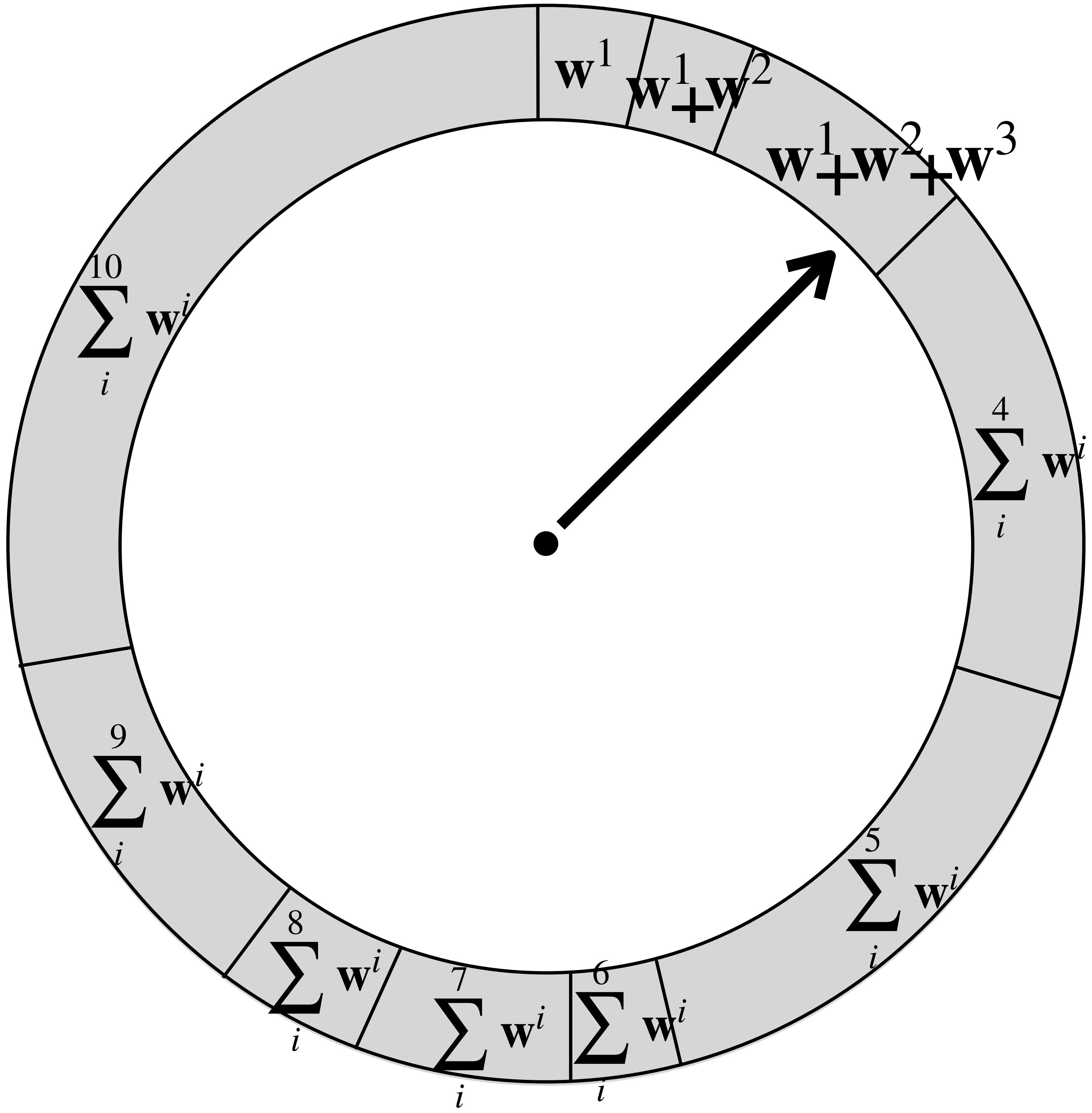
Generate N equally distributed samples starting at the random number



Stochastic universal resampling

4. Resample

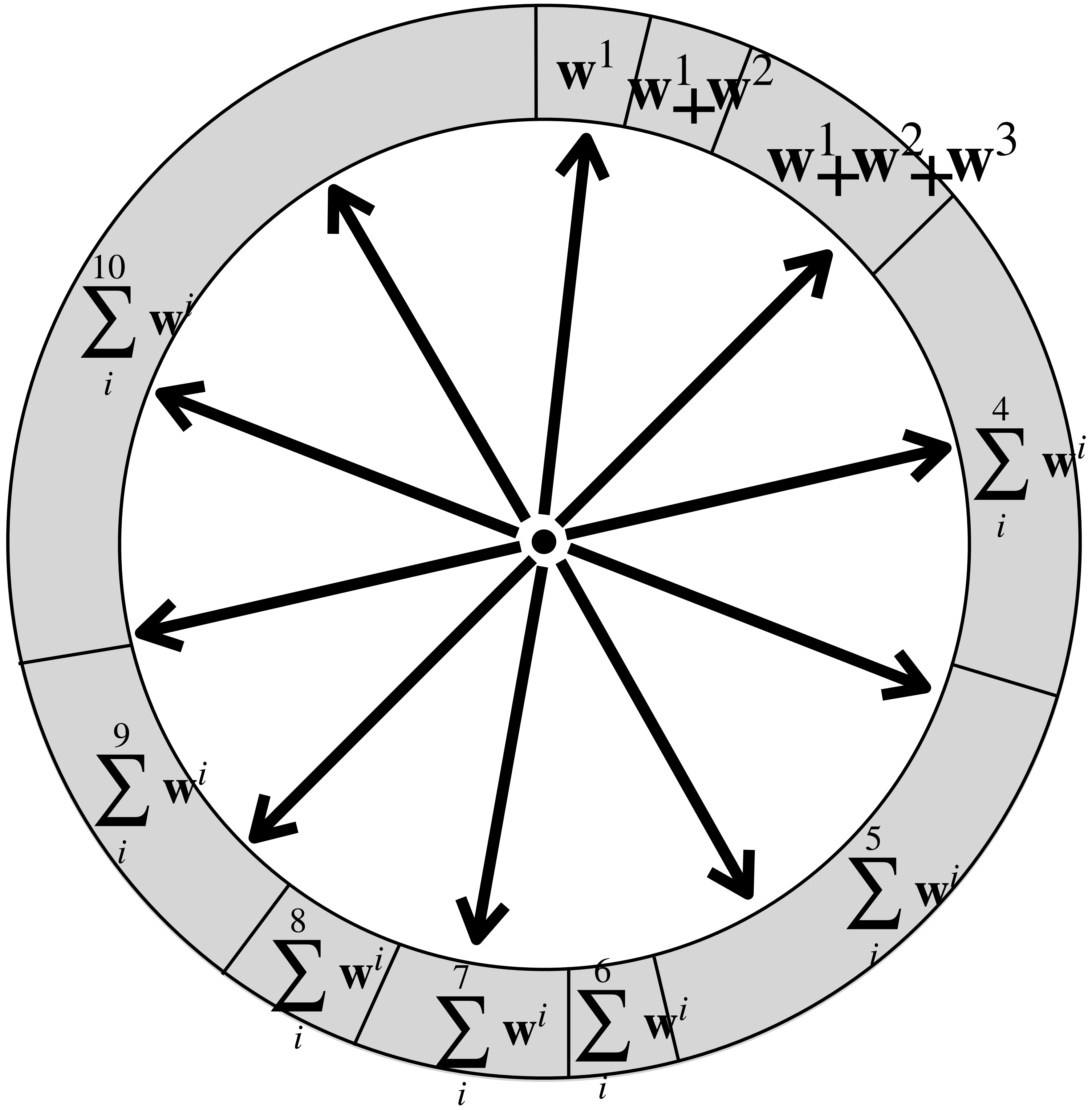
Draw $\mathscr{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Roulette wheel

- N particles $\mathcal{O}\big(N\log(N)\big)$
- easy to understand

Go through the wheel and update the slot if arrow is higher than cumsum value

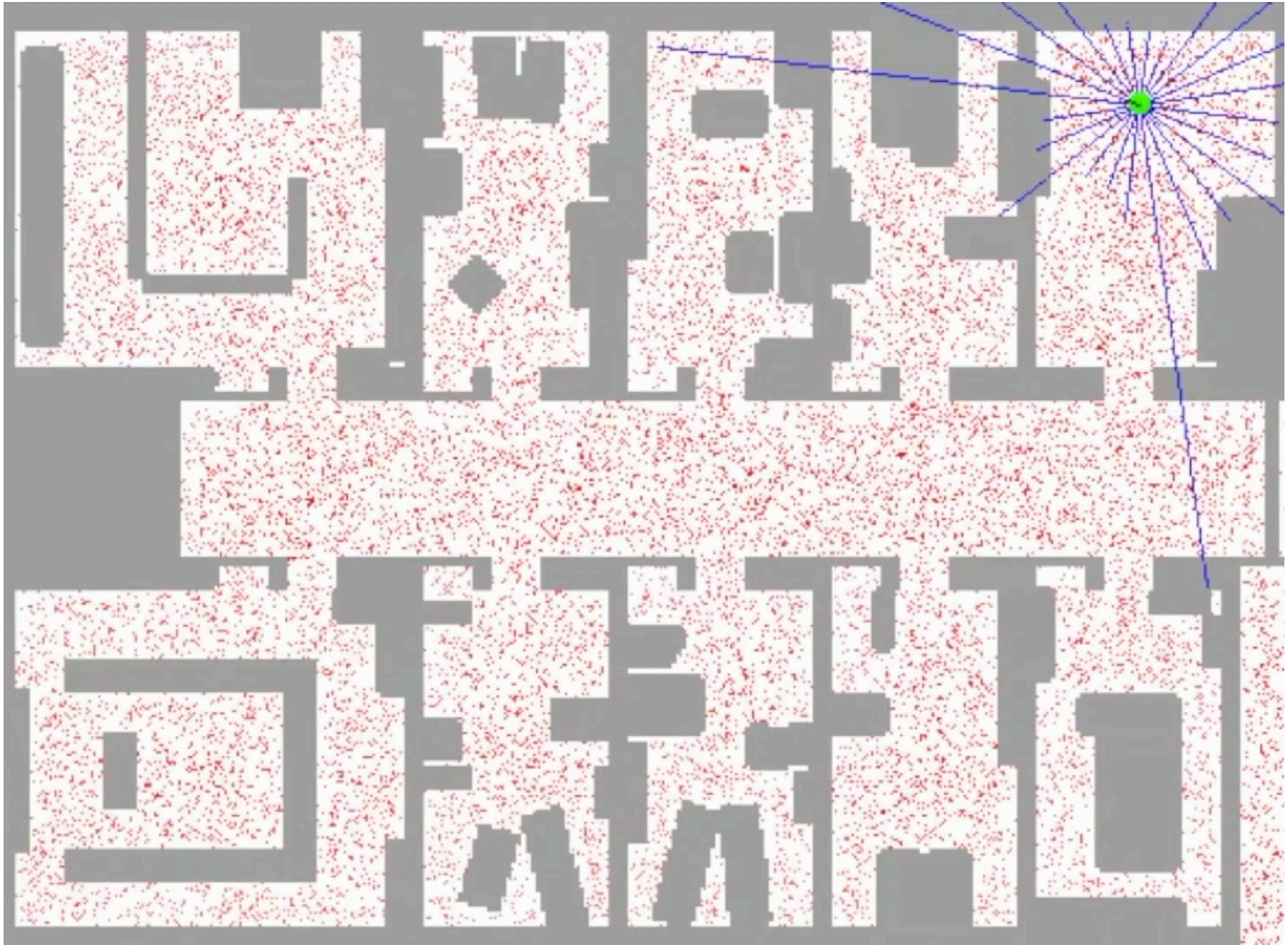


Stochastic universal resampling

- N particles $\mathcal{O}\big(N\big)$
- lower variance

Particle filter example [Dieter Fox]

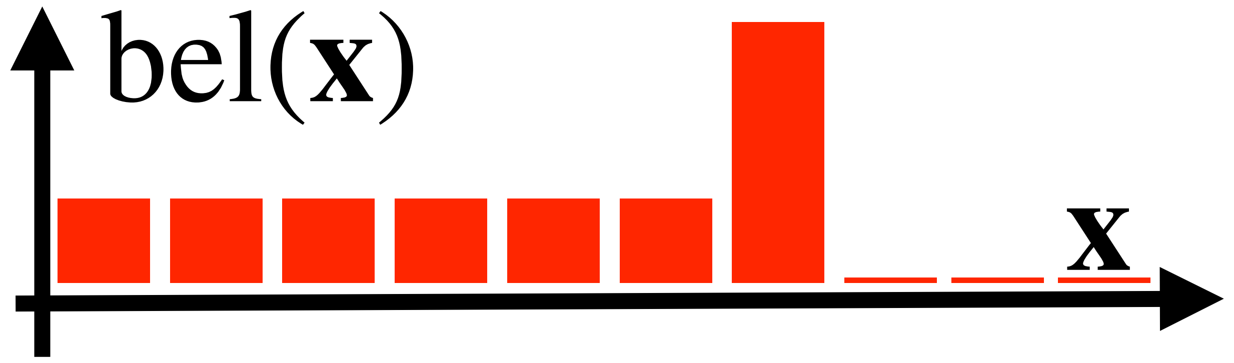
Particle filter example   [Dieter Fox]



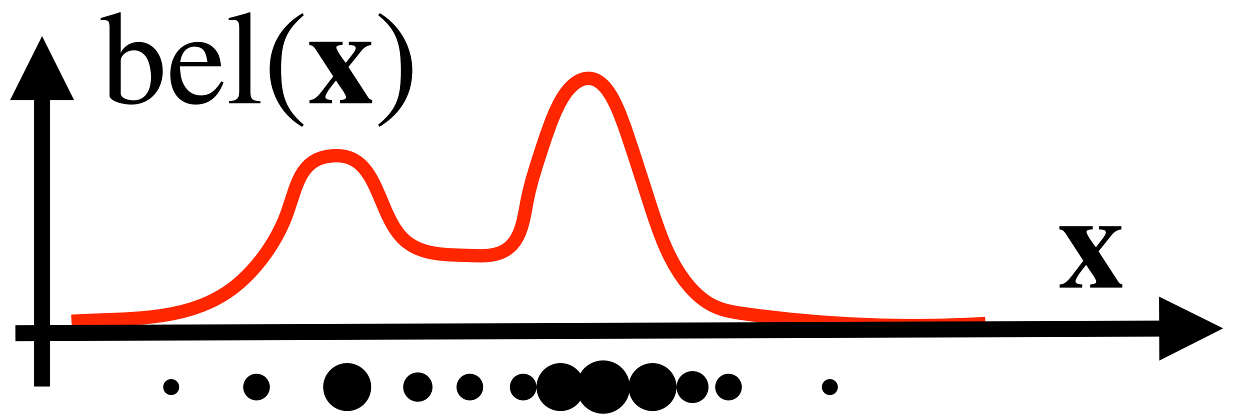
Summary

	Drawbacks	Advantages
Bayes filter	course of dimensionality; spatial discretization	represents arbitrary prob. distribution
Particle filter	course of dimensionality; partical quantization	represents arbitrary prob. distribution
Kalman filter	represent only gaussians; suffers from linearization	nicely scales with higher dimensions
Factorgraph	represents gaussians; grows to infinity	does not suffer from linearizations; allows for arbitrary conditional independences

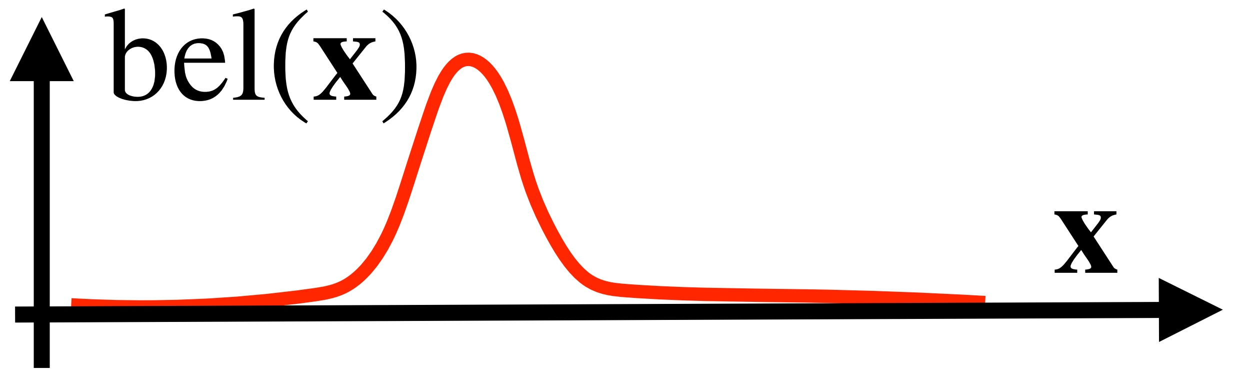
Bayes filter



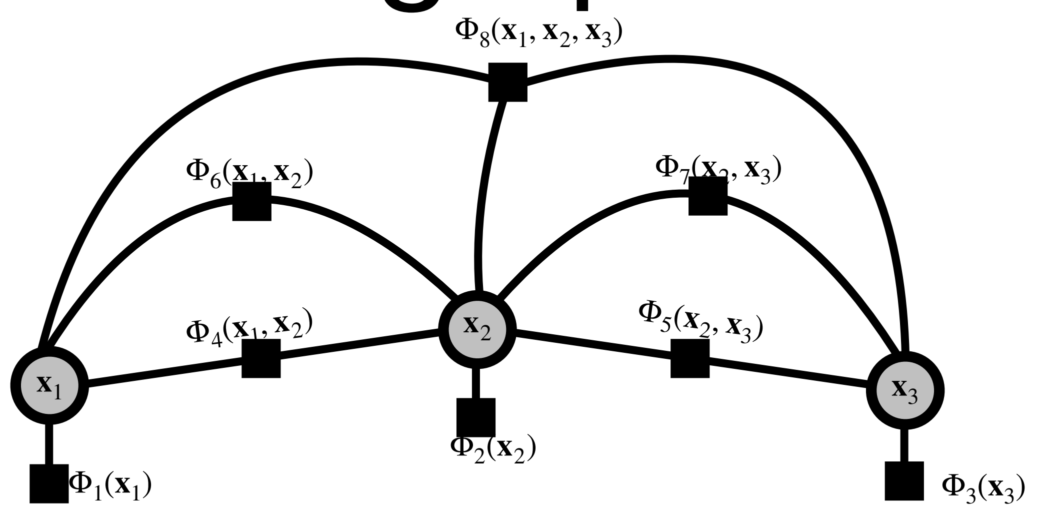
Particle filter



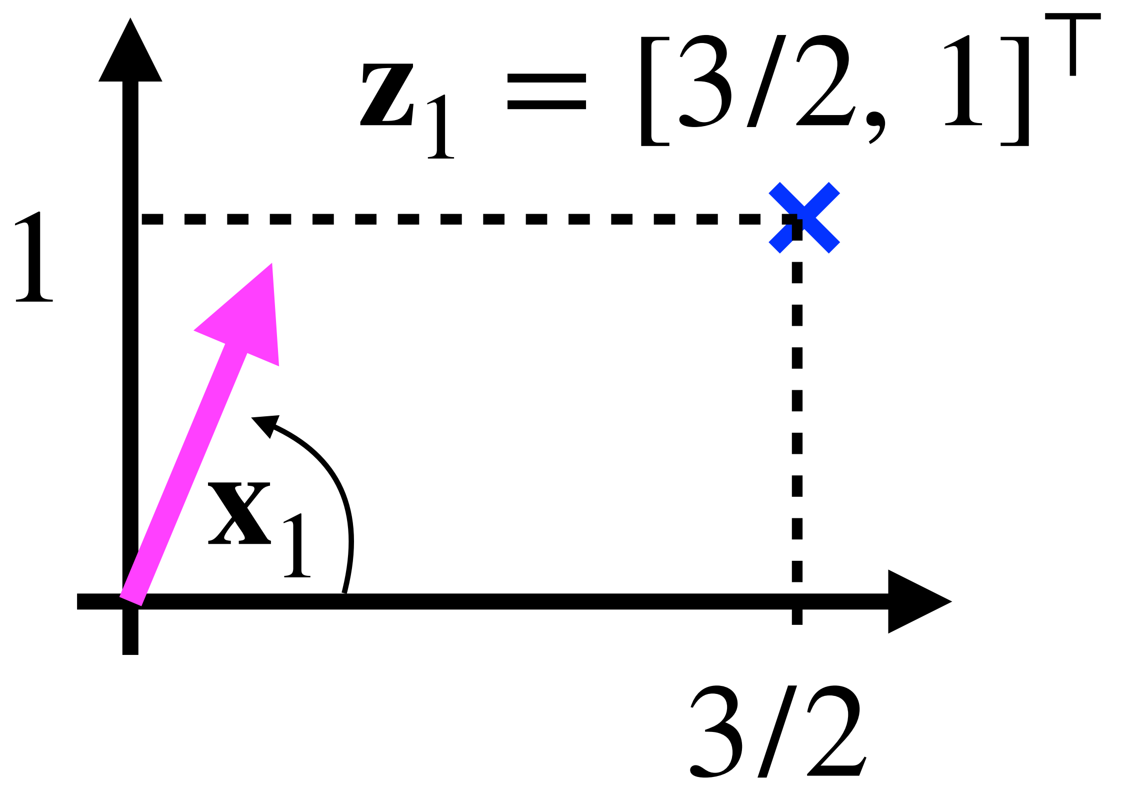
Kalman filter



Factorgraph



Assignment for solved example



- Robot is unit magenta arrow mounted to the origin of wcf by a swivel joint (i.e. it can only rotate around the point [0,0])
- State $\mathbf{x}_t \in \mathbb{R}$ is its (counter-clockwise) angle wrt x-axis
- Control $\mathbf{u}_t$ changes the state according to the motion model

  $$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{x}_{t-1} + \mathbf{u}_t$$

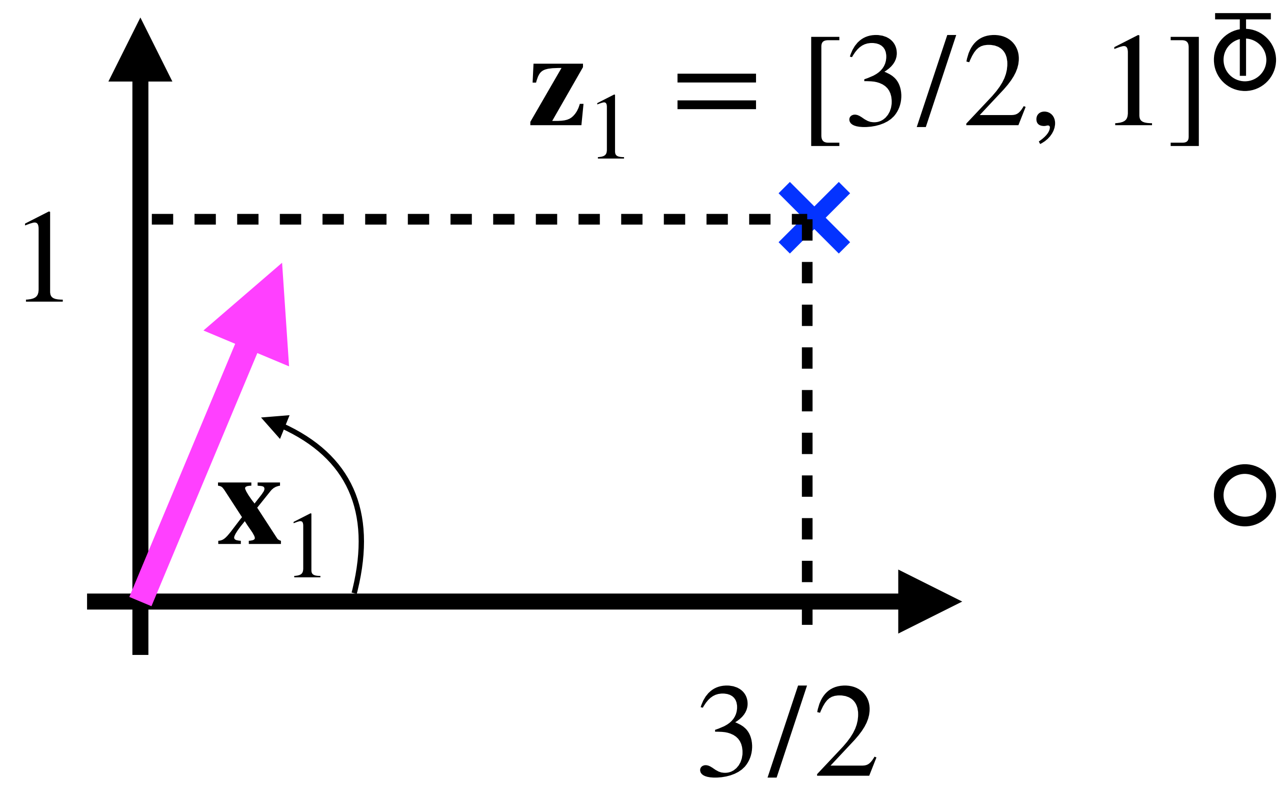
  with zero-mean gaussian noise with covariance $\mathbf{R}_t = 1$
- Measurement $\mathbf{z}_t \in \mathbb{R}^2$ is provided by GPS sensor with the measurement function

  $$\mathbf{z}_t = h(\mathbf{x}_t) = \begin{bmatrix} \cos \mathbf{x}_t \\ \sin \mathbf{x}_t \end{bmatrix}$$

  with zero-mean gaussian noise with covariance $\mathbf{Q}_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Consider two states example, where:

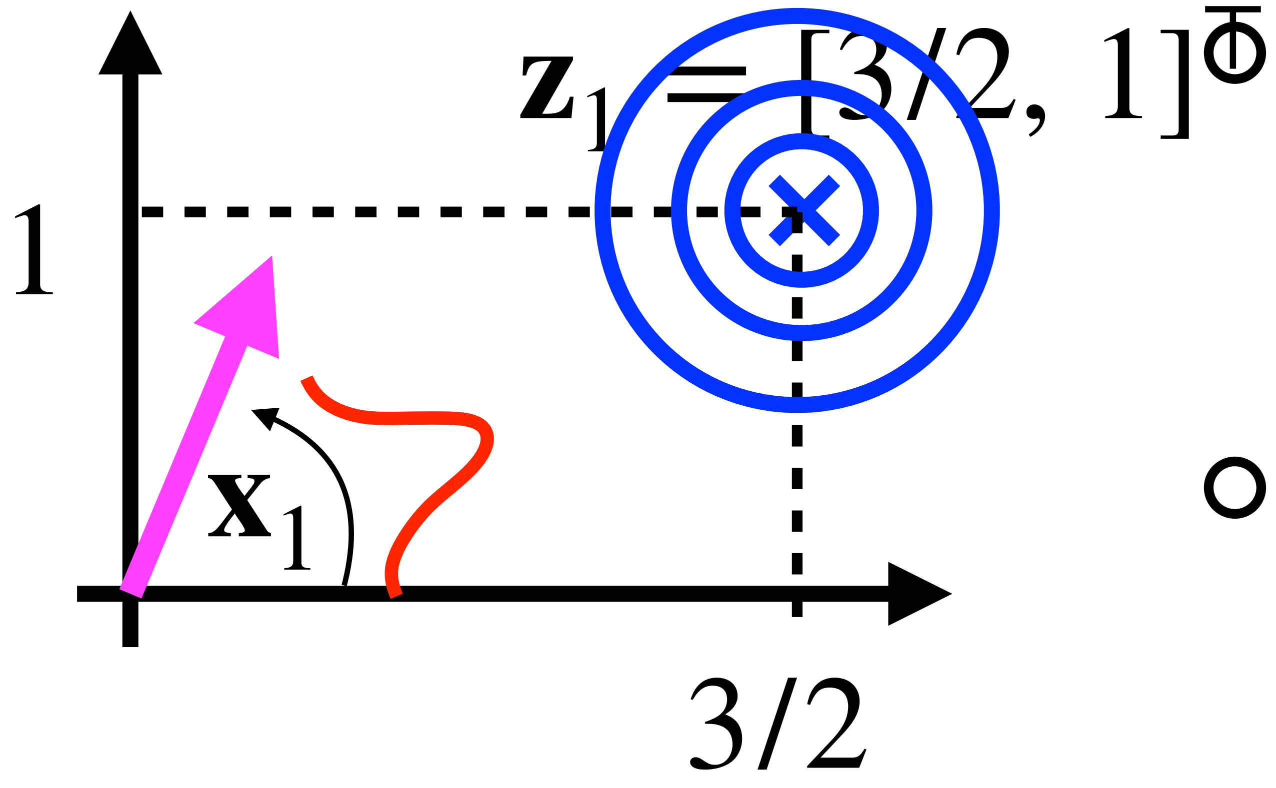
  $$\mathrm{bel}(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1;\ \mu_0 = 0,\ \Sigma_0 = 1), \quad \mathbf{z}_1 = \begin{bmatrix} 3/2 \\ 1 \end{bmatrix}, \quad \mathbf{u}_1 = \pi/2$$

Factorgraph



- Write down state-transition probability distribution
- Write down measurement probability distribution
- Outline distributions into the sketch
- Draw underlying factorgraph
- Write down MAP state estimation problem

Factorgraph (solution)



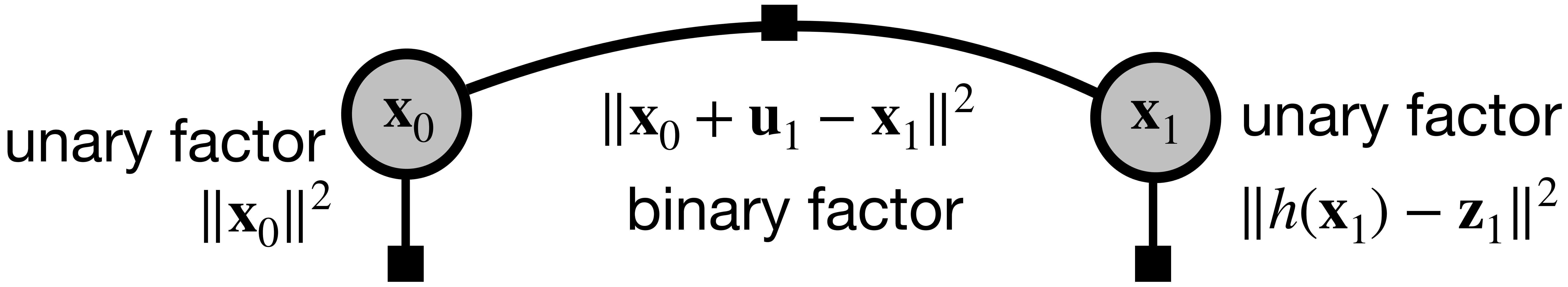
- Write down state-transition probability distribution

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_t;\ \mathbf{x}_{t-1} + \mathbf{u}_t,\ \mathbf{R}_t)$$

- Write down measurement probability distribution

$$p(\mathbf{z}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t;\ ,\ \mathbf{Q}_t)$$

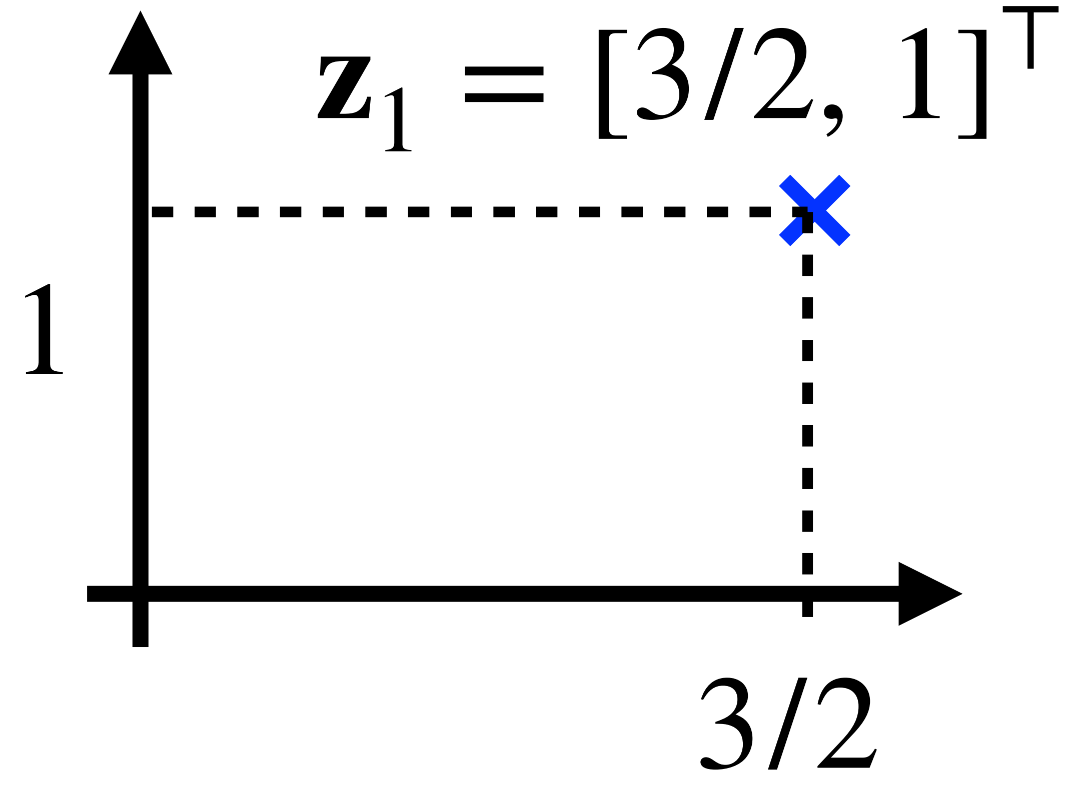
- Outline distributions into the sketch
- Draw underlying factorgraph



- Write down MAP state estimation problem

$$\arg\min_{\mathbf{x}_0, \mathbf{x}_1} \|\mathbf{x}_0 + \mathbf{u}_1 - \mathbf{x}_1\|^2 + \|h(\mathbf{x}_1) - \mathbf{z}_1\|^2 + \|\mathbf{x}_0\|^2$$

Extended Kalman Filter



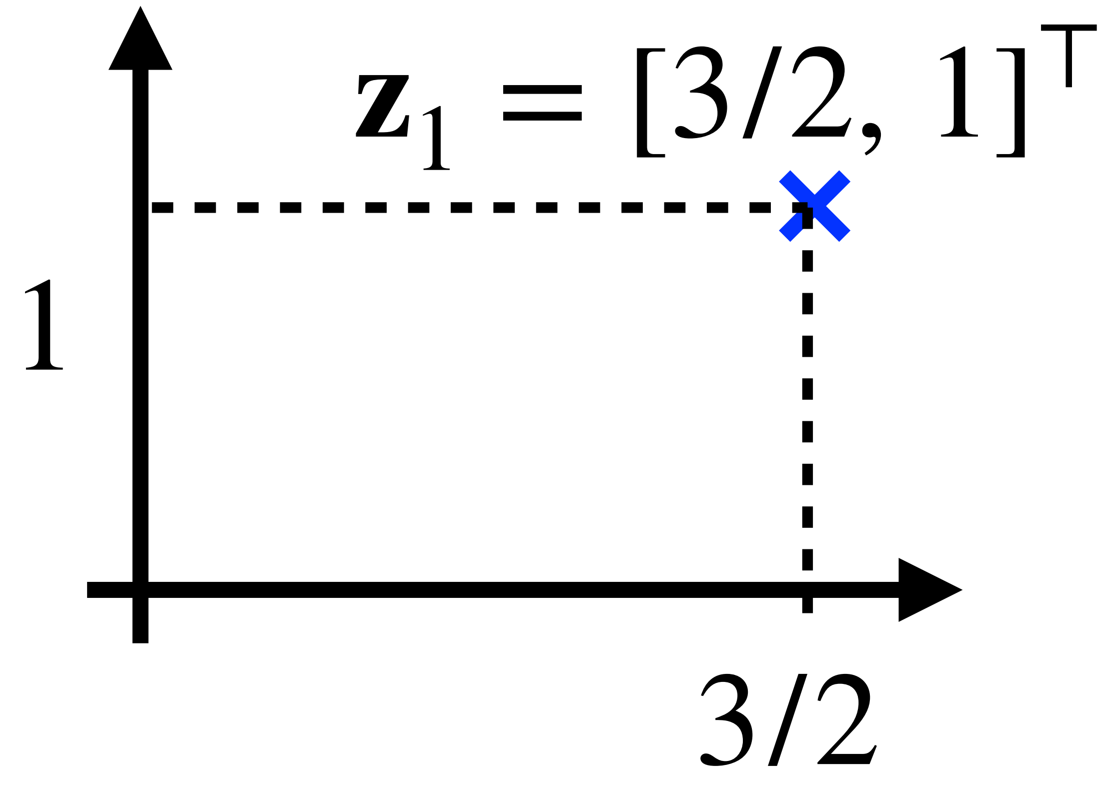
- **Perform prediction step of (E)KF,*** i.e. $\overline{\mathrm{bel}}(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1;\ \bar{\mu}_1, \bar{\mathbf{\Sigma}}_1)$

  $\bar{\mu}_1 = ?$

  $\bar{\mathbf{\Sigma}}_1 = ?$

- Linearize measurement function around $\bar{\mu}_1$ (outline it in sketch)

$$h(\mathbf{x}_1) = \begin{bmatrix} \cos(\mathbf{x}_1) \\ \sin(\mathbf{x}_1) \end{bmatrix} \approx ?$$



- **Perform measurement step of EKF**

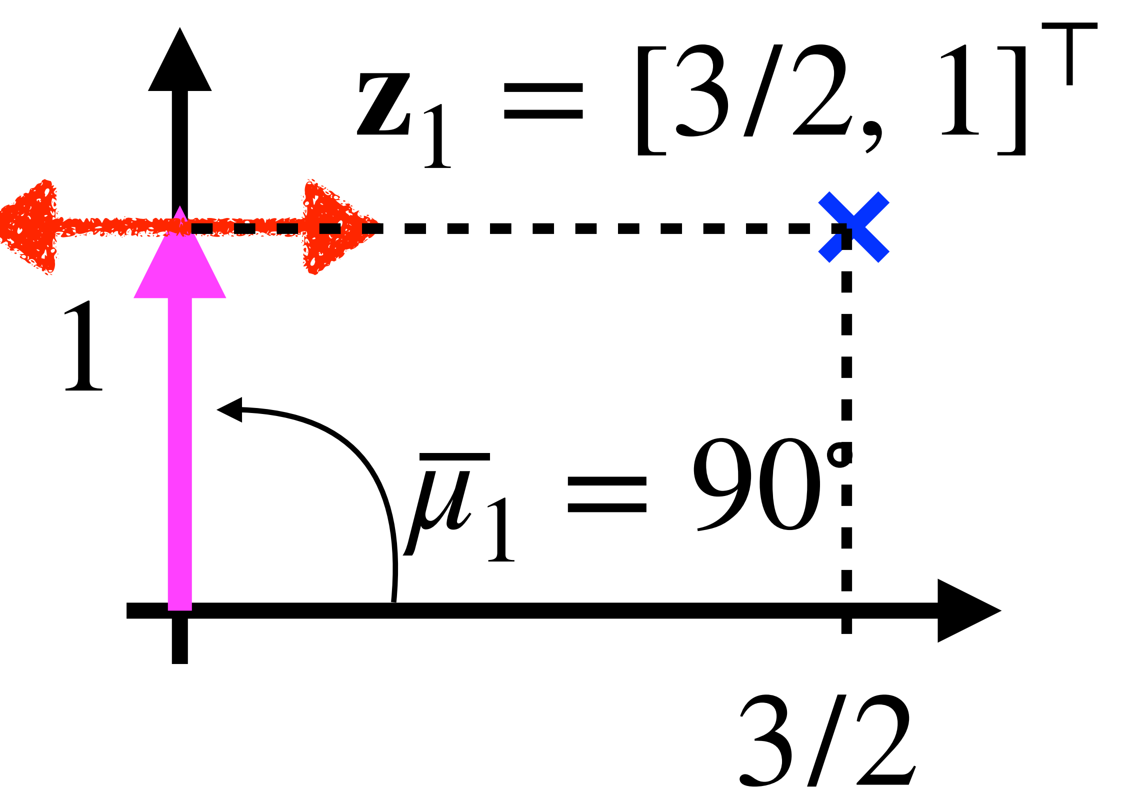
  $\mathbf{K}_1 = \bar{\mathbf{\Sigma}}_1 \mathbf{H}_1^\top (\mathbf{H}_t \bar{\mathbf{\Sigma}}_1 \mathbf{H}_t^\top + \mathbf{Q}_t)^{-1} = ?$

  $\mu_1 = \overline{\mu_1} + \mathbf{K}_1 \big(\mathbf{z}_1 - h(\bar{\mu}_1)\big) = ?$

  $\mathbf{\Sigma}_1 = (\mathbf{I} - \mathbf{K}_1 \mathbf{H}_1) \bar{\mathbf{\Sigma}}_1 = ?$

* It is strictly prohibited to memorize any EKF equations, however you are allowed to have it on your cheatsheet (and it will be also provided in the test assignment) ;-)

Extended Kalman Filter (solution)



- **Perform prediction step of (E)KF,*** i.e. $\overline{\mathrm{bel}}(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1;\ \bar{\mu}_1, \bar{\boldsymbol{\Sigma}}_1)$

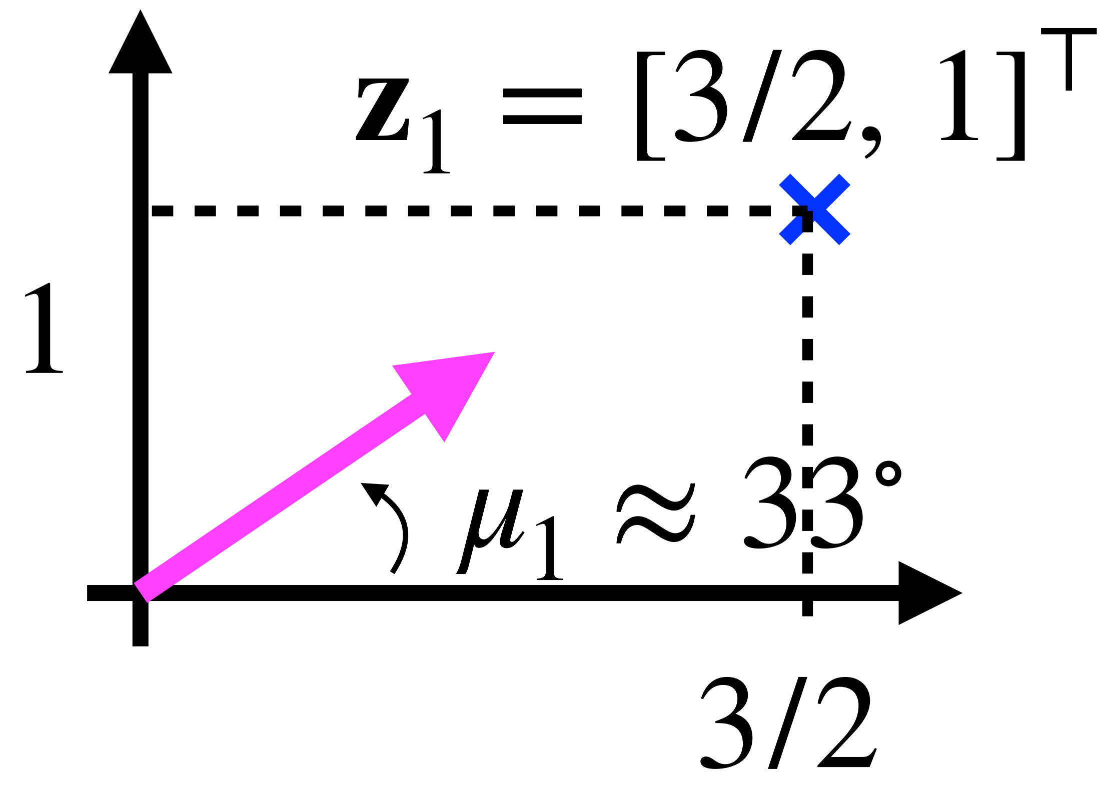
$$\bar{\mu}_1 = \bar{\mu}_0 + \mathbf{u}_1 = 0 + \frac{\pi}{2} = \frac{\pi}{2}$$

$$\bar{\boldsymbol{\Sigma}}_1 = \mathbf{G}_1 \boldsymbol{\Sigma}_0 \mathbf{G}_1^\top + \mathbf{R}_1 = 1 \cdot 1 \cdot 1 + 1 = 2$$

- Linearize measurement function around $\bar{\mu}_1$ (outline it in sketch)

$$h(\mathbf{x}_1) = \begin{bmatrix} \cos(\mathbf{x}_1) \\ \sin(\mathbf{x}_1) \end{bmatrix} \approx \begin{bmatrix} \cos(\bar{\mu}_1) \\ \sin(\bar{\mu}_1) \end{bmatrix} + \begin{bmatrix} -\sin(\bar{\mu}_1) \\ \cos(\bar{\mu}_1) \end{bmatrix} \cdot \mathbf{x}_1 = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{h(\bar{\mu}_1)} + \underbrace{\begin{bmatrix} -1 \\ 0 \end{bmatrix}}_{\mathbf{H}_1} \cdot \mathbf{x}_1$$

- **Perform measurement step of EKF**



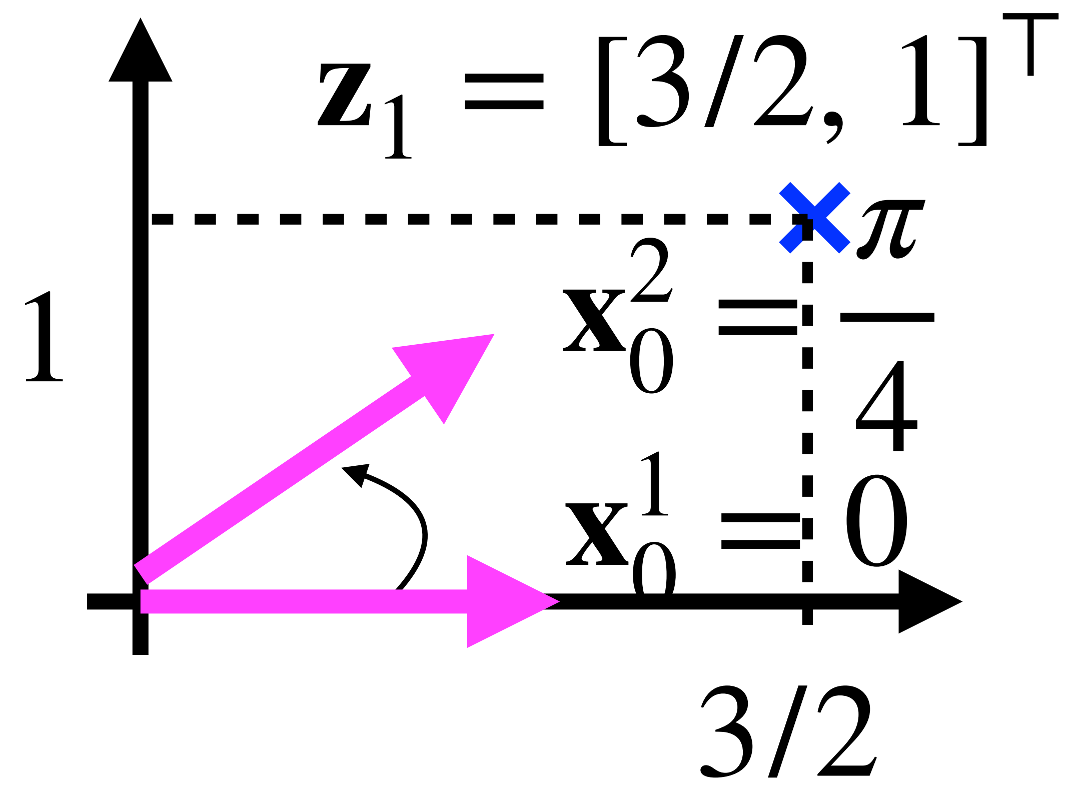
$$\mathbf{K}_1 = \bar{\boldsymbol{\Sigma}}_1 \mathbf{H}_1^\top (\mathbf{H}_t \bar{\boldsymbol{\Sigma}}_1 \mathbf{H}_t^\top + \mathbf{Q}_t)^{-1} = [-2 \quad 0] \cdot \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = [-2/3 \quad 0]$$

$$\mu_1 = \bar{\mu}_1 + \mathbf{K}_1 \big(\mathbf{z}_1 - h(\bar{\mu}_1) \big) = \frac{\pi}{2} - 1 \approx 33°$$

$$\boldsymbol{\Sigma}_1 = (\mathbf{I} - \mathbf{K}_1 \mathbf{H}_1) \bar{\boldsymbol{\Sigma}}_1 = (1 - 2/3) \cdot 2 = 2/3$$

* It is strictly prohibited to memorize any EKF equations, however you are allowed to have it on your cheatsheet (and it will be also provided in the test assignment) ;-)

Partical filter



Prediction step of PF:

- Particles representing $\overline{\mathrm{bel}}(\mathbf{x}_1)$ are drawn from this distribution:

  $\bar{\mathbf{x}}_1^1 \sim\ ?$

  $\bar{\mathbf{x}}_1^2 \sim\ ?$

- Assume zero noise and generate particles in the mean values

  $\bar{\mathbf{x}}_1^1 =\ ?$ $\bar{\mathbf{x}}_1^2 =\ ?$

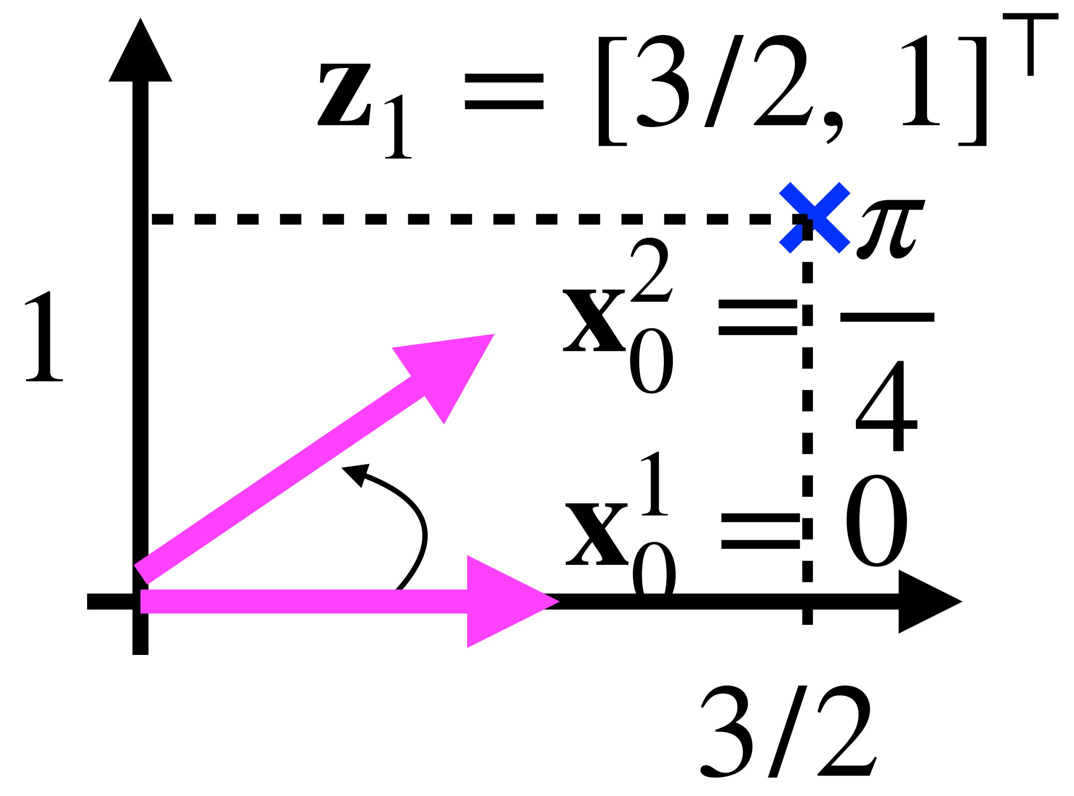
Measurement step of PF:

- Update weights of particles to represent $\mathrm{bel}(\mathbf{x}_1)$

  $\mathbf{w}_1^1 =\ ?$ $\mathbf{w}_1^2 =\ ?$

Which particle has a higher chance to survive the resampling?

Partical filter (solution)



Prediction step of PF:

- Particles representing $\overline{\text{bel}}(\mathbf{x}_1)$ are drawn from this distribution:

$$\bar{\mathbf{x}}_1^1 \sim p(\mathbf{x}_1 \,|\, \mathbf{x}_0^1, \mathbf{u}_1) = \mathcal{N}(\mathbf{x}_1;\ \mathbf{x}_0^1 + \mathbf{u}_1, \mathbf{R}_1) = \mathcal{N}(\mathbf{x}_1;\ \frac{\pi}{2}, 1)$$

$$\bar{\mathbf{x}}_1^2 \sim p(\mathbf{x}_1 \,|\, \mathbf{x}_0^2, \mathbf{u}_1) = \mathcal{N}(\mathbf{x}_1;\ \mathbf{x}_0^2 + \mathbf{u}_1, \mathbf{R}_1) = \mathcal{N}(\mathbf{x}_1;\ \frac{3\pi}{4}, 1)$$

- Assume zero noise and generate particles in the mean values

$$\bar{\mathbf{x}}_1^1 = \frac{\pi}{2}\ , \qquad \bar{\mathbf{x}}_1^2 = \frac{3\pi}{4}$$

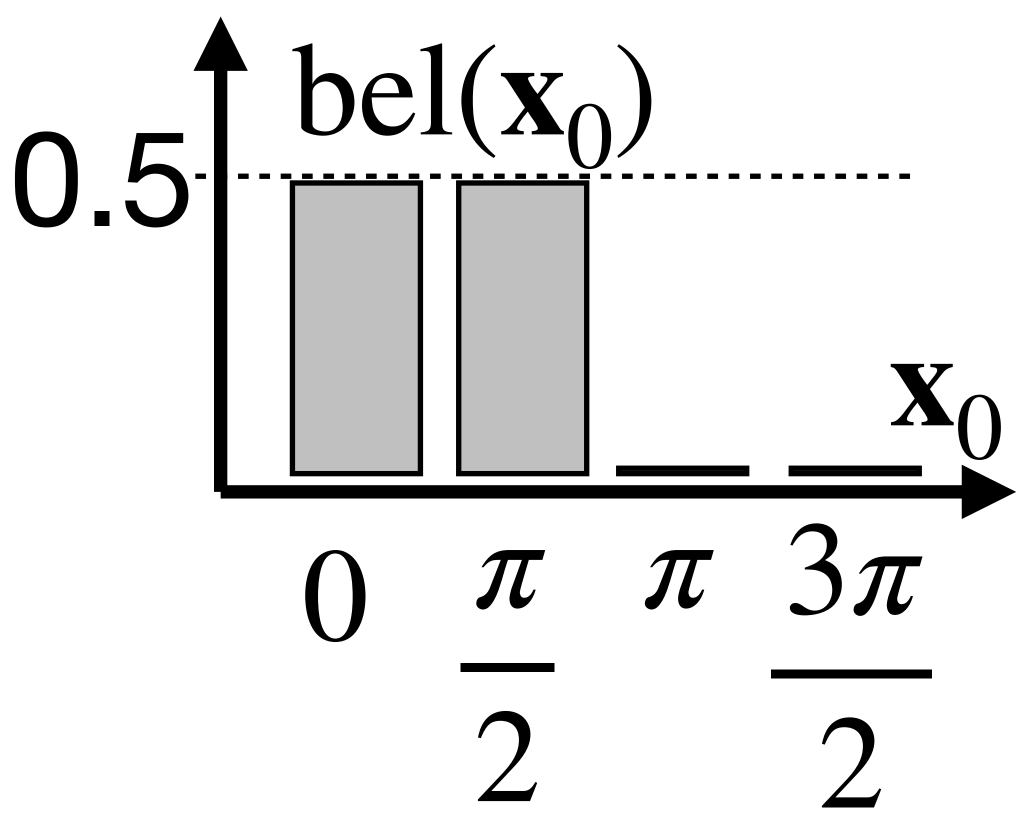
Measurement step of PF:

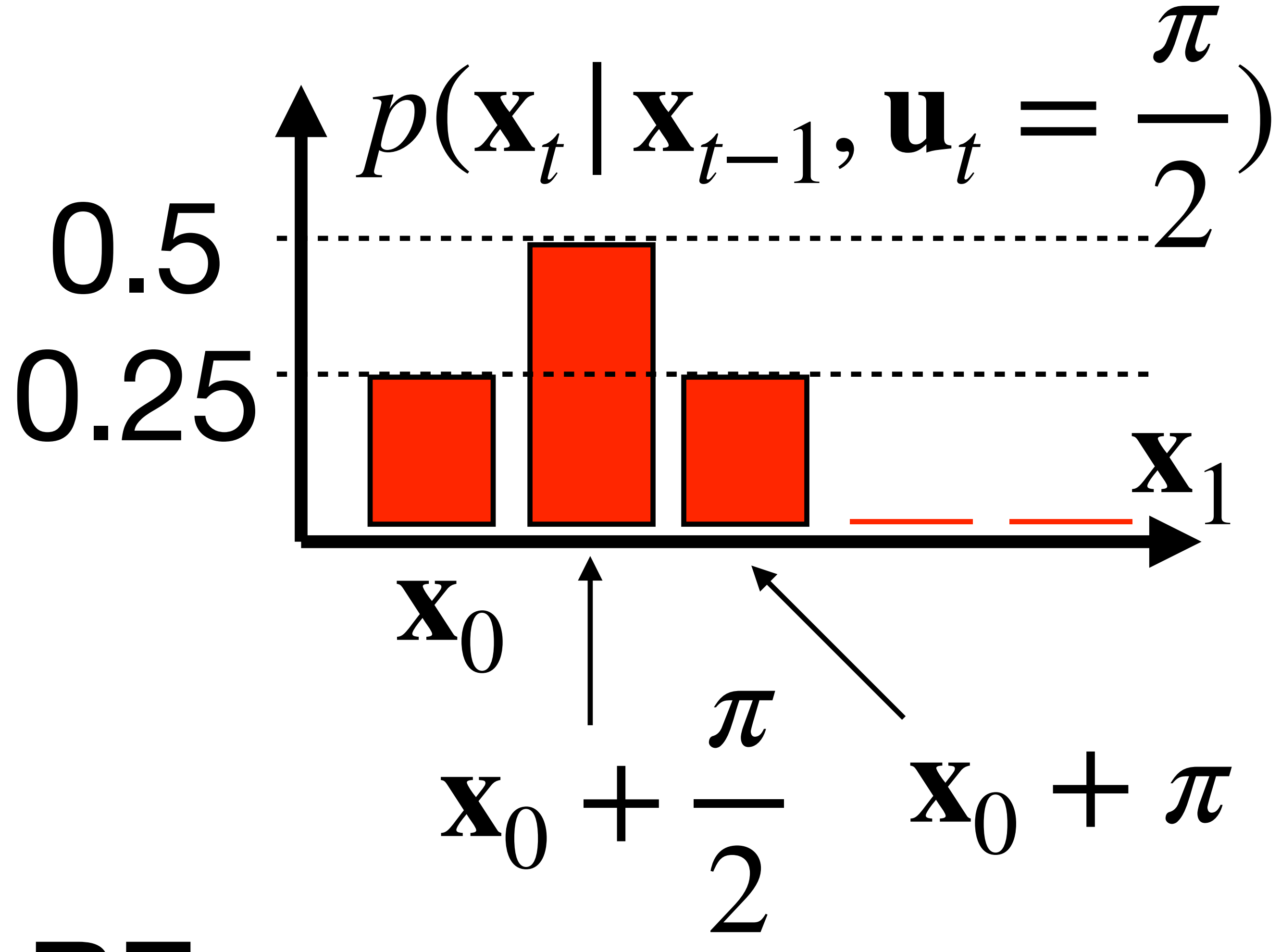
- Update weights of particles to represent $\text{bel}(\mathbf{x}_1)$

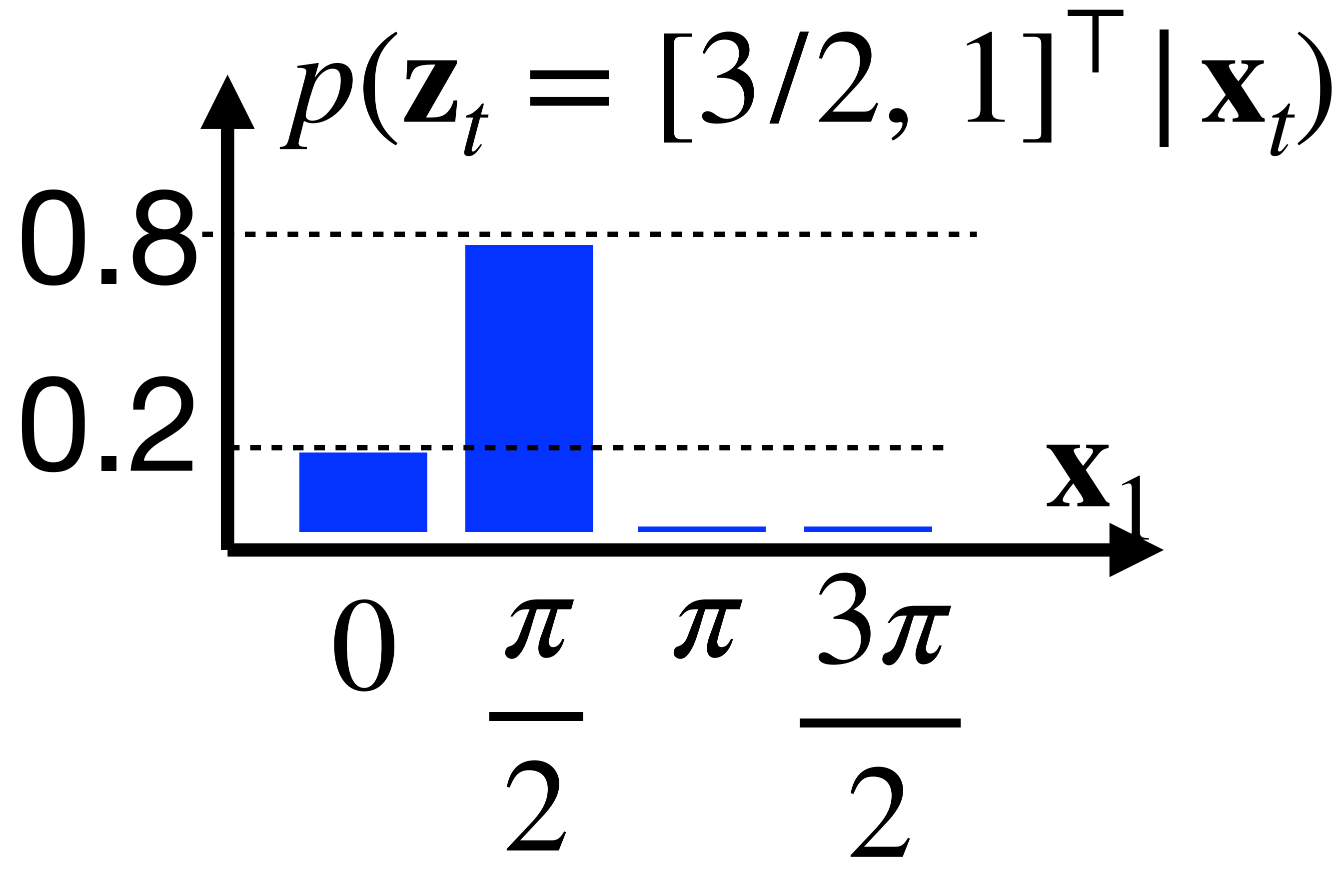
$$\mathbf{w}_1^1 = \mathcal{N}\left(\underbrace{\begin{bmatrix} 3/2 \\ 1 \end{bmatrix}}_{\mathbf{z}_1};\ \underbrace{\begin{bmatrix} \cos\frac{\pi}{2} \\ \sin\frac{\pi}{2} \end{bmatrix}}_{h(\mathbf{x}_1^1)},\ \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{Q}_1}\right) \qquad \mathbf{w}_1^2 = \mathcal{N}\left(\underbrace{\begin{bmatrix} 3/2 \\ 1 \end{bmatrix}}_{\mathbf{z}_1};\ \underbrace{\begin{bmatrix} \cos\frac{3\pi}{4} \\ \sin\frac{3\pi}{4} \end{bmatrix}}_{h(\mathbf{x}_1^2)},\ \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{Q}_1}\right)$$

Which particle has a higher chance to survive the resampling? $\mathbf{w}_1^1$

Discrete Bayes filter

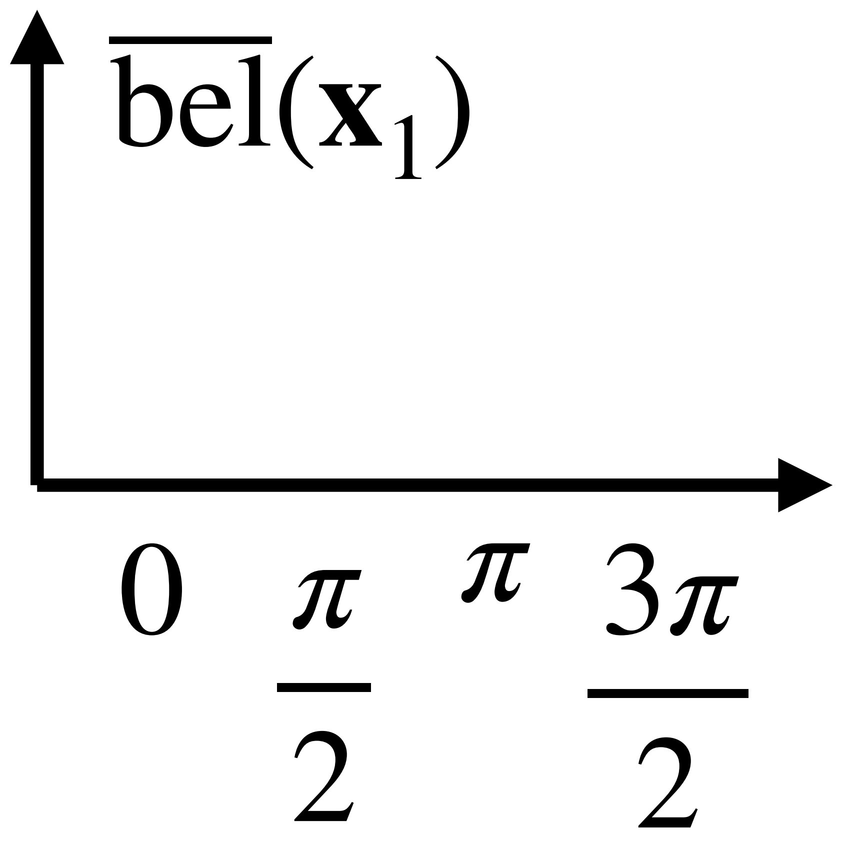






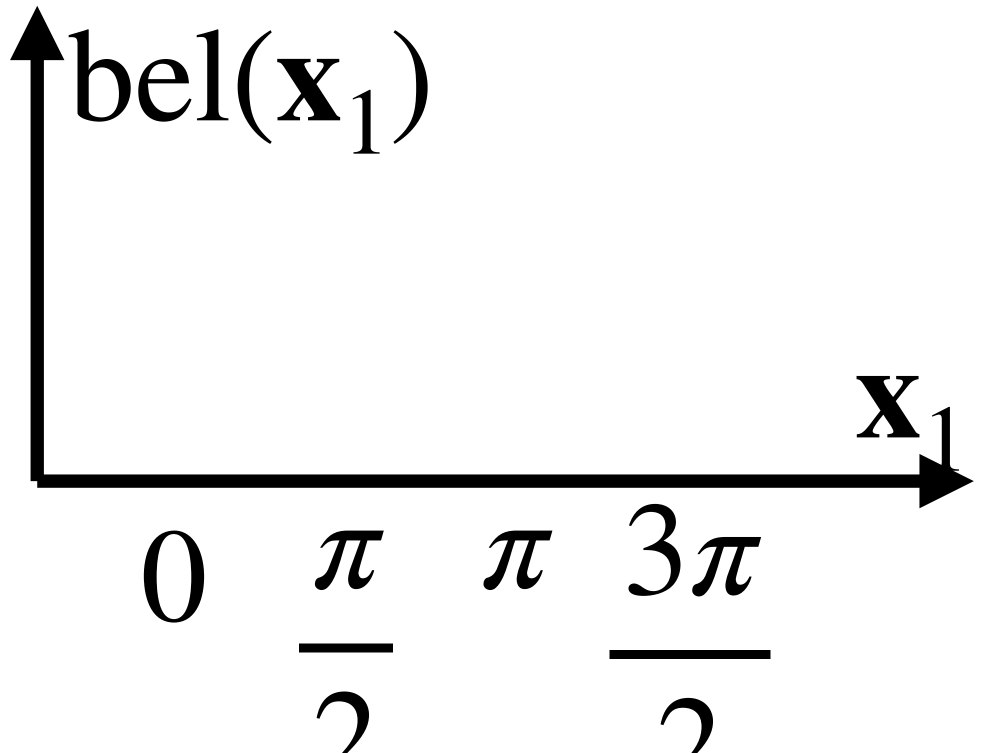
Prediction step of BF:

$$\overline{\text{bel}}(\mathbf{x}_t) = \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot \text{bel}(\mathbf{x}_{t-1})$$

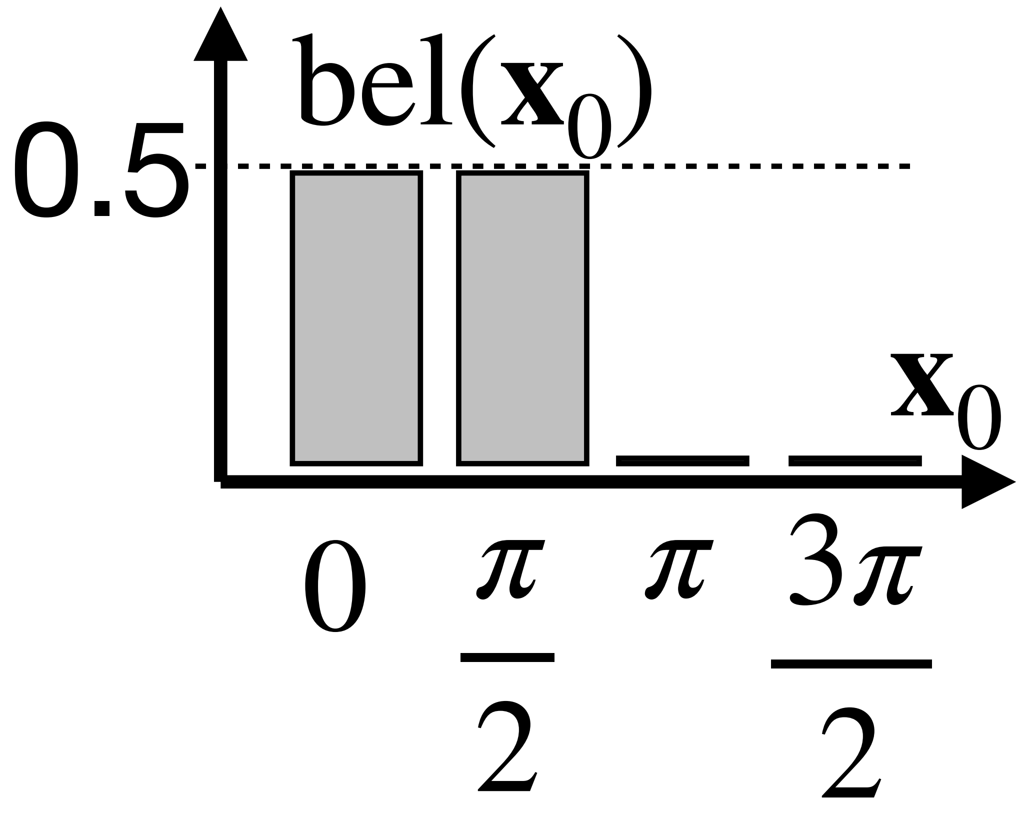


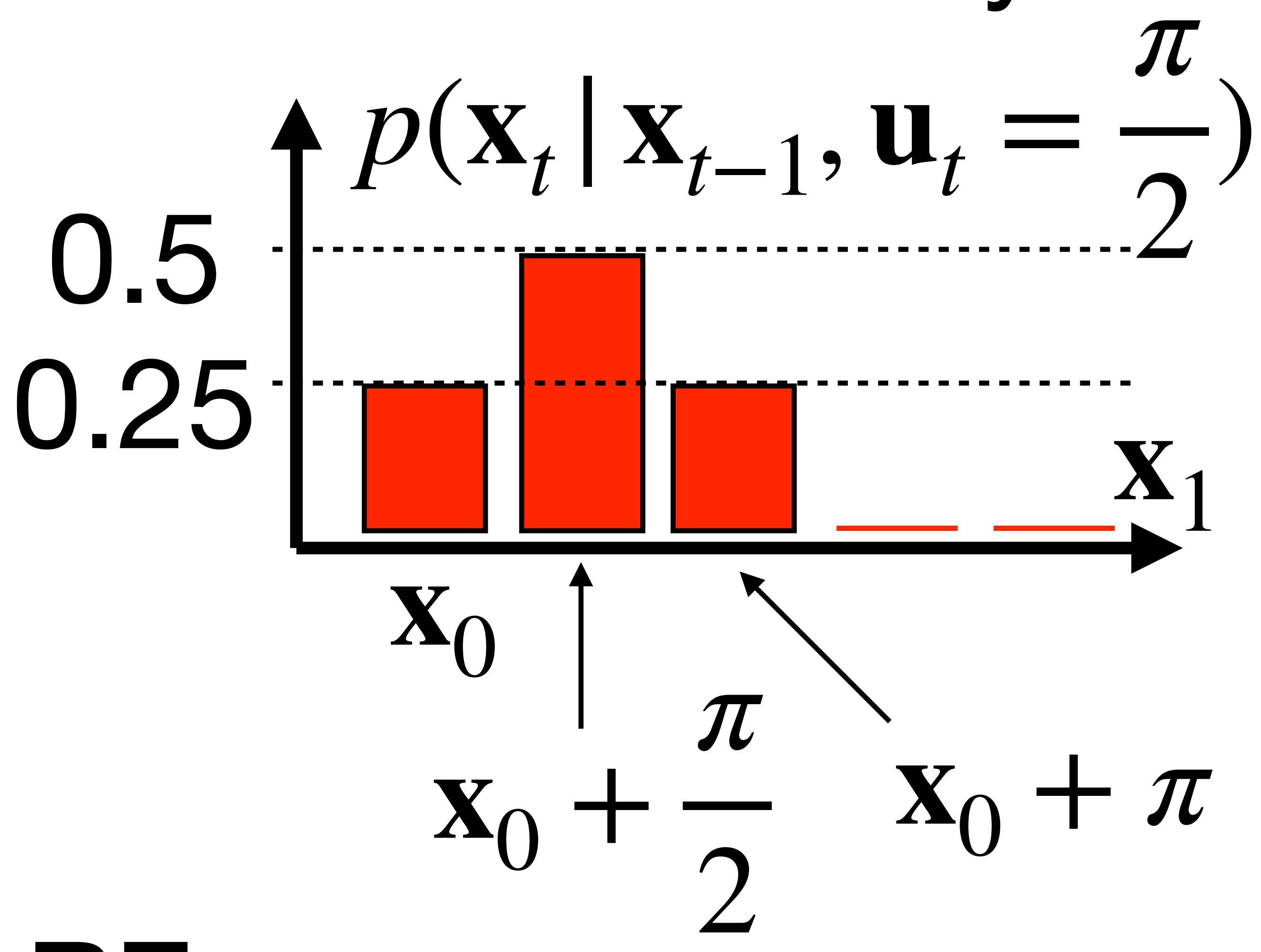
Measurement step of BF:

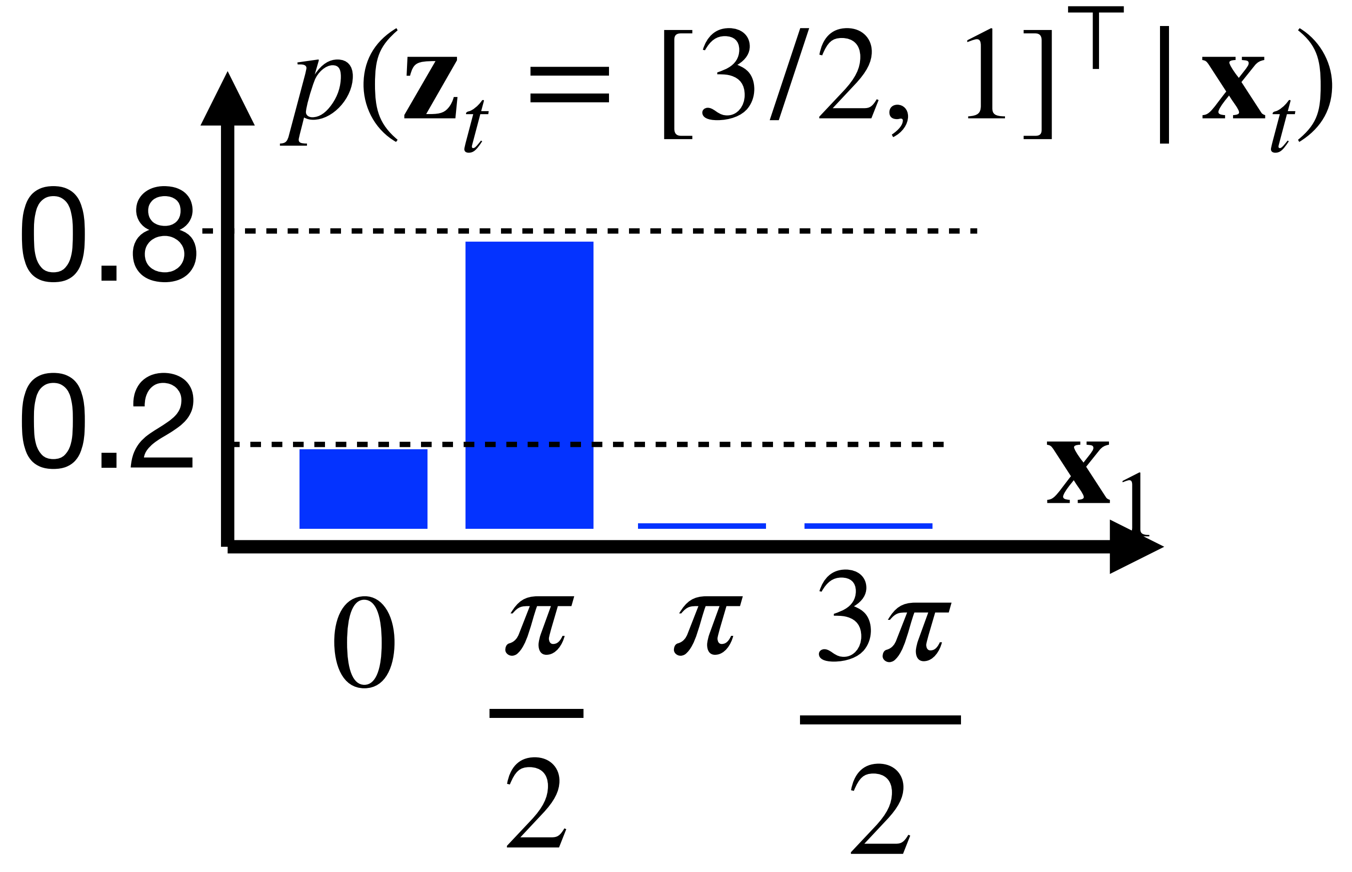
$$\text{bel}(\mathbf{x}_t) = \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t)\overline{\text{bel}}(\mathbf{x}_t)$$



Discrete Bayes filter (solution)

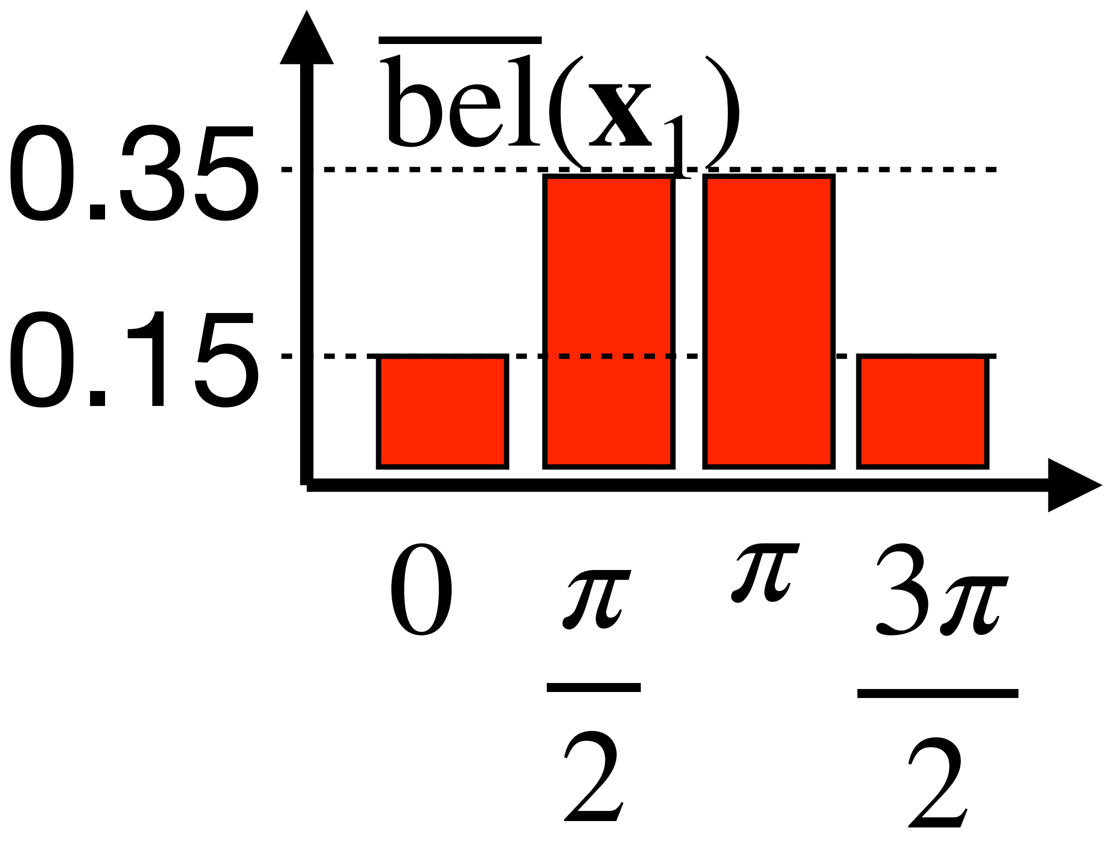






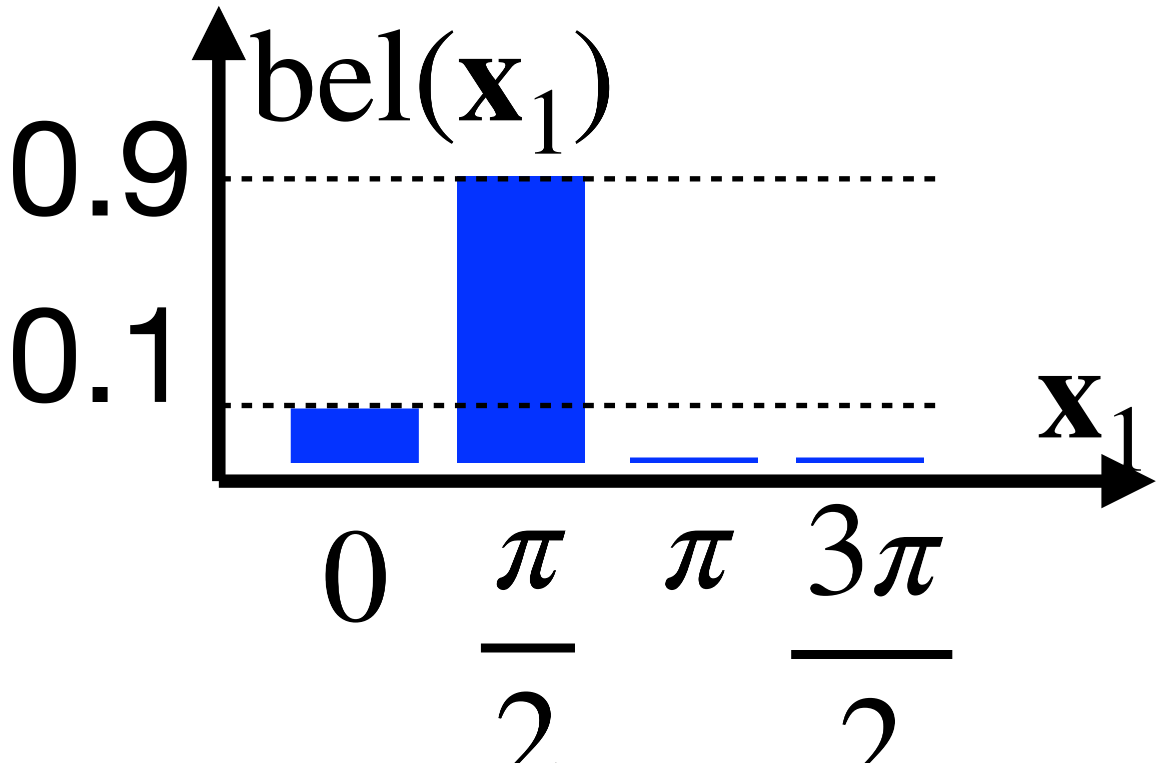
Prediction step of BF:

$$\overline{\text{bel}}(\mathbf{x}_t) = \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \cdot \text{bel}(\mathbf{x}_{t-1})$$

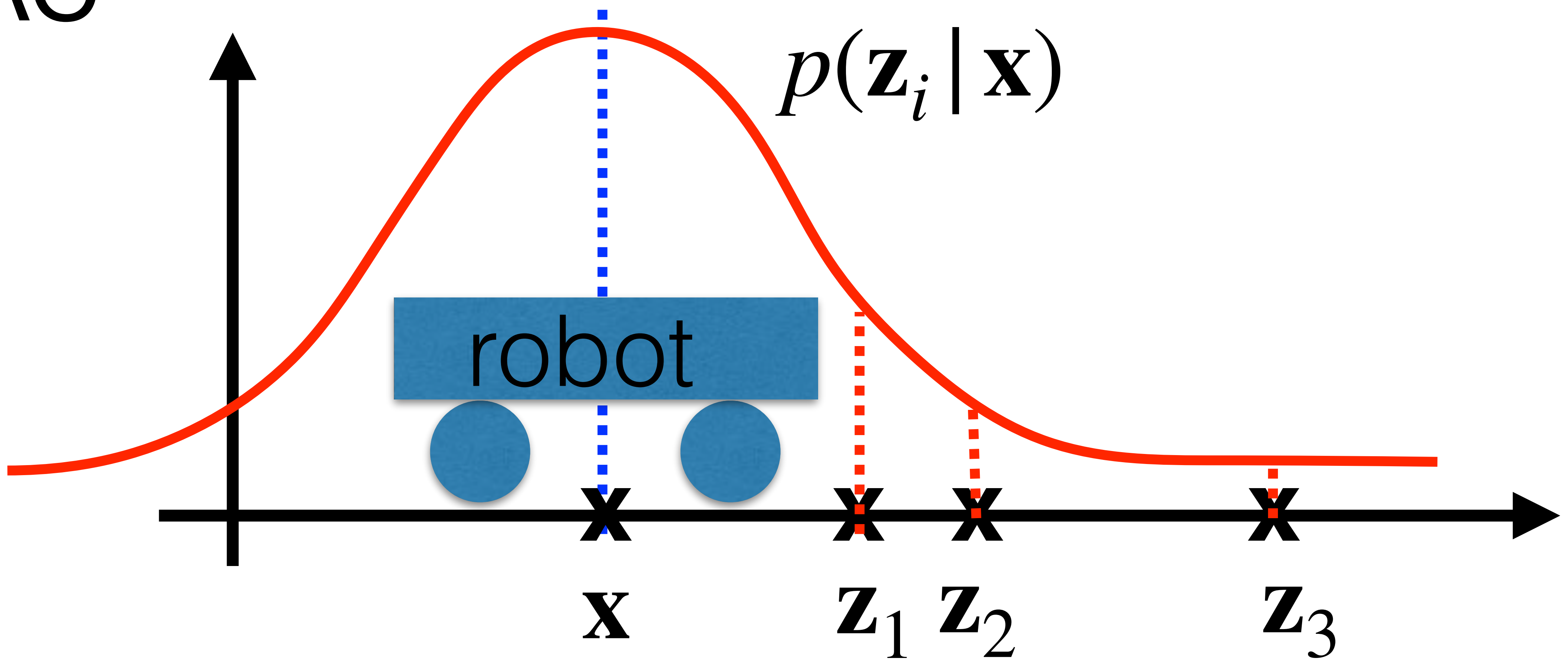


Measurement step of BF:

$$\text{bel}(\mathbf{x}_t) = \eta \cdot p(\mathbf{z}_t \mid \mathbf{x}_t)\overline{\text{bel}}(\mathbf{x}_t)$$



RANSAC



- Assume that
  - no motion model is applied,
  - no prior probability distribution
- GPS position is measured three-times:

  $\mathbf{z}_1 = 2 \qquad \mathbf{z}_2 = 3 \qquad \mathbf{z}_3 = 7$

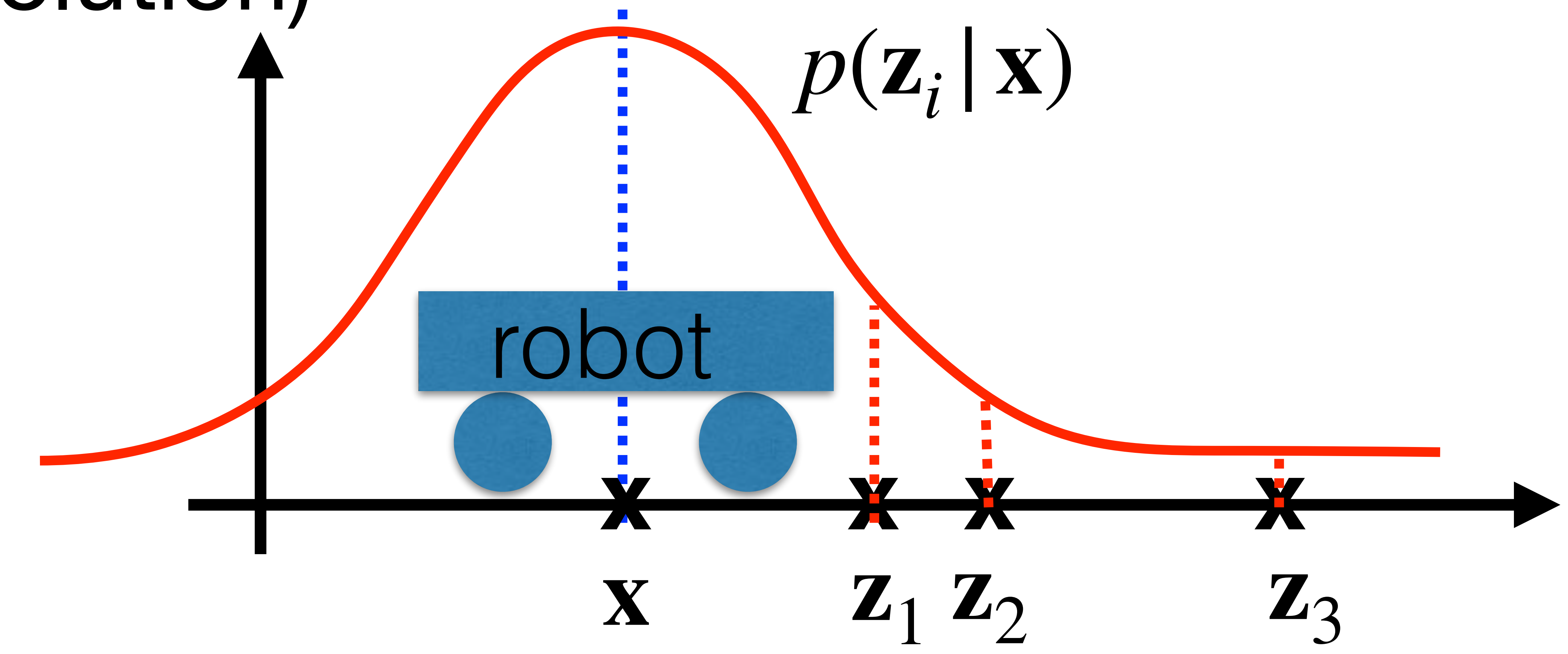
- What is MLE of state $\mathbf{x}$ under the gaussian noise? $p(\mathbf{z}_i | \mathbf{x}) = \mathcal{N}(\mathbf{z}_i;\ \mathbf{x}, 1)$ $\quad \mathbf{x}^\star = ?$
- What is MLE of state $\mathbf{x}$ under the gaussian noise? $p(\mathbf{z}_i | \mathbf{x}) = \mathcal{N}(\mathbf{z}_i;\ \mathbf{x}, 100)$ $\quad \mathbf{x}^\star = ?$
- What is MLE of state $\mathbf{x}$ under the gaussian noise?

  $$p(\mathbf{z}_1 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_1;\ \mathbf{x}, 4)$$
  $$p(\mathbf{z}_2 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_2;\ \mathbf{x}, 1)$$
  $$p(\mathbf{z}_3 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_3;\ \mathbf{x}, 1)$$

  $\mathbf{x}^\star = ?$

- How can you get MLE of the state under the heavy-tail-gaussian noise?

RANSAC (solution)



- Assume that
  - no motion model is applied,
  - no prior probability distribution
- GPS position is measured three-times:

  $\mathbf{z}_1 = 2 \quad \mathbf{z}_2 = 3 \quad \mathbf{z}_3 = 7$

- What is MLE of state $\mathbf{x}$ under the gaussian noise? $p(\mathbf{z}_i|\mathbf{x}) = \mathcal{N}(\mathbf{z}_i;\ \mathbf{x}, 1)$ $\quad \mathbf{x}^\star = 4$
- What is MLE of state $\mathbf{x}$ under the gaussian noise? $p(\mathbf{z}_i|\mathbf{x}) = \mathcal{N}(\mathbf{z}_i;\ \mathbf{x}, 100)$ $\quad \mathbf{x}^\star = 4$
- What is MLE of state $\mathbf{x}$ under the gaussian noise?

$$p(\mathbf{z}_1|\mathbf{x}) = \mathcal{N}(\mathbf{z}_1;\ \mathbf{x}, 4)$$
$$p(\mathbf{z}_2|\mathbf{x}) = \mathcal{N}(\mathbf{z}_2;\ \mathbf{x}, 1)$$
$$p(\mathbf{z}_3|\mathbf{x}) = \mathcal{N}(\mathbf{z}_3;\ \mathbf{x}, 1)$$

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = \arg\max_{\mathbf{x}} \left(\prod_i p(\mathbf{z}_i|\mathbf{x}) \right)$$

$$= \arg\min_{\mathbf{x}} \sum_i 1/\sigma_i^2 \cdot (\mathbf{x} - \mathbf{z}_i)^2 = \frac{\sum_i \mathbf{z}_i/\sigma_i^2}{\sum_i 1/\sigma_i^2} = \frac{0.25 \cdot 2 + 1 \cdot 3 + 1 \cdot 7}{2.25} = 4.66$$

- How can you get MLE of the state under the heavy-tail-gaussian noise?

  RANSAC (result depends on tolerance margin and implementation) $\mathbf{x}^\star \in <2, 3>$

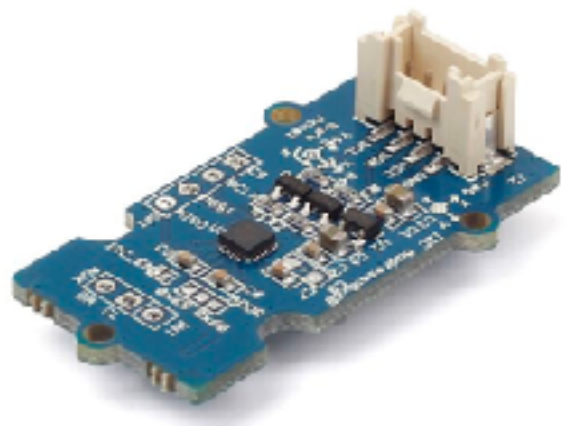
You should be able to use all measurement and transition models in all discussed concepts (EKF, PF, FG,…) including their first order approximations

Examples of measurement probabilities

$$p\Big(\underbrace{\begin{bmatrix} z_t^{\mathrm{GPS},x} \\ z_t^{\mathrm{GPS},y} \end{bmatrix}}_{\mathbf{z}_t^{\mathrm{GPS}}} \mid \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}\Big) = \mathcal{N}\Big(\mathbf{z}_t^{\mathrm{GPS}};\ \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{h^{\mathrm{GPS}}(\mathbf{x}_t)},\ \mathbf{Q}_t^{\mathrm{GPS}}\Big)$$

GPS

$$p\Big(\underbrace{\begin{bmatrix} z_t^{x} \\ z_t^{y} \\ z_t^{\theta} \end{bmatrix}}_{\mathbf{z}_t^{\mathrm{odom}}} \mid \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}, \underbrace{\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix}}_{\mathbf{x}_{t+1}}\Big) = \mathcal{N}\Big(\mathbf{z}_t^{\mathrm{odom}};\ \underbrace{\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x_t})}_{h^{\mathrm{odom}}(\mathbf{x}_t)},\ \mathbf{Q}_t^{\mathrm{odom}}\Big)$$



IMU

$$p\Big(\underbrace{\begin{bmatrix} z_t^{x} \\ z_t^{y} \\ z_t^{\theta} \end{bmatrix}}_{\mathbf{z}_t^{\mathbf{m}}} \mid \underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t}, \underbrace{\begin{bmatrix} m^{x} \\ m^{y} \\ m^{\theta} \end{bmatrix}}_{\mathbf{m}}\Big) = \mathcal{N}\Big(\mathbf{z}_t^{\mathbf{m}};\ \underbrace{\mathrm{w2r}(\mathbf{m}, \mathbf{x_t})}_{h^{\mathbf{m}}(\mathbf{x}_t)},\ \mathbf{Q}_t^{\mathbf{m}}\Big)$$

Marker detector

Examples of state-transition probabilities

Differential-drive model

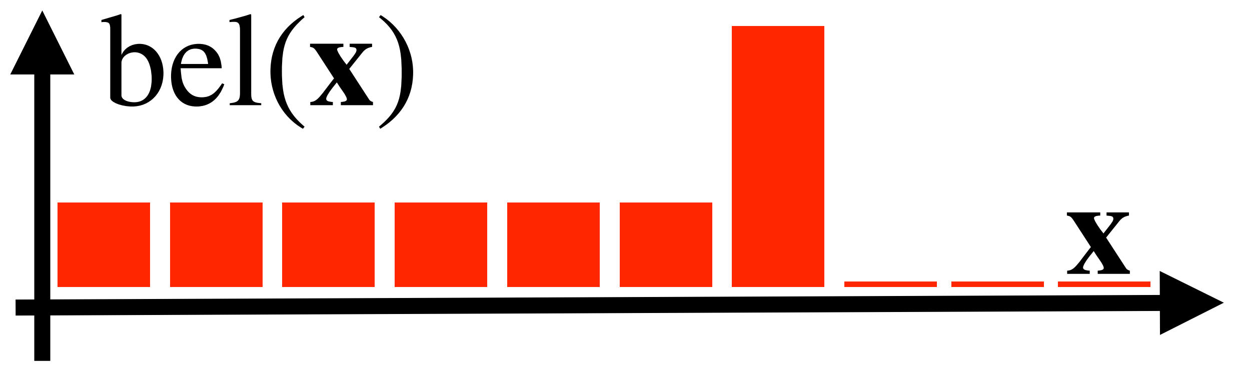
$$p\left(\underbrace{\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}}_{\mathbf{x}_t} \middle| \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\mathbf{u}_t}\right) = \mathcal{N}\left(\mathbf{x}_t; \underbrace{\begin{bmatrix} x_{t-1} + \frac{v_t}{\omega_t}\big(+ \sin(\theta_{t-1} + \omega_t \Delta t) - \sin(\theta_{t-1})\big) \\ y_{t-1} + \frac{v_t}{\omega_t}\big(- \cos(\theta_{t-1} + \omega_t \Delta t) + \cos(\theta_{t-1})\big) \\ \theta_{t-1} + \omega_t \Delta t \end{bmatrix}}_{g(\mathbf{x}_{t-1}, \mathbf{u}_t)}, \ \mathbf{R}_t\right)$$

Balistic trajectory

$$p\left(\underbrace{\begin{bmatrix} x_t \\ y_t \end{bmatrix}}_{\mathbf{x}_t} \middle| \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix}}_{\mathbf{x}_{t-1}}, \underbrace{\begin{bmatrix} v_t \\ \omega_t \end{bmatrix}}_{\mathbf{u}_t}\right) = \mathcal{N}\left(\mathbf{x}_t; \underbrace{\begin{bmatrix} x_{t-1} + v_t \Delta t \cos(\omega_t) \\ y_{t-1} + v_t \Delta t \sin(\omega_t) - \frac{1}{2} g \Delta t^2 \end{bmatrix}}_{g(\mathbf{x}_{t-1}, \mathbf{u}_t)}, \ \mathbf{R}_t\right)$$

You should also understand reasoning behind this table

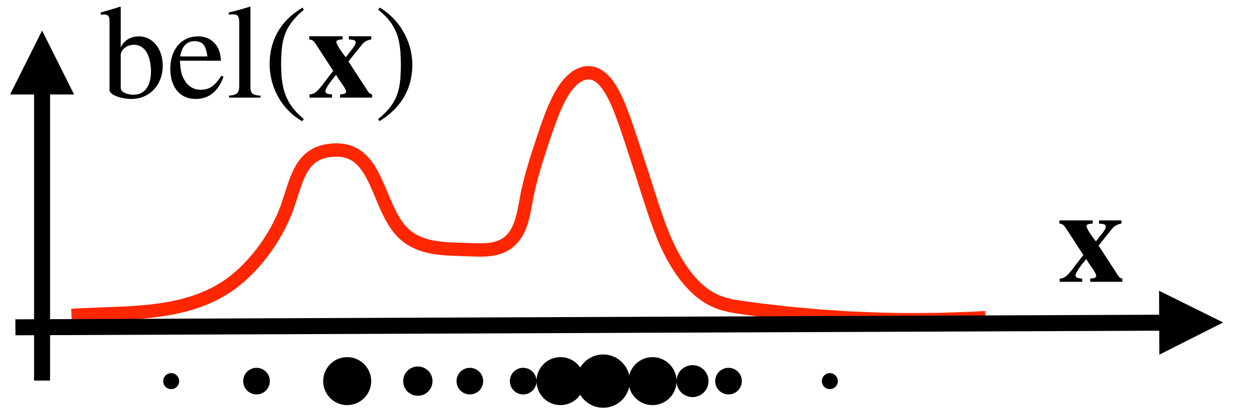
Bayes filter



Drawbacks
- course of dimensionality
- spatial discretization

Advantages
- represents arbitrary prob. distribution

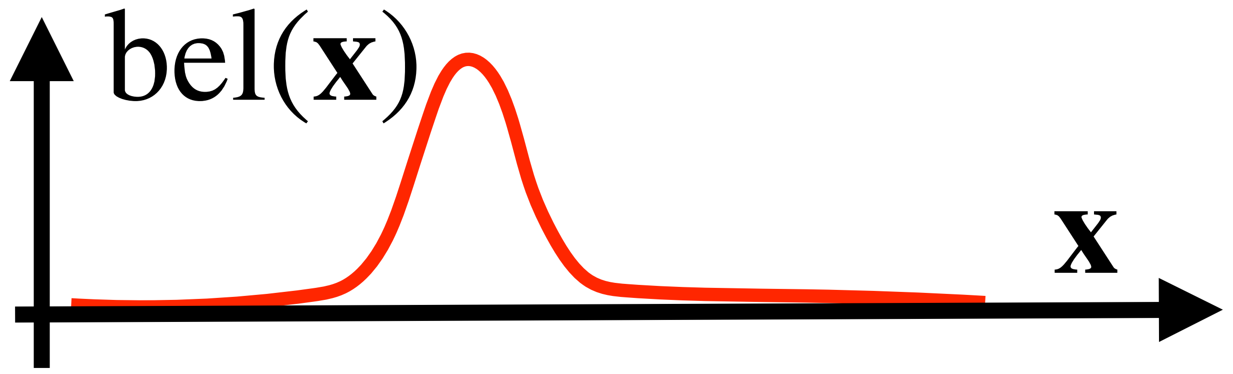
Particle filter



Drawbacks
- course of dimensionality
- partical quantization

Advantages
- represents arbitrary prob. distribution

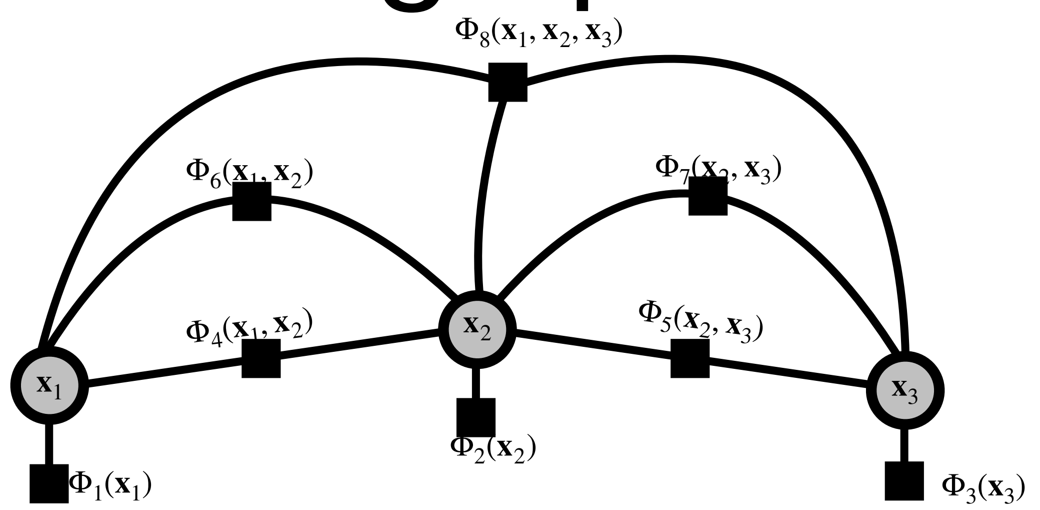
Kalman filter



Drawbacks
- represent only gaussians
- suffers from linearization

Advantages
- nicely scales with higher dimensions

Factorgraph



Drawbacks
- represents gaussians
- grows to infinity

Advantages
- does not suffer from linearizations
- allows for arbitrary conditional independences