

Particle filter

Karel Zimmermann

Drawbacks

Advantages

Bayes filter

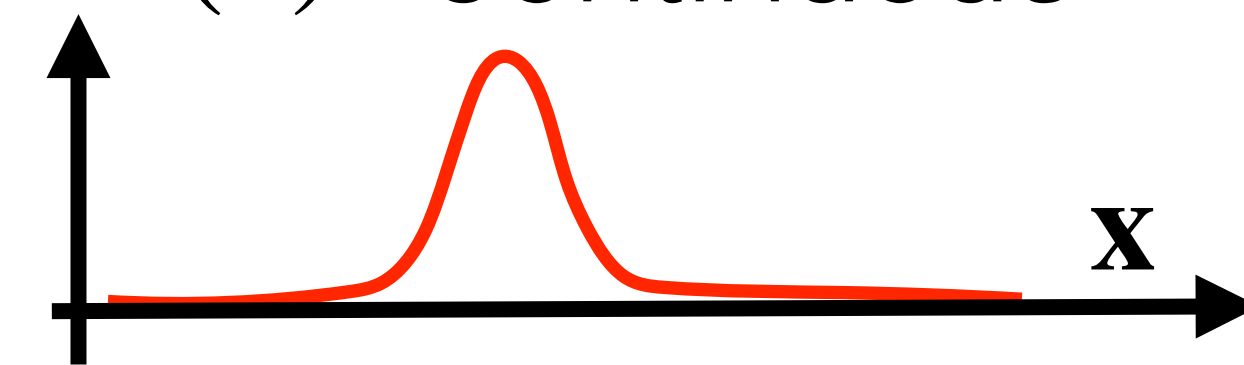


- course of dimensionality
- spatial discretization

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

- represents arbitrary prob. distribution

Kalman filter

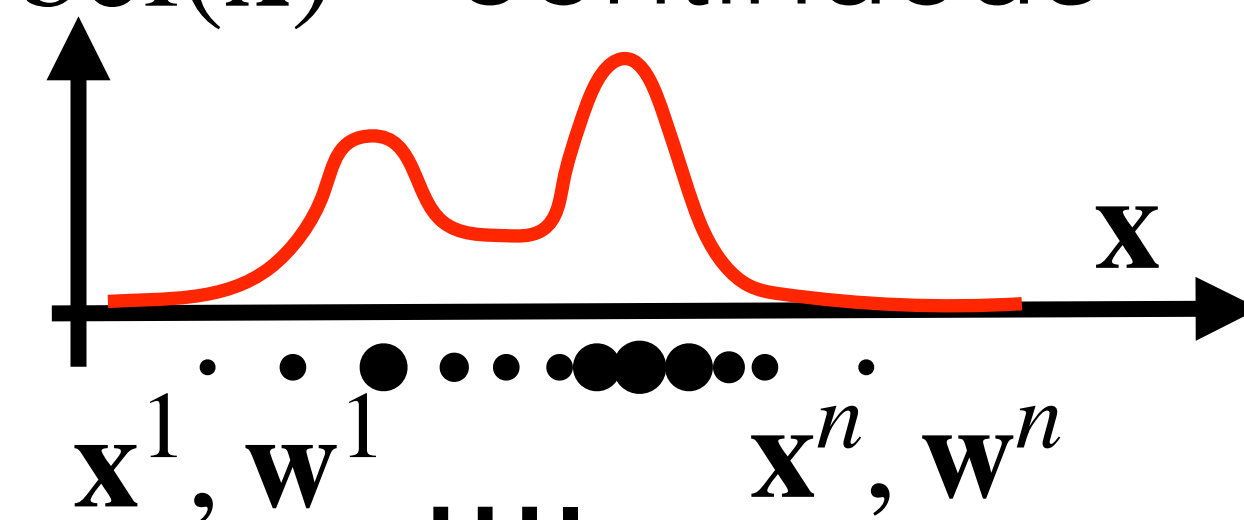


- represent only gaussians
- suffers from linearization

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

- nicely scales with higher dimensions

Partical filter



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

Dirac impulse function

Particle filter

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

Kidnapped robot problem

Particles = hypothesis about the current state



Particle filter

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

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

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



Particle filter

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

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

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

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

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

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

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



Particle filter

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

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

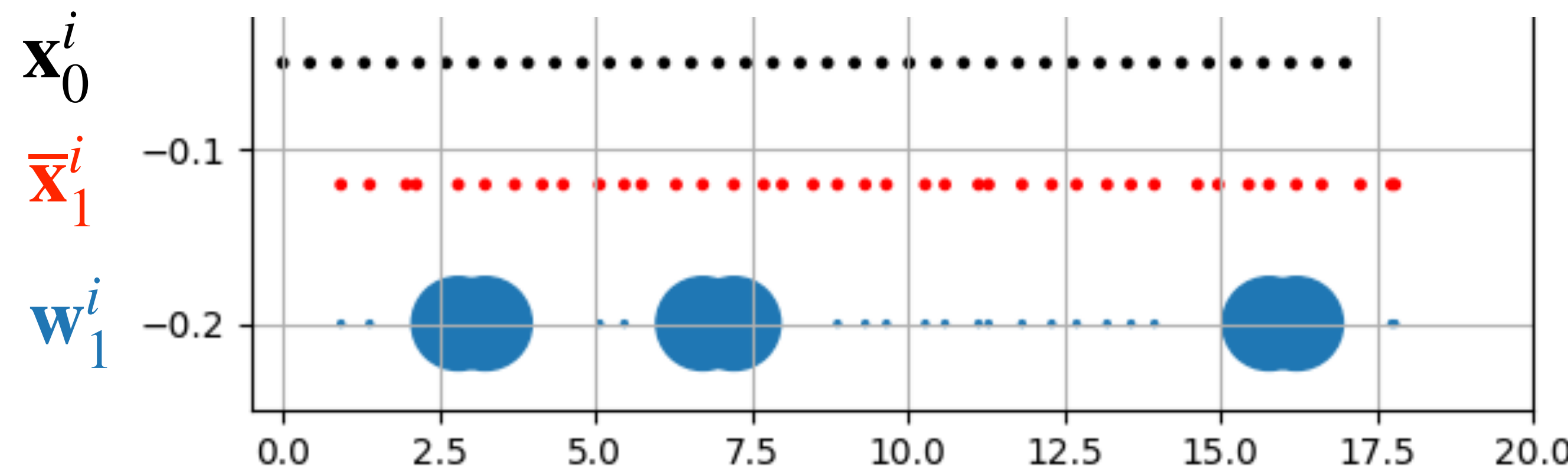
$$\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$

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

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

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

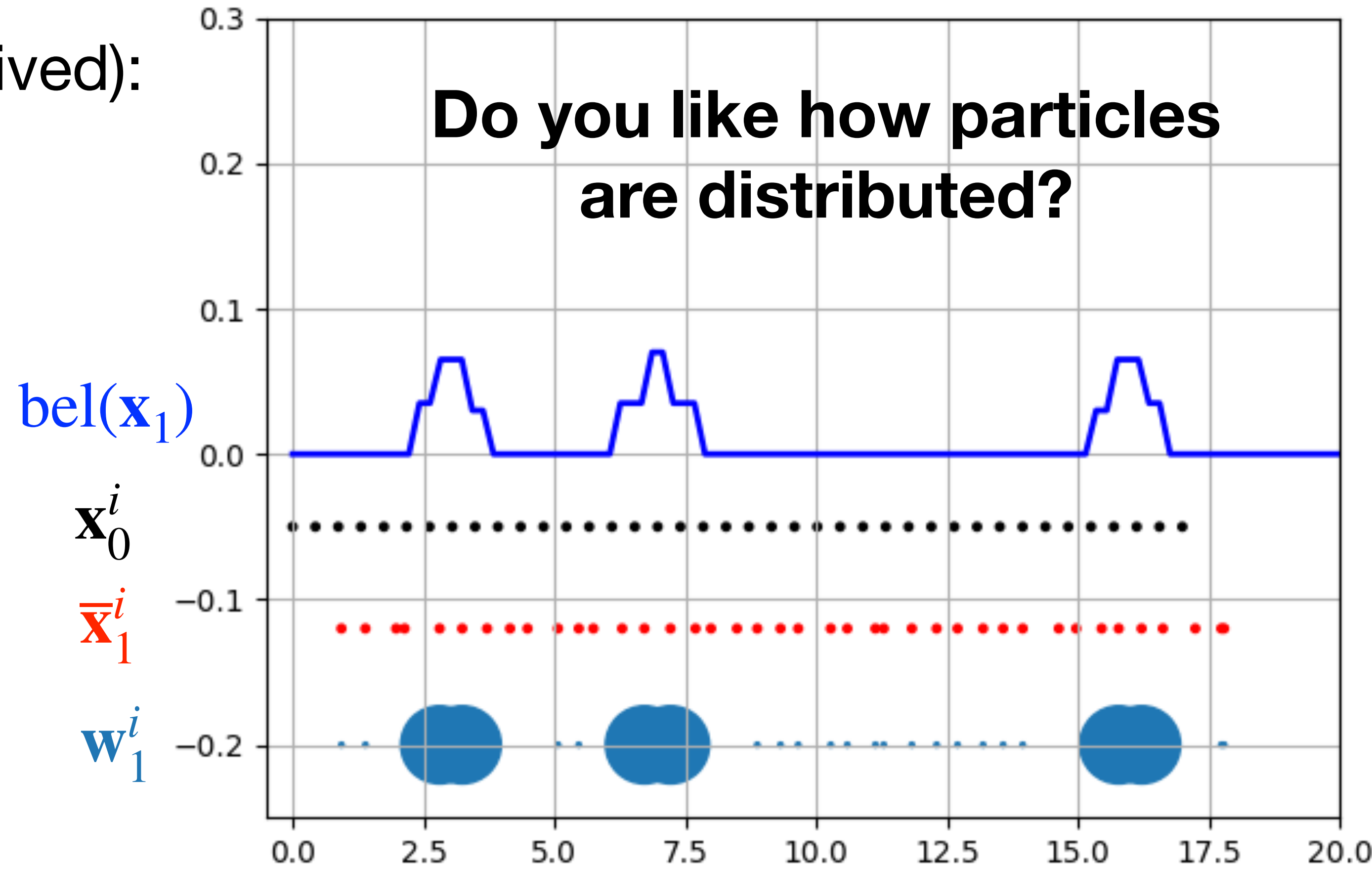
$$p(\mathbf{z}_t = 1 | \mathbf{x}_t)$$



Particle filter

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

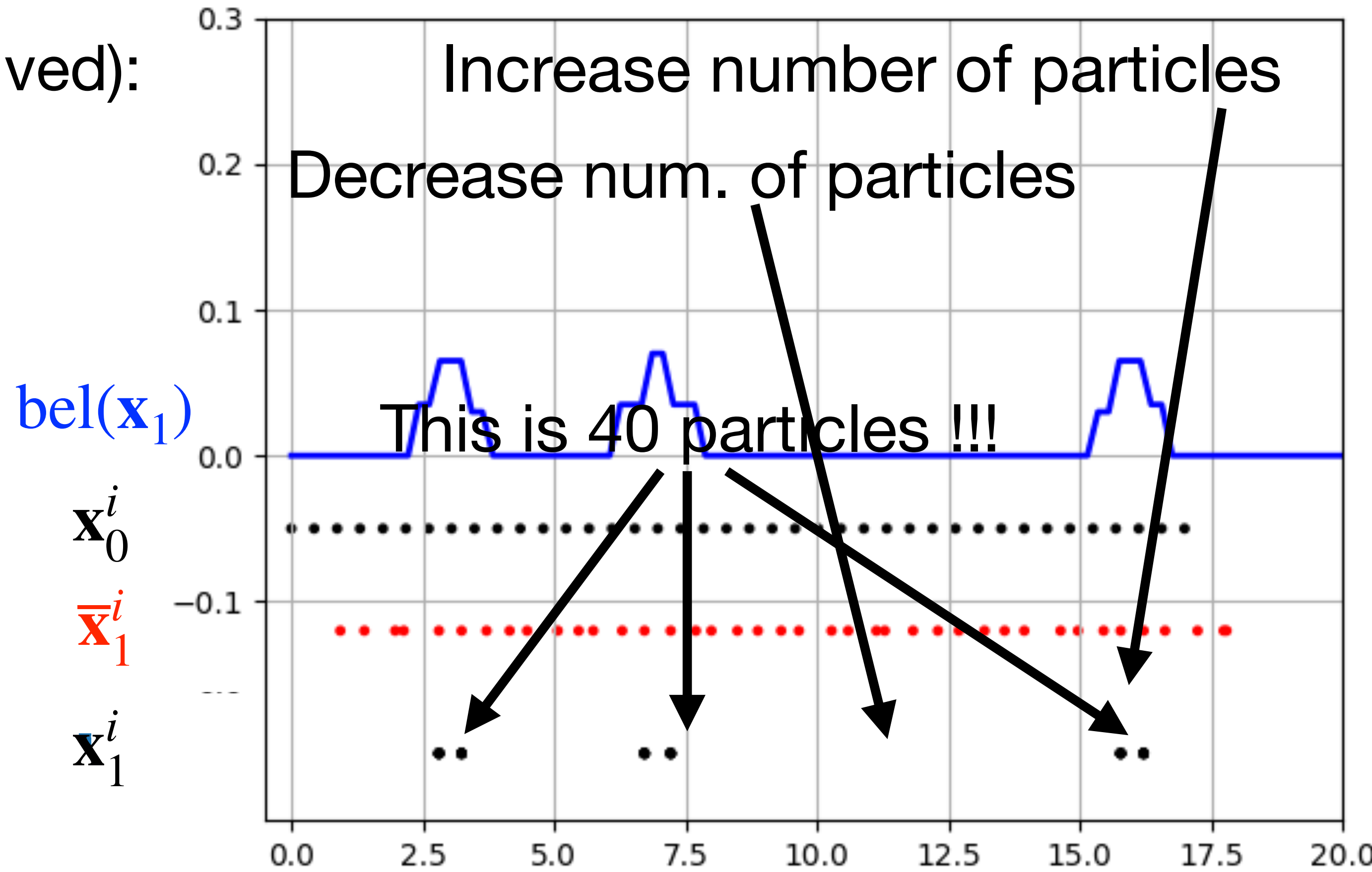
$$p(\mathbf{z}_t = 1 | \mathbf{x}_t)$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

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



Particle filter

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

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

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

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

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

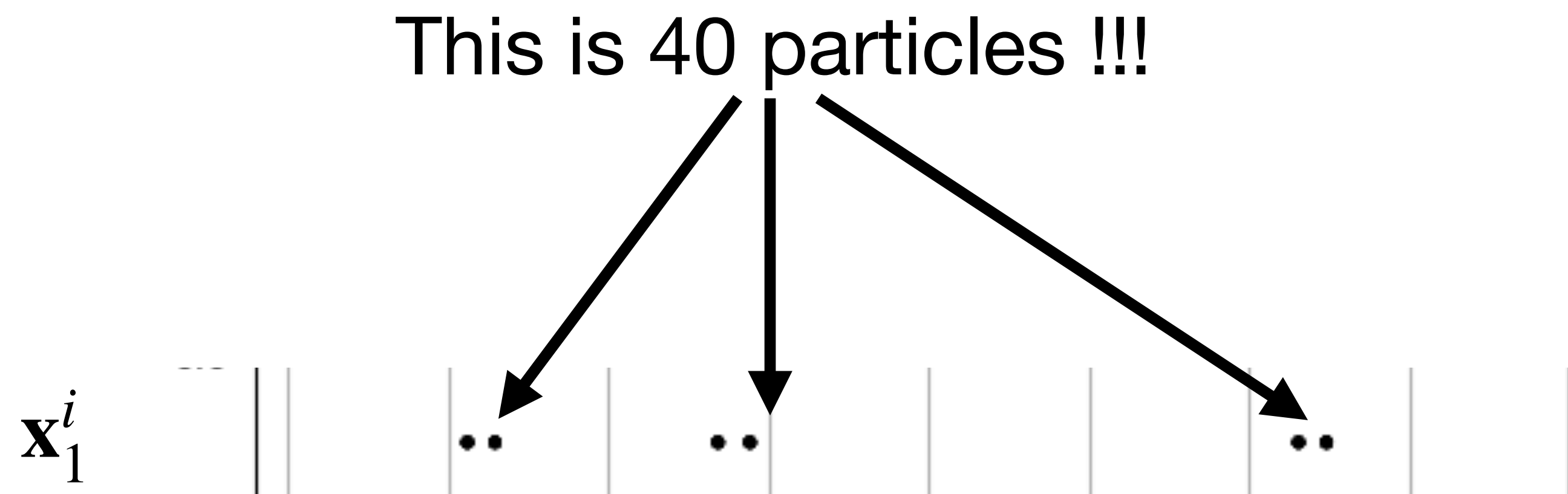
$$\mathbf{w}_t^i = p(\mathbf{z}_t | \bar{\mathbf{x}}_t^i)$$

4. Resample

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

5. Repeat from 2:

$$t = t + 1$$



Particle filter

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

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

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

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

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

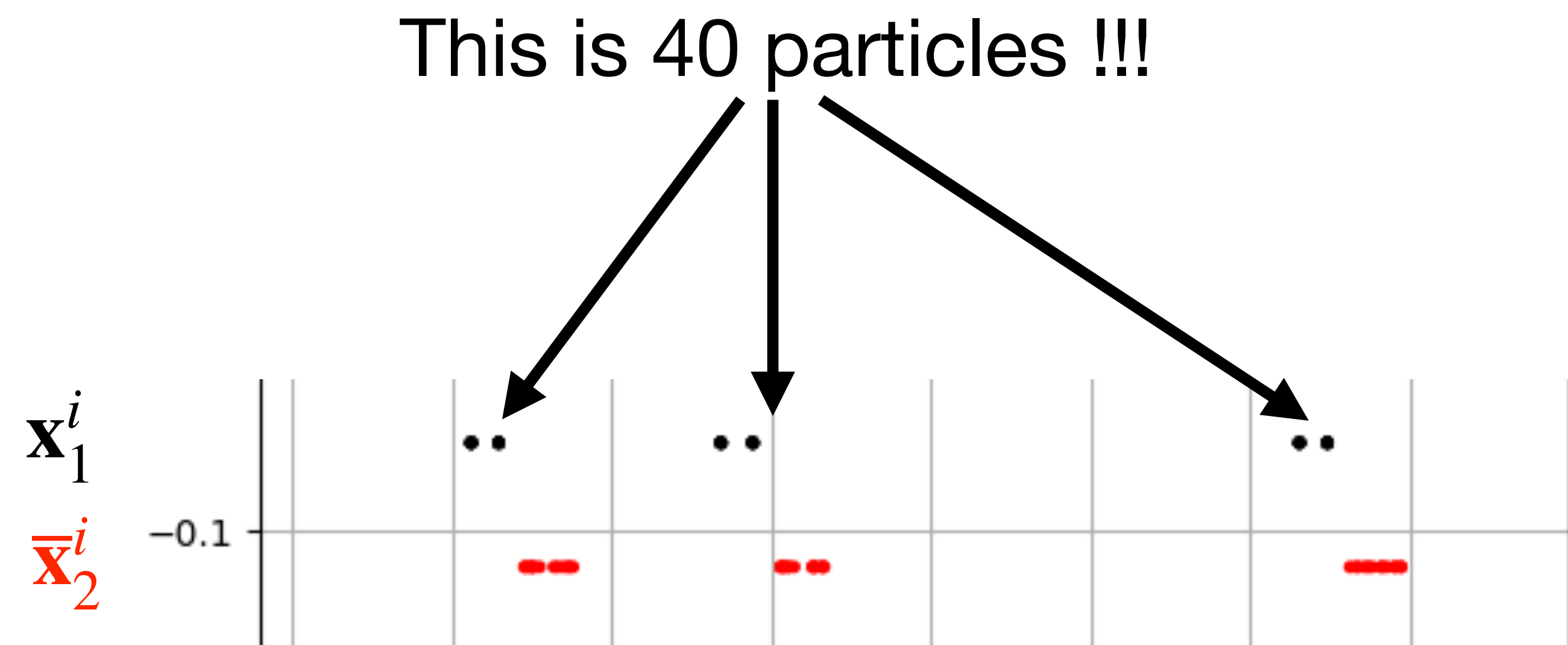
$$\mathbf{w}_t^i = p(\mathbf{z}_t | \bar{\mathbf{x}}_t^i)$$

4. Resample

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

5. Repeat from 2:

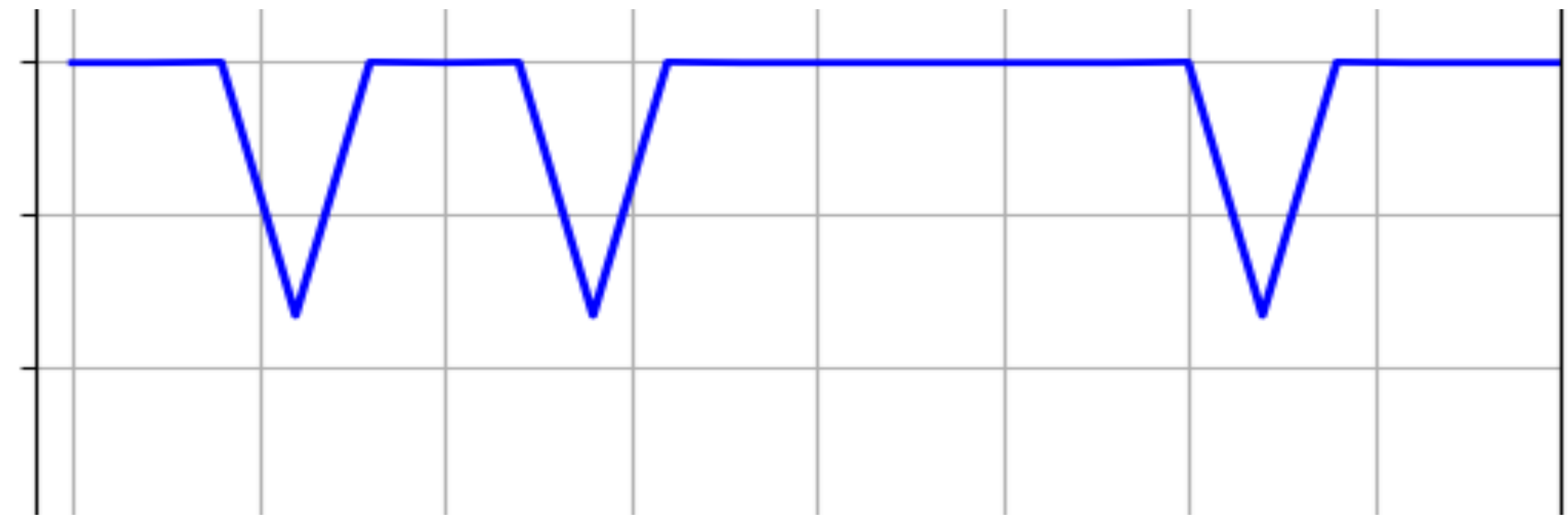
$$t = t + 1$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

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

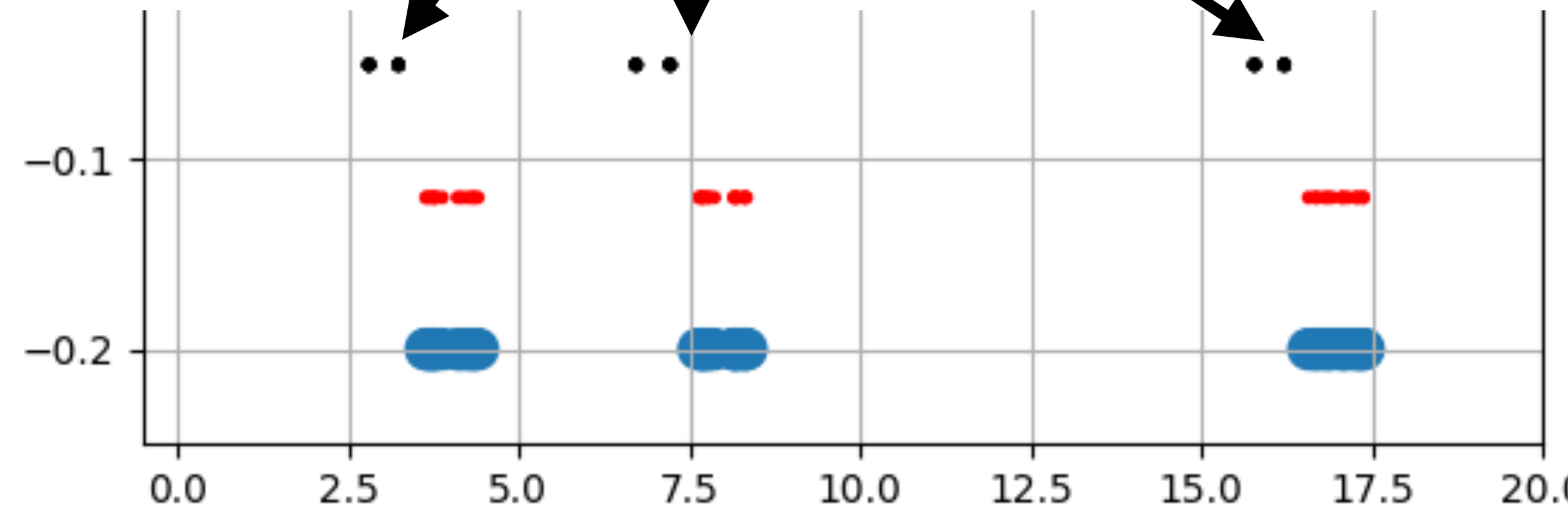


This is 40 particles !!!

\mathbf{x}_1^i

$\bar{\mathbf{x}}_2^i$

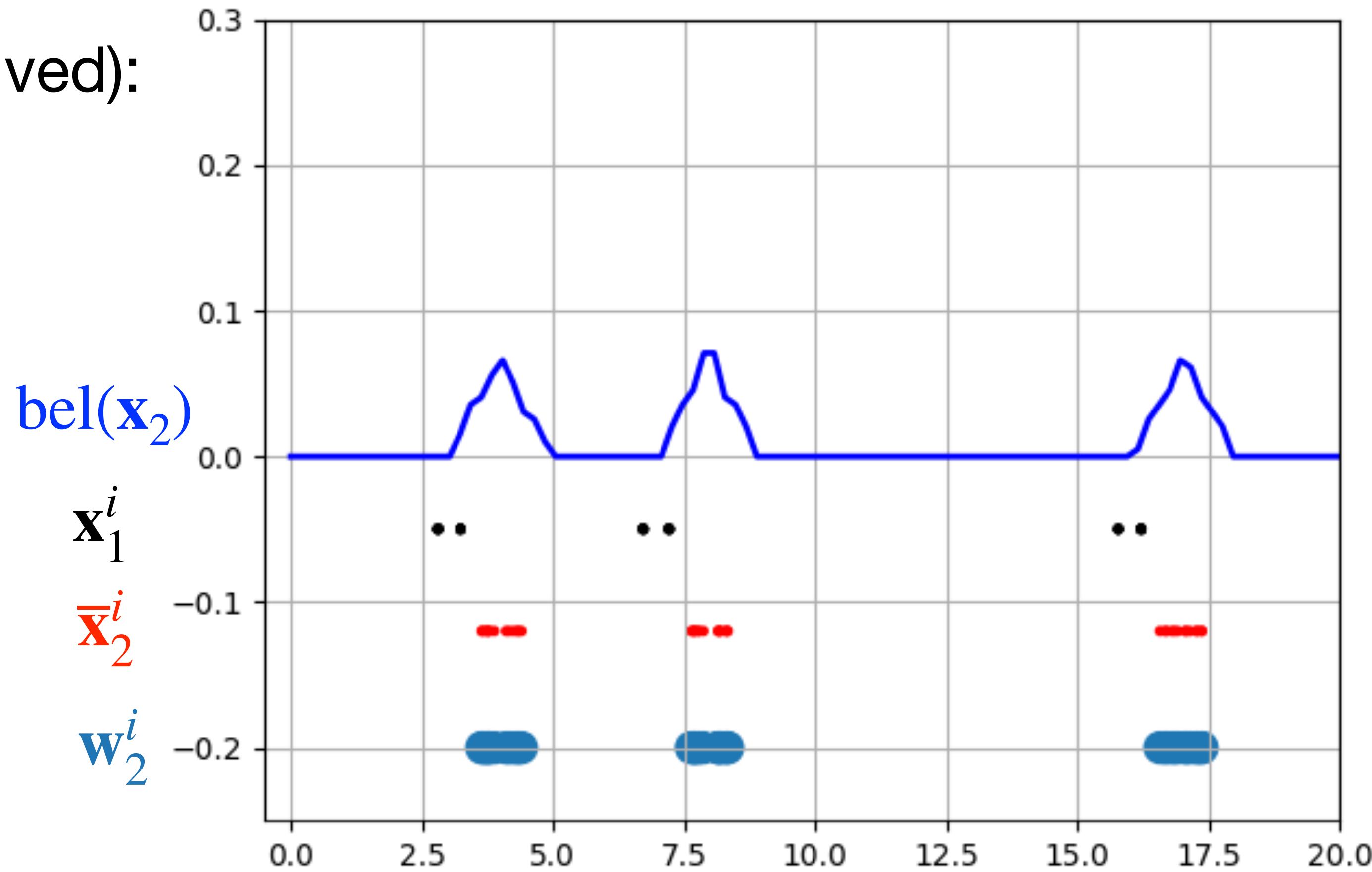
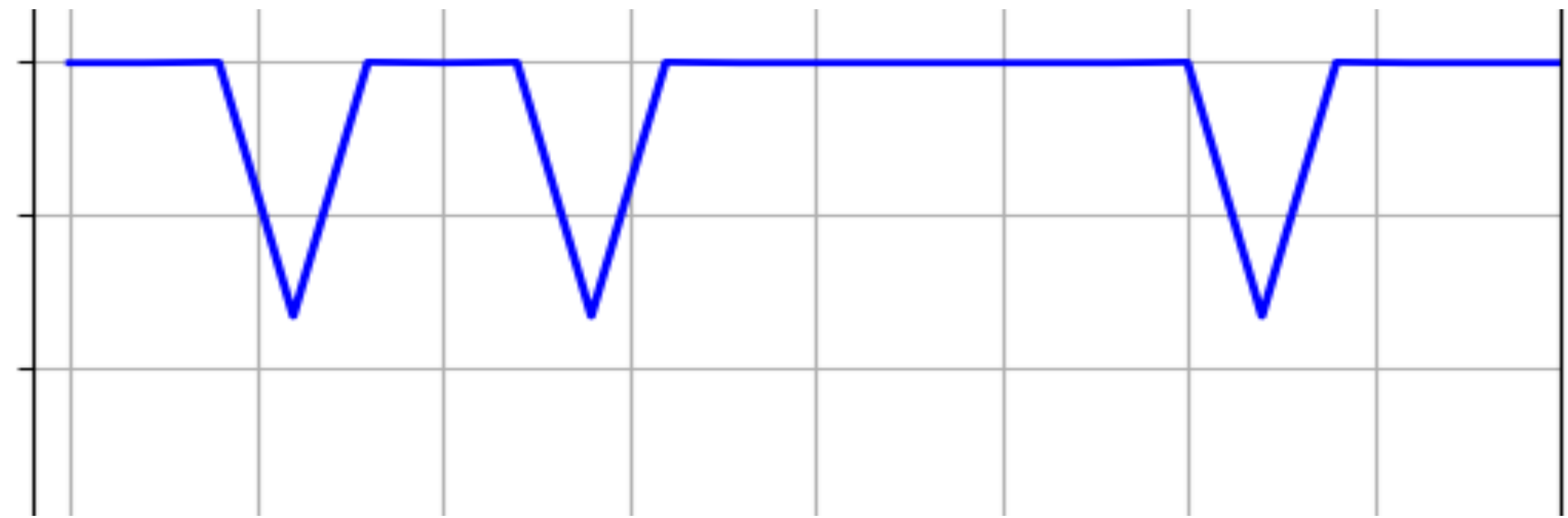
\mathbf{w}_2^i



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

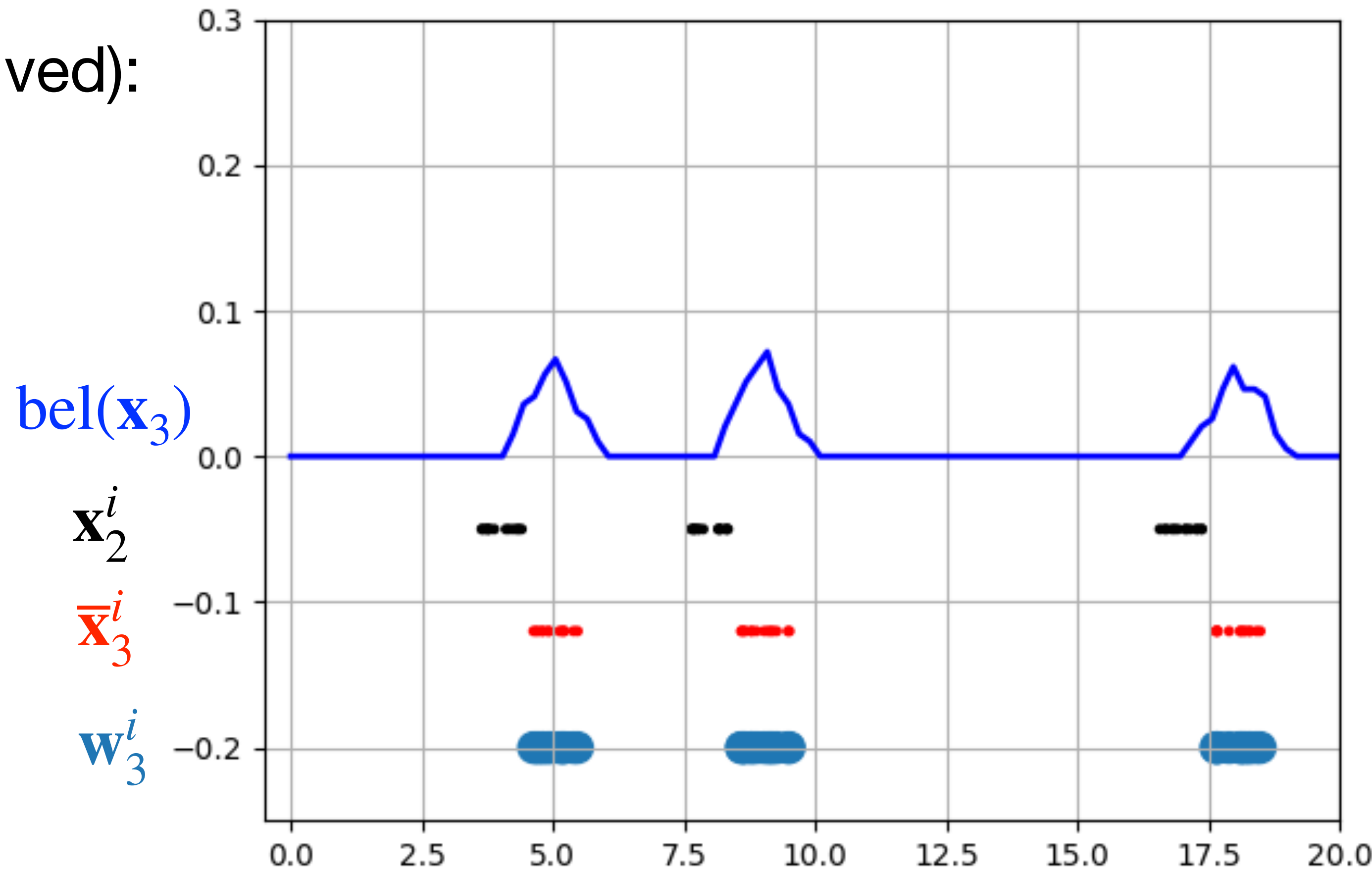
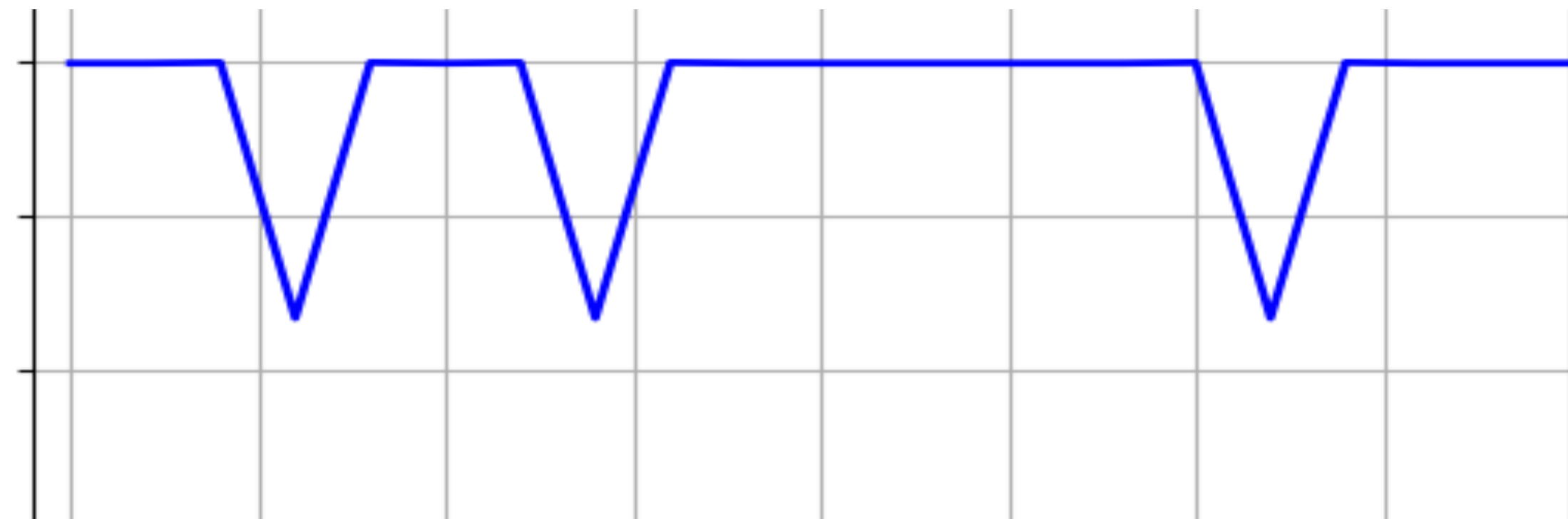
$$p(\mathbf{z}_t = 0 | \mathbf{x}_t)$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
$$\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

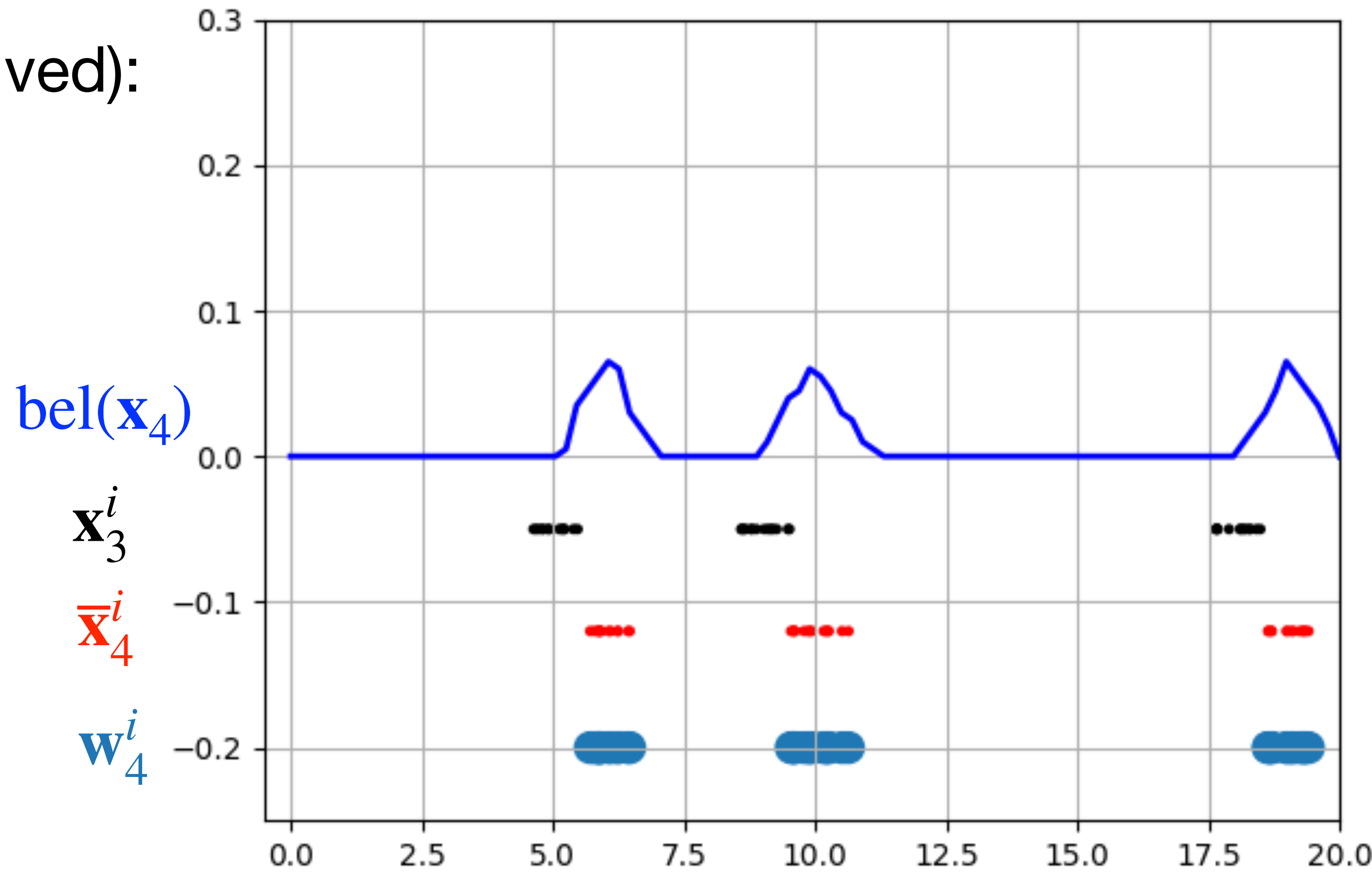
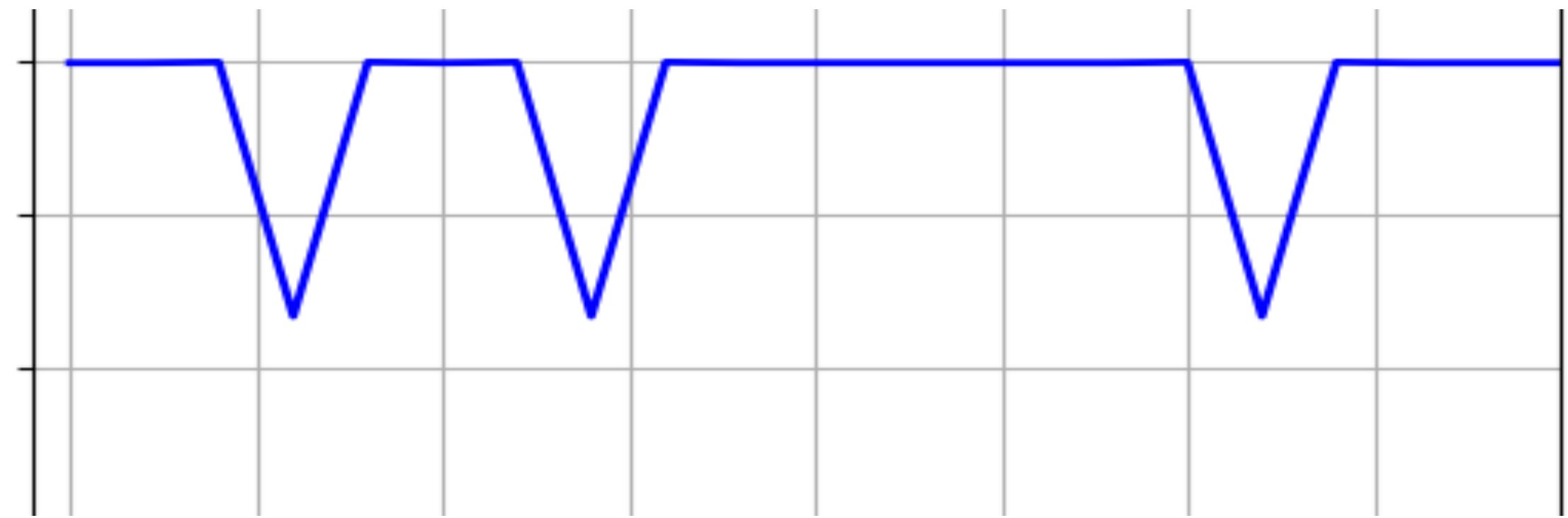
$$p(\mathbf{z}_t = 0 | \mathbf{x}_t)$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

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



Particle filter

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

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i

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

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

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

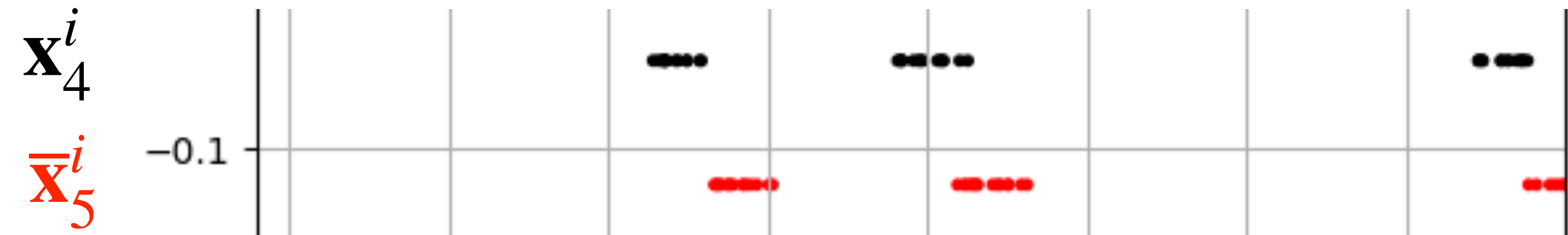
$$\mathbf{w}_t^i = p(\mathbf{z}_t | \bar{\mathbf{x}}_t^i)$$

4. Resample

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

5. Repeat from 2:

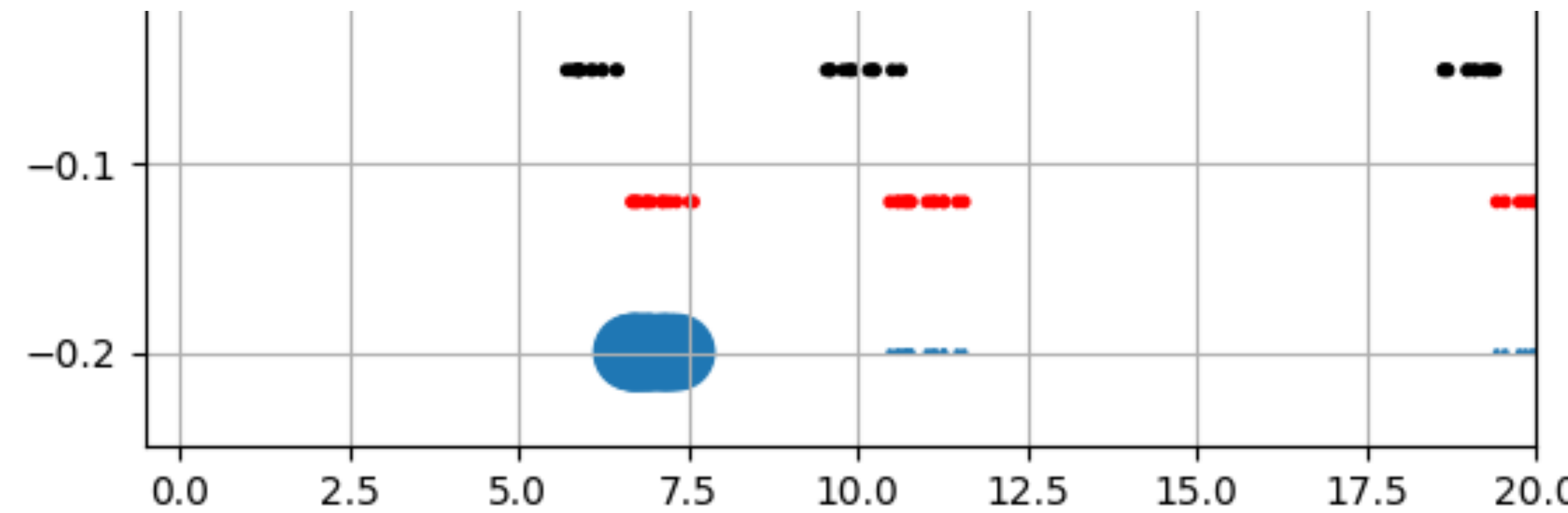
$$t = t + 1$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

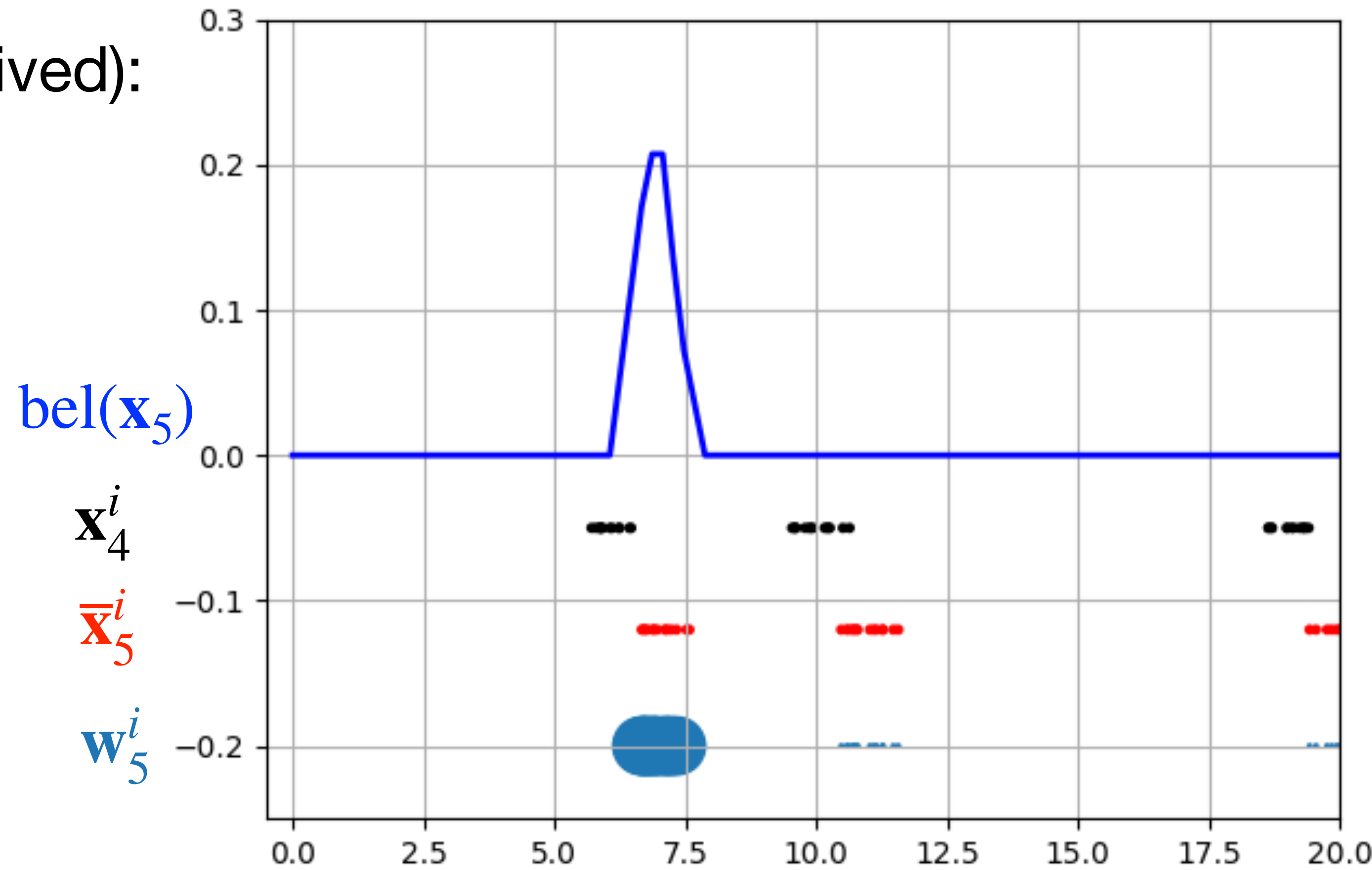
$$p(\mathbf{z}_t = 1 | \mathbf{x}_t)$$

 \mathbf{x}_4^i $\bar{\mathbf{x}}_5^i$ \mathbf{w}_5^i 

Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

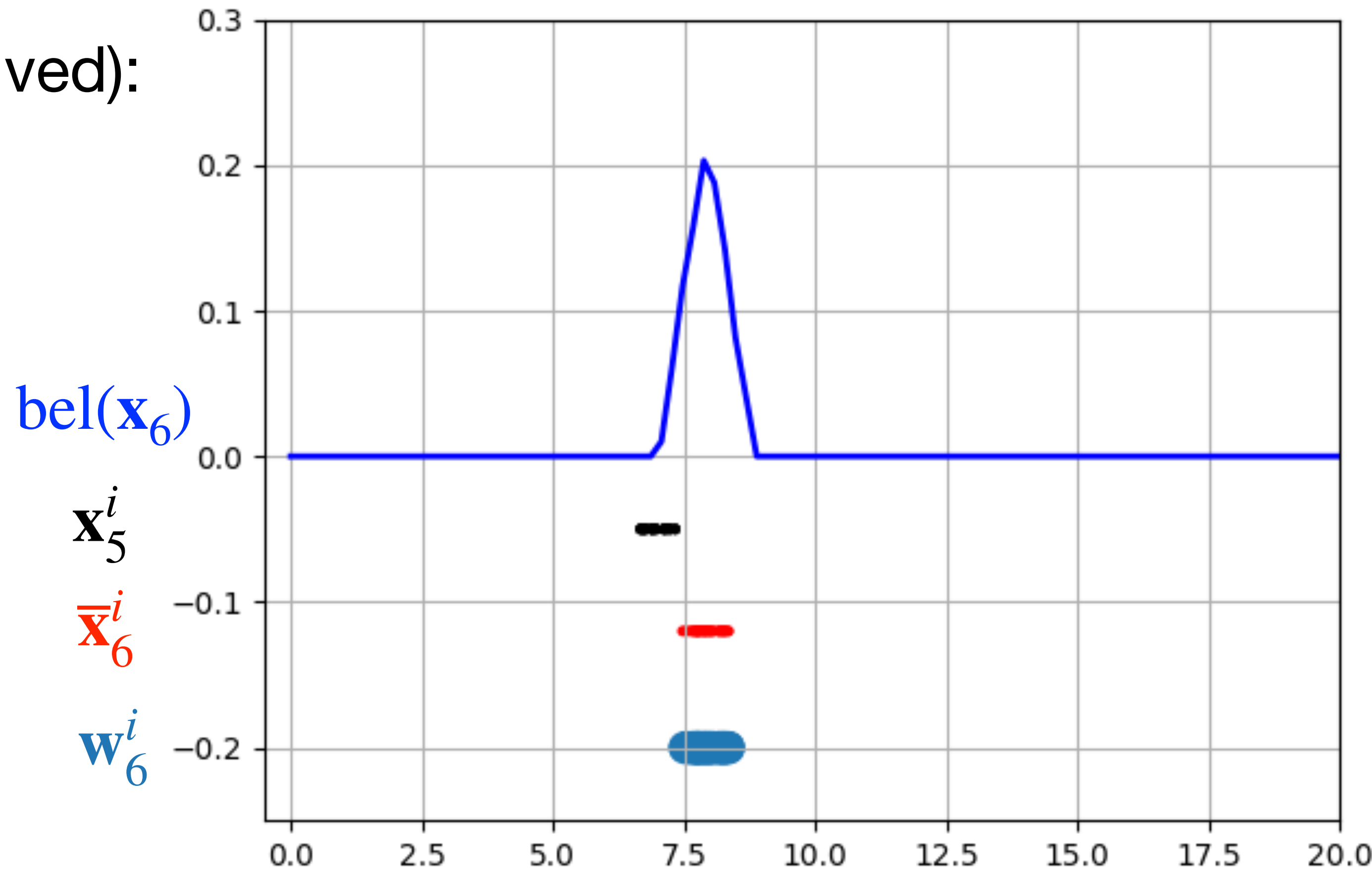
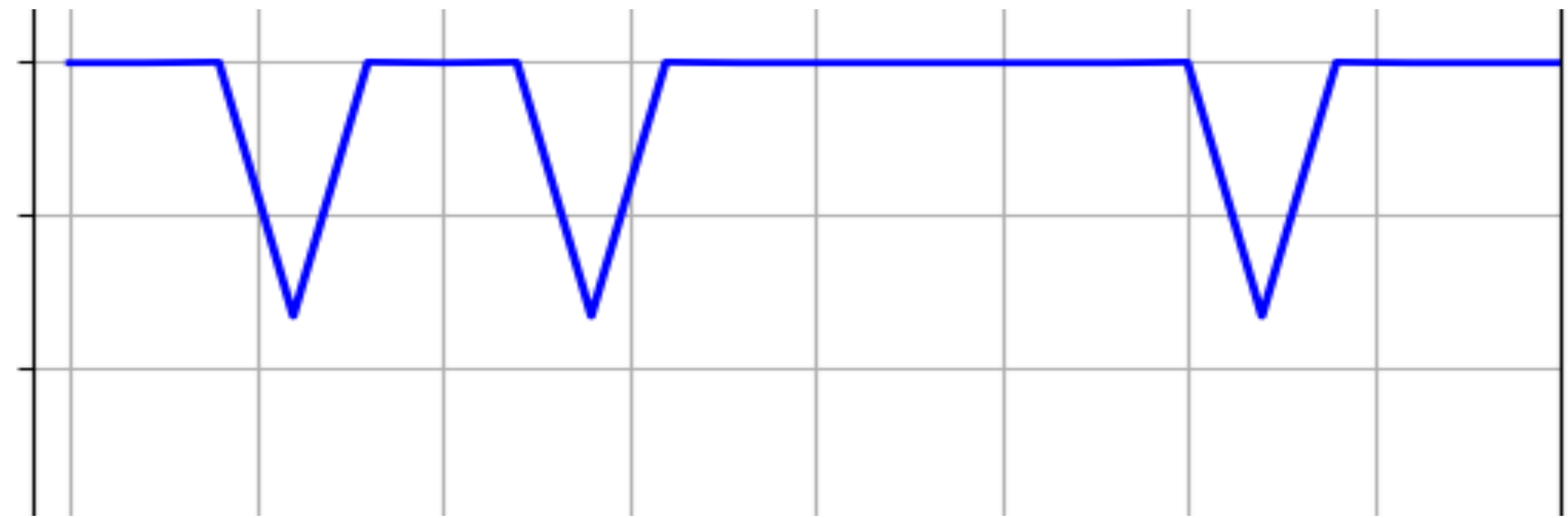
$$p(\mathbf{z}_t = 1 | \mathbf{x}_t)$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
 $\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

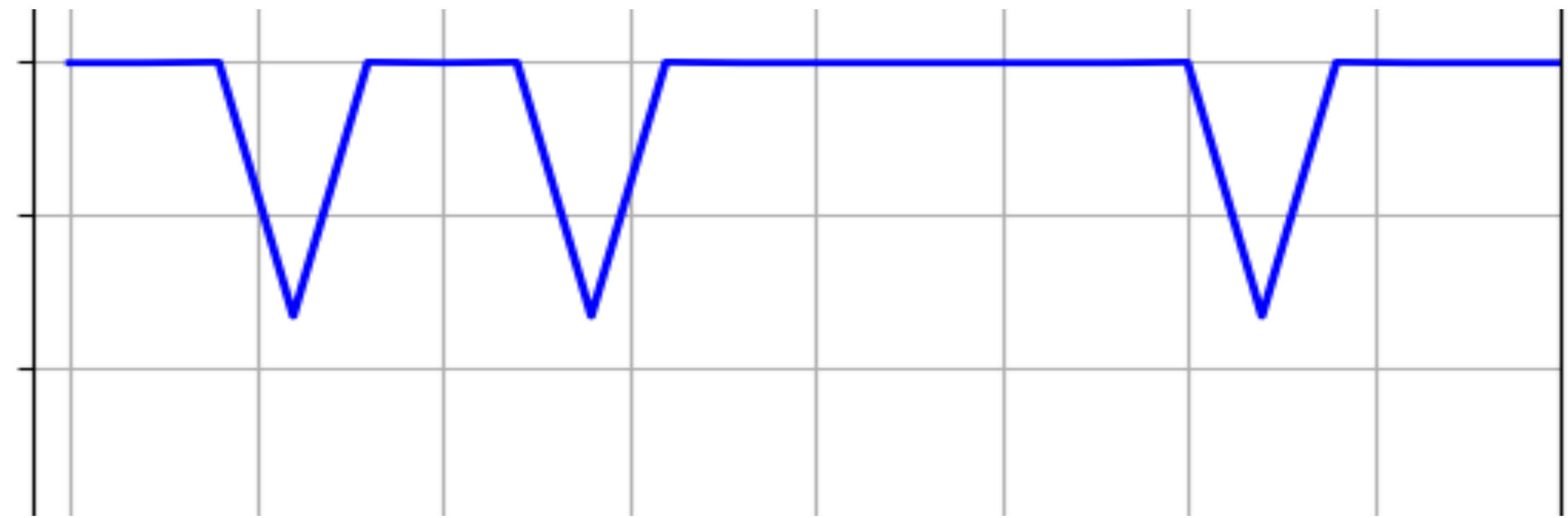
$$p(\mathbf{z}_t = 0 | \mathbf{x}_t)$$



Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
$$\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

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

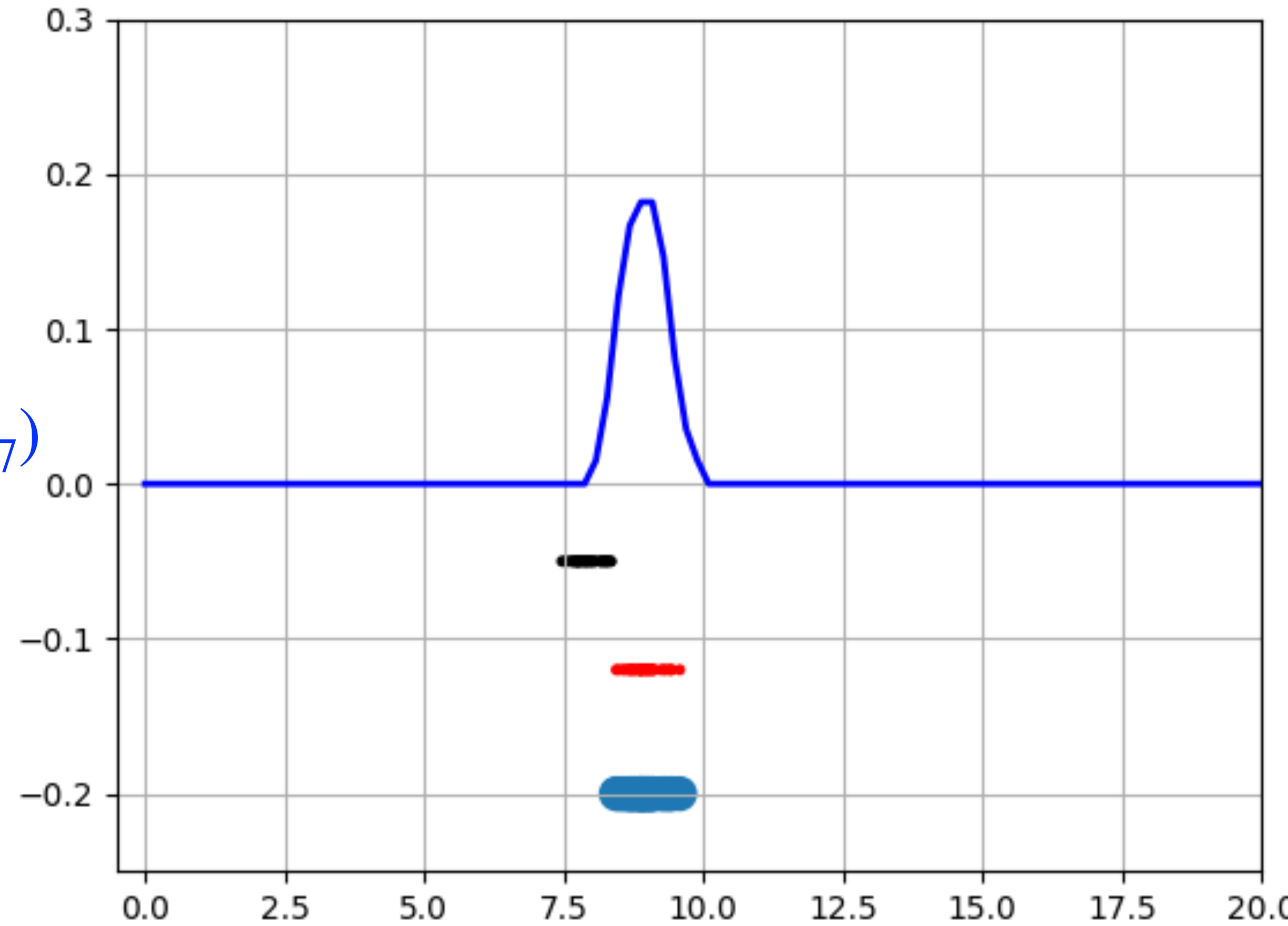


$$\text{bel}(\mathbf{x}_7)$$

$$\mathbf{x}_6^i$$

$$\bar{\mathbf{x}}_7^i$$

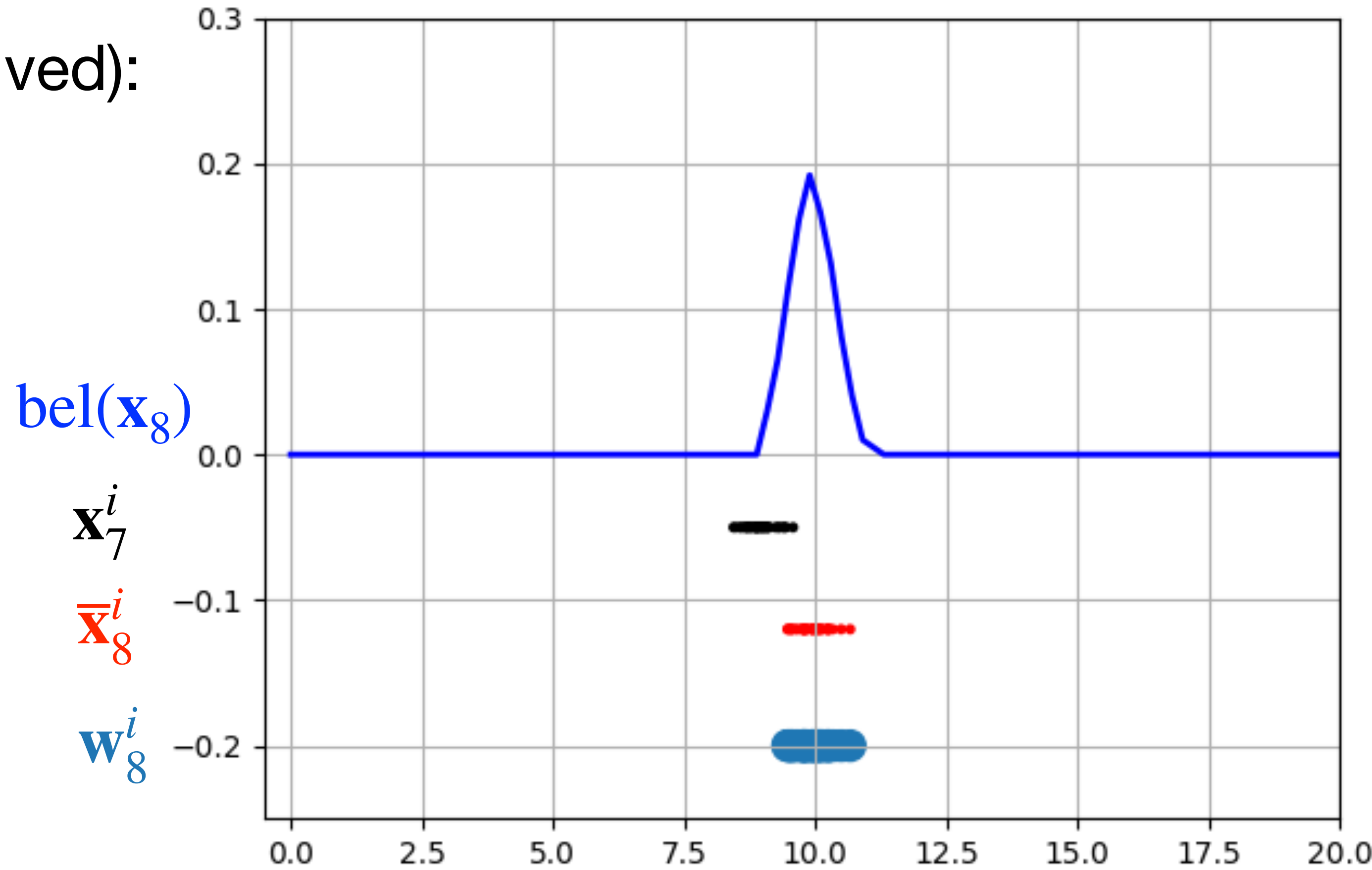
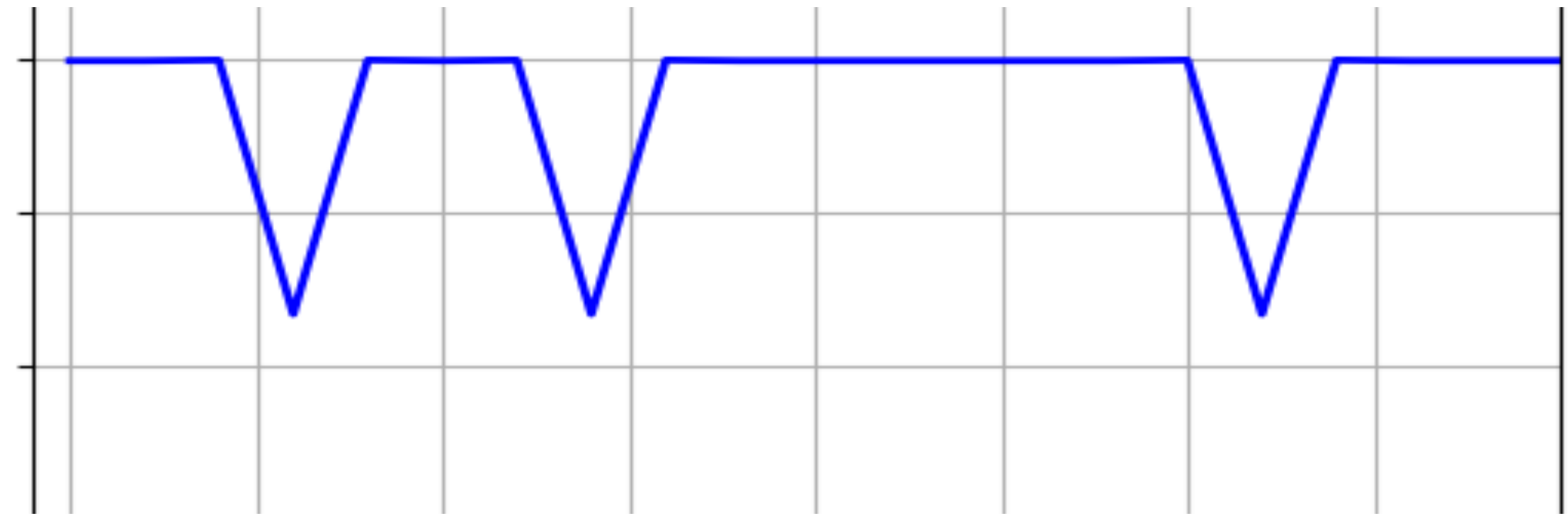
$$\mathbf{w}_7^i$$



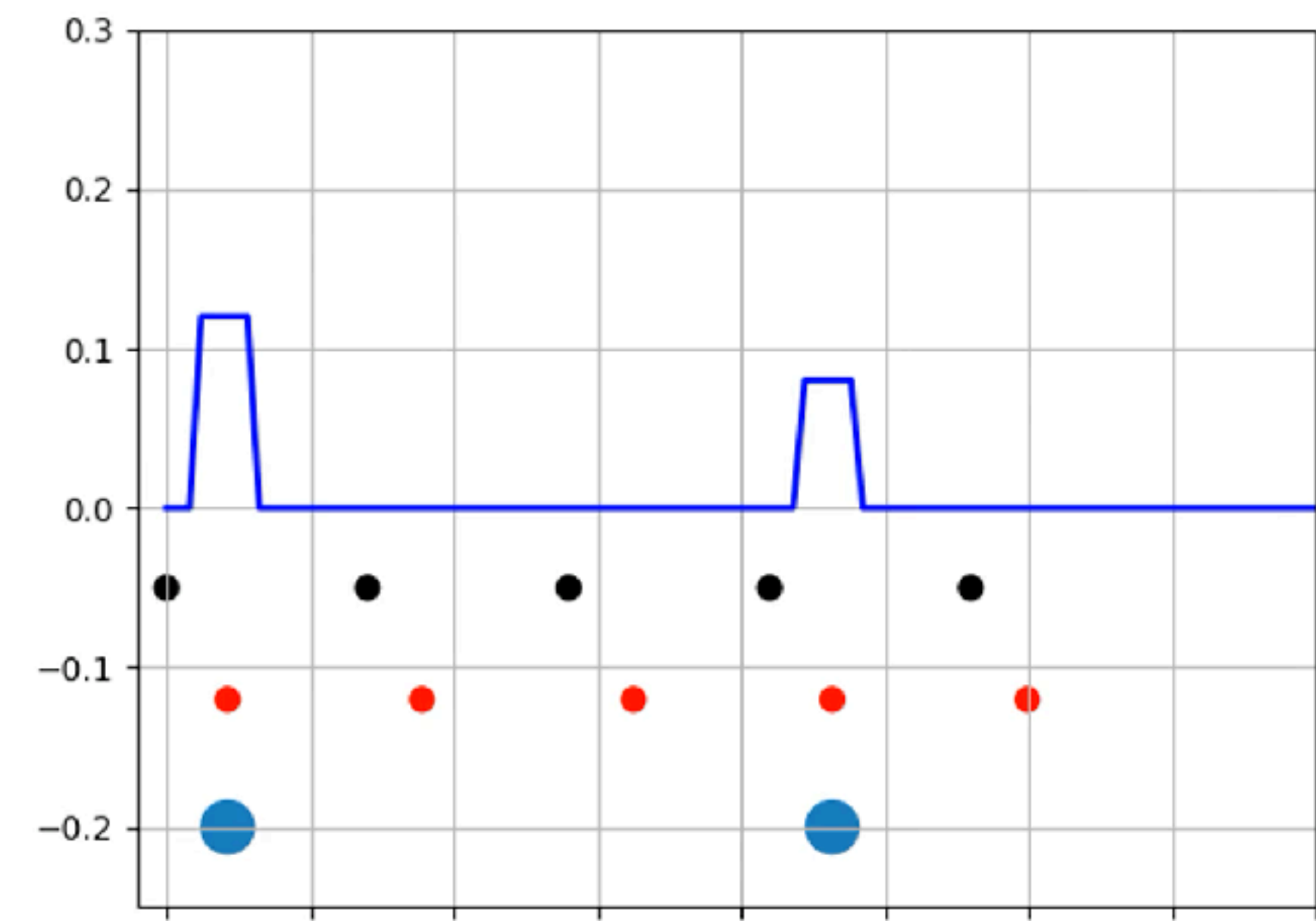
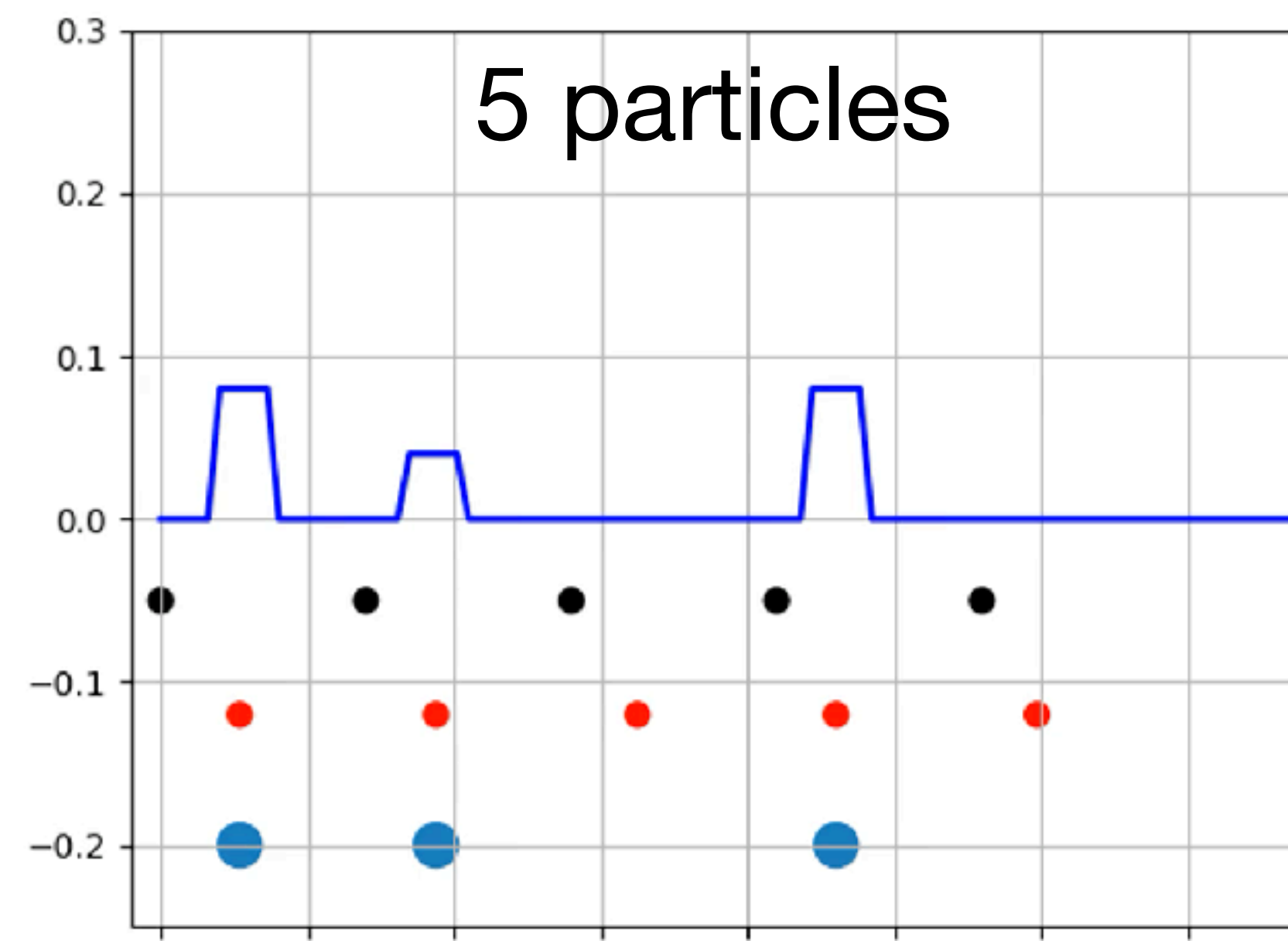
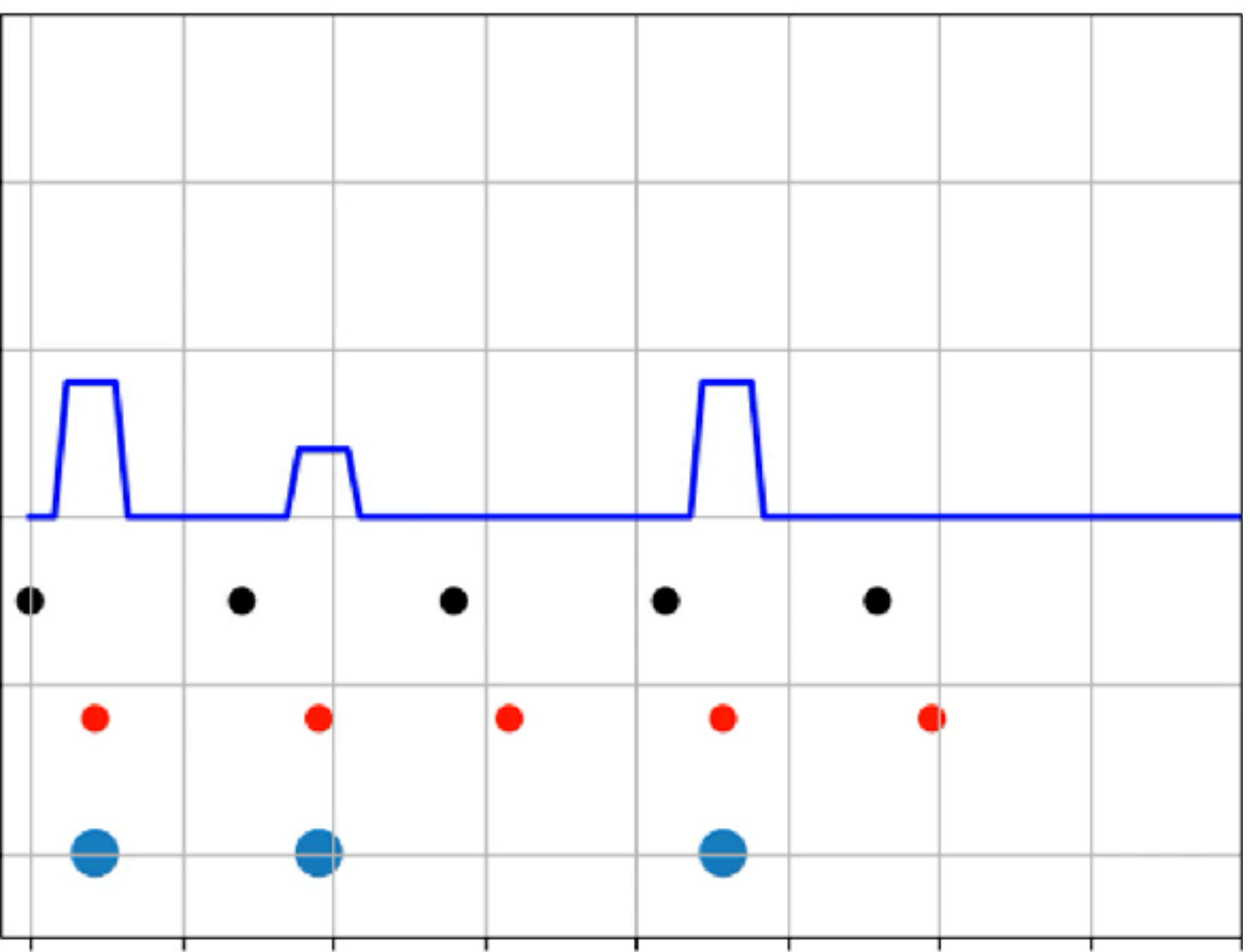
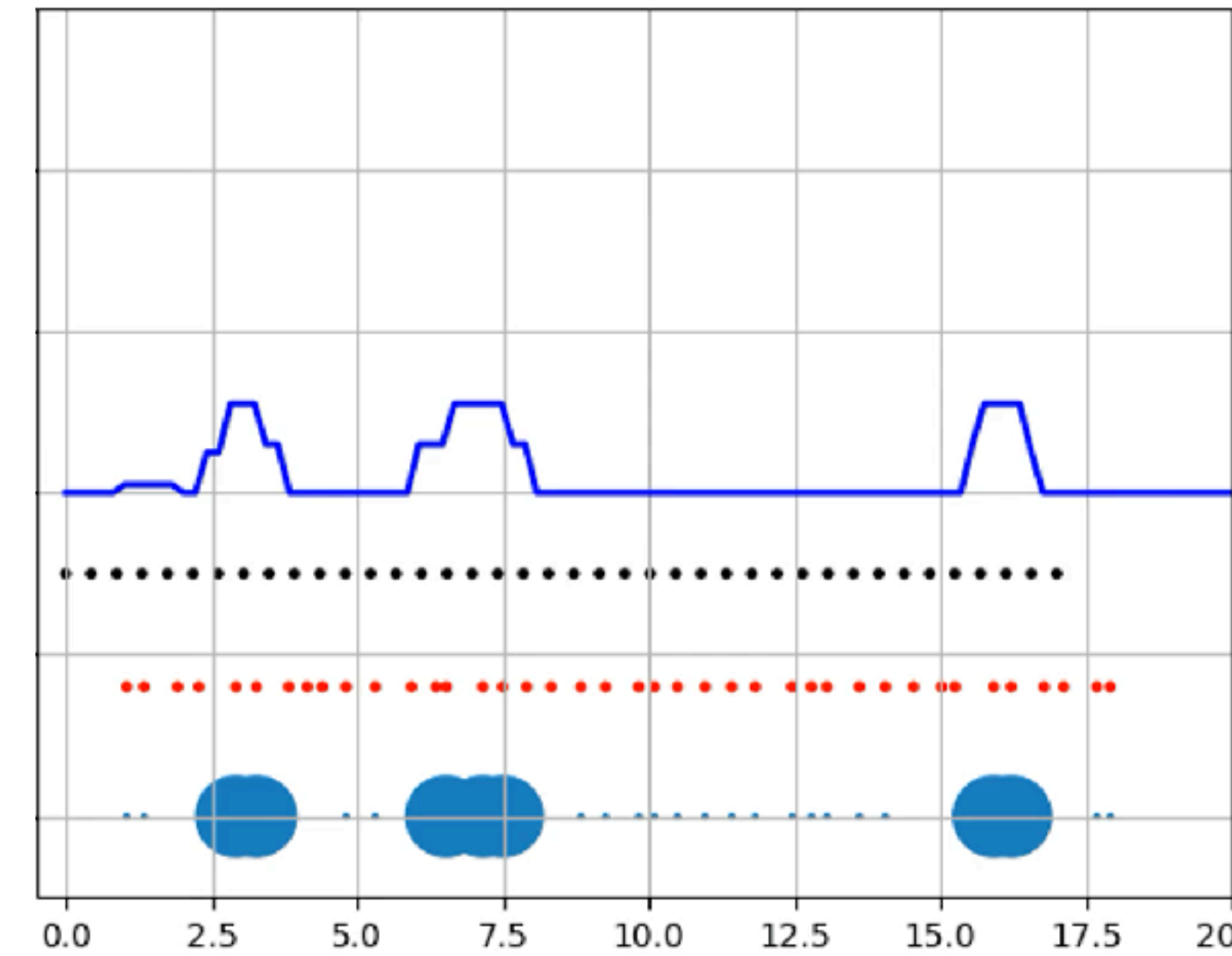
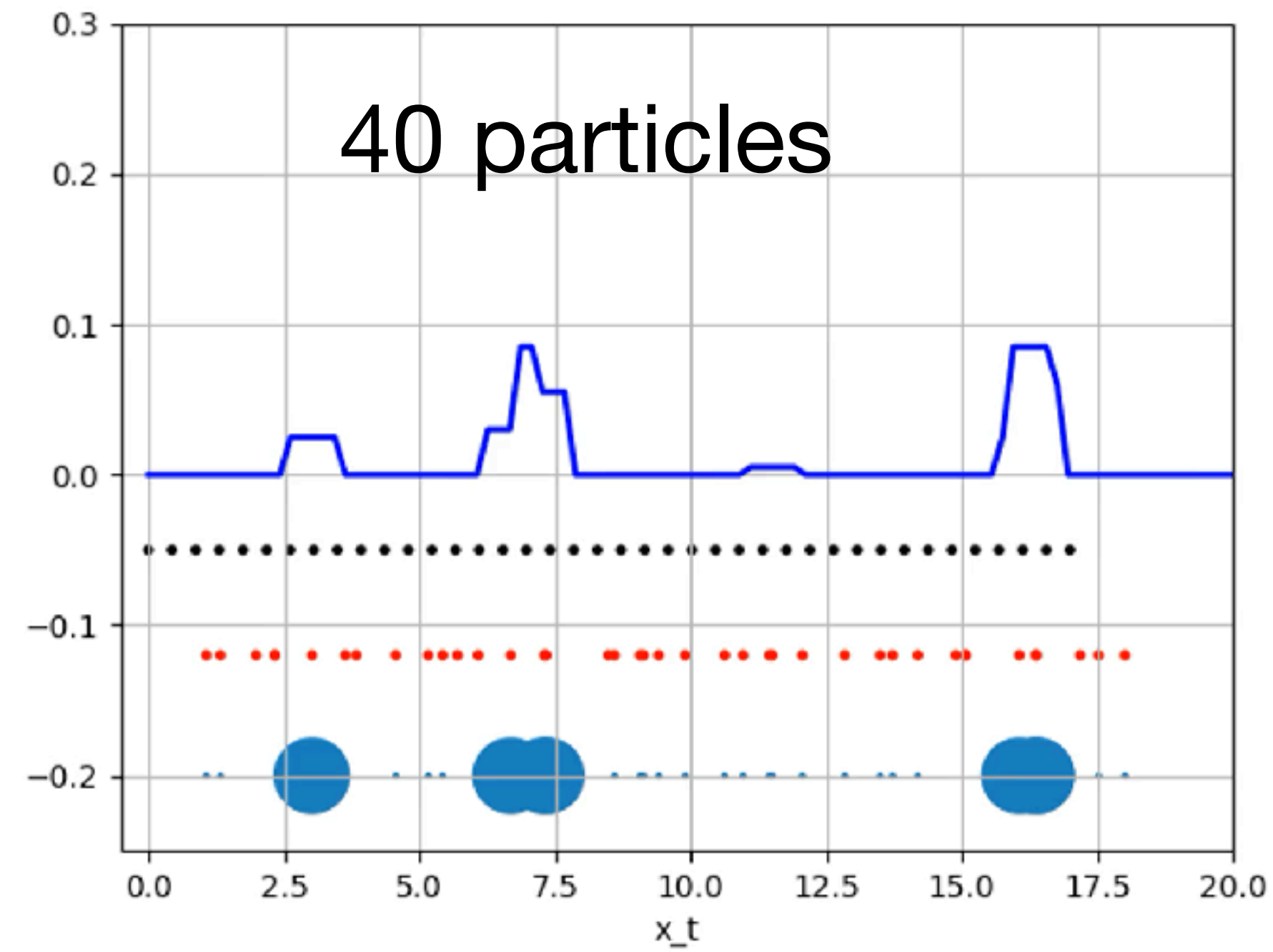
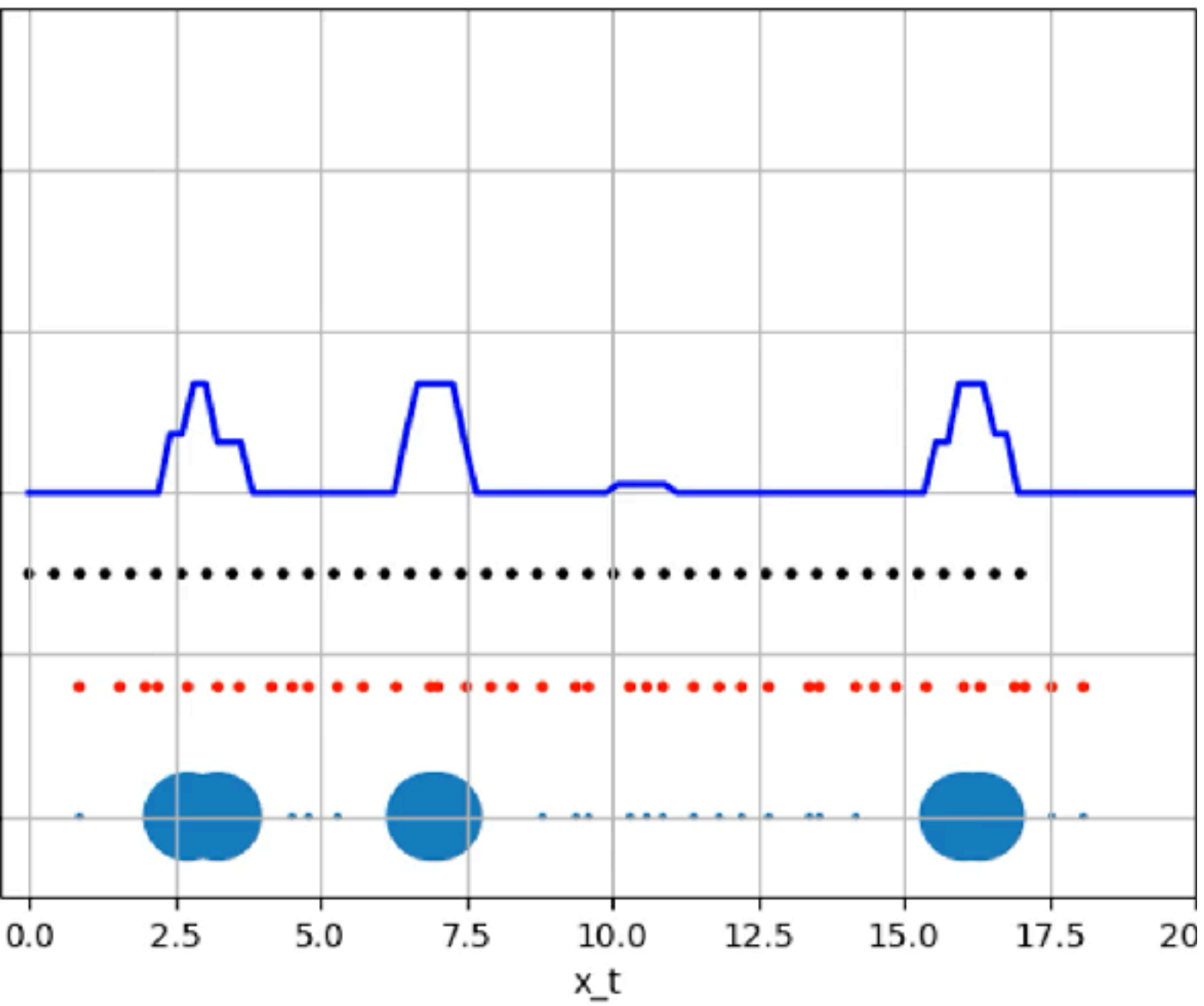
Particle filter

1. Initialize particles: $\mathcal{X}_0 = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n\}$
2. Prediction step (new action \mathbf{u}_t performed):
For all \mathbf{x}_{t-1}^i
$$\bar{\mathbf{x}}_t^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$$
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 $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$
5. Repeat from 2:
 $t = t + 1$

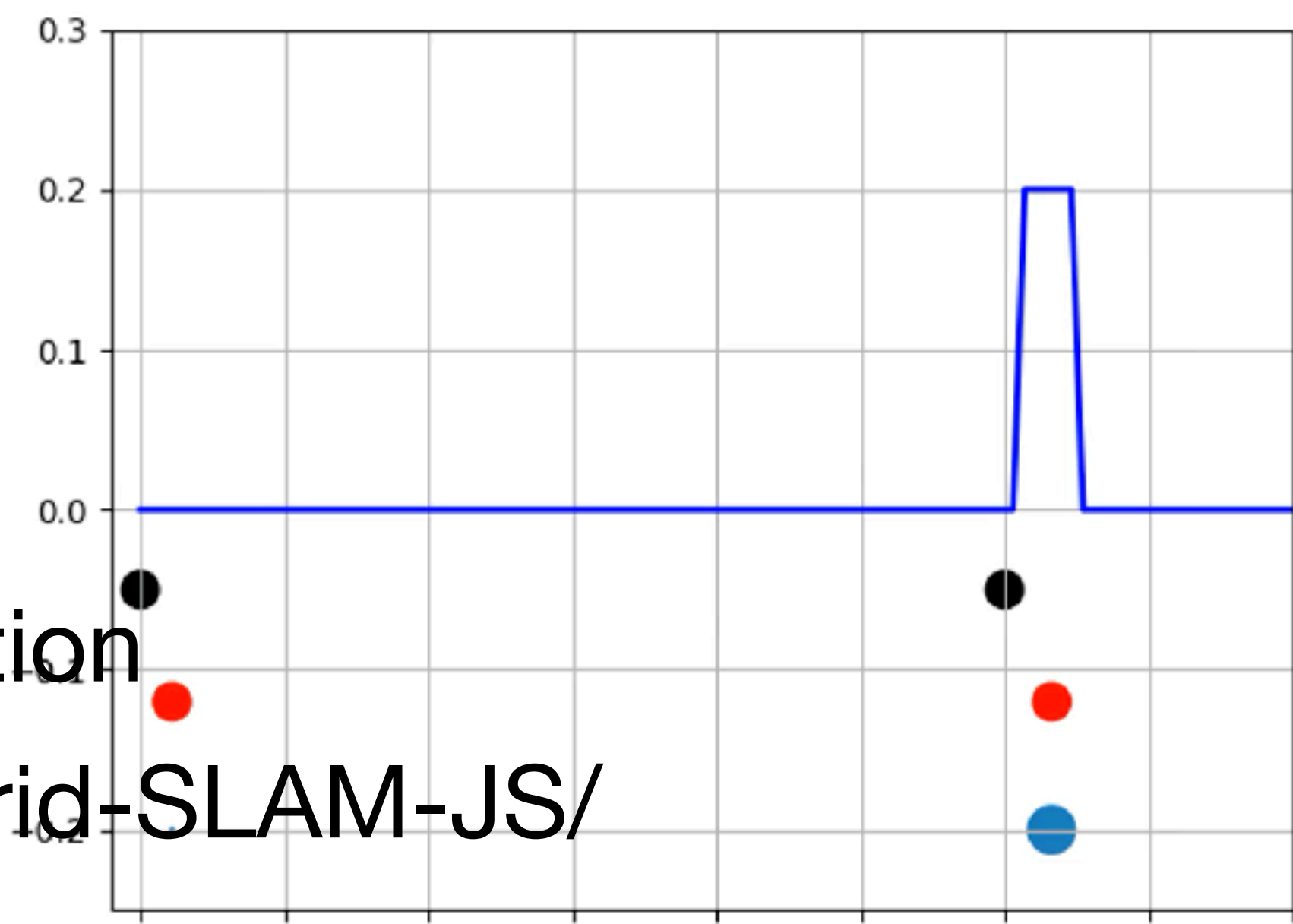
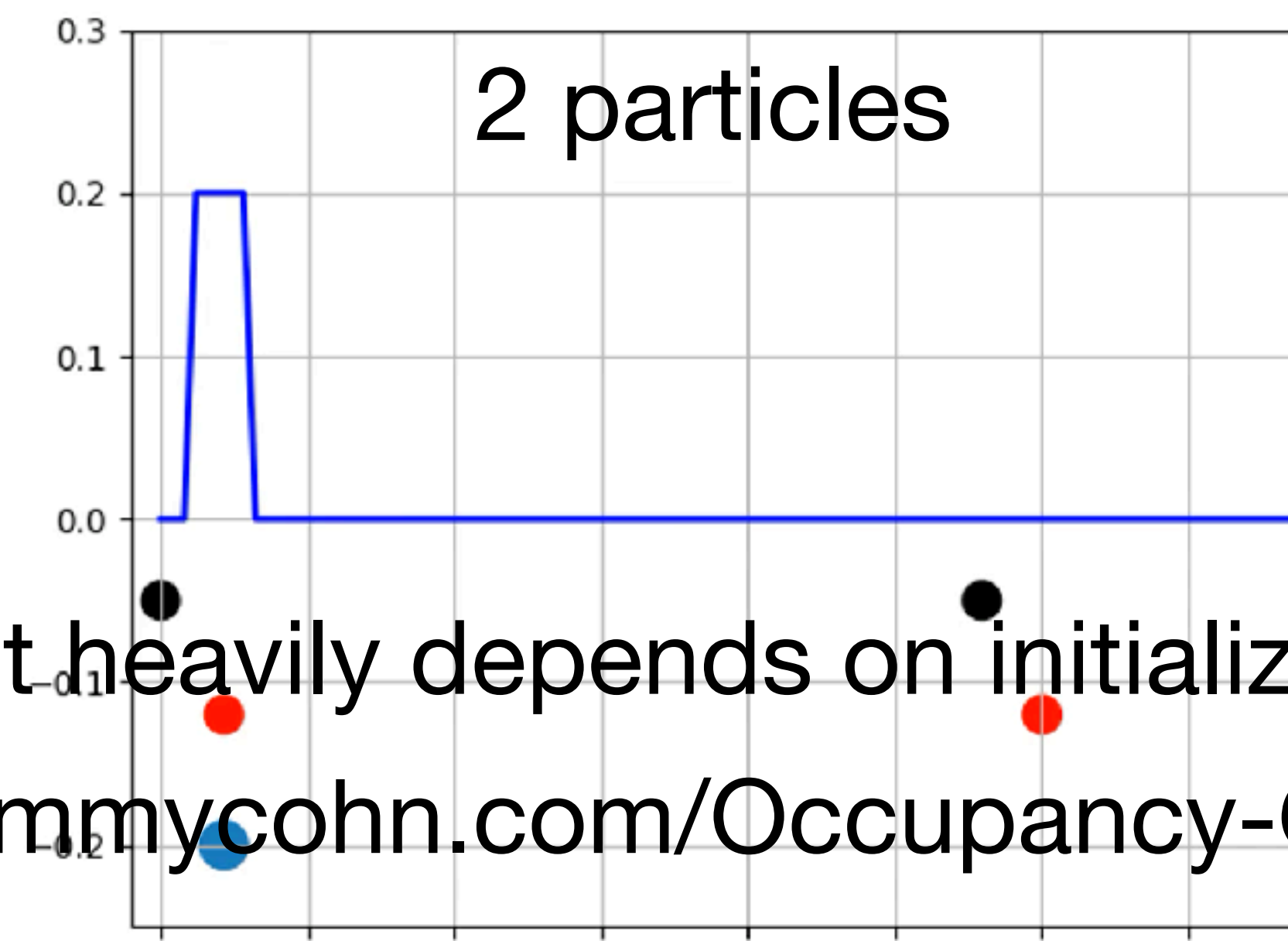
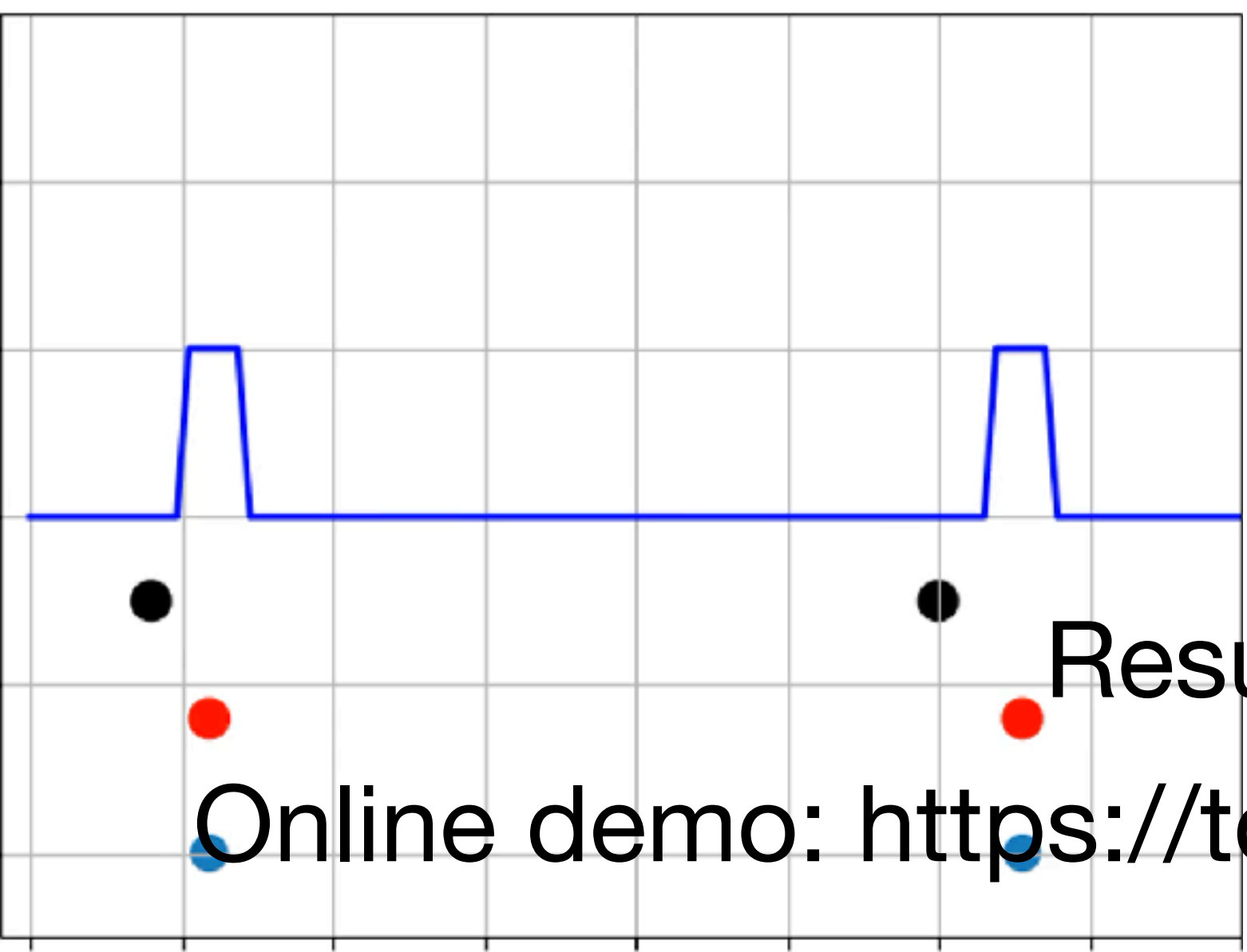
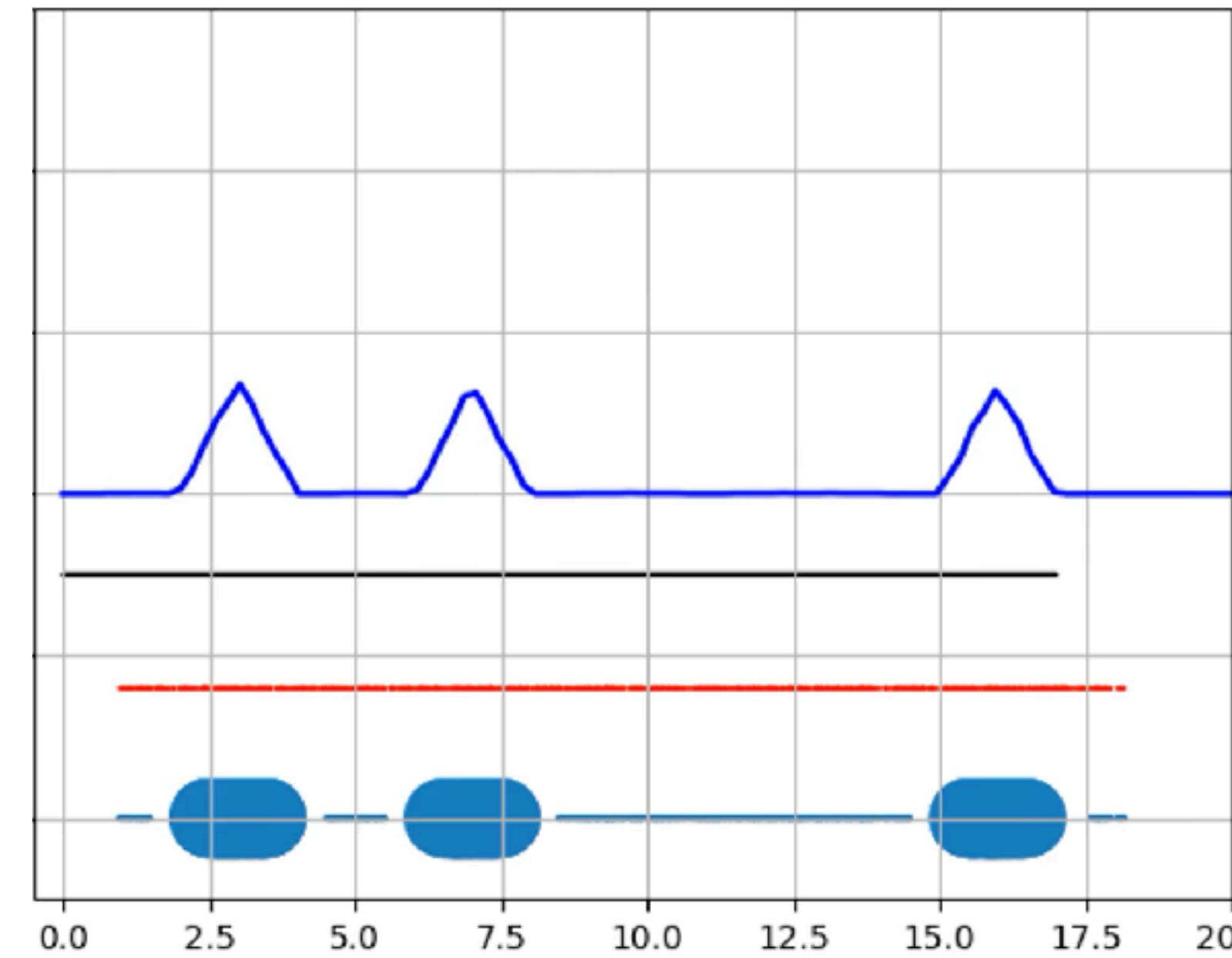
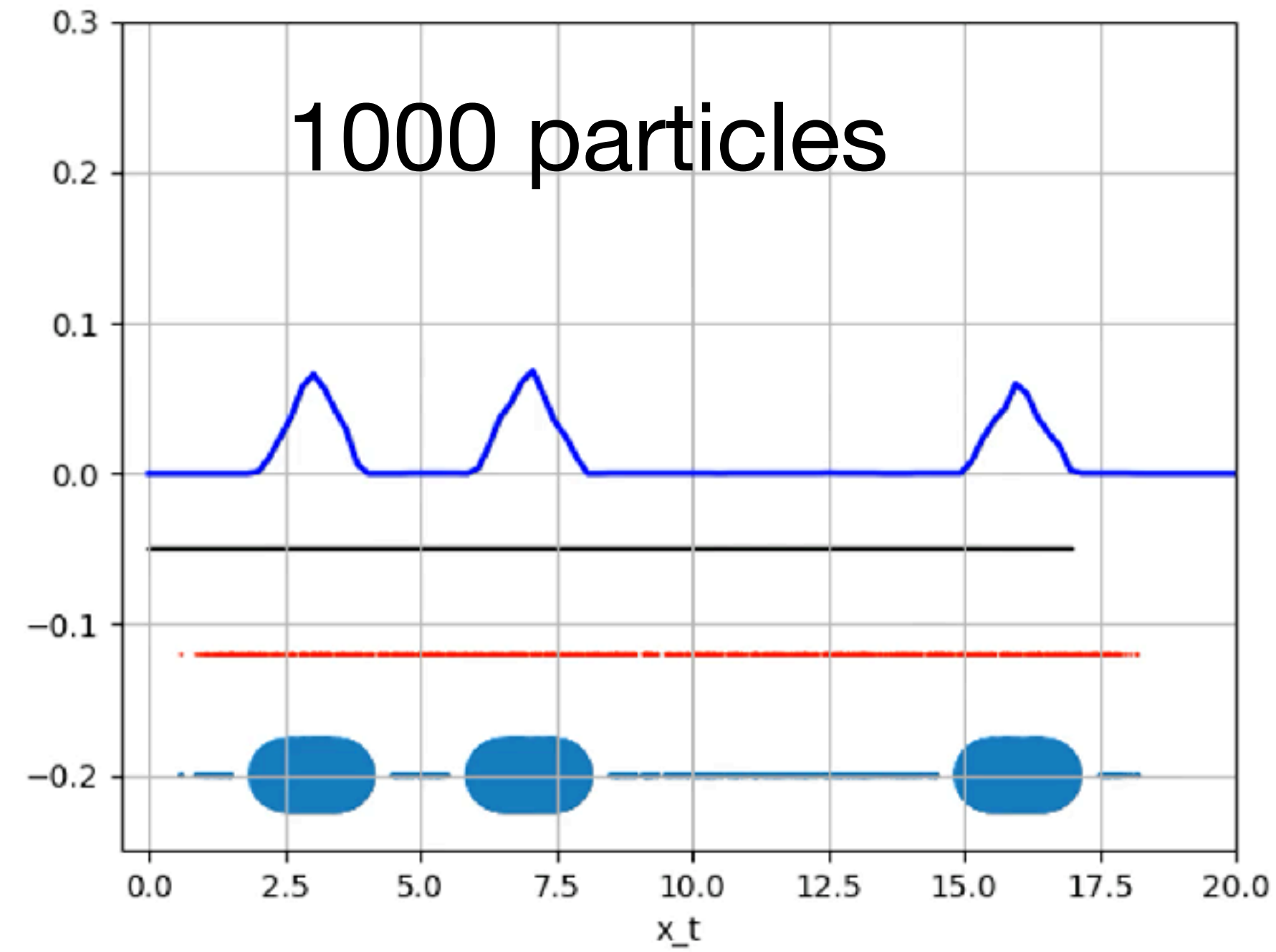
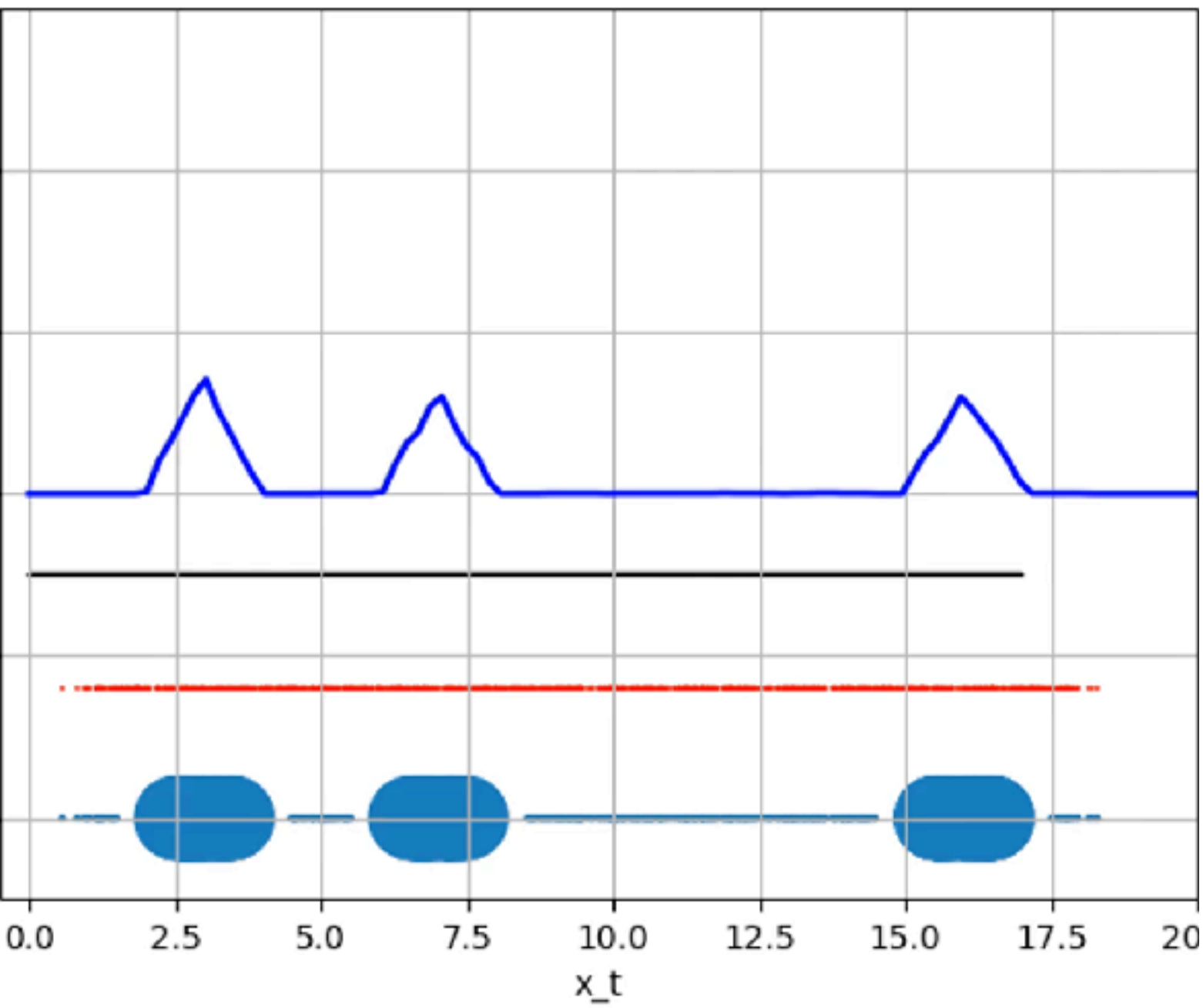
$$p(\mathbf{z}_t = 0 | \mathbf{x}_t)$$



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/>

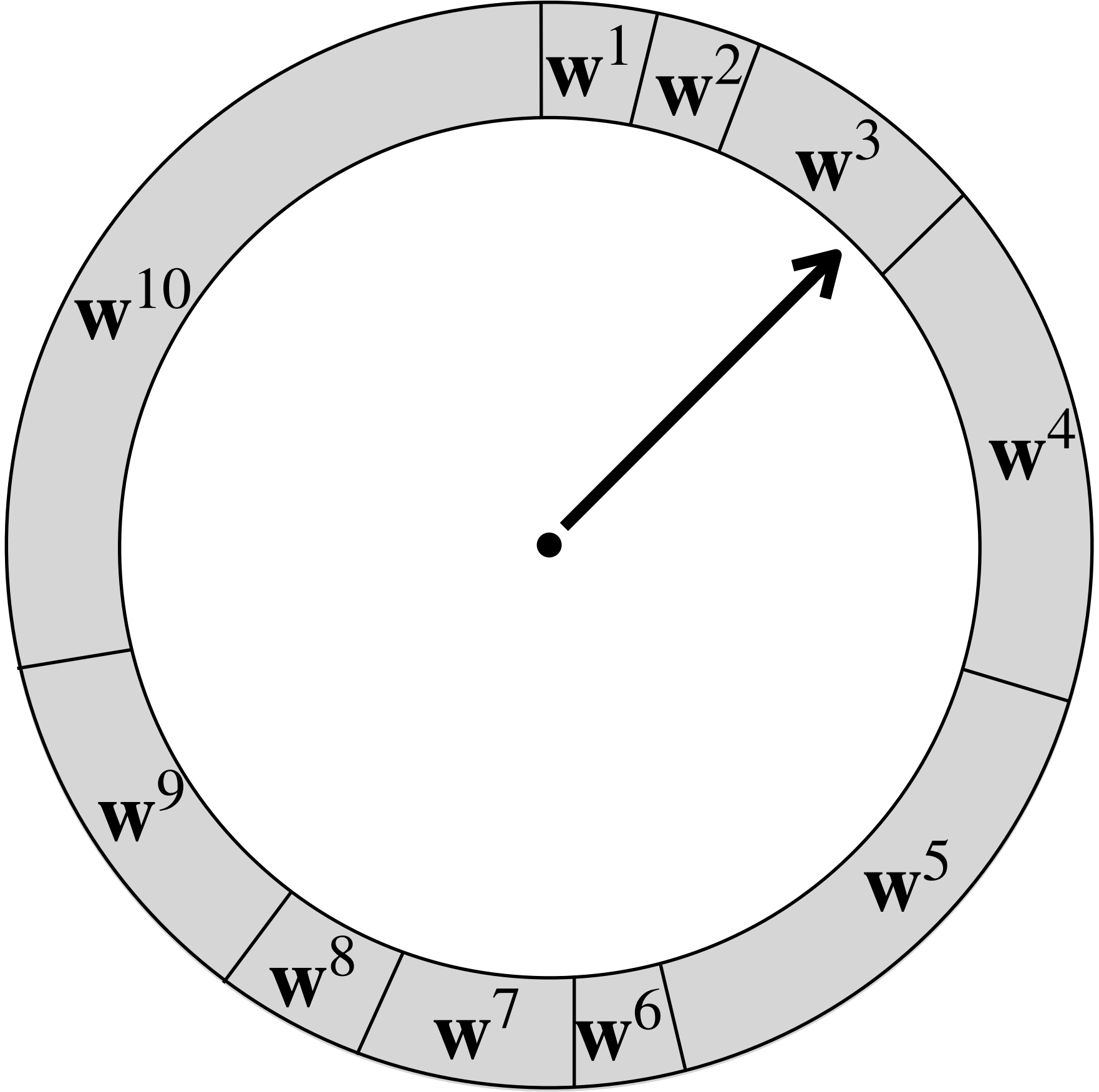
4. Resample

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



4. Resample

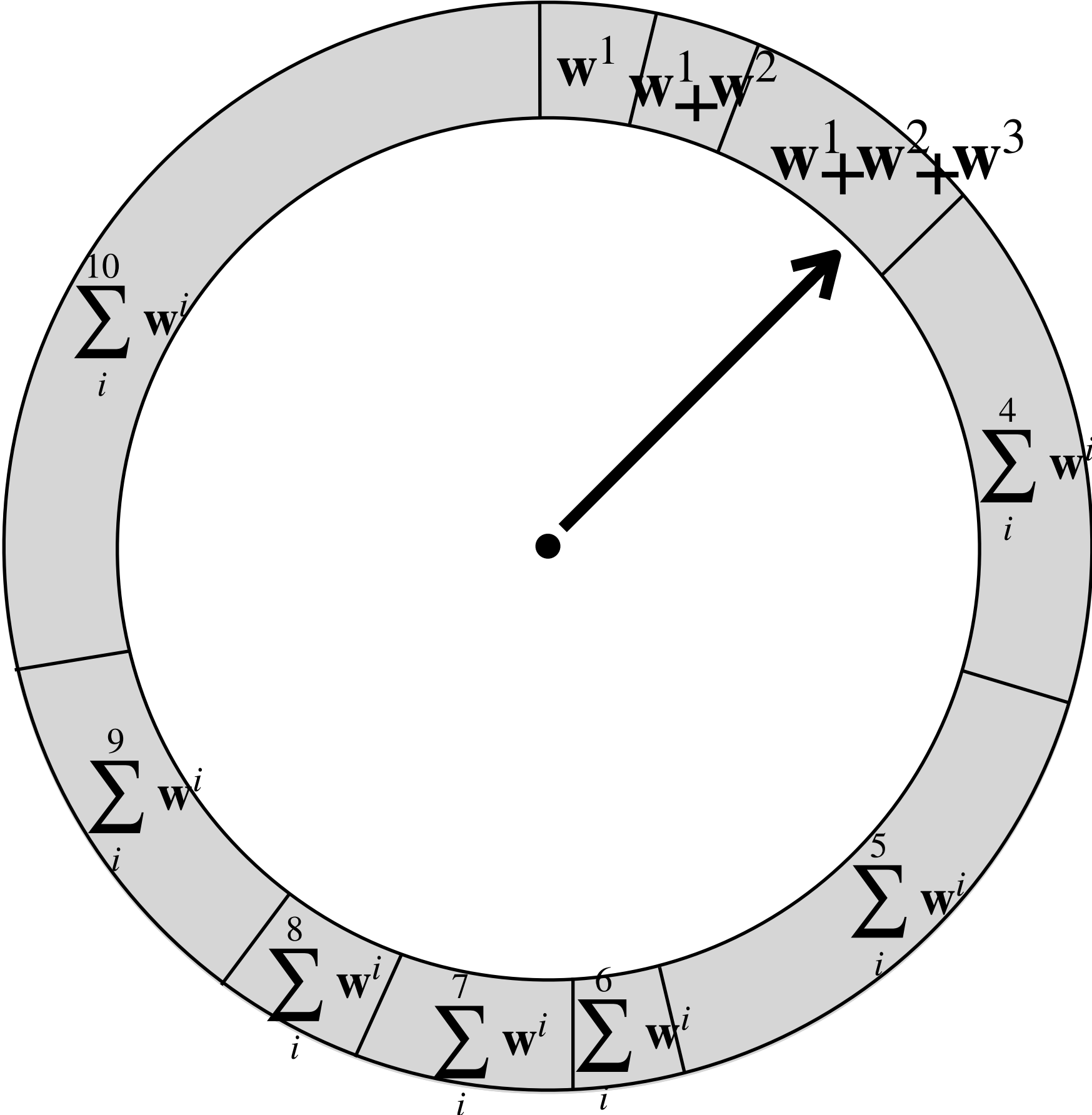
Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Roulette wheel

4. Resample

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

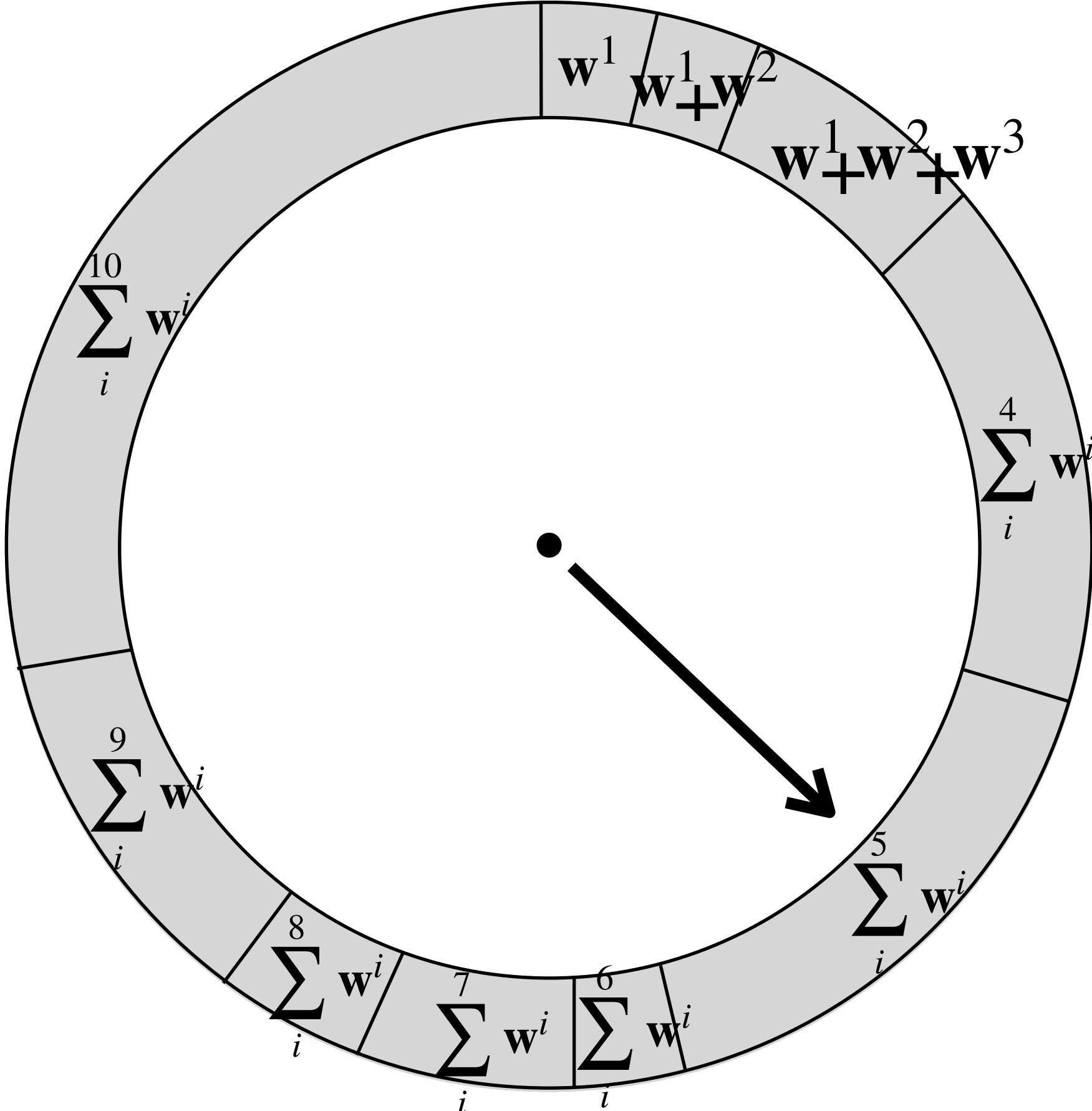


Replace values by cumsum

Roulette wheel

4. Resample

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



Generate random number

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

Repeat is N times

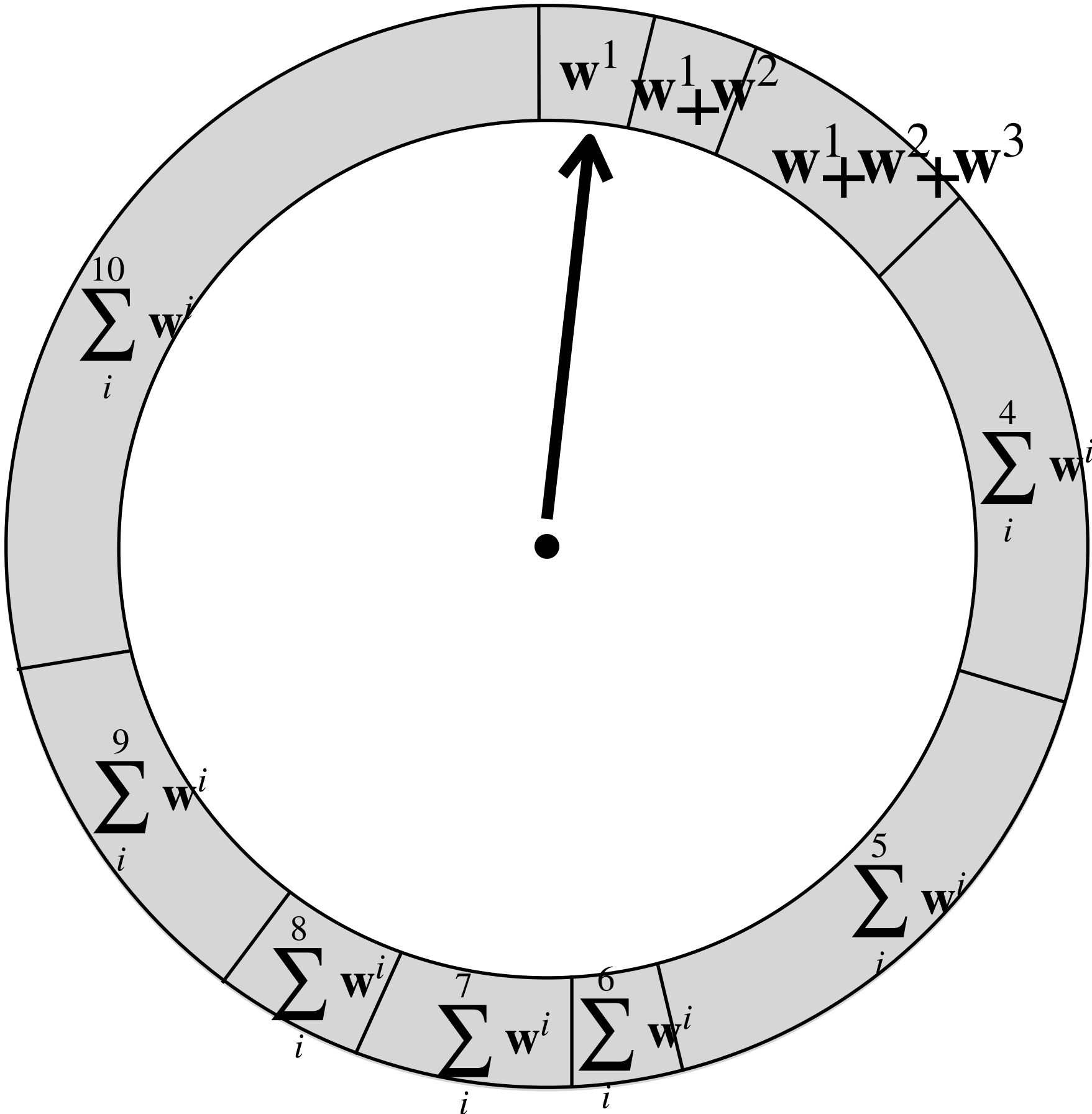
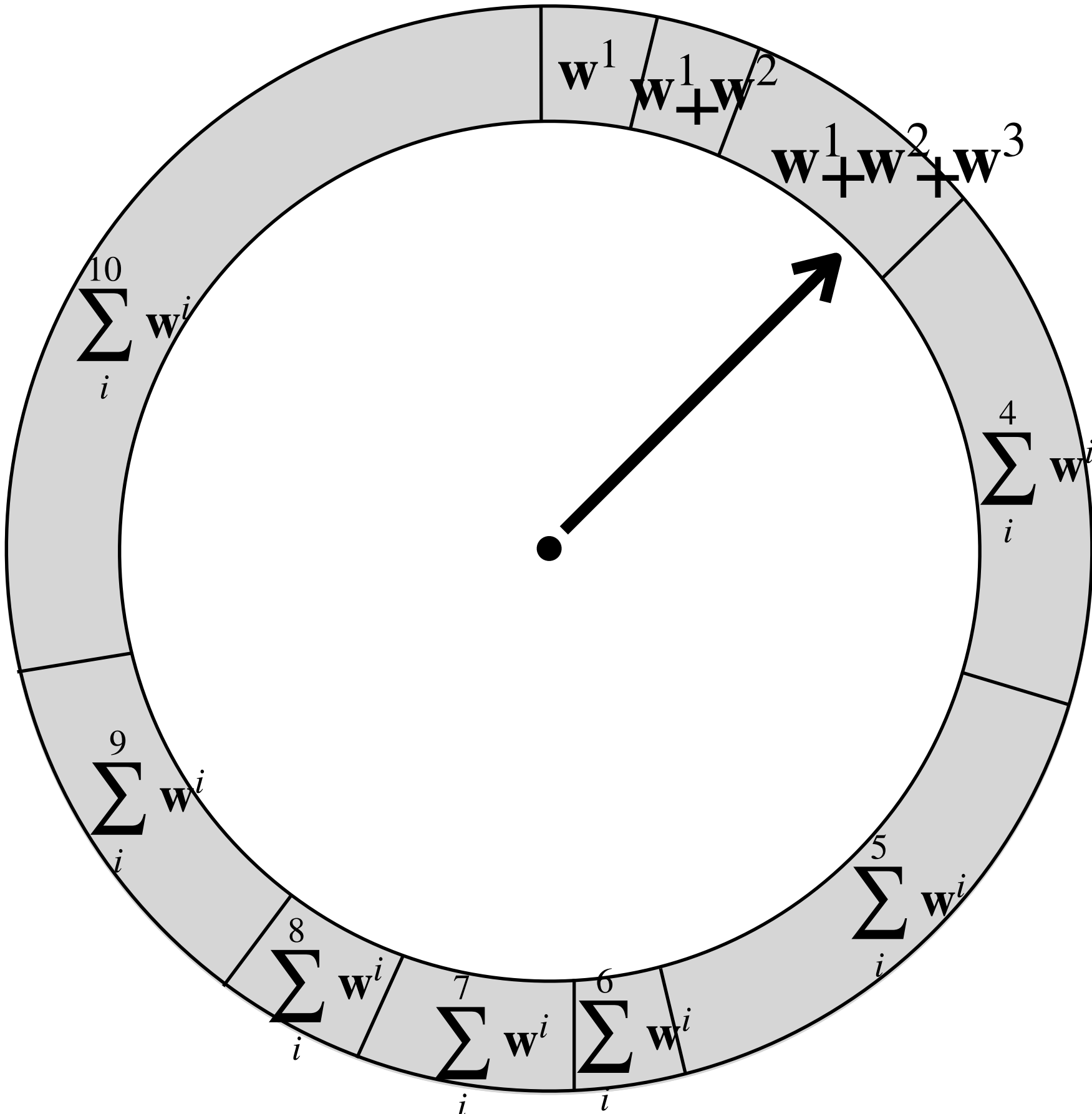
Roulette wheel

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

4. Resample

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

Generate only one small random number



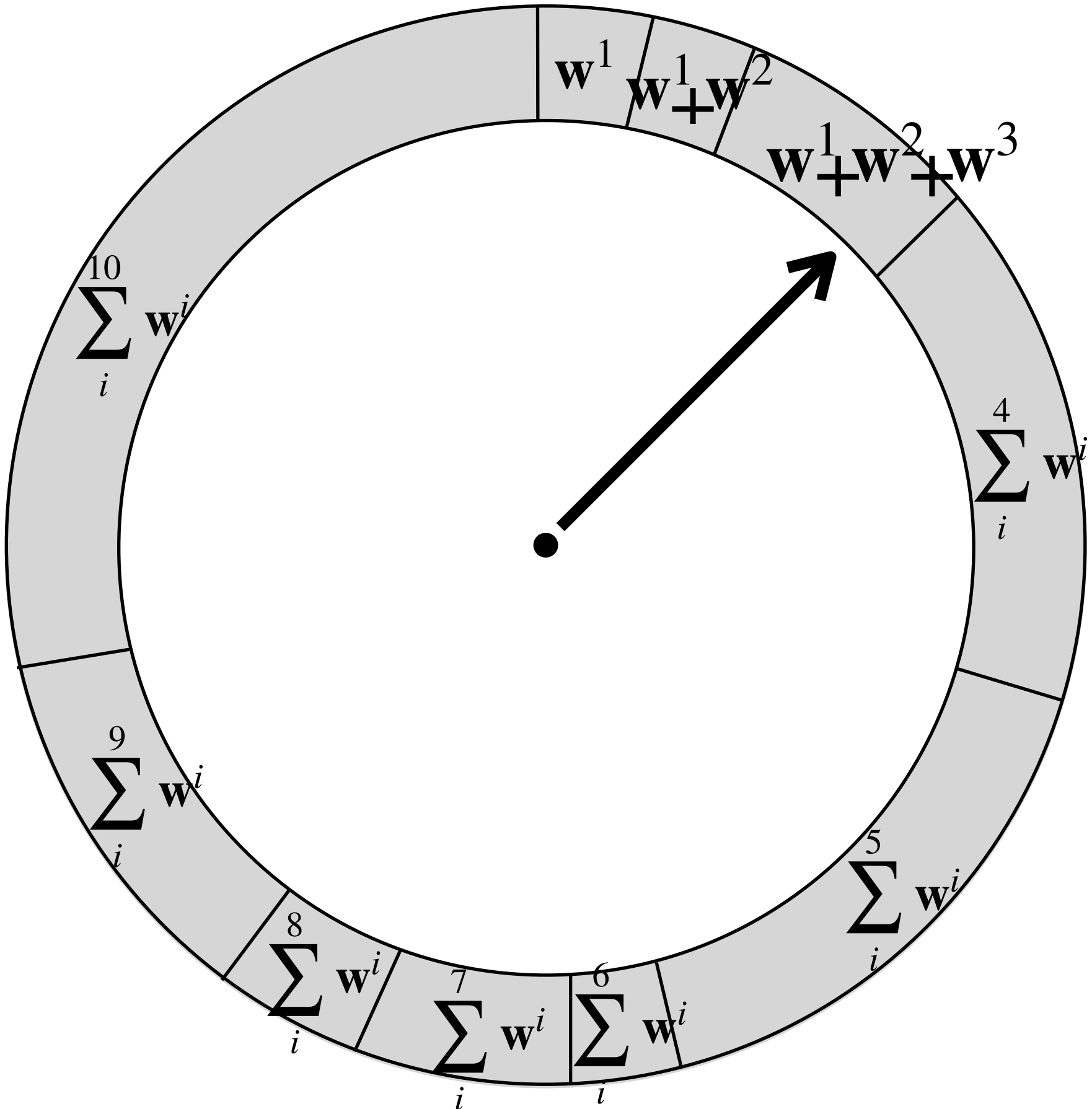
Roulette wheel

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

Stochastic universal resampling

4. Resample

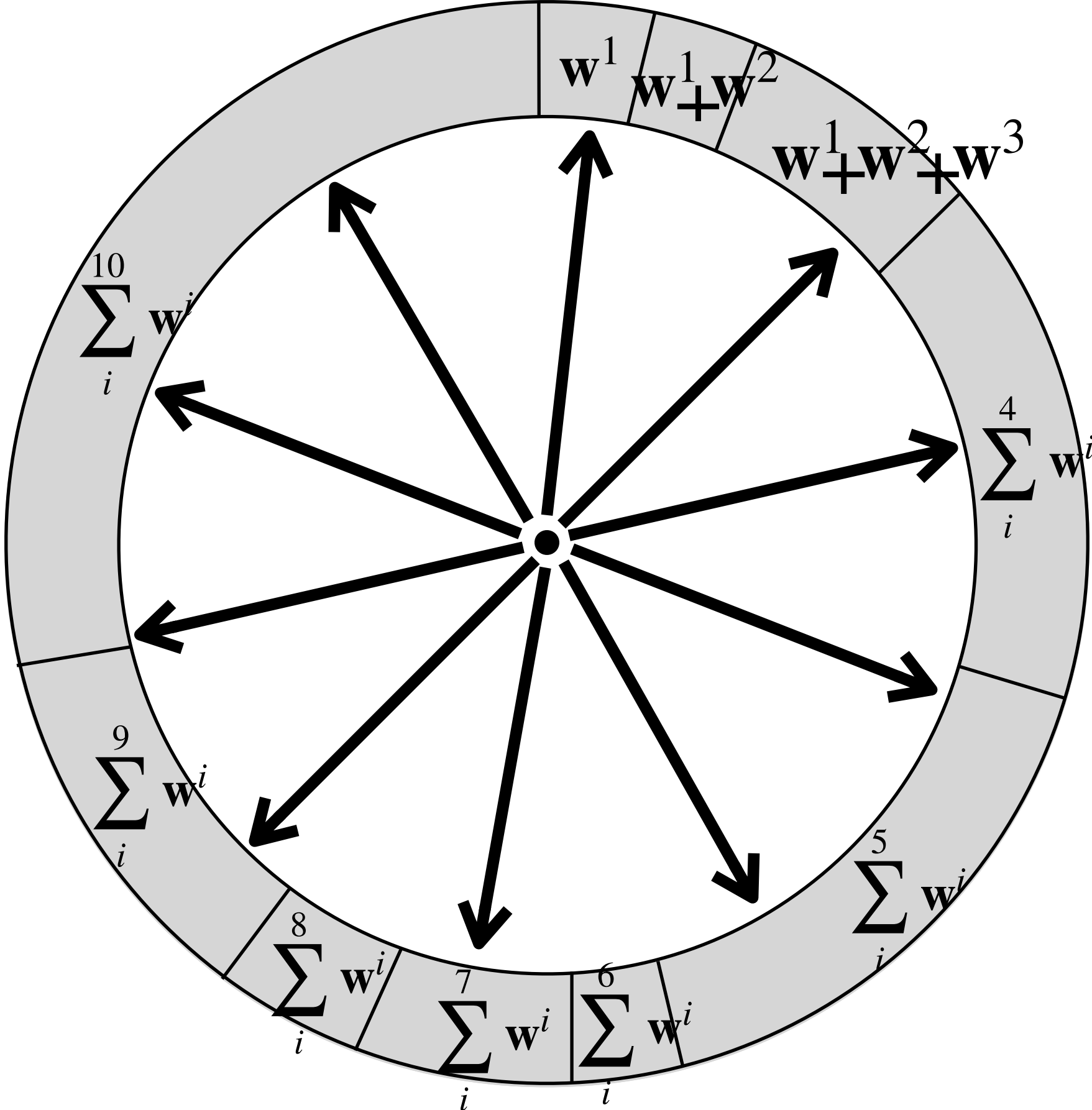
Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Roulette wheel

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

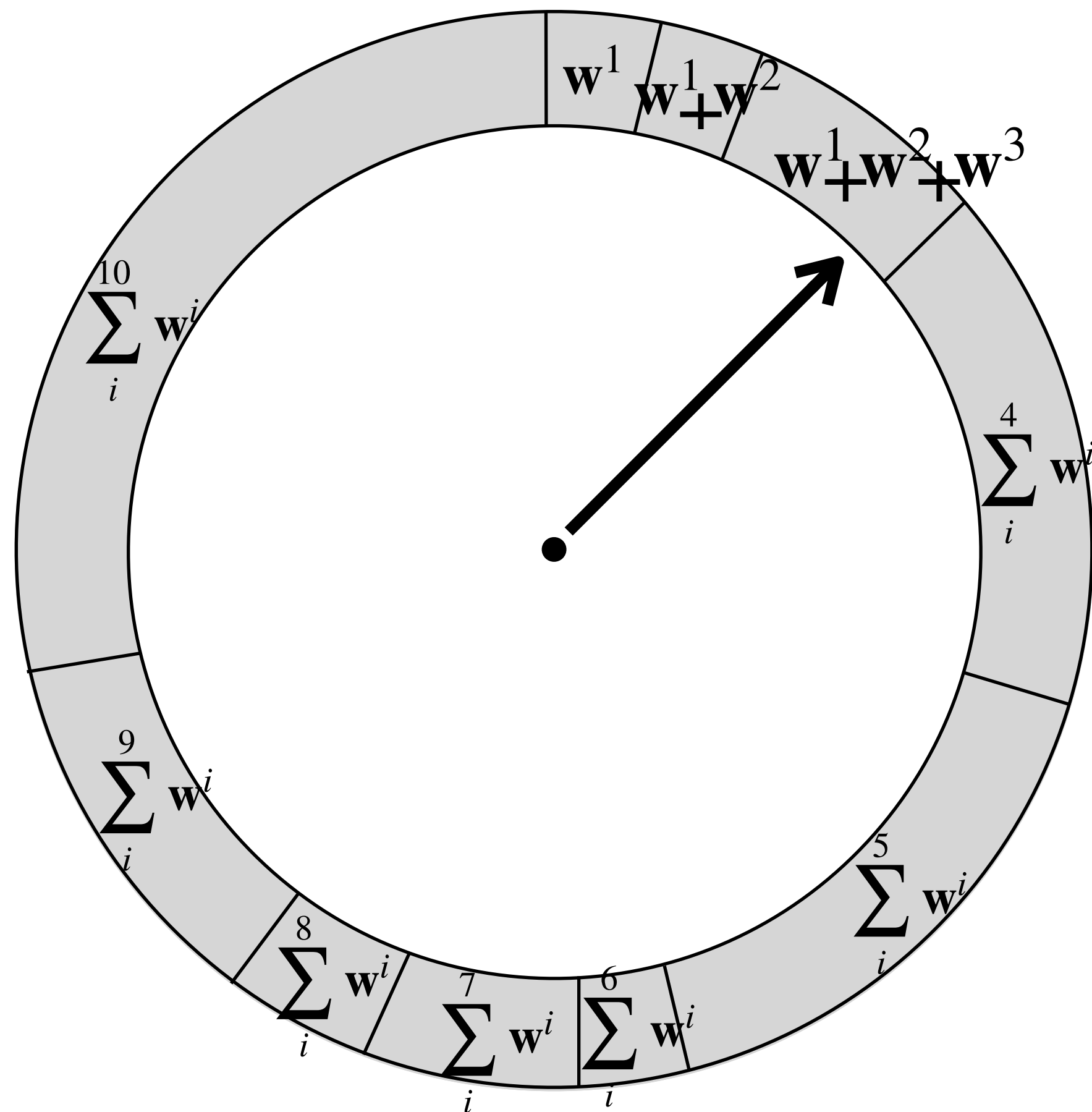
Generate N equally distributed samples starting at the random number



Stochastic universal resampling

4. Resample

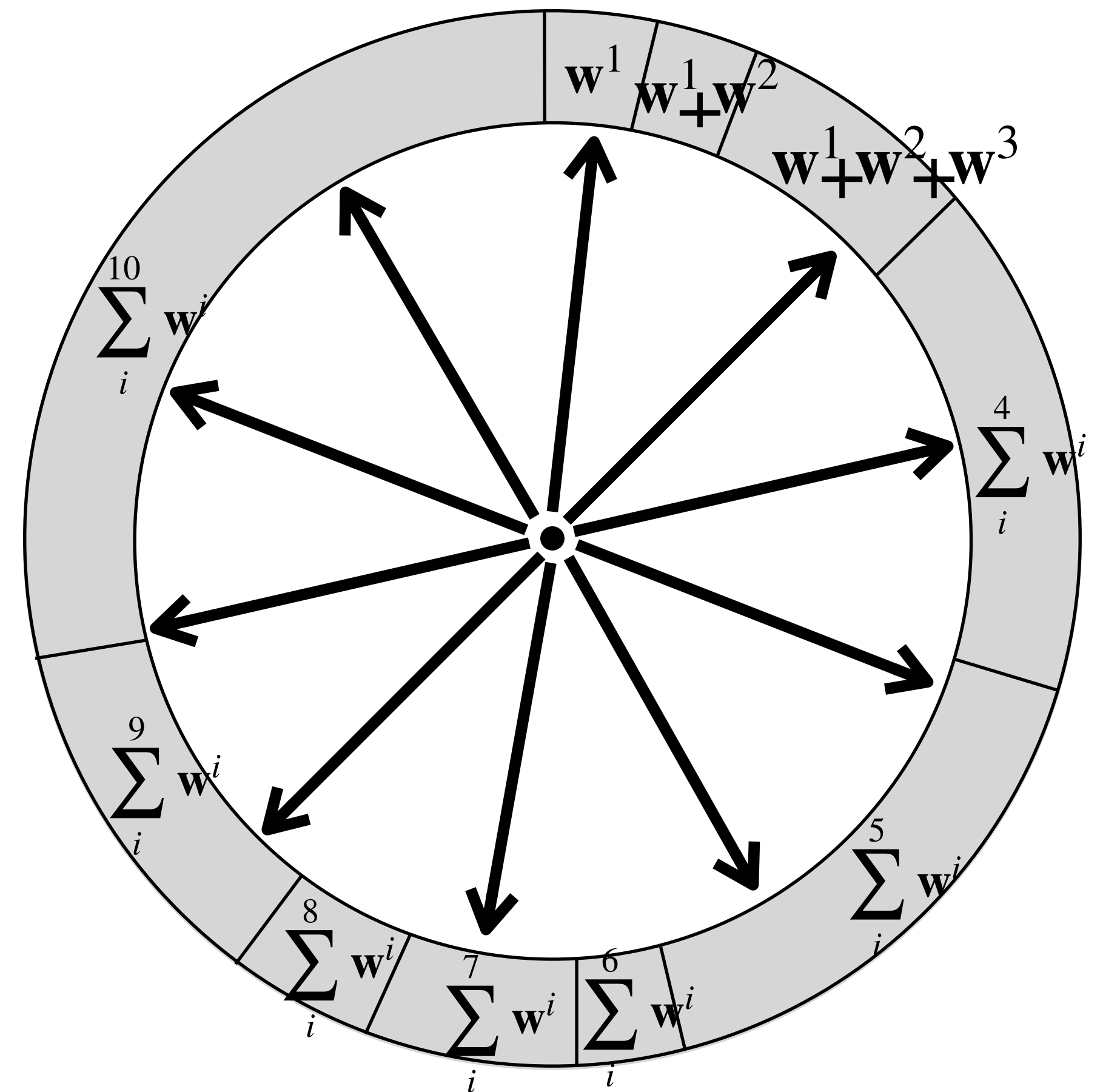
Draw $\mathcal{X}_t = \{\mathbf{x}_t^1, \dots, \mathbf{x}_t^n\} \propto \mathbf{w}_t^i$



Roulette wheel

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

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

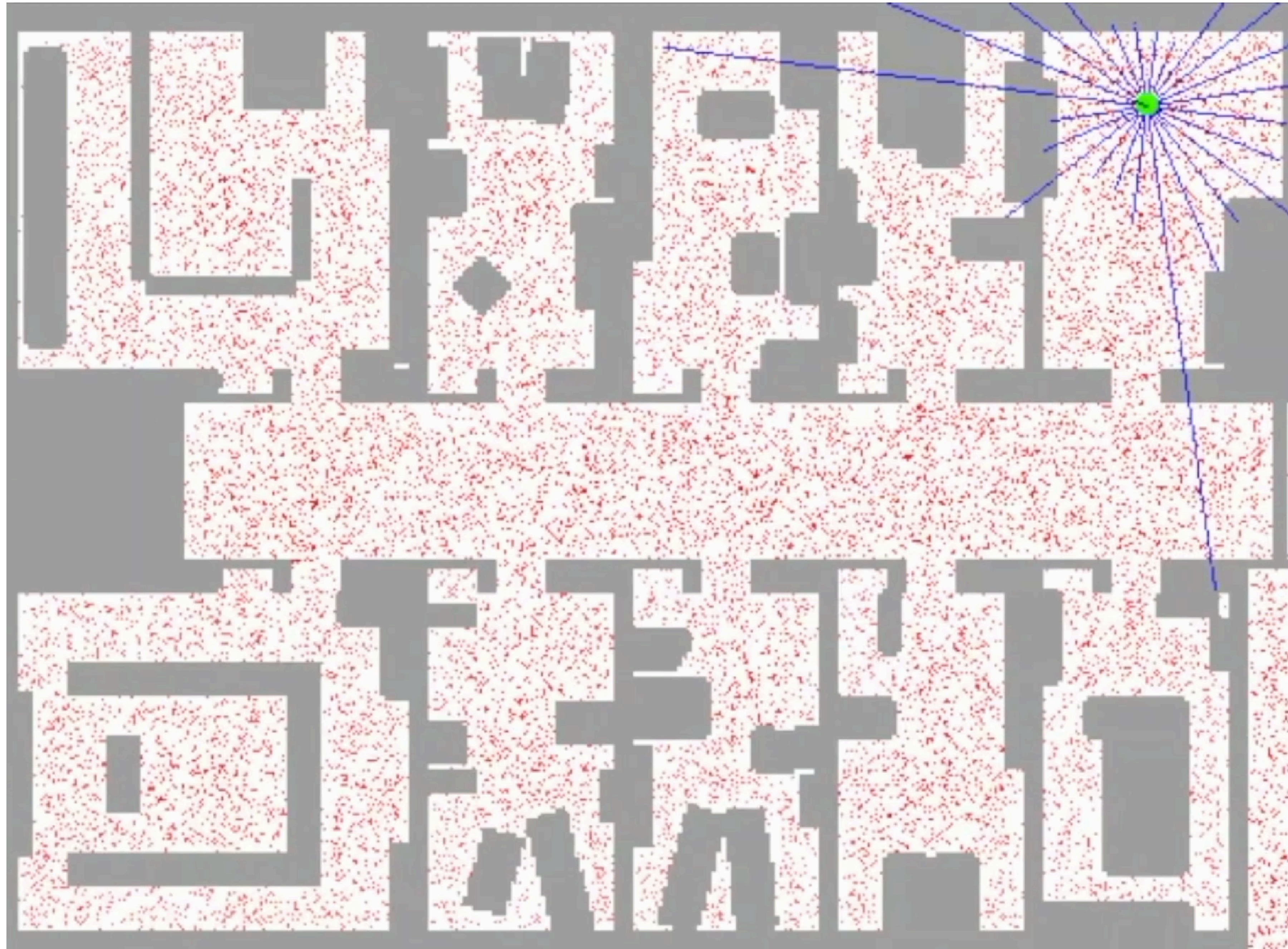


Stochastic universal resampling

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

Particle filter example [Dieter Fox]

Particle filter example [Dieter Fox]



Summary

Drawbacks

- course of dimensionality
- spatial discretization

- course of dimensionality
- partical quantization

- represent only gaussians
- suffers from linearization

- represents gaussians
- grows to infinity

Advantages

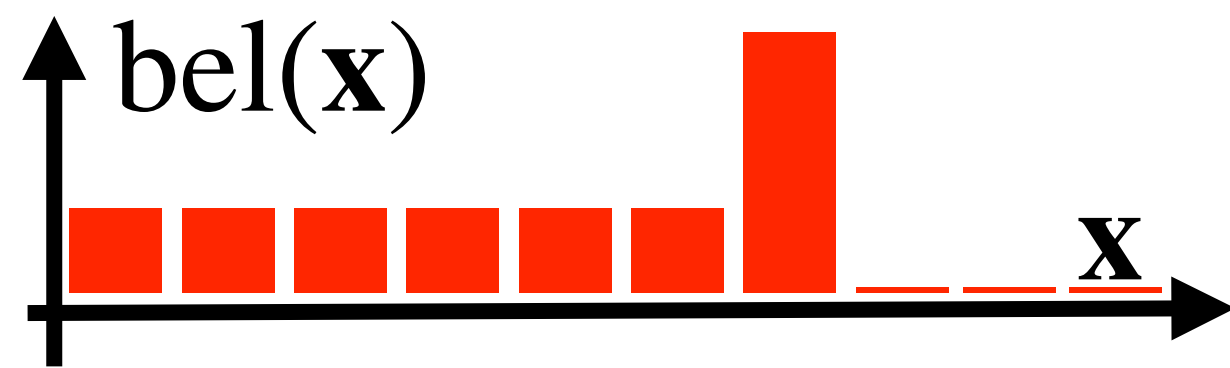
- represents arbitrary prob. distribution

- represents arbitrary prob. distribution

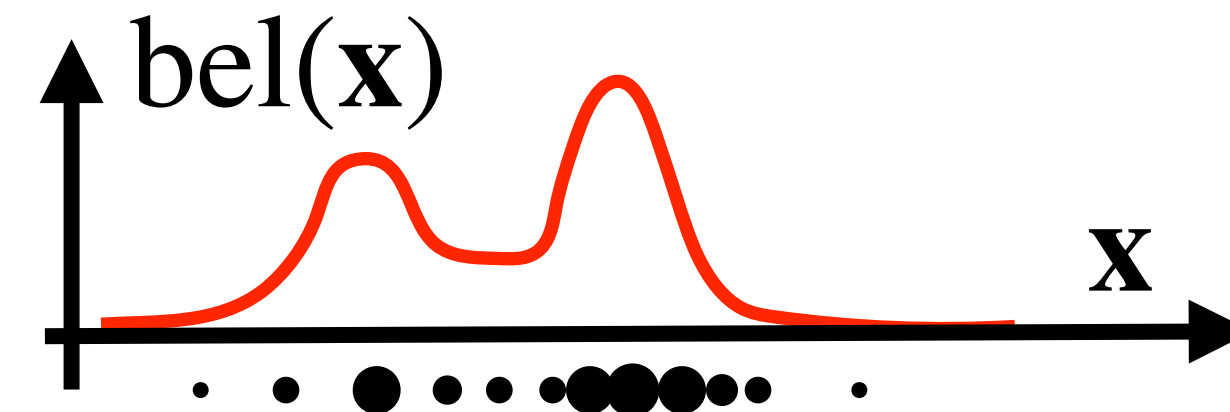
- nicely scales with higher dimensions

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

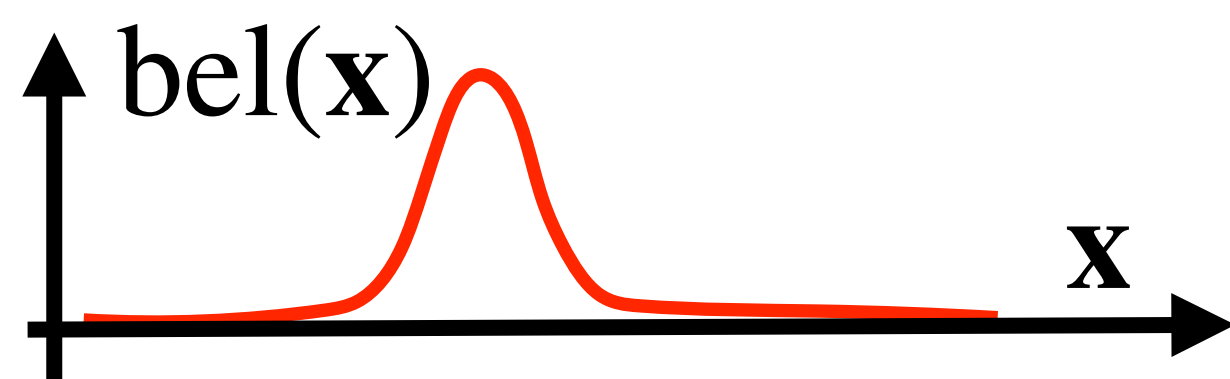
Bayes filter



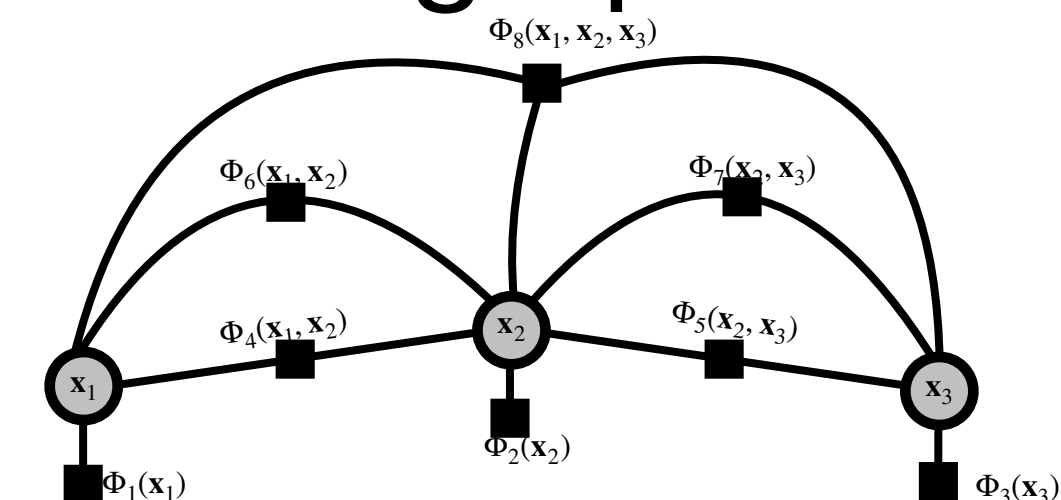
Particle filter



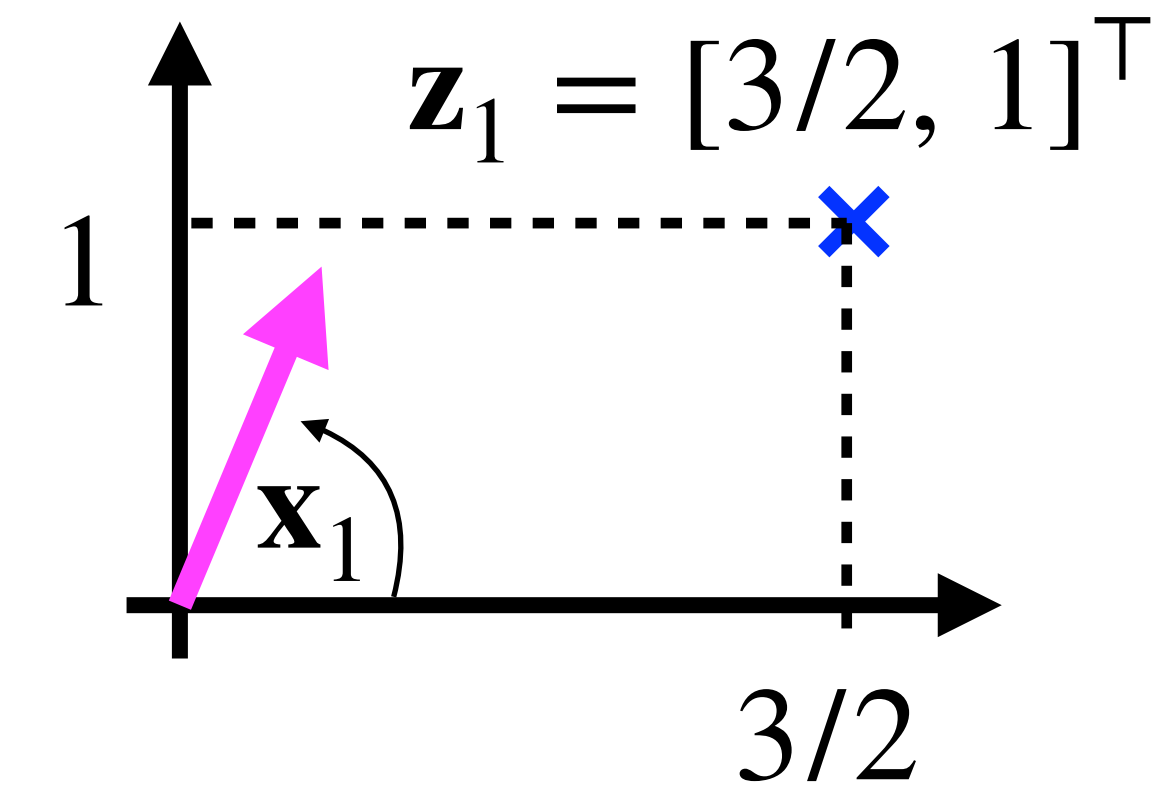
Kalman filter



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

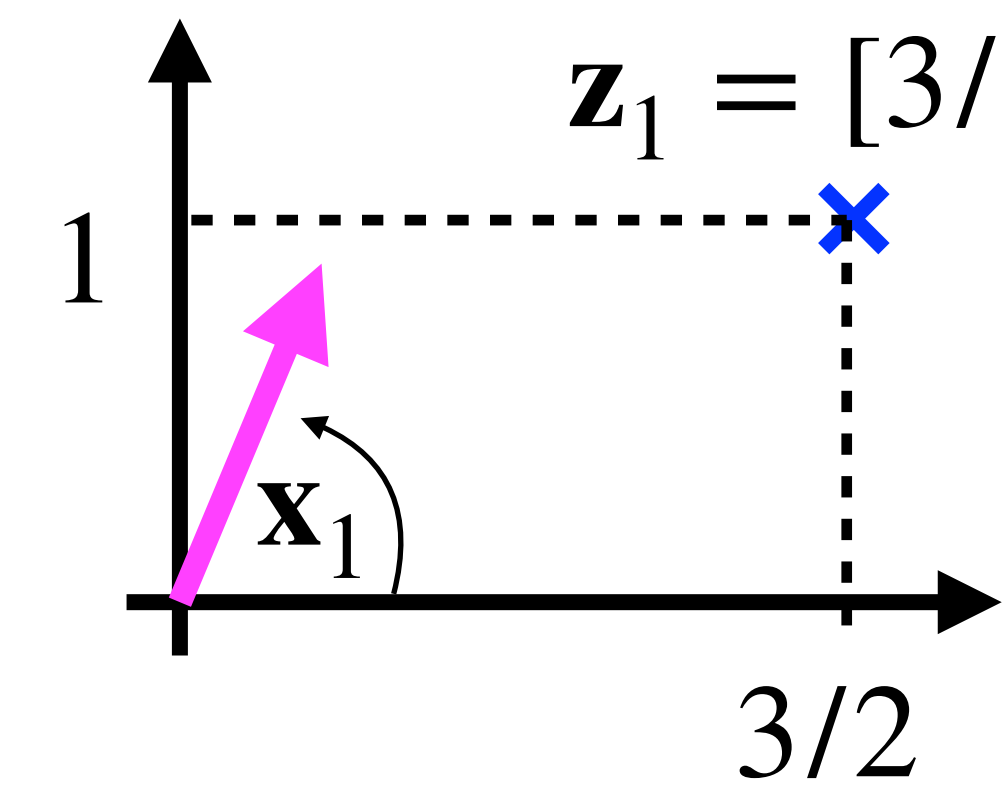
$$\mathbf{z}_t = h(\mathbf{x}_t) = \begin{bmatrix} \cos \mathbf{x}_t \\ \sin \mathbf{x}_t \end{bmatrix}$$

with zero-mean gaussian noise with covariance $\mathbf{Q}_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

- Consider two states example, where:

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

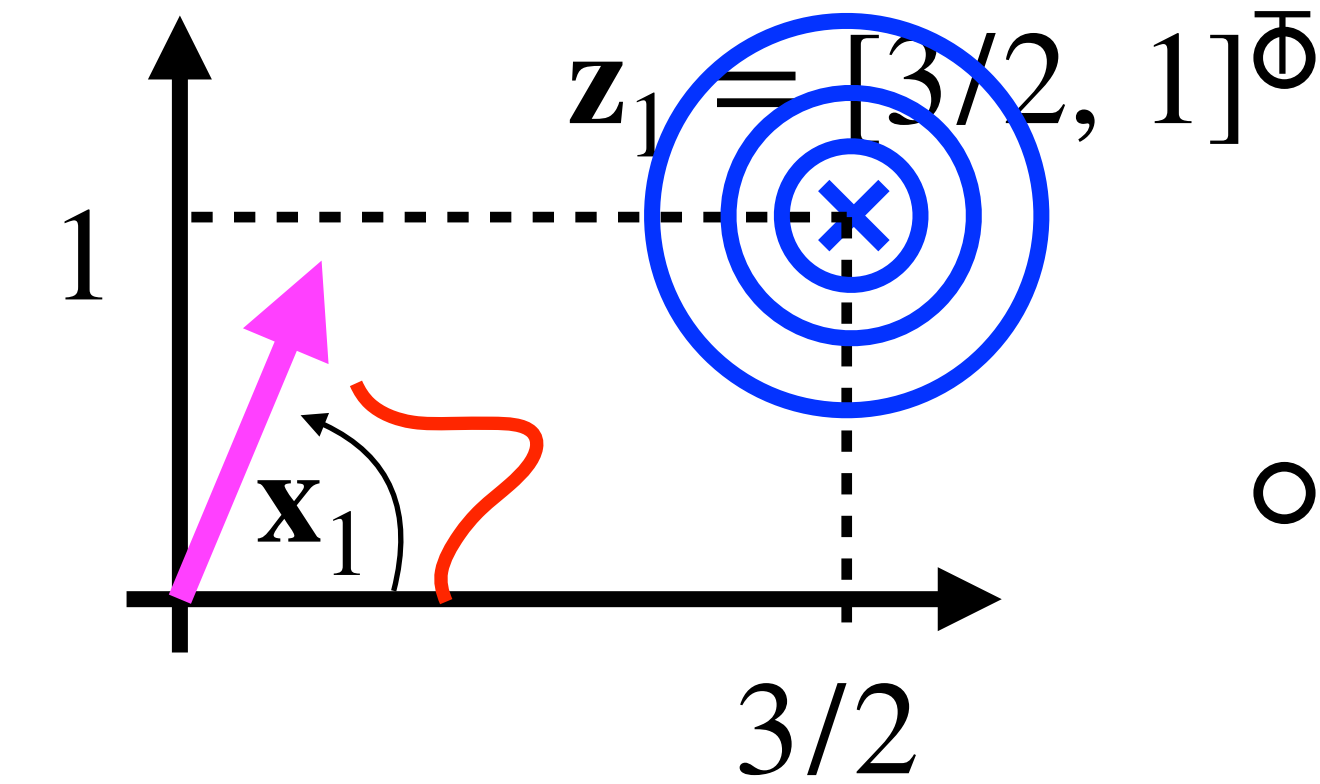
Factorgraph



$\mathbf{z}_1 = [3/2, 1]^T$ 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

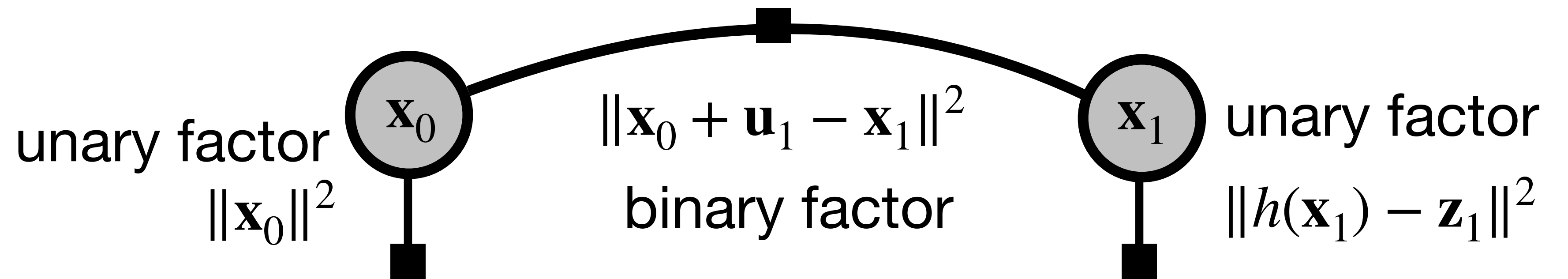
$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1} + \mathbf{u}_t, \mathbf{R}_t)$$

Write down measurement probability distribution

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

Outline distributions into the sketch

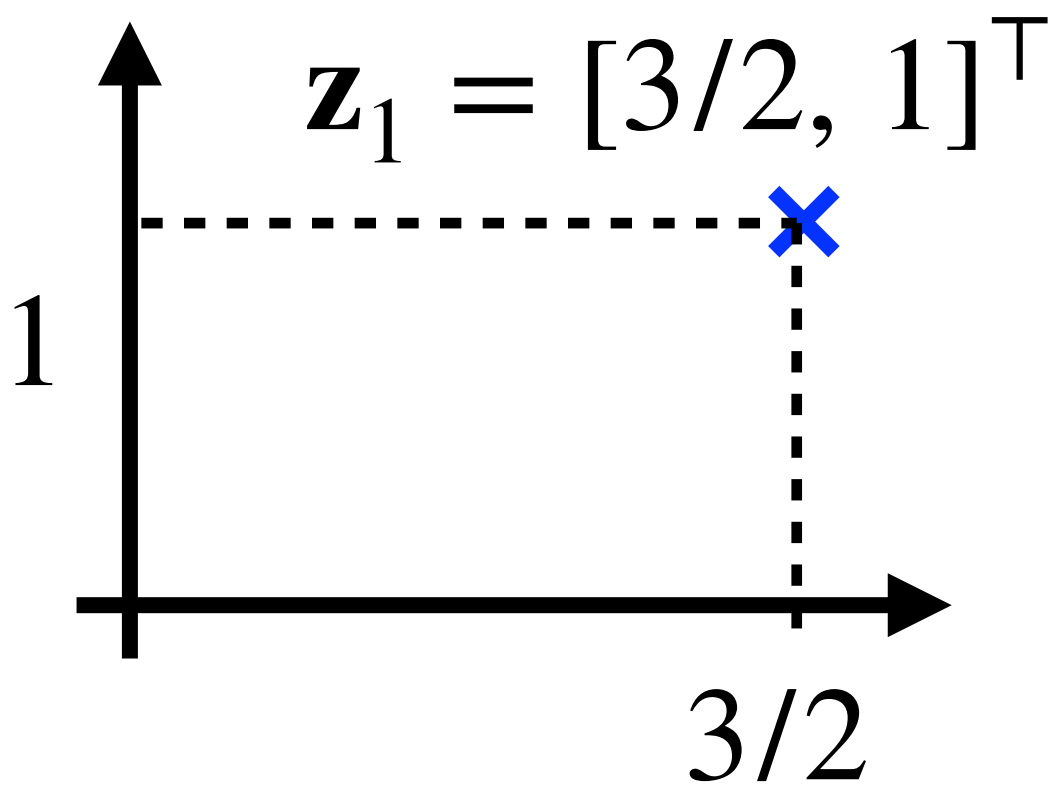
Draw underlying factorgraph



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. $\overline{\text{bel}}(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1; \bar{\mu}_1, \bar{\Sigma}_1)$

$\bar{\mu}_1 = ?$

$\bar{\Sigma}_1 = ?$

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

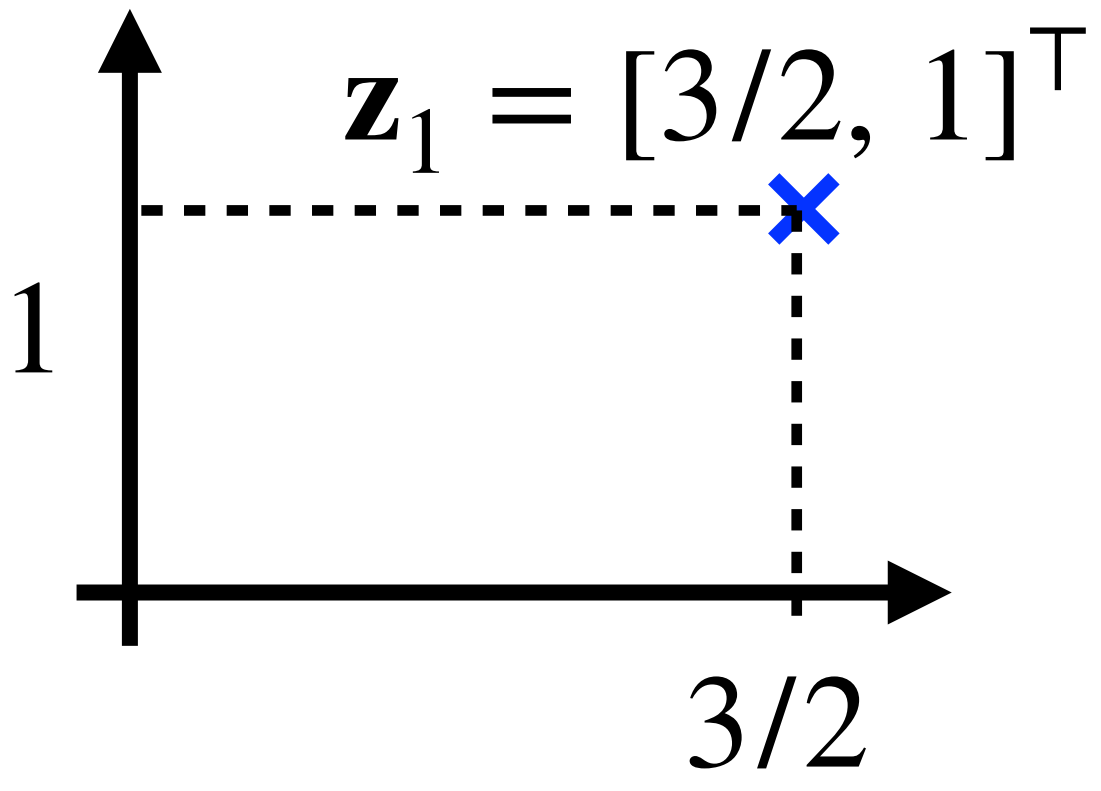
↔ $h(\mathbf{x}_1) = \begin{bmatrix} \cos(\mathbf{x}_1) \\ \sin(\mathbf{x}_1) \end{bmatrix} \approx ?$

○ **Perform measurement step of EKF**

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

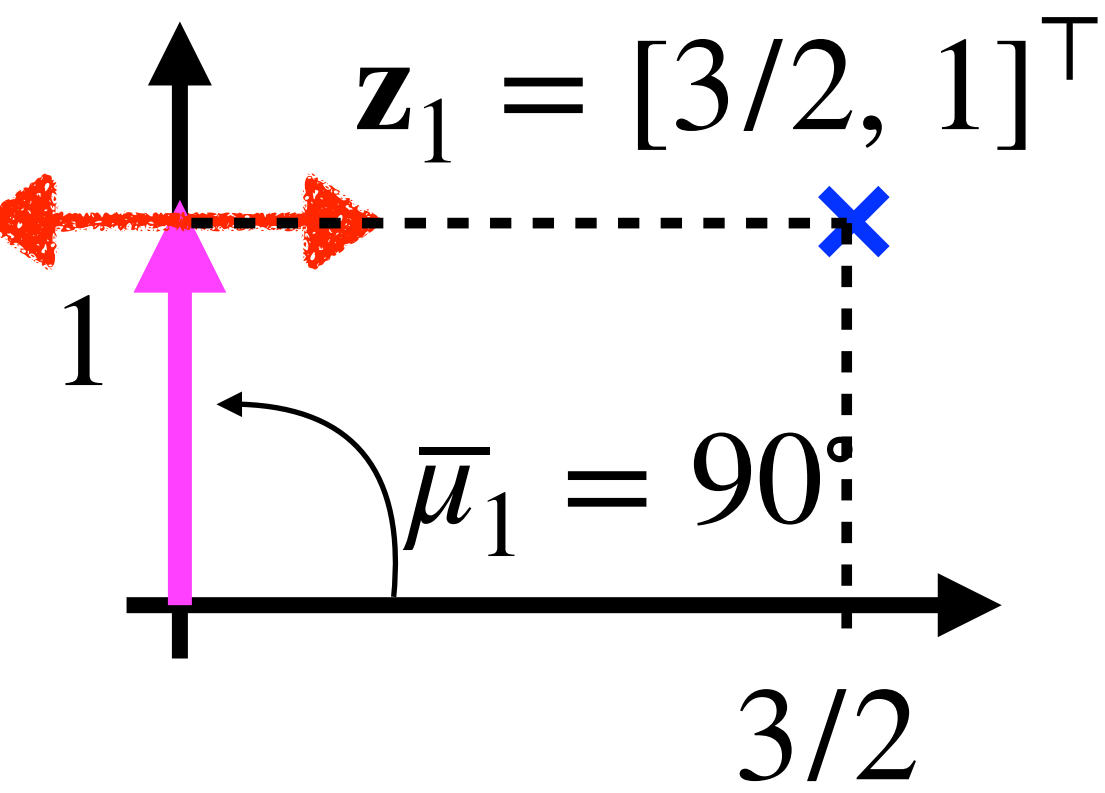
$\mu_1 = \bar{\mu}_1 + \mathbf{K}_1 (\mathbf{z}_1 - h(\bar{\mu}_1)) = ?$

$\Sigma_1 = (\mathbf{I} - \mathbf{K}_1 \mathbf{H}_1) \bar{\Sigma}_1 = ?$



* 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. $\overline{\text{bel}}(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_1; \bar{\mu}_1, \bar{\Sigma}_1)$

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

$$\bar{\Sigma}_1 = \mathbf{G}_1 \Sigma_0 \mathbf{G}_1^\top + \mathbf{R}_1 = 1 \cdot 1 \cdot 1 + 1 = 2$$

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

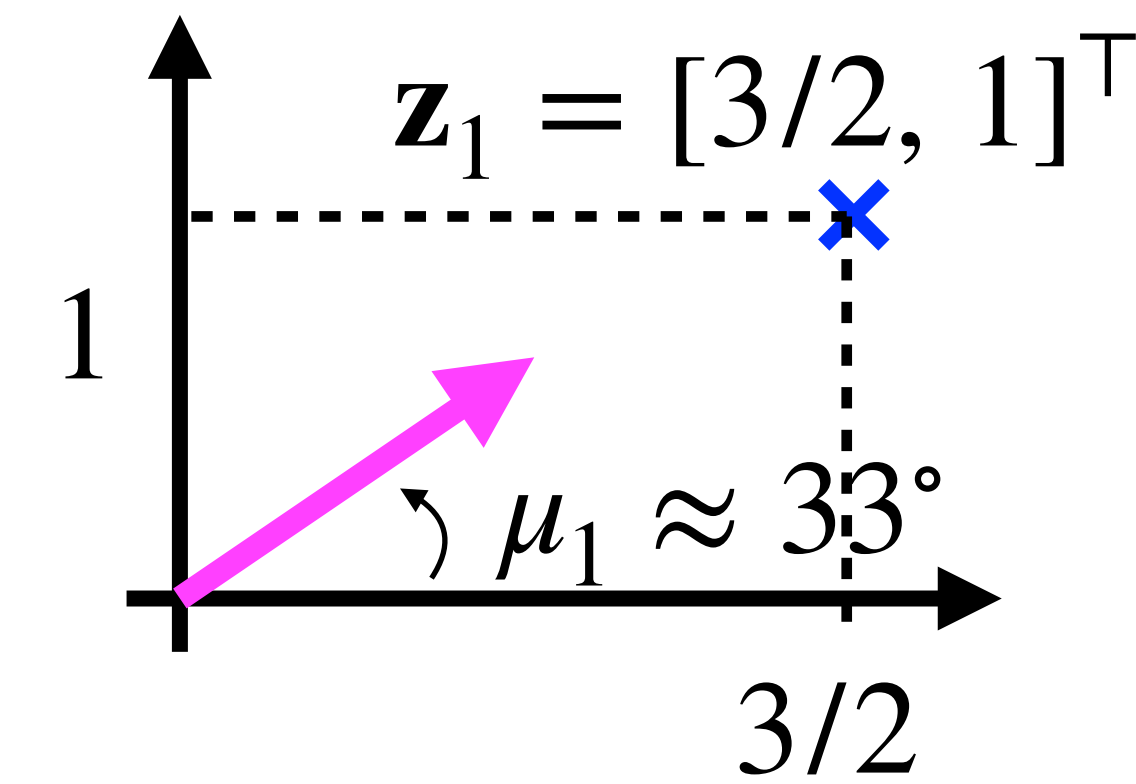
$$\Leftrightarrow h(\mathbf{x}_1) = \begin{bmatrix} \cos(\mathbf{x}_1) \\ \sin(\mathbf{x}_1) \end{bmatrix} \approx \begin{bmatrix} \cos(\bar{\mu}_1) \\ \sin(\bar{\mu}_1) \end{bmatrix} + \begin{bmatrix} -\sin(\bar{\mu}_1) \\ \cos(\bar{\mu}_1) \end{bmatrix} \cdot \mathbf{x}_1 = \underbrace{\begin{bmatrix}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

- Perform measurement step of **EKF**

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

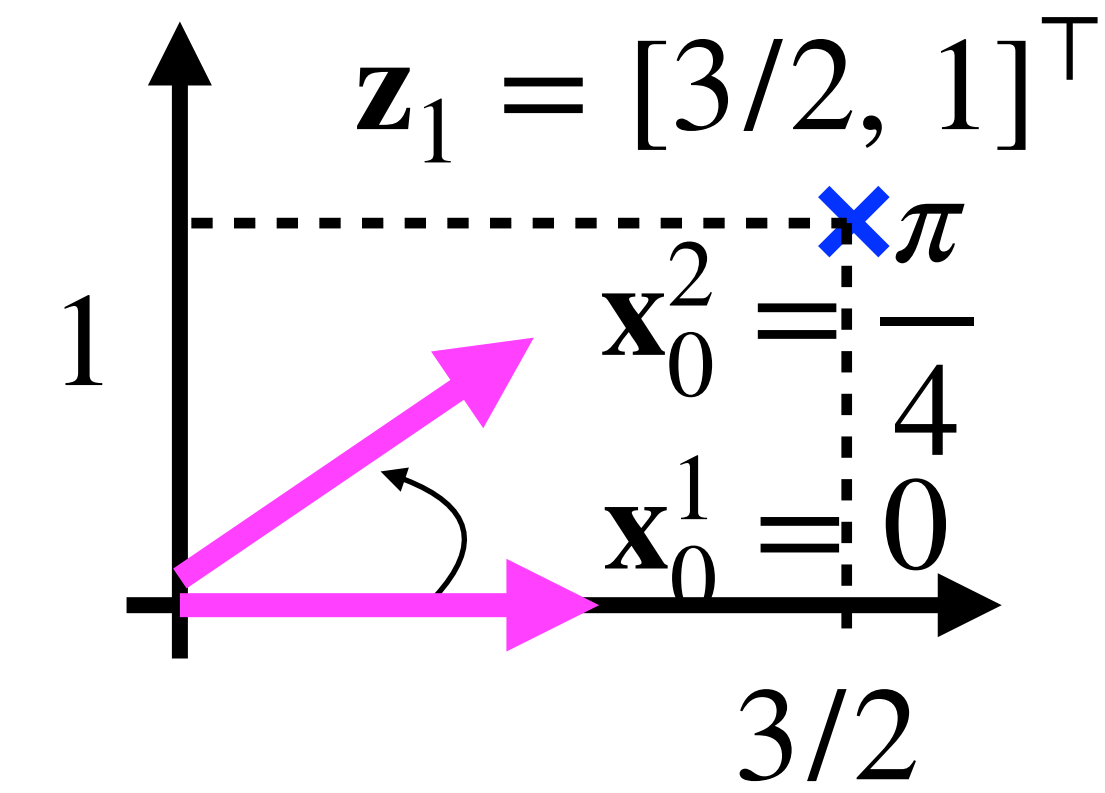
$$\mu_1 = \bar{\mu}_1 + \mathbf{K}_1 (\mathbf{z}_1 - h(\bar{\mu}_1)) = \frac{\pi}{2} - 1 \approx 33^\circ$$

$$\Sigma_1 = (\mathbf{I} - \mathbf{K}_1 \mathbf{H}_1) \bar{\Sigma}_1 = (1 - 2/3) \cdot 2 = 2/3$$



* 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 $\overline{\text{bel}}(\mathbf{x}_1)$ 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:

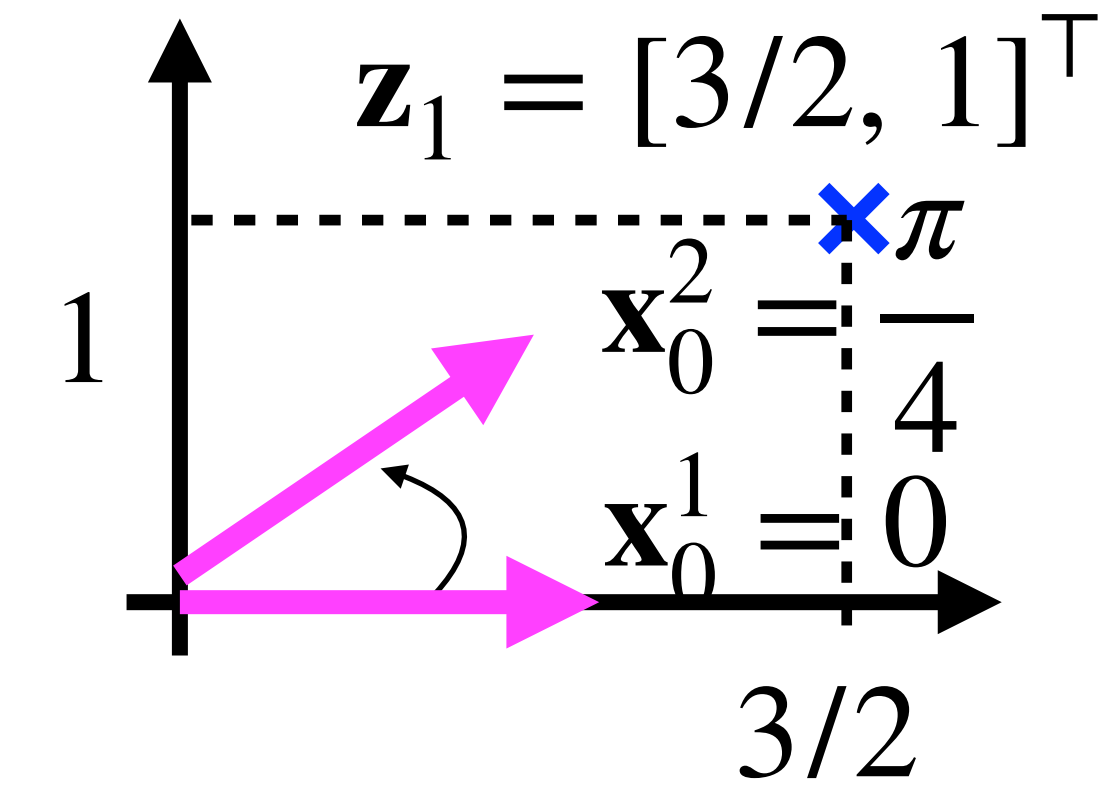
- Update weights of particles to represent $\text{bel}(\mathbf{x}_1)$

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

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

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

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

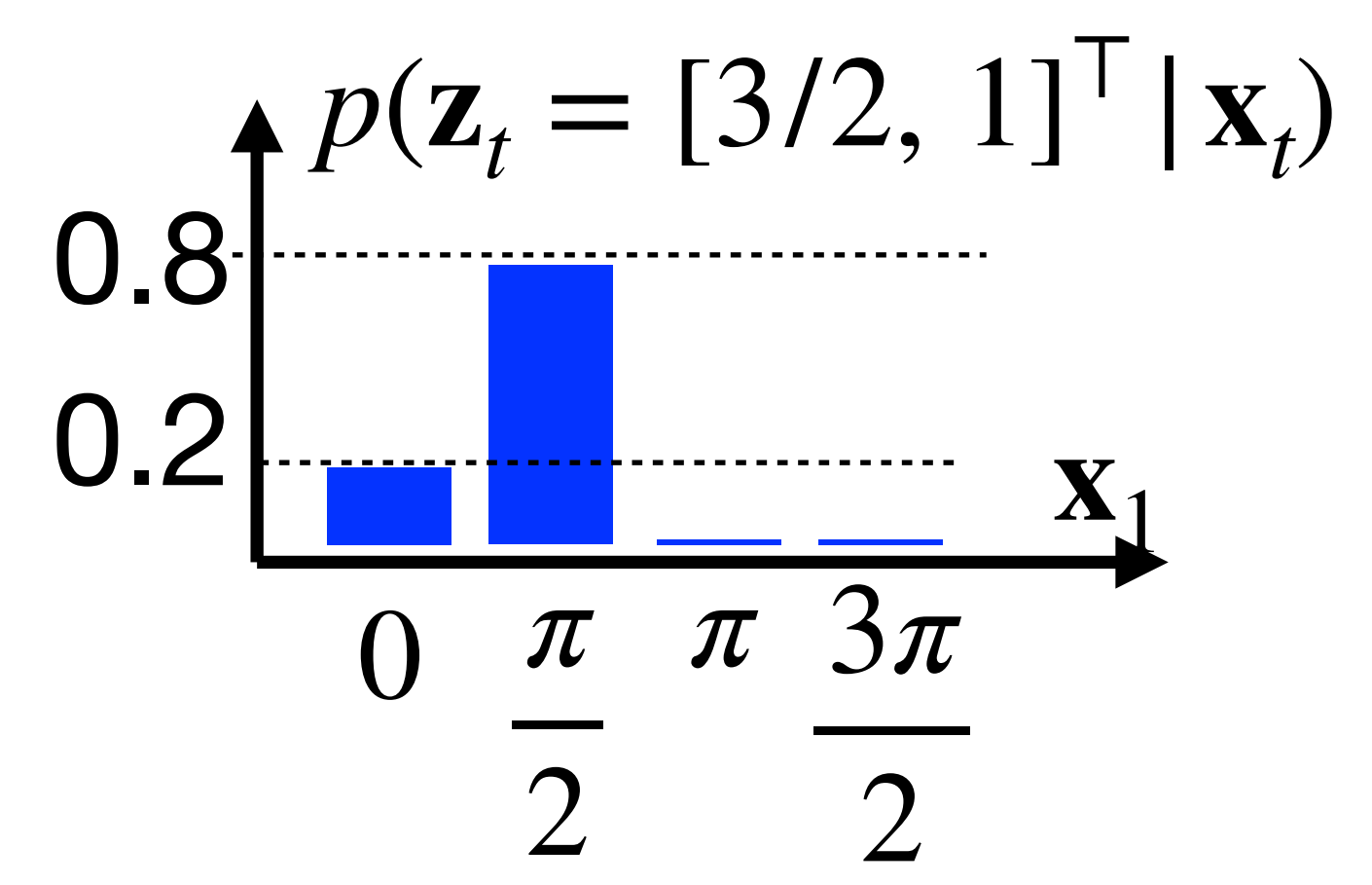
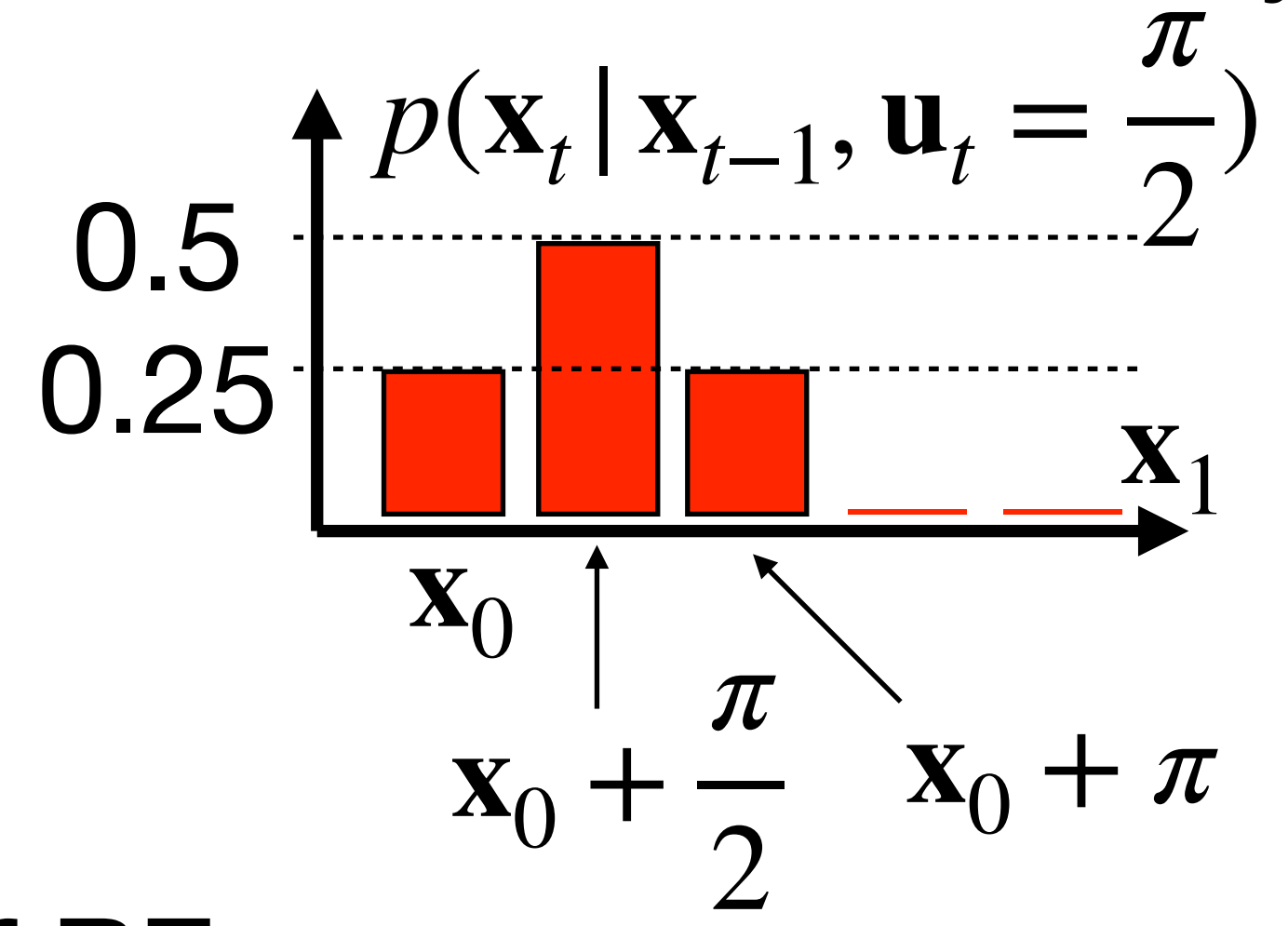
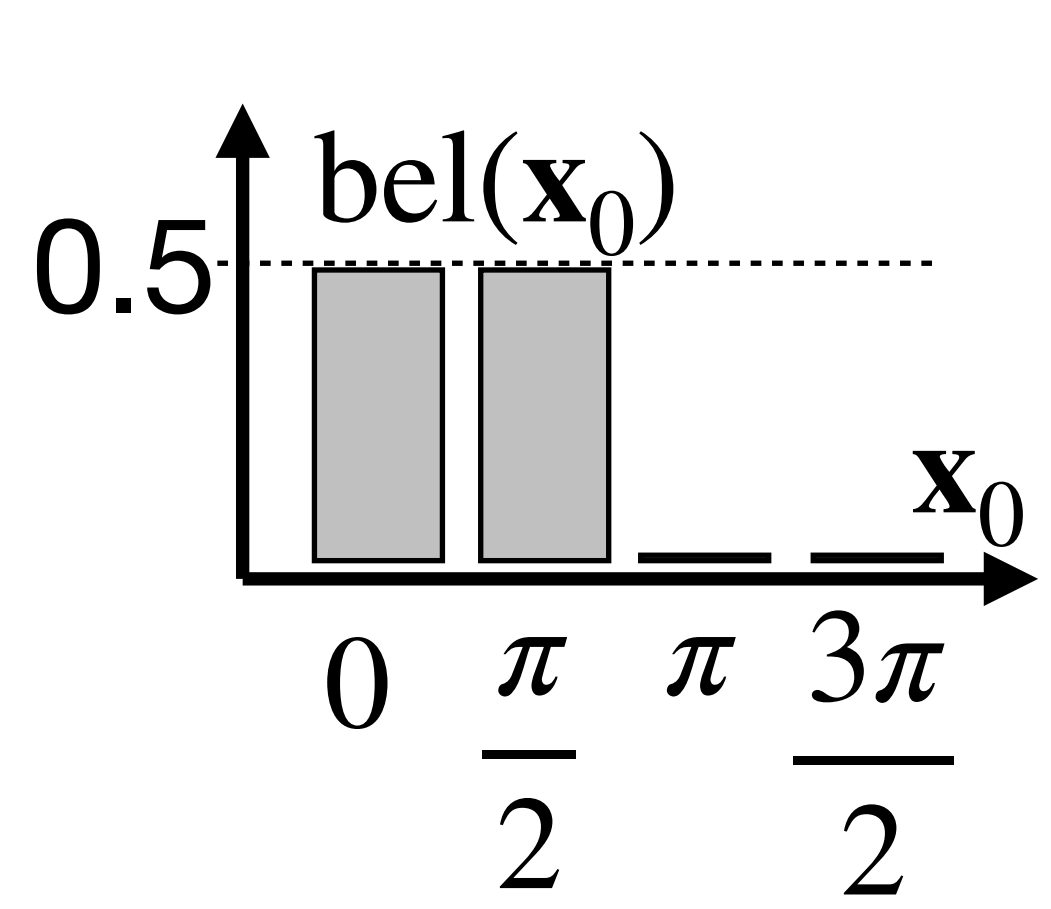
Measurement step of PF:

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

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

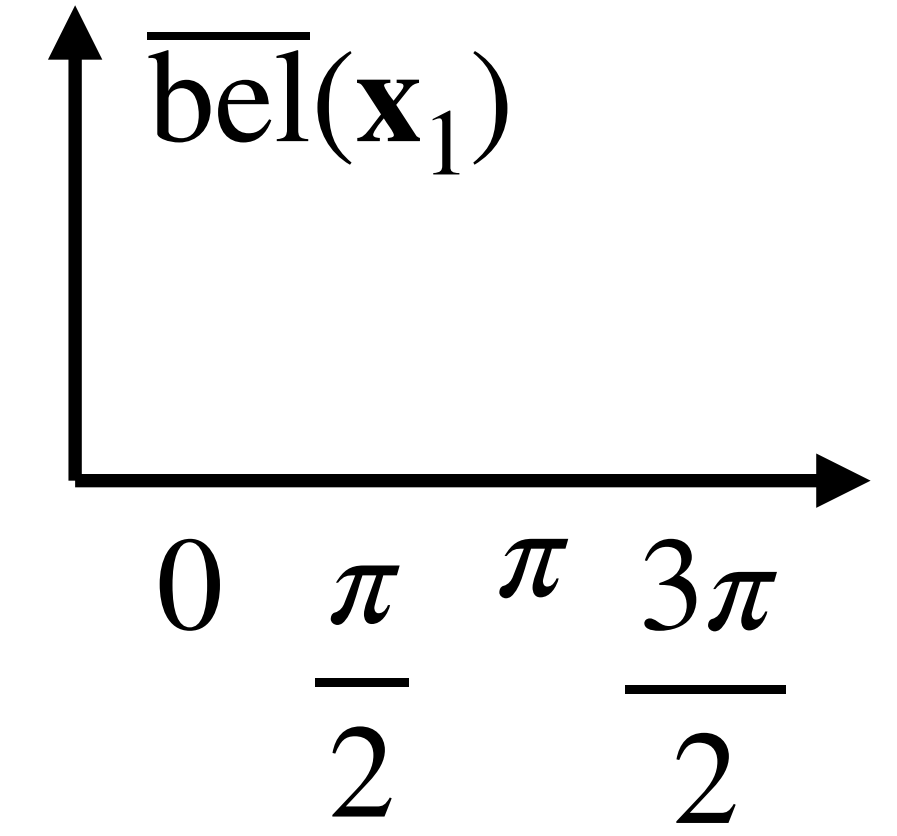
Which particle has a higher chance to survive the resampling? \mathbf{w}_1^1

Discrete Bayes filter



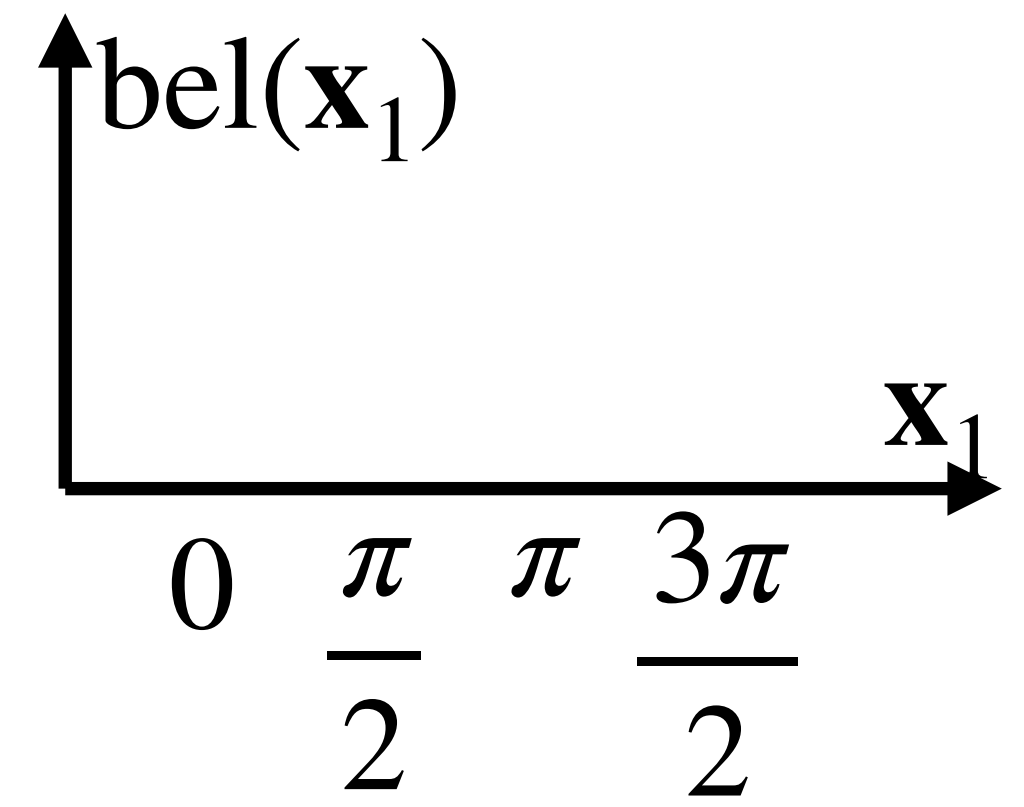
Prediction step of BF:

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

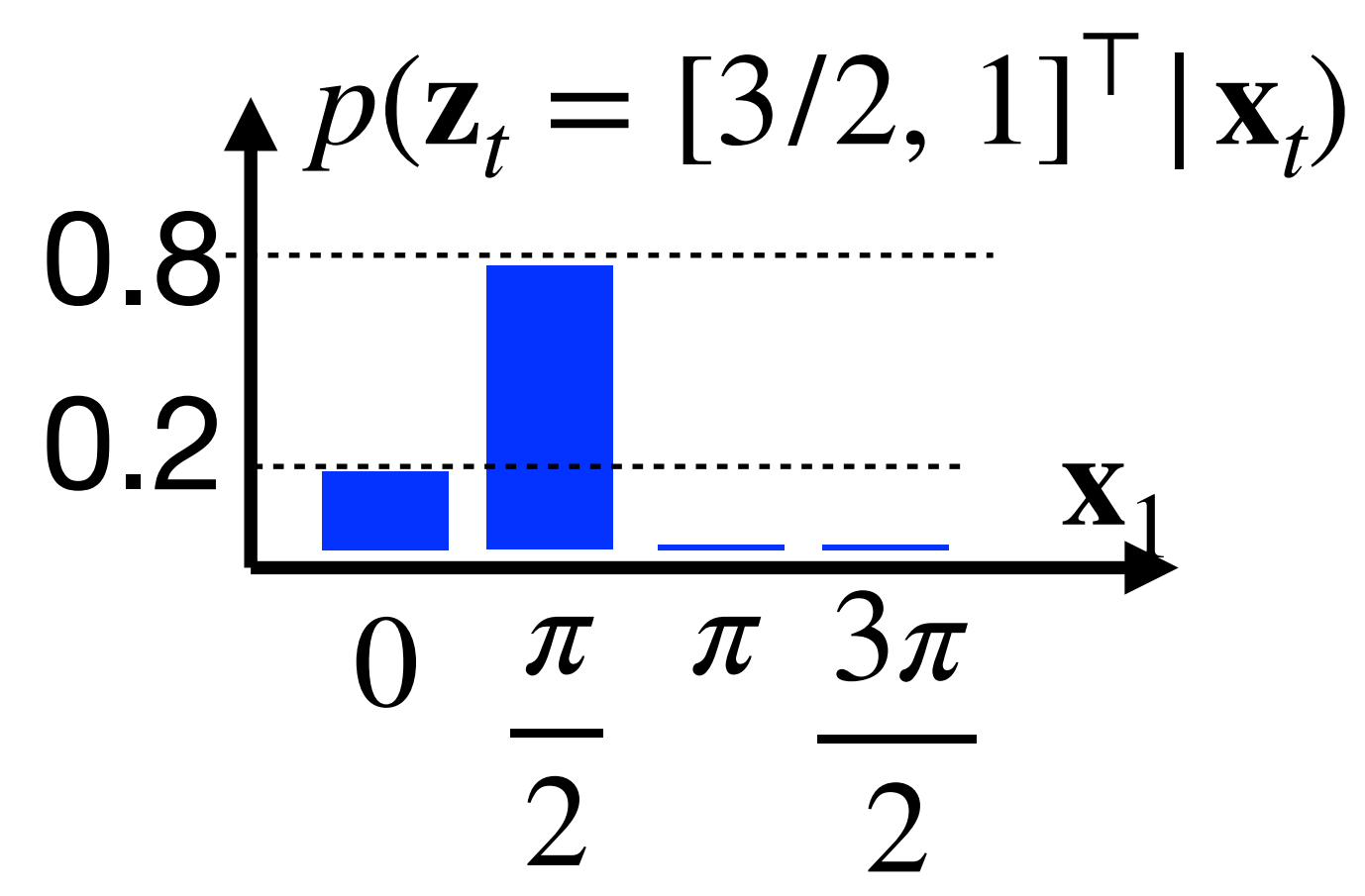
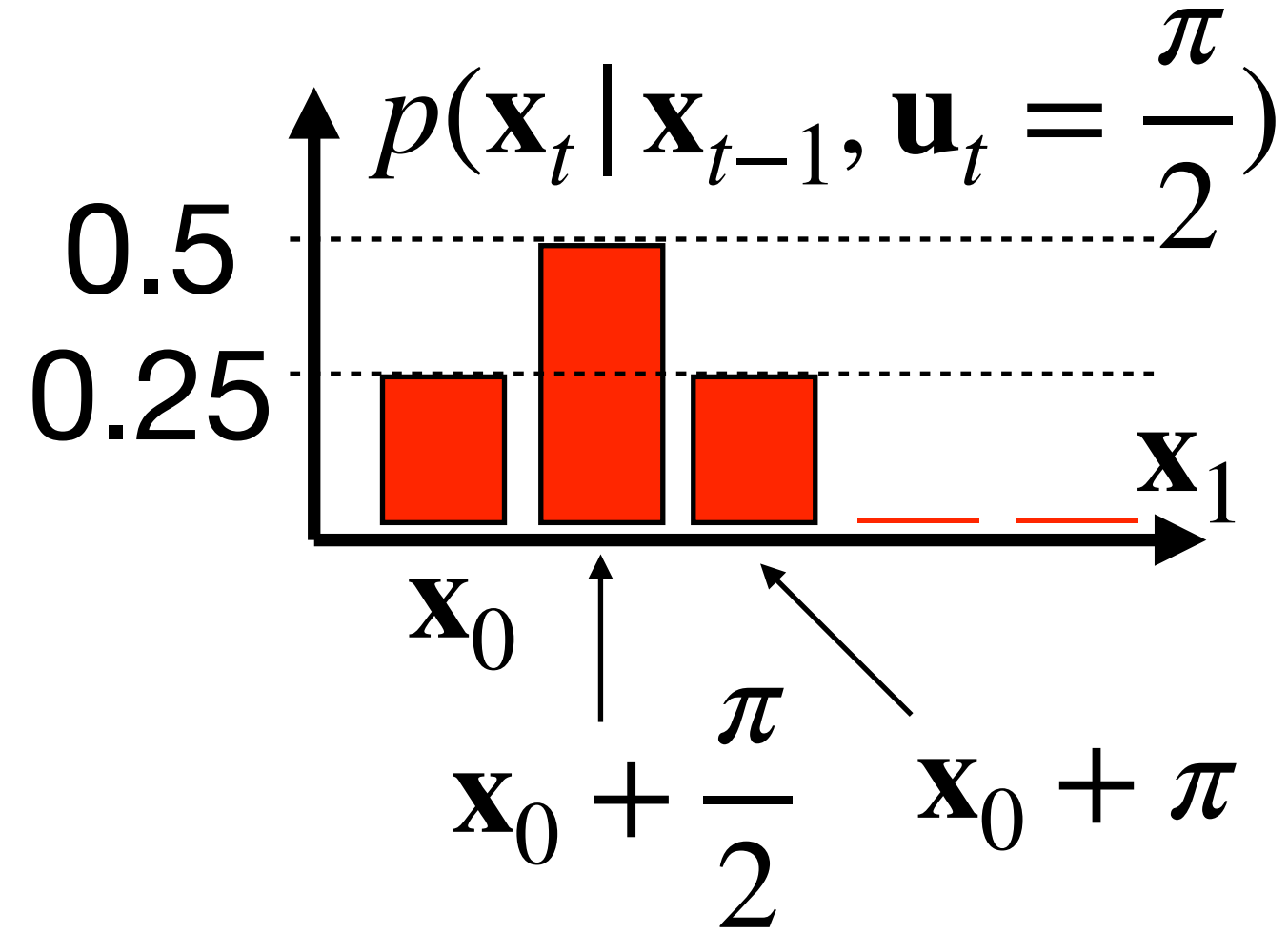
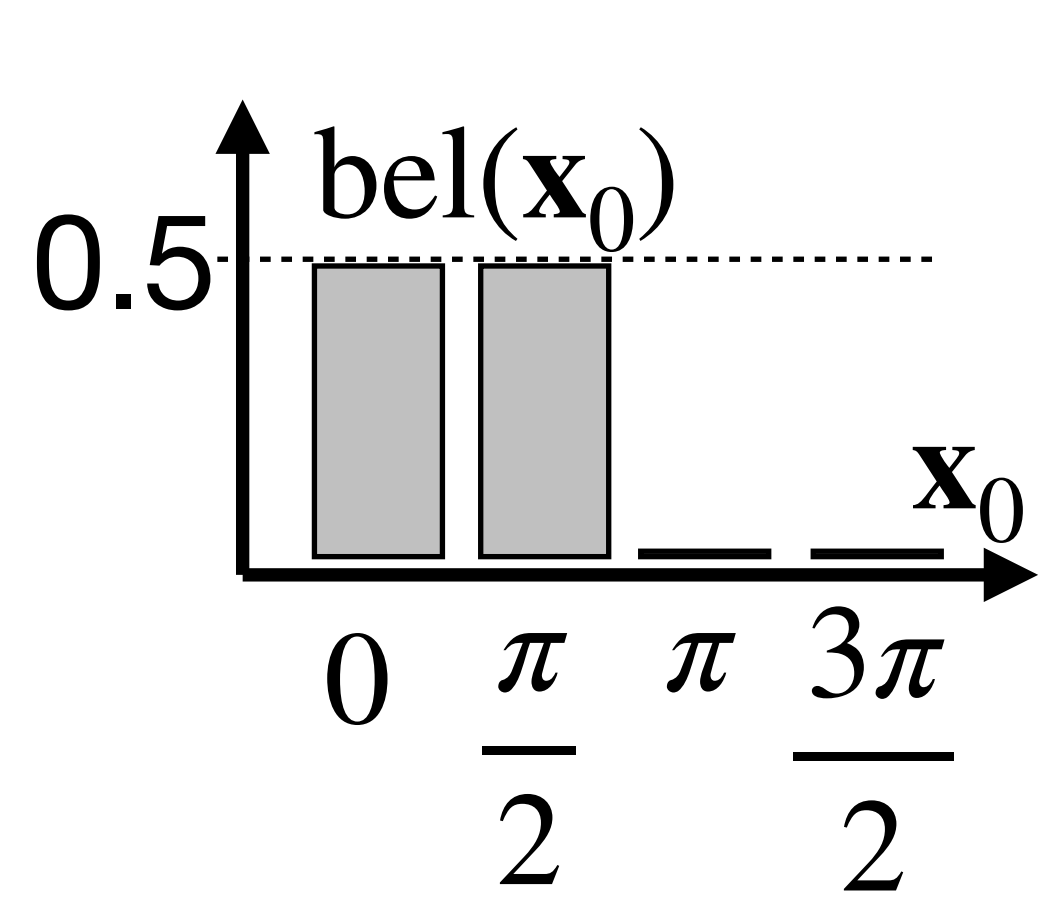


Measurement step of BF:

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

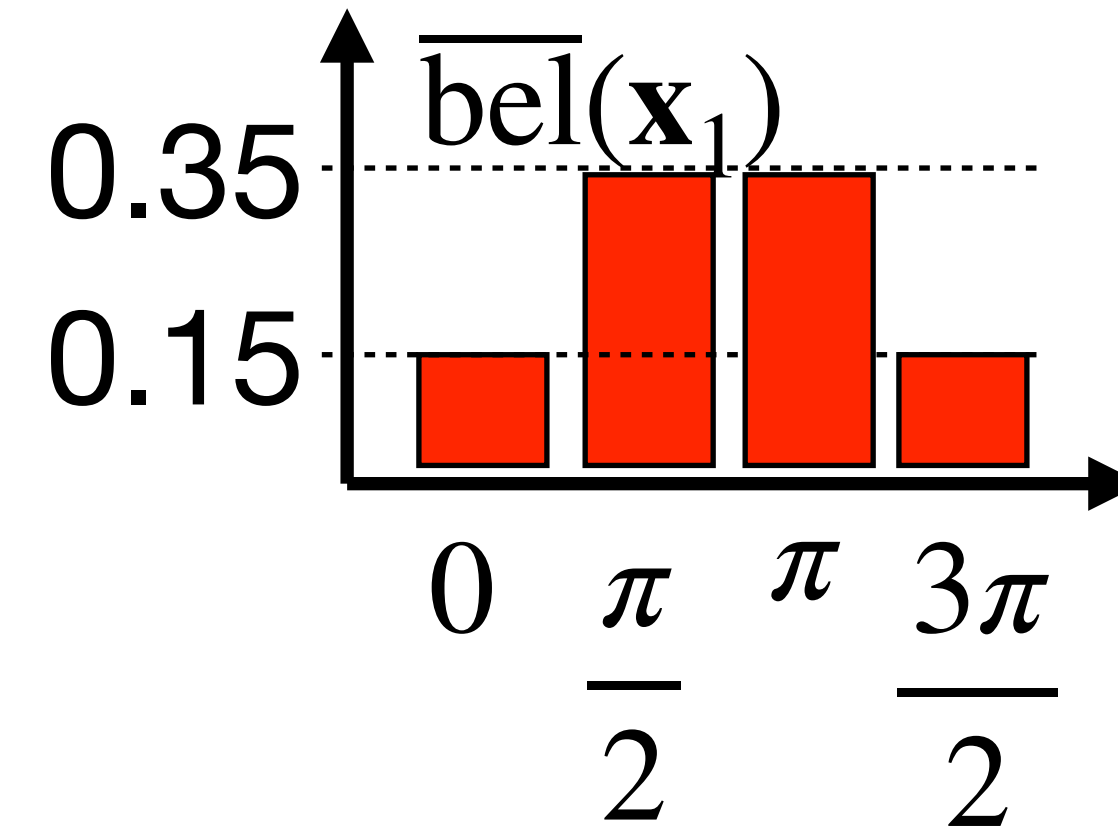


Discrete Bayes filter (solution)



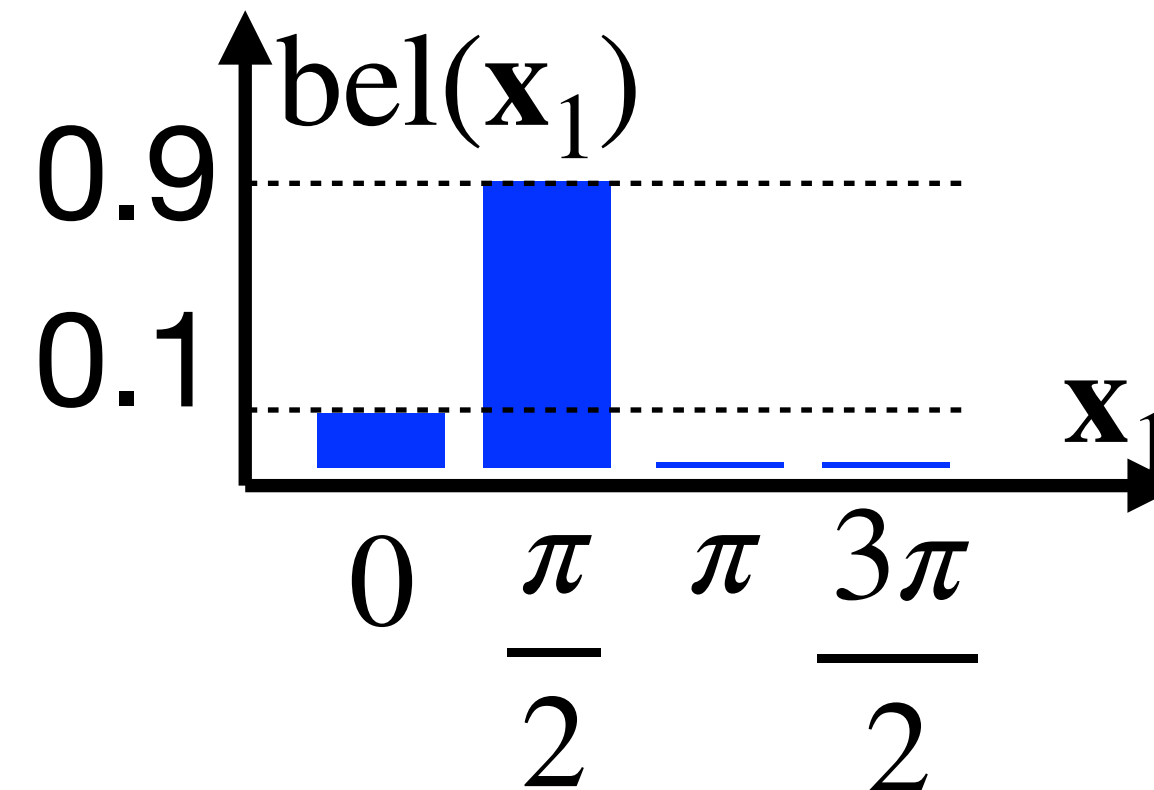
Prediction step of BF:

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



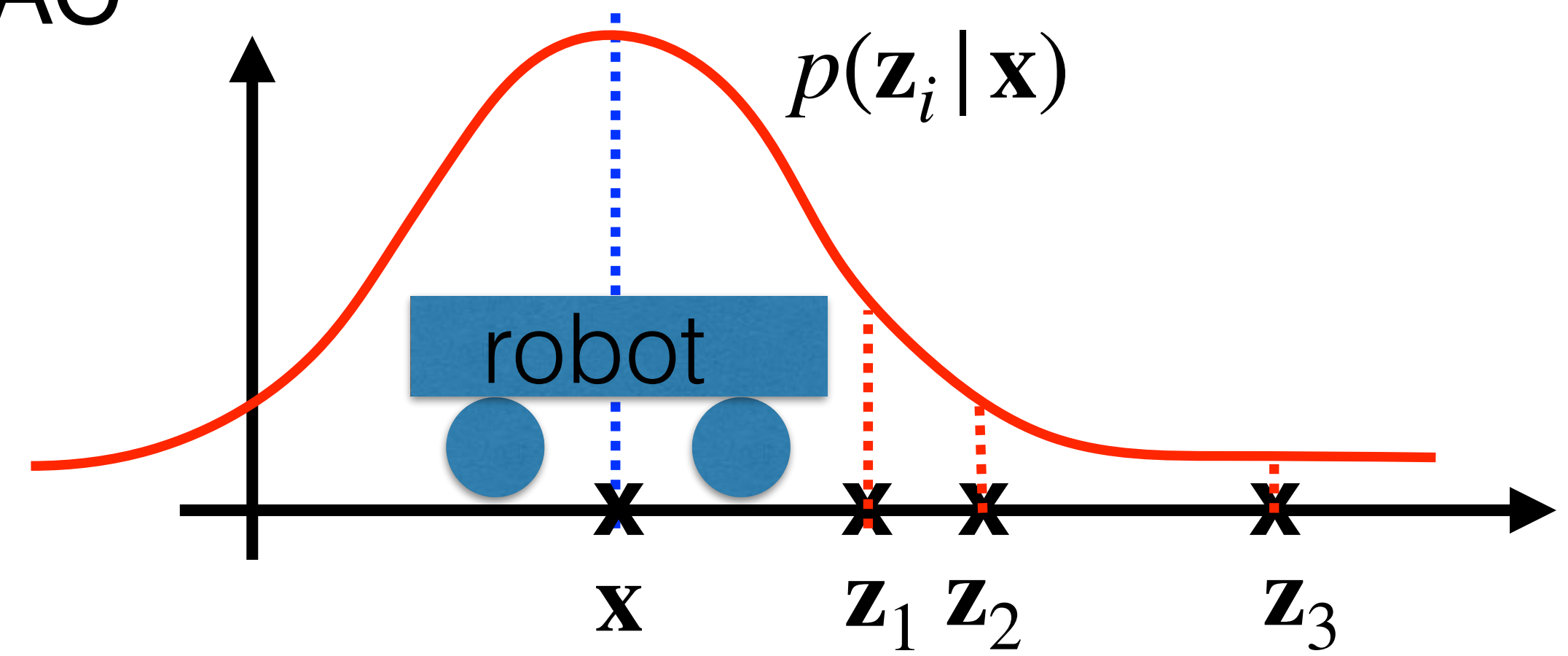
Measurement step of BF:

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



RANSAC

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



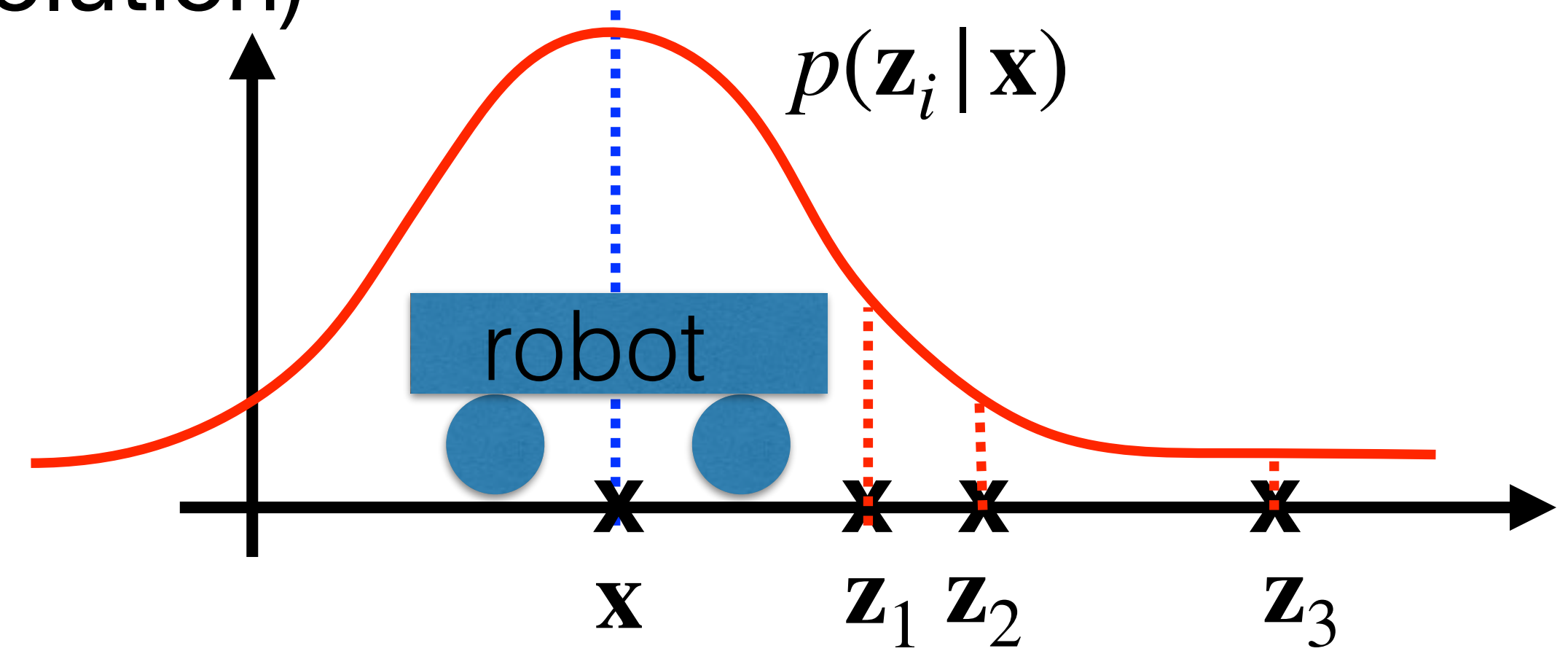
$$\mathbf{z}_1 = 2 \quad \mathbf{z}_2 = 3 \quad \mathbf{z}_3 = 7$$

- What is MLE of state \mathbf{x} under the gaussian noise? $p(\mathbf{z}_i | \mathbf{x}) = \mathcal{N}(\mathbf{z}_i; \mathbf{x}, 1) \quad \mathbf{x}^* = ?$
- What is MLE of state \mathbf{x} under the gaussian noise? $p(\mathbf{z}_i | \mathbf{x}) = \mathcal{N}(\mathbf{z}_i; \mathbf{x}, 100) \quad \mathbf{x}^* = ?$
- What is MLE of state \mathbf{x} under the gaussian noise? $\mathbf{x}^* = ?$
 - $p(\mathbf{z}_1 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_1; \mathbf{x}, 4)$
 - $p(\mathbf{z}_2 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_2; \mathbf{x}, 1)$
 - $p(\mathbf{z}_3 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_3; \mathbf{x}, 1)$

- 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? $p(\mathbf{z}_i | \mathbf{x}) = \mathcal{N}(\mathbf{z}_i; \mathbf{x}, 1) \quad \mathbf{x}^* = 4$

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

○ What is MLE of state \mathbf{x} under the gaussian noise? $p(\mathbf{z}_1 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_1; \mathbf{x}, 4)$

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

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

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

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

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

RANSAC (result depends on tolerance margin and implementation) $\mathbf{x}^* \in \langle 2, 3 \rangle$

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

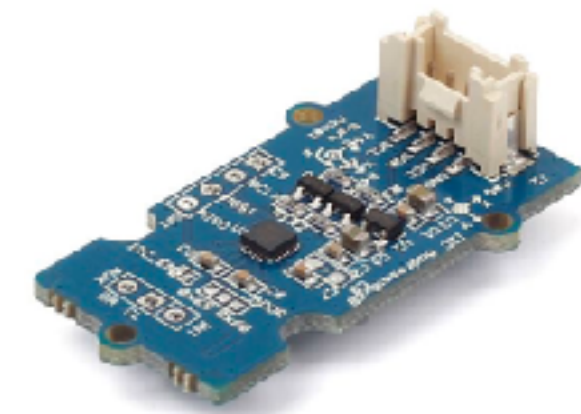
Examples of measurement probabilities

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



GPS

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



IMU

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



Marker detector

Examples of state-transition probabilities



Differential-drive model

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



Balistic trajectory

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

You should also understand reasoning behind this table

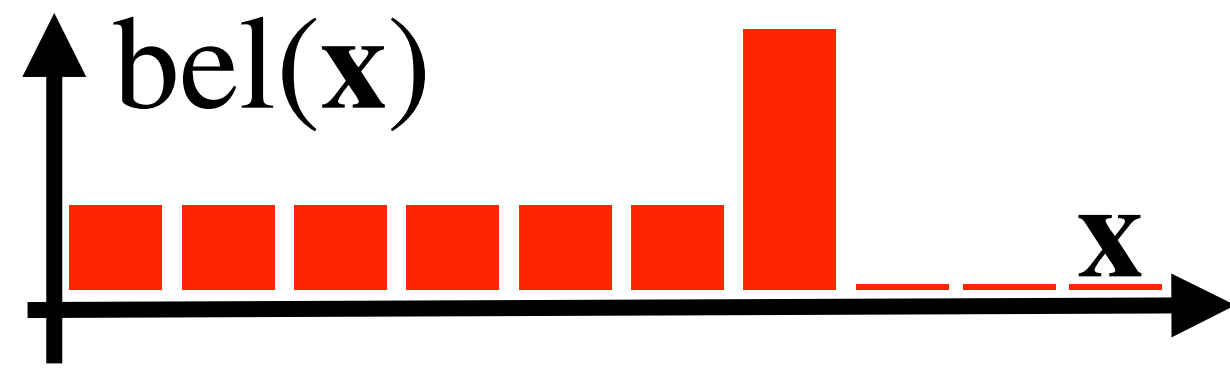
Drawbacks

- course of dimensionality
- spatial discretization

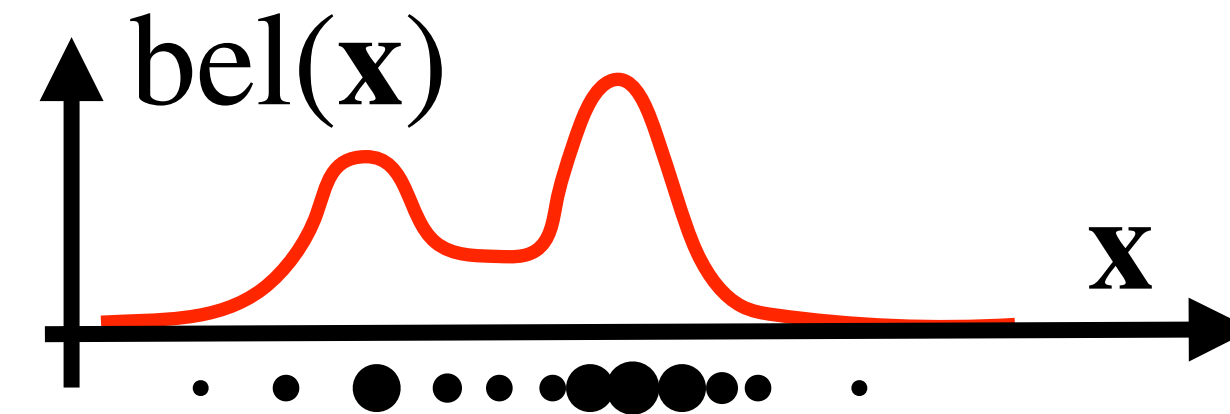
Advantages

- represents arbitrary prob. distribution

Bayes filter



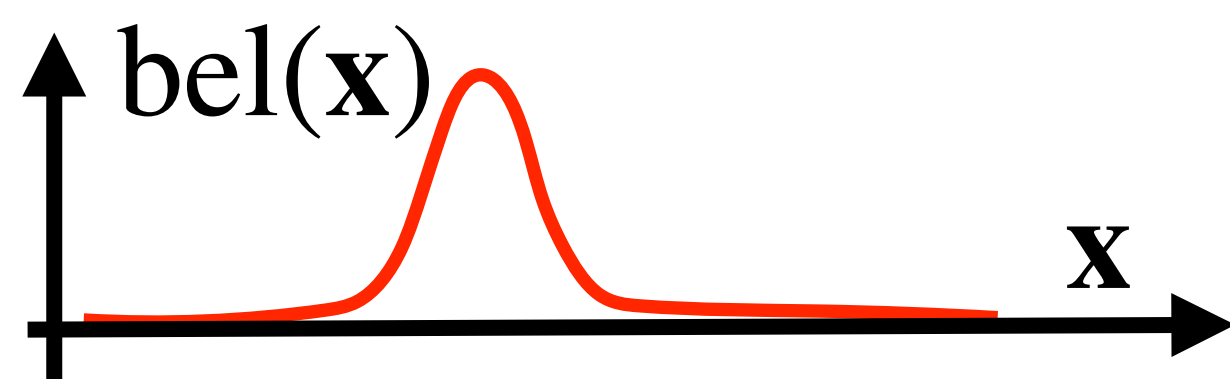
Particle filter



- course of dimensionality
- partical quantization

- represents arbitrary prob. distribution

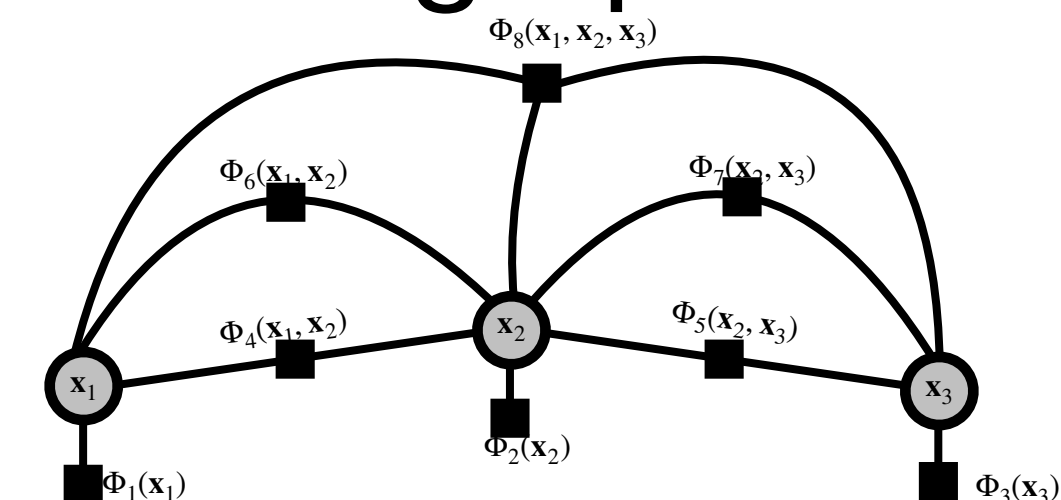
Kalman filter



- 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