Моделирование колесных роботов

Лекция 2. Алгоритмы локализации

Олег Шипитько

ИППИ РАН

MIPT
MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

Москва, 2022

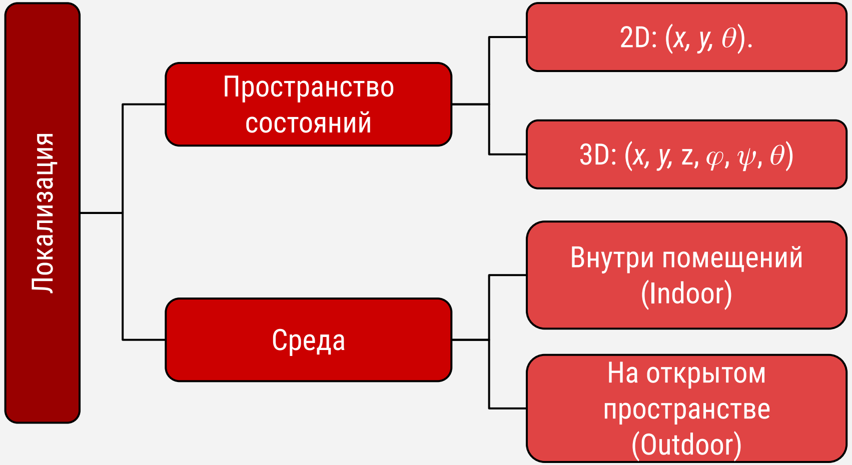
https://qrgo.page.link/eiY1e

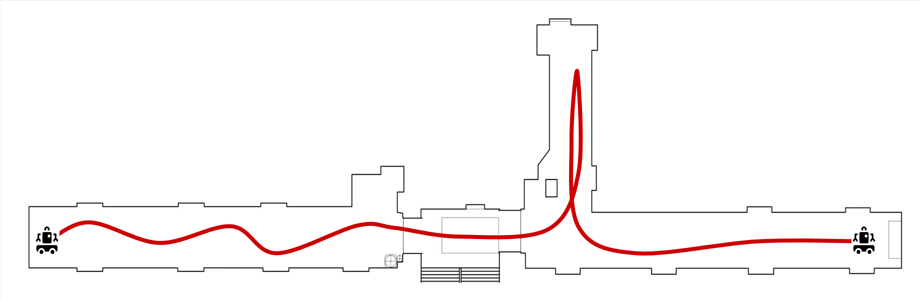
СОДЕРЖАНИЕ ЛЕКЦИИ

1. Определение и постановка задачи локализации
2. Вероятностная постановка
    a. Рекурсивная байесовская оценка
3. Фильтр Калмана
4. Фильтр частиц

ЧТО ТАКОЕ ЛОКАЛИЗАЦИЯ?

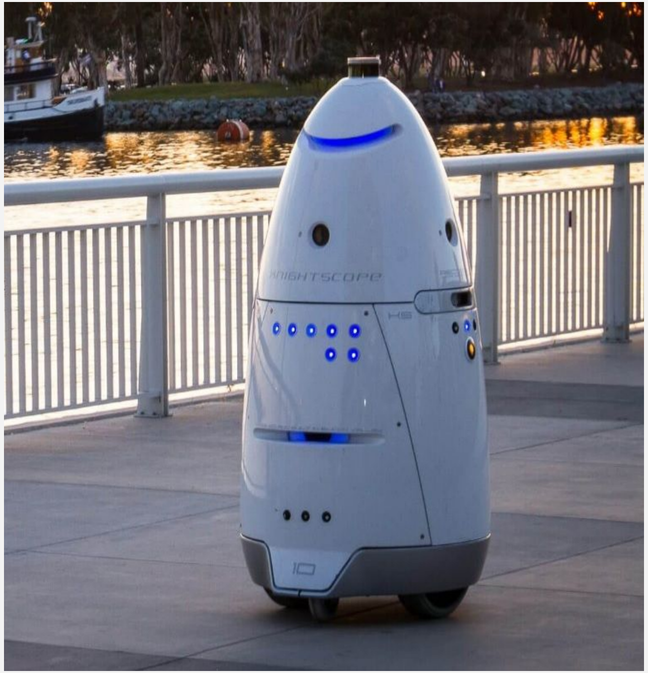
Локализация (в робототехнике) обозначает процесс трекинга (отслеживания) положения робота (позы и ориентации) в пространстве (фиксированной системе координат)*.





* Ziegler, Julius; Henning Lategahn; Markus Schreiber; Christoph G Keller; Carsten Knoppel; Jochen Hipp; Martin Haueis; and Christoph Stiller. 2014. “Video based localization for bertha.” In Intelligent Vehicles Symposium Proceedings, 2014 IEEE, 1231–1238.

ПОЧЕМУ ВАЖНА ЛОКАЛИЗАЦИЯ?



Знание точного положения в пространстве является предпосылкой для точного планирования движения и его осуществления.

ФОРМАЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ ЛОКАЛИЗАЦИИ

Даны:

$\mathbf{x}_{1:t-1}$ – все предыдущие состояния (положение)

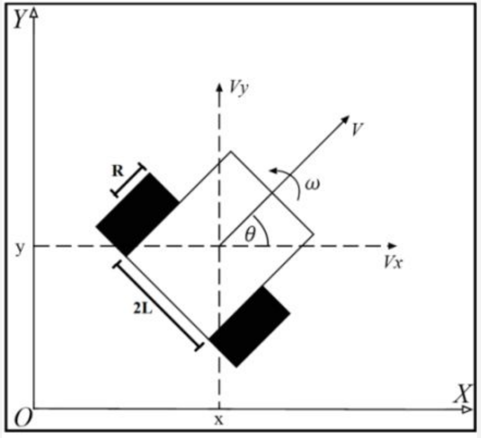
$\mathbf{u}_{1:t}$ – история применяемых сигналов управления

$\mathbf{z}_{1:t}$ – измерения датчиков

map – карта

Определить:

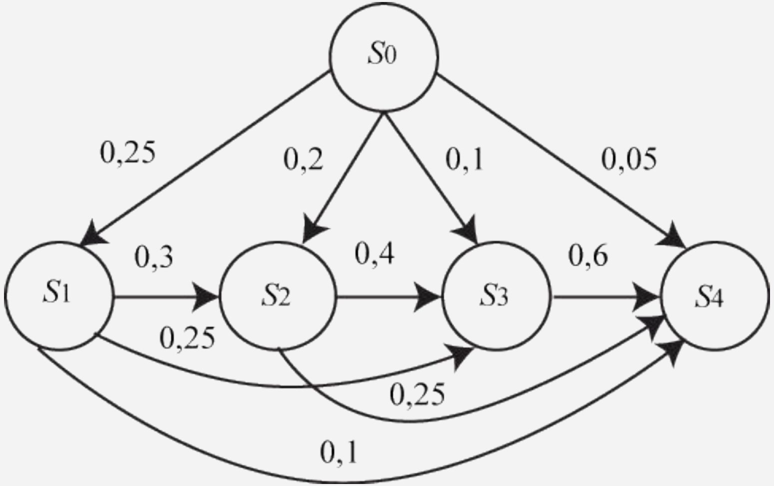
$\mathbf{x}_t$ – текущее положение робота



МАРКОВСКИЕ ПРОЦЕССЫ

Марковский процесс – стохастический процесс который:

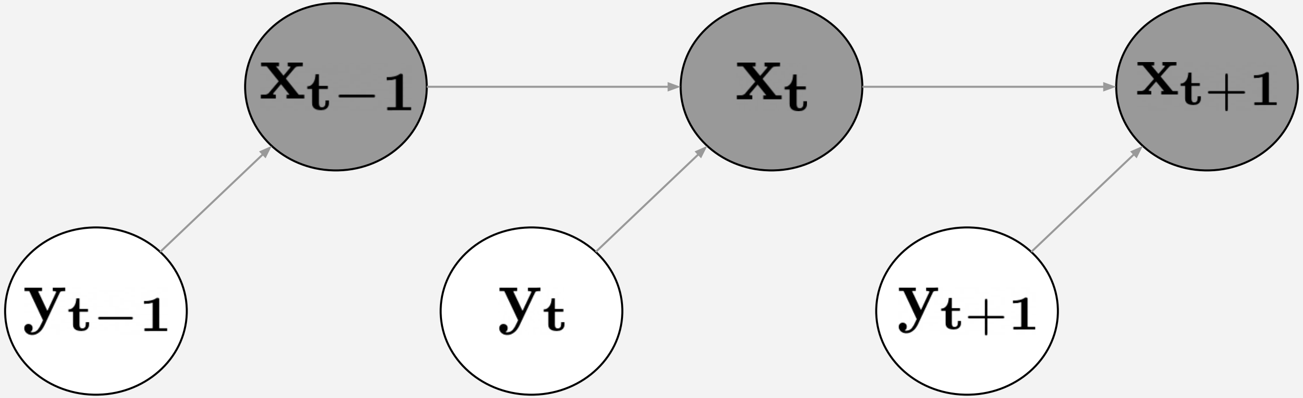
- ❏ после любого заданного значения временного параметра *t* не зависит от эволюции, предшествовавшей *t*, при условии, что значение процесса в этот момент фиксировано
- ❏ «будущее» процесса не зависит от «прошлого» при известном «настоящем»
- ❏ удовлетворяет **марковскому свойству**

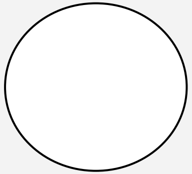


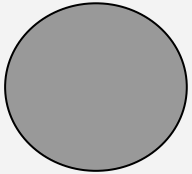
Марковское свойство (для дискретного случая):

$$p(X_n = \mathbf{x_n} | X_{n-1} = \mathbf{x_{n-1}}, X_{n-2} = \mathbf{x_{n-2}}, ..., X_0 = \mathbf{x_0}) = p(X_n = \mathbf{x_n} | X_{n-1} = \mathbf{x_{n-1}})$$

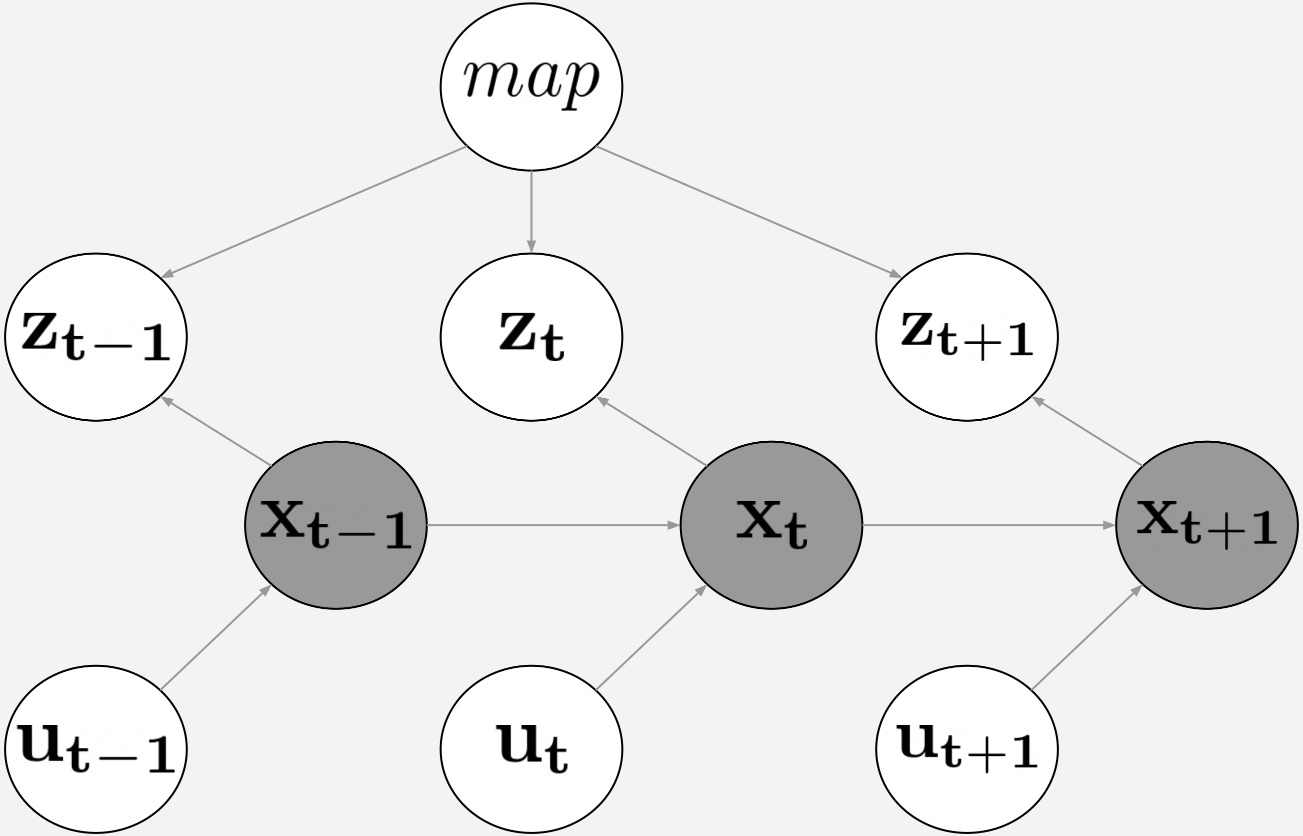
СКРЫТЫЕ МАРКОВСКИЕ ПРОЦЕССЫ



 — наблюдаемый процесс

 — скрытый процесс

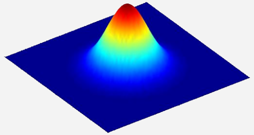
ЛОКАЛИЗАЦИЯ КАК скрытый МАРКОВСКИЙ ПРОЦЕСС



ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

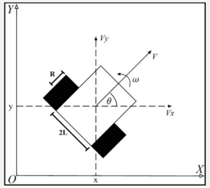
$$p(\mathbf{x_t} | map, \mathbf{z_{1:t}}, \mathbf{u_{1:t}}) = ?$$

Вероятностное распределение



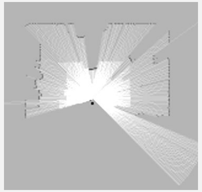
Поза робота

$$\mathbf{x_t} = (x_t, y_t, \theta_t)$$



Карта

Измерение



Вектор управляющих сигналов

$$\mathbf{u_t} = (\Delta x_t, \Delta y_t, \Delta\theta_t)$$

$\Delta x_t, \Delta y_t, \Delta\theta_t$ - изменение положения за текущий шаг времени

Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic robotics*. MIT press.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}} p(\mathbf{x}_t | map, \mathbf{z}_{\mathbf{1:t}}, \mathbf{u}_{\mathbf{1:t}})$$

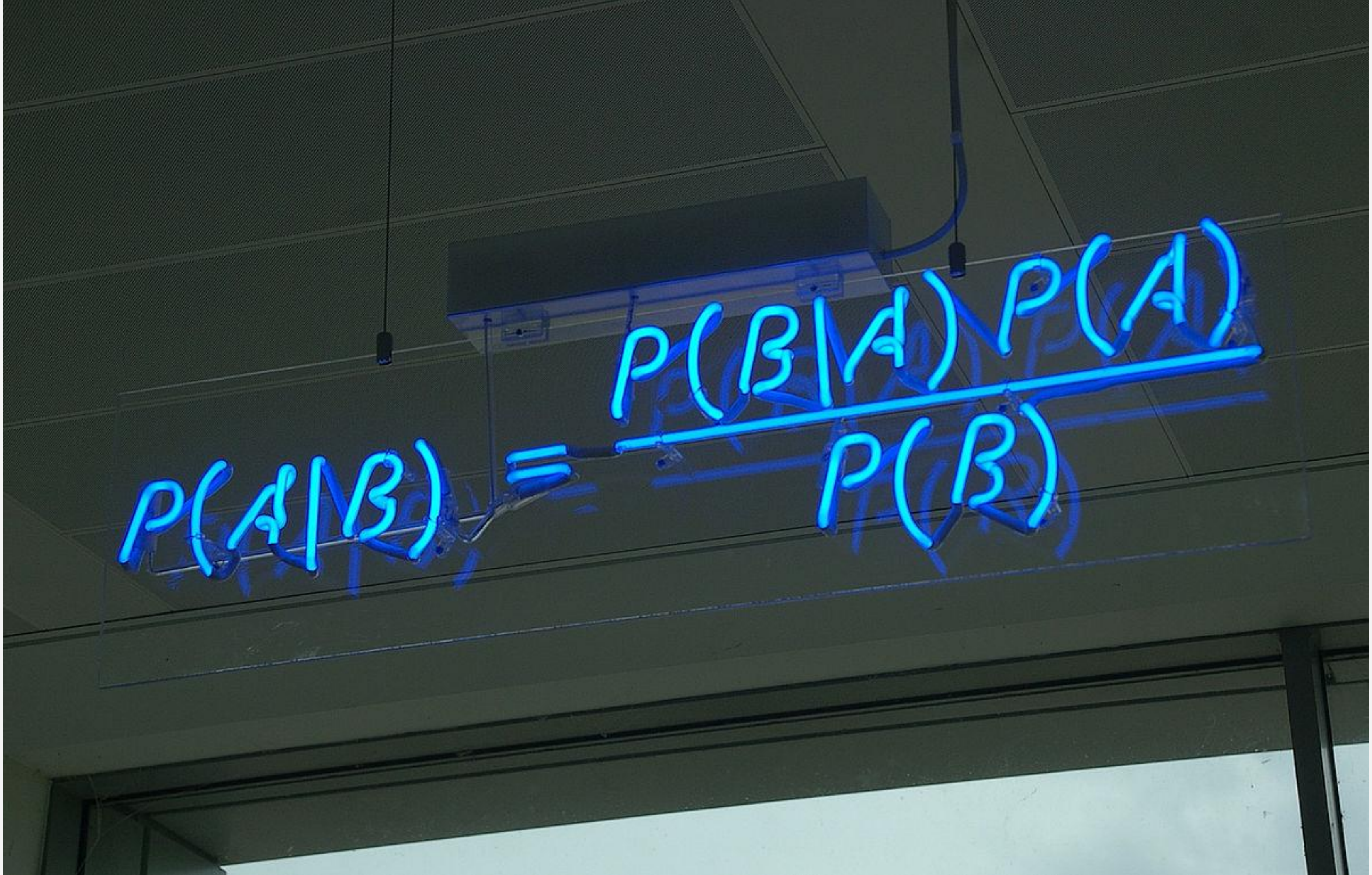
Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic robotics*. MIT press.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

Проблема локализации может быть сформулирована следующим образом: имея вектор всех последовательных измерений сенсоров $\mathbf{z}_{1:t} = \mathbf{z}_0 \ldots \mathbf{z}_t$, и управляющих сигналов $\boldsymbol{u}_{1:t} = \boldsymbol{u}_0 \ldots \boldsymbol{u}_t$, необходимо восстановить апостериорное распределение $\boldsymbol{x}_t$ в любой заданный момент времени t.

(Sebastian Thrun 2002)

Thrun, Sebastian. 2002. “Particle filters in robotics.” In Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, 511–518.


$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

ТЕОРЕМА БАЙЕСА

$$p(a|b) = \frac{p(b|a)\,p(a)}{p(b)}$$

Апостериорная вероятность
Как вероятно, что наша гипотеза a правдива с учетом полученных данных b

Правдоподобие
Насколько вероятен исход b, при условии, что наша гипотеза a верна

Априорная вероятность
Насколько вероятна была гипотеза a до получения новых данных b

Частная (маргинальная) вероятность
Вероятность исхода b, не зависящая от a

ФОРМУЛА ПОЛНОЙ ВЕРОЯТНОСТИ

$$p(b) = \sum_{i=1}^{N} p(a_i)p(b|a_i)$$

ТЕОРЕМА БАЙЕСА
ПРИМЕР

Один из трех стрелков вызывается на линию огня и производит два выстрела. Вероятность попадания в мишень при одном выстреле для первого стрелка равна **0.3**, для второго – **0.5**, для третьего – **0.8**. Мишень не поражена. Найти вероятность того, что выстрелы произведены первым стрелком.

A_1 – на линию огня вызван первый стрелок

A_2 – на линию огня вызван второй стрелок

A_3 – на линию огня вызван третий стрелок

Так как вызов на линию огня любого стрелка равновозможен, то

$$p(A_1) = p(A_2) = p(A_3) = \frac{1}{3}$$

Источник: https://www.matburo.ru/

ТЕОРЕМА БАЙЕСА
ПРИМЕР

В результате опыта наблюдалось событие B - после произведенных выстрелов мишень не поражена. Условные вероятности этого события при сделанных гипотезах равны:

$$p(B|A_1) = (1 - 0.3) * (1 - 0.3) = 0.49$$
$$p(B|A_2) = (1 - 0.5) * (1 - 0.5) = 0.25$$
$$p(B|A_3) = (1 - 0.8) * (1 - 0.8) = 0.04$$

По формуле Байеса находим вероятность гипотезы A_1 после опыта:

$$p(A_1|B) = \frac{0.49 * 1/3}{0.49 * 1/3 + 0.25 * 1/3 + 0.4 * 1/3} = 0.628$$

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\mathrm{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$$

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\mathrm{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) =$$

Теорема Байеса

$$= \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$$

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\mathrm{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

Марковское свойство $$= \eta p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$$

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\mathrm{bel}(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

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

Формула полной вероятности

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\mathrm{bel}(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

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

Марковское свойство $$= \eta p(\mathbf{z}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1}$$

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\begin{aligned}
\text{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \\
= \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \\
= \eta p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = \\
= \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} = \\
= \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} = \\
= \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1}
\end{aligned}$$

Марковское свойство

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА

$$\mathrm{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

$$= \eta p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

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

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

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

Рекурсивный член $$= \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \mathrm{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$$

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА ПОЗЫ

$$p(\mathbf{x_t}|map, \mathbf{z_t}, \mathbf{u_t}) = C \cdot p(\mathbf{z_t}|\mathbf{x_t}, map) \int_S p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}}) p(\mathbf{x_{t-1}}|map, \mathbf{z_{t-1}}, \mathbf{u_{t-1}}) d\mathbf{x_{t-1}}$$

C – коэффициент нормализации

S – пространство поз робота

$p(\mathbf{z_t}|\mathbf{x_t}, map)$ – модель измерения

$p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}})$ – модель движения

$p(\mathbf{x_{t-1}}|map, \mathbf{z_{t-1}}, \mathbf{u_{t-1}})$ – предыдущее состояние системы (поза робота)

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА ПОЗЫ

$$p(\mathbf{x_t}|map, \mathbf{z_t}, \mathbf{u_t}) = C \cdot p(\mathbf{z_t}|\mathbf{x_t}, map) \int_S p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}}) p(\mathbf{x_{t-1}}|map, \mathbf{z_{t-1}}, \mathbf{u_{t-1}}) d\mathbf{x_{t-1}}$$

C – коэффициент нормализации

S – пространство поз робота

$p(\mathbf{z_t}|\mathbf{x_t}, map)$ – модель измерения

$p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}})$ – модель движения

$p(\mathbf{x_{t-1}}|map, \mathbf{z_{t-1}}, \mathbf{u_{t-1}})$ – предыдущее состояние системы (поза робота)

РЕКУРСИВНАЯ БАЙЕСОВСКАЯ ОЦЕНКА ПОЗЫ

$$p(\mathbf{x_t}|map, \mathbf{z_t}, \mathbf{u_t}) = C \cdot \underbrace{p(\mathbf{z_t}|\mathbf{x_t}, map)}_{\text{Коррекция}} \int_S \underbrace{p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}})p(\mathbf{x_{t-1}}|map, \mathbf{z_{t-1}}, \mathbf{u_{t-1}})}_{\text{Предсказание}} d\mathbf{x_{t-1}}$$

C – коэффициент нормализации

S – пространство поз робота

$p(\mathbf{z_t}|\mathbf{x_t}, map)$ – модель измерения

$p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}})$ – модель движения

$p(\mathbf{x_{t-1}}|map, \mathbf{z_{t-1}}, \mathbf{u_{t-1}})$ – предыдущее состояние системы (поза робота)

РЕАЛИЗАЦИЯ РЕКУРСИВНОЙ БАЙЕСОВСКОЙ ОЦЕНКИ

- Рекурсивная байесовская оценка – фреймворк
- Существует множество алгоритмов-реализаций:
  - Линейный и нелинейные модели измерения и движения
  - Нормальные и произвольные (мультимодальные) распределения ошибок
  - Параметрические и непараметрические алгоритмы
  - ….

- Фильтр Калмана
- Информационный фильтр
- Гистограммный фильтр
- Фильтр частиц
- ….

РЕАЛИЗАЦИЯ РЕКУРСИВНОЙ БАЙЕСОВСКОЙ ОЦЕНКИ

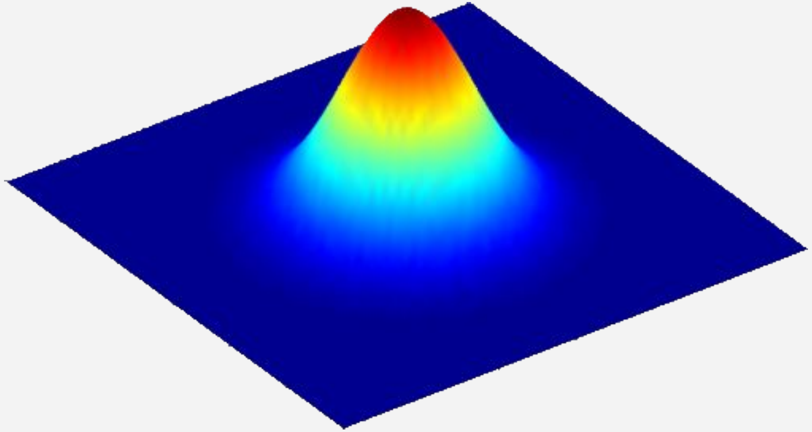
- Рекурсивная байесовская оценка – фреймворк
- Существует множество алгоритмов-реализаций:
  - Линейный и нелинейные модели измерения и движения
  - Нормальные и произвольные (мультимодальные) распределения ошибок
  - Параметрические и непараметрические алгоритмы
  - ….

- **Фильтр Калмана**
- Информационный фильтр
- Гистограммный фильтр
- **Фильтр частиц**
- ….

ФИЛЬТР КАЛМАНА

Доказуемо оптимален в случае:

- Линейных моделей движения и измерения (наблюдения)
- Нормально распределенных ошибок движения и измерения



ЛИНЕЙНАЯ СИСТЕМА

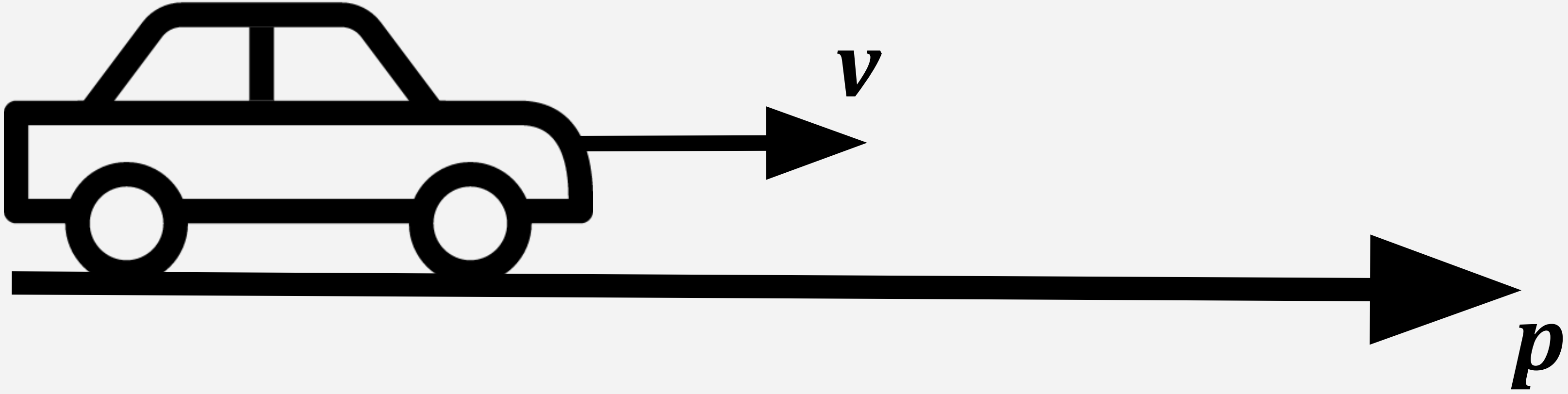
Система линейна если:

- отклик системы на сумму воздействий равен сумме откликов на каждое воздействие

Необходимые условия линейности:

- **Гомогенность** – при изменении амплитуды входного сигнала в k раз также в k раз изменяется и амплитуда выходного сигнала
- **Аддитивность** – при суммировании входных сигналов результирующий сигнал на выходе будет равен сумме реакций от исходных сигналов
- **Инвариантность** – смещение входного сигнала во времени вызывает аналогичное смещение выходного сигнала.
- **Статическая линейность** – основные законы в системе описываются линейными уравнениями
- **Гармоническая верность** – если на вход системы подать синусоидальный сигнал, то на выходе будет сигнал той же частоты

ФИЛЬТР КАЛМАНА
ПРИМЕР



ФИЛЬТР КАЛМАНА. СОСТОЯНИЕ СИСТЕМЫ

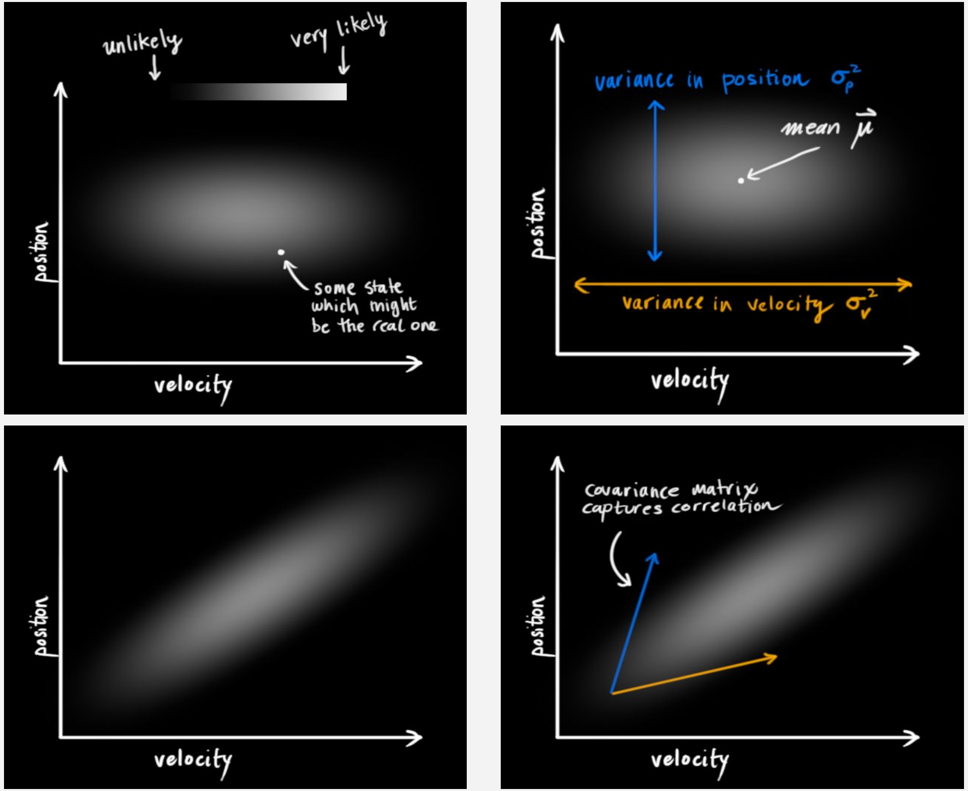
Наше знание о текущем состоянии системы описывается 2 элементами:

1. Оценкой текущего вектора состояния:

$$\hat{\mathbf{x}}_{\mathbf{t}} = \begin{bmatrix} p \\ v \end{bmatrix}$$

2. Оценкой матрицы ковариации:

$$\hat{\mathbf{\Sigma}}_{\mathbf{t}} = \begin{bmatrix} \sigma_{pp} & \sigma_{pv} \\ \sigma_{vp} & \sigma_{vv} \end{bmatrix} = \begin{bmatrix} \sigma_p^2 & \sigma_{pv} \\ \sigma_{vp} & \sigma_v^2 \end{bmatrix}$$

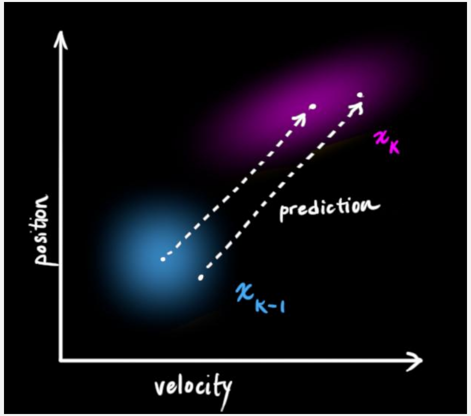


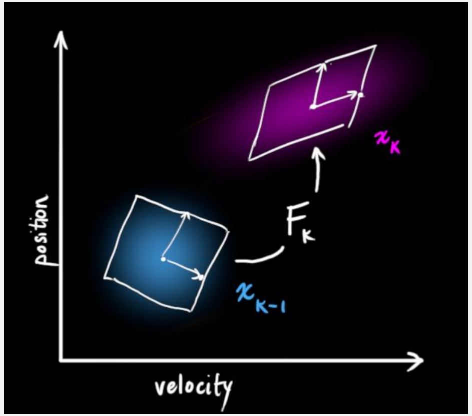
Источник: https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

ФИЛЬТР КАЛМАНА. ПРЕДСКАЗАНИЕ

Предсказание о состоянии системы в следующий момент времени делается на основе модели движения:

$$p_t = p_{t-1} + \Delta t v_{t-1}$$
$$v_t = v_{t-1}$$





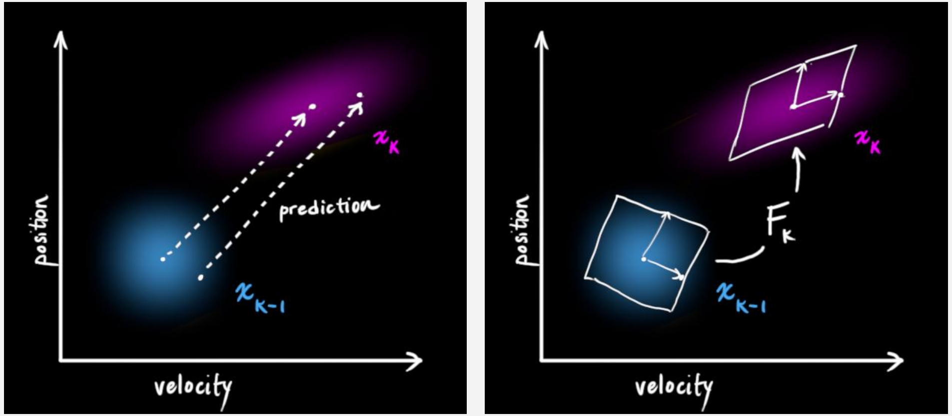
В матричной форме:

$$\hat{\mathbf{x}}_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{\mathbf{x}}_{t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1}$$

$\mathbf{F}_t$ – матрица эволюции процесса / системы (в нашем случае – модель движения) из *t-1* в *t*

ФИЛЬТР КАЛМАНА. ПРЕДСКАЗАНИЕ

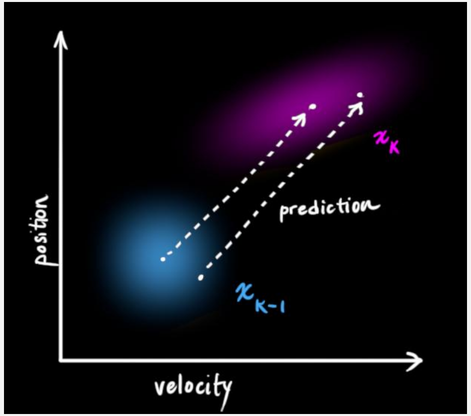
$$cov(x) = \Sigma$$
$$cov(\mathbf{A}x) = \mathbf{A}\Sigma\mathbf{A}^T$$

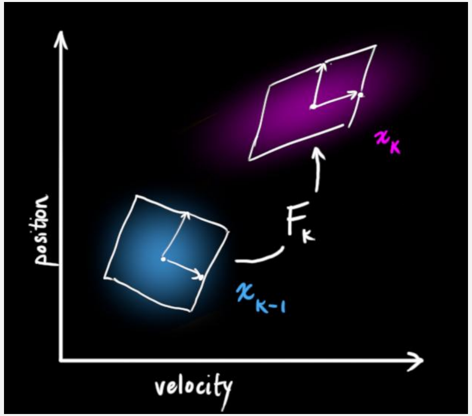


$$\hat{\mathbf{x}}_t = \mathbf{F}_t\hat{\mathbf{x}}_{t-1}$$
$$\hat{\Sigma}_t = \mathbf{F}_t\Sigma_{t-1}\mathbf{F}_t^T$$

ФИЛЬТР КАЛМАНА. СИГНАЛЫ УПРАВЛЕНИЯ

$$p_t = p_{t-1} + \Delta t v_{t-1} + \frac{1}{2} a \Delta t^2$$
$$v_t = v_{t-1} + a \Delta t$$





В матричной форме:

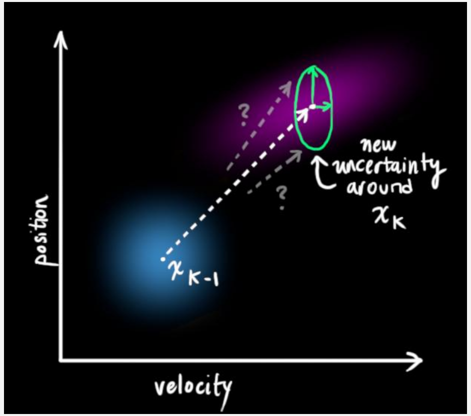
$$\hat{\mathbf{x}}_t = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_t \vec{\mathbf{u}_t}$$

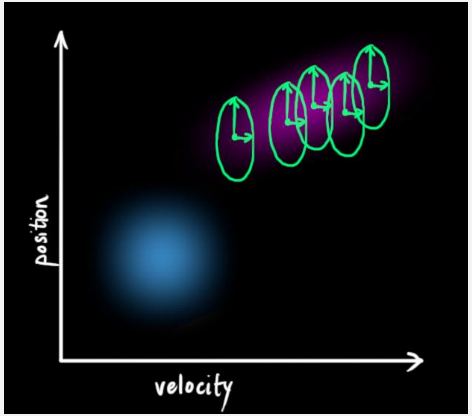
$\mathbf{B}_t$ – матрица описывающая изменение управляющих сигналов из t-1 в t

ФИЛЬТР КАЛМАНА. НЕОПРЕДЕЛЕННОСТЬ ДВИЖЕНИЯ

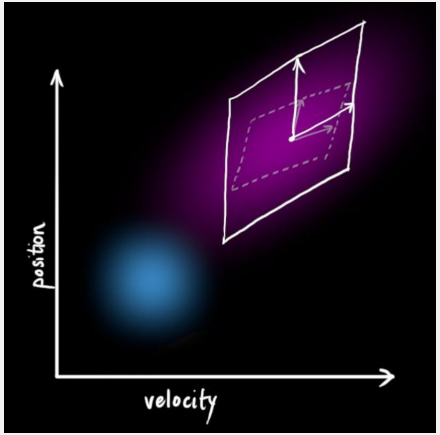
$$\hat{\mathbf{x}}_t = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_k \vec{\mathbf{u}}_t$$

$$\hat{\Sigma}_t = \mathbf{F}_t \Sigma_{t-1} \mathbf{F}_t^T + \mathbf{Q}_t$$





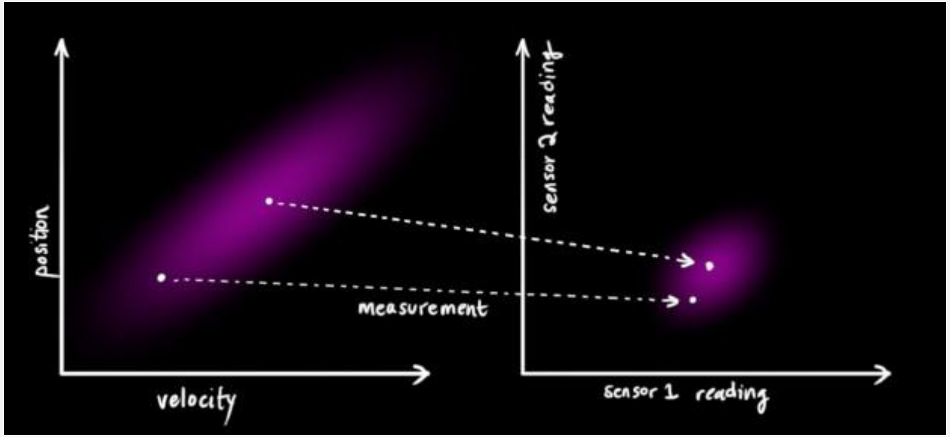
$\mathbf{Q}_t$ – матрица ковариации, которая описывает случайный характер эволюции системы

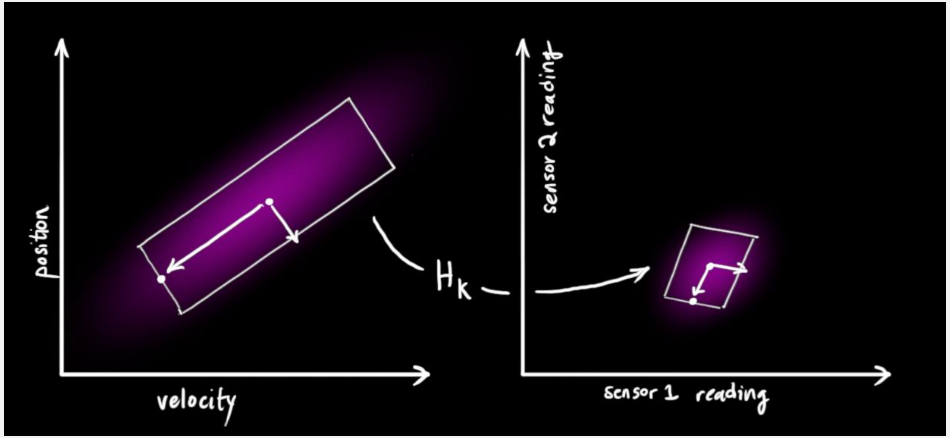


ФИЛЬТР КАЛМАНА. КОРРЕКЦИЯ

$$\vec{\mu}_{\text{expected}} = \mathbf{H}_t \hat{\mathbf{x}}_t$$

$$\mathbf{\Sigma}_{\text{expected}} = \mathbf{H}_t \mathbf{\Sigma}_t \mathbf{H}_t^T$$



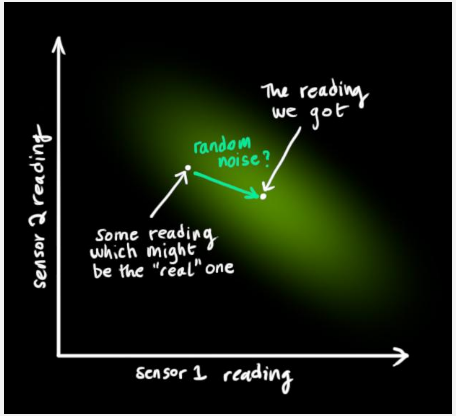


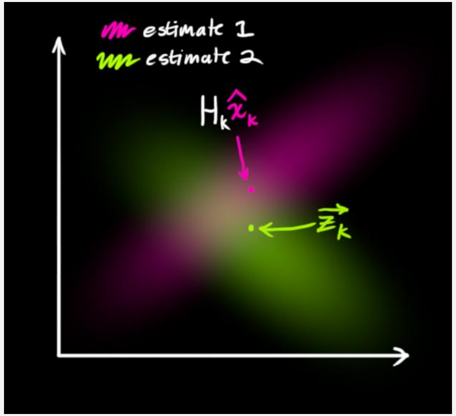
$\mathbf{H}_t$ – матрица, описывающая модель измерения: как из состояния $\hat{\mathbf{x}}_t$ получить измерение $\mathbf{z}_t$

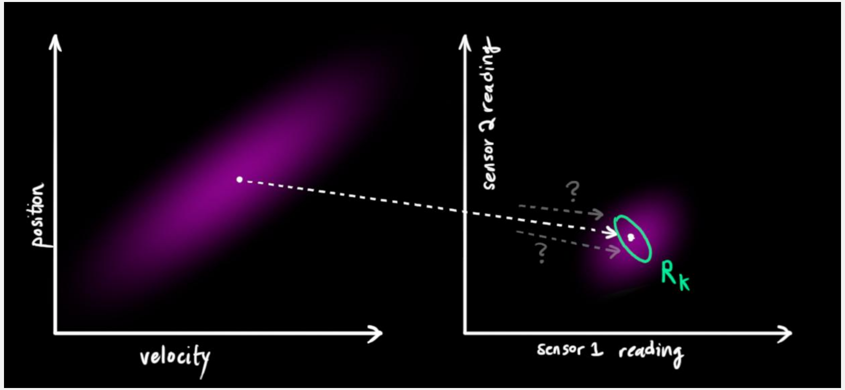
ФИЛЬТР КАЛМАНА. НЕОПРЕДЕЛЕННОСТЬ ИЗМЕРЕНИЙ

$$\vec{\mu}_{\text{measured}} = \vec{z_t}$$

$$\Sigma_{\text{measured}} = R_t$$







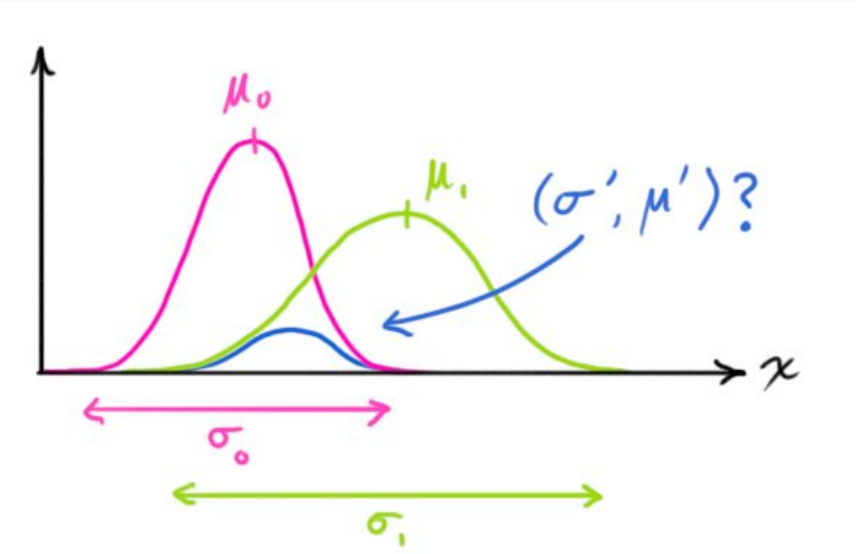
R_t – матрица ковариации, которая описывает случайный характер измерений

ОТСТУПЛЕНИЕ. НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ

$$\mathcal{N}(x, \mu_0, \sigma_0) \cdot \mathcal{N}(x, \mu_1, \sigma_1) \stackrel{?}{=} \mathcal{N}(x, \mu, \sigma)$$

$$\mu = \mu_0 + \frac{\sigma_0^2(\mu_1 \mu_0)}{\sigma_0^2 + \sigma_1^2}$$

$$\sigma^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2}$$



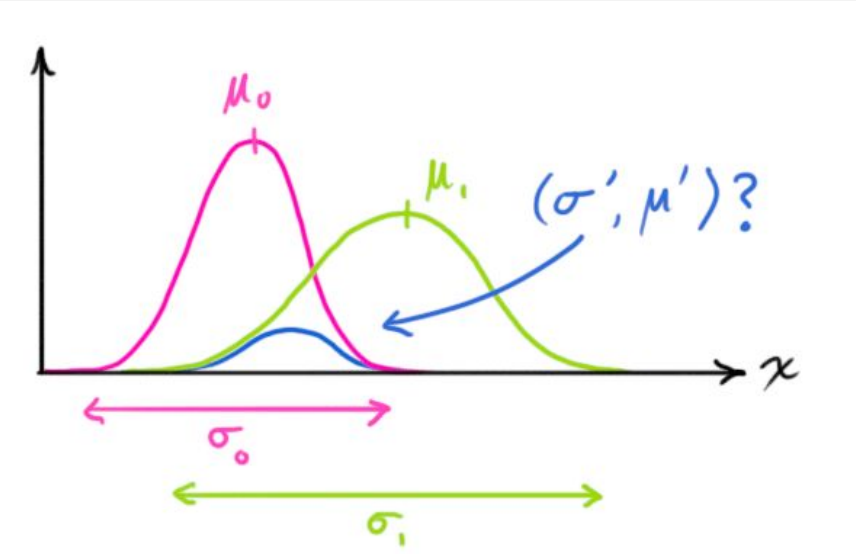
ОТСТУПЛЕНИЕ. НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ

$$\mathcal{N}(x, \mu_0, \sigma_0) \cdot \mathcal{N}(x, \mu_1, \sigma_1) \stackrel{?}{=} \mathcal{N}(x, \mu, \sigma)$$

$$\mathbf{K} = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$$

$$\vec{\mu} = \vec{\mu_0} + \mathbf{K}(\vec{\mu_1} - \vec{\mu_0})$$

$$\Sigma = \Sigma_0 - \mathbf{K}\Sigma_0$$



ФИЛЬТР КАЛМАНА. СОБИРАЕМ ВСЕ ВМЕСТЕ

$$\boxed{\begin{aligned} \mathbf{K} &= \Sigma_0(\Sigma_0 + \Sigma_1)^{-1} \\ \vec{\mu} &= \vec{\mu_0} + \mathbf{K}(\vec{\mu_1} - \vec{\mu_0}) \\ \Sigma &= \Sigma_0 - \mathbf{K}\Sigma_0 \end{aligned}}$$

$$(\mu_0, \Sigma_0) = (\mathbf{H}_t \hat{\mathbf{x}}_t, \mathbf{H}_t \hat{\mathbf{\Sigma}}_t \mathbf{H}_t^T)$$

$$(\mu_1, \Sigma_1) = (\vec{\mathbf{z}_t}, \mathbf{R}_t)$$

$$\mathbf{H}_t \hat{\mathbf{x}}'_t = \mathbf{H}_t \hat{\mathbf{x}}_t + \mathbf{K}(\vec{\mathbf{z}_t} - \mathbf{H}_t \hat{\mathbf{x}}_t)$$

$$\mathbf{H}_t \mathbf{\Sigma}'_t \mathbf{H}_t^T = \mathbf{H}_t \hat{\mathbf{\Sigma}}_t \mathbf{H}_t^T - \mathbf{K} \mathbf{H}_t \hat{\mathbf{\Sigma}}_t \mathbf{H}_t^T$$

ФИЛЬТР КАЛМАНА. СОБИРАЕМ ВСЕ ВМЕСТЕ

$$\mathbf{K} = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$$

$$\vec{\mu} = \vec{\mu_0} + \mathbf{K}(\vec{\mu_1} - \vec{\mu_0})$$

$$\Sigma = \Sigma_0 - \mathbf{K}\Sigma_0$$

$$(\mu_0, \Sigma_0) = (\mathbf{H}_t\hat{\mathbf{x}}_t, \mathbf{H}_t\hat{\mathbf{\Sigma}}_t\mathbf{H}_t^T)$$

$$(\mu_1, \Sigma_1) = (\vec{\mathbf{z}_t}, \mathbf{R}_t)$$

$$\mathbf{K} = \mathbf{H}_t\hat{\mathbf{\Sigma}}_t\mathbf{H}_t^T(\mathbf{H}_t\hat{\mathbf{\Sigma}}_t\mathbf{H}_t^T + \mathbf{R}_t)^{-1}$$

$$\mathbf{H}_t\mathbf{x}'_t = \mathbf{H}_t\hat{\mathbf{x}}_t + \mathbf{K}(\vec{\mathbf{z}_t} - \mathbf{H}_t\hat{\mathbf{x}}_t)$$

$$\mathbf{H}_t\mathbf{\Sigma}'_t\mathbf{H}_t^T = \mathbf{H}_t\hat{\mathbf{\Sigma}}_t\mathbf{H}_t^T - \mathbf{K}\mathbf{H}_t\hat{\mathbf{\Sigma}}_t\mathbf{H}_t^T$$

ФИЛЬТР КАЛМАНА. СОБИРАЕМ ВСЕ ВМЕСТЕ

$$\mathbf{K} = \Sigma_0(\Sigma_0 + \Sigma_1)^{-1}$$

$$\vec{\mu} = \vec{\mu_0} + \mathbf{K}(\vec{\mu_1} - \vec{\mu_0})$$

$$\Sigma = \Sigma_0 - \mathbf{K}\Sigma_0$$

$$(\mu_0, \Sigma_0) = (\mathbf{H}_t\hat{\mathbf{x}}_t, \mathbf{H}_t\hat{\boldsymbol{\Sigma}}_t\mathbf{H}_t^T)$$

$$(\mu_1, \Sigma_1) = (\vec{\mathbf{z}_t}, \mathbf{R}_t)$$

$$\mathbf{K}' = \hat{\boldsymbol{\Sigma}}_t\mathbf{H}_t^T(\mathbf{H}_t\hat{\boldsymbol{\Sigma}}_t\mathbf{H}_t^T + \mathbf{R}_t)^{-1}$$

$$\mathbf{x}'_t = \hat{\mathbf{x}}_t + \mathbf{K}'(\vec{\mathbf{z}_t} - \mathbf{H}_t\hat{\mathbf{x}}_t)$$

$$\boldsymbol{\Sigma}'_t = \hat{\boldsymbol{\Sigma}}_t - \mathbf{K}'\mathbf{H}_t\hat{\boldsymbol{\Sigma}}_t$$

ФИЛЬТР КАЛМАНА

Предсказание:

$$\hat{\mathbf{x}}_t = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_k \vec{\mathbf{u}_t}$$

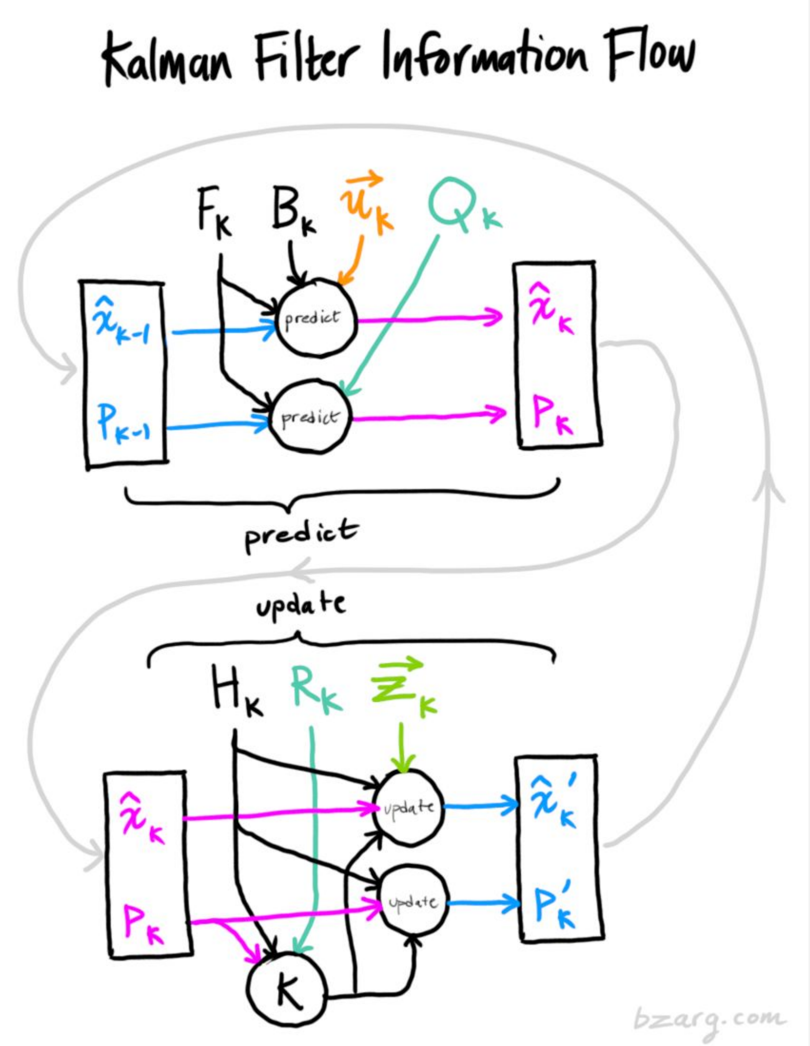
$$\hat{\mathbf{\Sigma}}_t = \mathbf{F}_t \mathbf{\Sigma}_{t-1} \mathbf{F}_t^T + \mathbf{Q}_t$$

Коррекция:

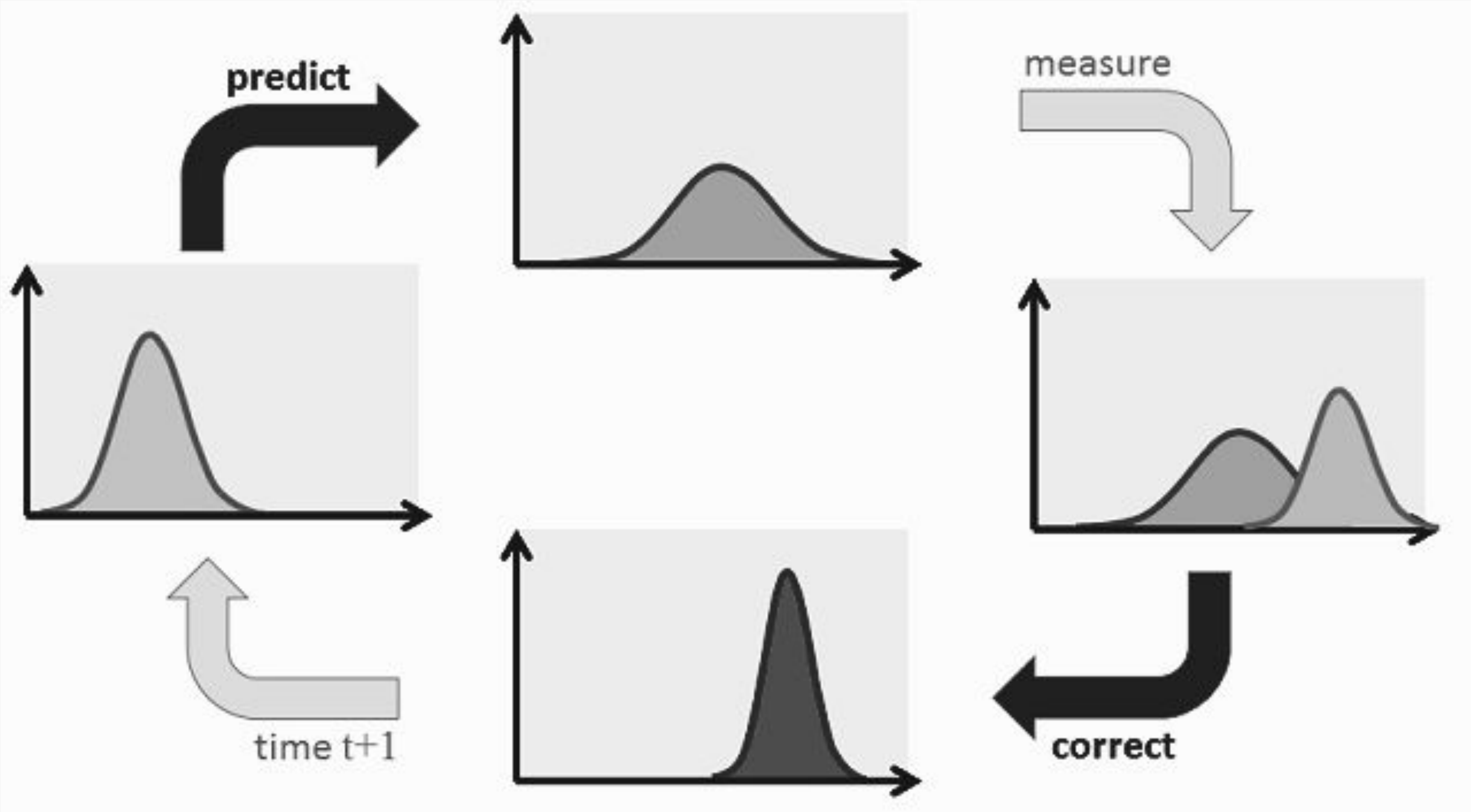
$$\mathbf{K}' = \hat{\mathbf{\Sigma}}_t \mathbf{H}_t^T (\mathbf{H}_t \hat{\mathbf{\Sigma}}_t \mathbf{H}_t^T + \mathbf{R}_t)^{-1}$$

$$\mathbf{x}'_t = \hat{\mathbf{x}}_t + \mathbf{K}'(\vec{\mathbf{z}_t} - \mathbf{H}_t \hat{\mathbf{x}}_t)$$

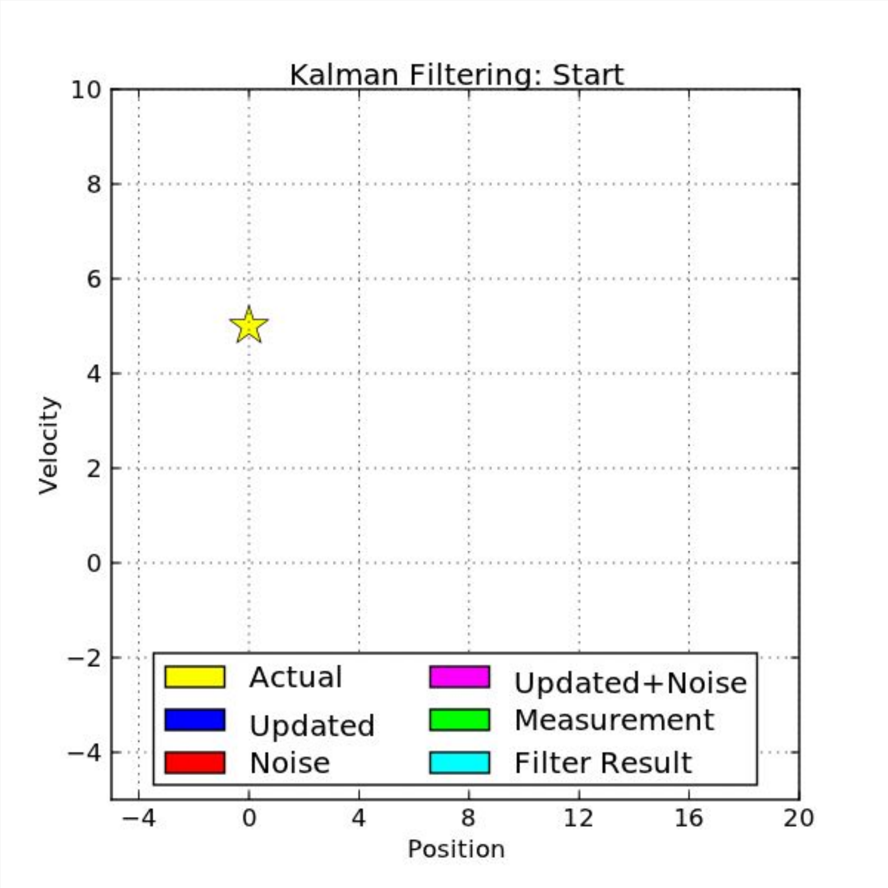
$$\mathbf{\Sigma}'_t = \hat{\mathbf{\Sigma}}_t - \mathbf{K}' \mathbf{H}_t \hat{\mathbf{\Sigma}}_t$$



ФИЛЬТР КАЛМАНА. СОБИРАЕМ ВСЕ ВМЕСТЕ



ФИЛЬТР КАЛМАНА. СОБИРАЕМ ВСЕ ВМЕСТЕ



ЧТО ДЕЛАТЬ С НЕЛИНЕЙНОСТЬЮ?

Представим, что теперь мы хотим оценивать еще и ориентацию робота в пространстве:

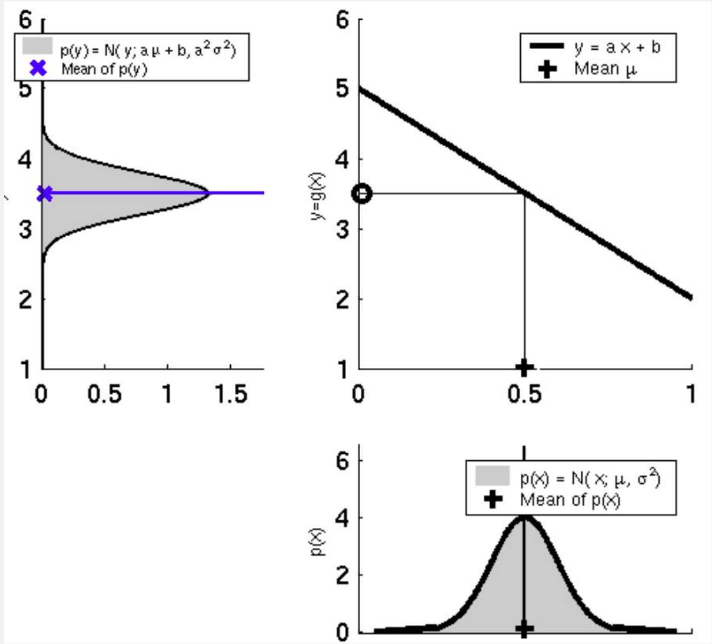
- Это приведет к появлению тригонометрических функций в модели системы
- На выходе нелинейной системы больше **не** нормальное распределение
- Фильтр Калмана больше неприменим

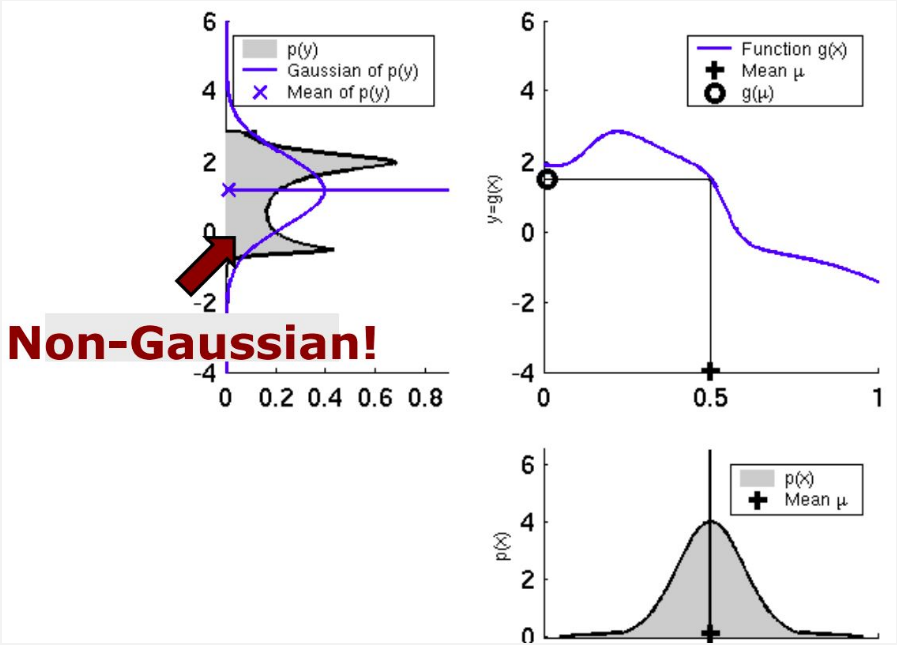
Решение:

- Локальная линеаризация моделей
- Сигма-точечная аппроксимация

ЧТО ДЕЛАТЬ С НЕЛИНЕЙНОСТЬЮ?

Линейная система **VS** Нелинейная система





Источник: Thrun, Burgard, Fox

ЧТО ДЕЛАТЬ С НЕЛИНЕЙНОСТЬЮ?

Линейная система **VS** Нелинейная система

$$\hat{\mathbf{x}}_t = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_t \vec{\mathbf{u}}_t \qquad \hat{\mathbf{x}}_t = g(\hat{\mathbf{x}}_{t-1}, \vec{\mathbf{u}}_t)$$

$$\vec{\mu}_{\text{expected}} = \mathbf{H}_t \hat{\mathbf{x}}_t \qquad \vec{\mu}_{\text{expected}} = h(\hat{\mathbf{x}}_t)$$

ЧТО ДЕЛАТЬ С НЕЛИНЕЙНОСТЬЮ?

Линейная система **VS** Нелинейная система

$$\hat{\mathbf{x}}_t = \mathbf{F}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_t \vec{\mathbf{u}}_t \qquad \hat{\mathbf{x}}_t = g(\hat{\mathbf{x}}_{t-1}, \vec{\mathbf{u}}_t)$$

$$\vec{\mu}_{\text{expected}} = \mathbf{H}_t \hat{\mathbf{x}}_t \qquad \vec{\mu}_{\text{expected}} = h(\hat{\mathbf{x}}_t)$$

Нелинейные функции

РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА
(EXTENDED KALMAN FILTER)

$$g(\hat{\mathbf{x}}_{t-1}, \vec{\mathbf{u}}_t) \approx g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t) + \frac{\partial g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t)}{\partial \mathbf{x}_{t-1}}(\hat{\mathbf{x}}_{t-1} - \vec{\mu}_{t-1})$$

$$h(\hat{\mathbf{x}}_t) \approx h(\vec{\mu}_t) + \frac{\partial h(\vec{\mu}_t)}{\partial \mathbf{x}_t}(\mathbf{x}_t - \vec{\mu}_t)$$

РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА
(EXTENDED KALMAN FILTER)

$$g(\hat{\mathbf{x}}_{t-1}, \vec{\mathbf{u}}_t) \approx g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t) + \frac{\partial g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t)}{\partial \mathbf{x}_{t-1}} (\hat{\mathbf{x}}_{t-1} - \vec{\mu}_{t-1})$$

$$h(\hat{\mathbf{x}}_t) \approx h(\vec{\mu}_t) + \frac{\partial h(\vec{\mu}_t)}{\partial \mathbf{x}_t} (\mathbf{x}_t - \vec{\mu}_t)$$

Матрицы Якоби

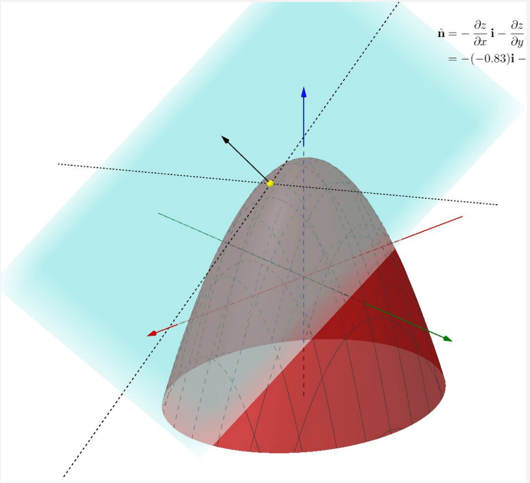
РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА
(EXTENDED KALMAN FILTER)

Пусть задана вектор-функция:

$$g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix}$$

Тогда матрица Якоби определяется как:

$$G_x = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}$$



Задает ориентацию плоскости нормальной к векторной функции в заданной точке.

Обобщает понятие градиента скалярной функции на многомерный случай.

РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА
(EXTENDED KALMAN FILTER)

$$g(\hat{\mathbf{x}}_{t-1}, \vec{\mathbf{u}}_t) \approx g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t) + \frac{\partial g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t)}{\partial \mathbf{x}_{t-1}}(\hat{\mathbf{x}}_{t-1} - \vec{\mu}_{t-1})$$

$$h(\hat{\mathbf{x}}_t) \approx h(\vec{\mu}_t) + \frac{\partial h(\vec{\mu}_t)}{\partial \mathbf{x}_t}(\mathbf{x}_t - \vec{\mu}_t)$$

Матрицы Якоби

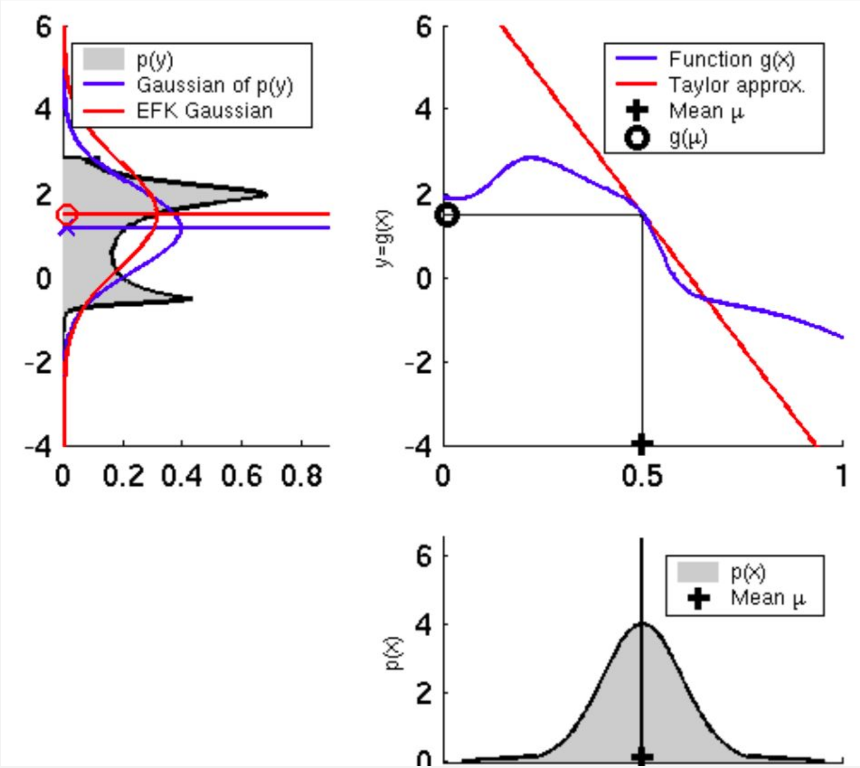
РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА
(EXTENDED KALMAN FILTER)

$$g(\hat{\mathbf{x}}_{t-1}, \vec{\mathbf{u}}_t) \approx g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t) + \frac{\partial g(\vec{\mu}_{t-1}, \vec{\mathbf{u}}_t)}{\partial \mathbf{x}_{t-1}}(\hat{\mathbf{x}}_{t-1} - \vec{\mu}_{t-1})$$

$$h(\hat{\mathbf{x}}_t) \approx h(\vec{\mu}_t) + \frac{\partial h(\vec{\mu}_t)}{\partial \mathbf{x}_t}(\mathbf{x}_t - \vec{\mu}_t)$$

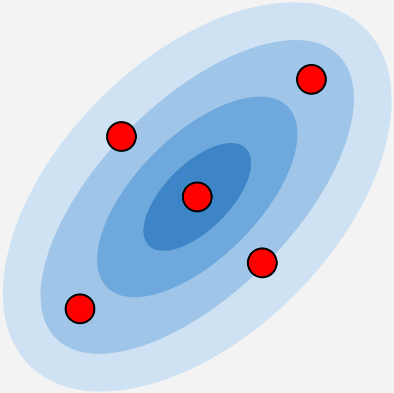
Линейные функции!

РАСШИРЕННЫЙ ФИЛЬТР КАЛМАНА
(EXTENDED KALMAN FILTER)

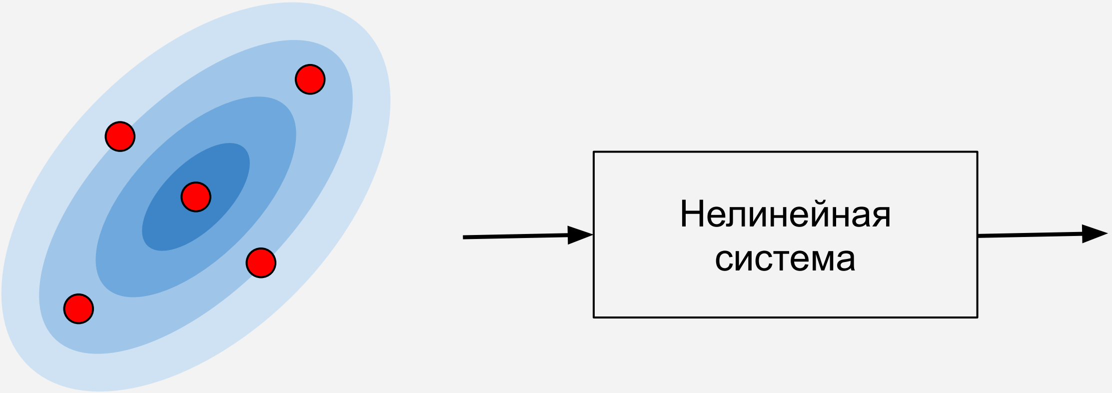


СИГМА-ТОЧЕЧНЫЙ ФИЛЬТР КАЛМАНА
(UNSCENTED KALMAN FILTER)

СИГМА-ТОЧЕЧНЫЙ ФИЛЬТР КАЛМАНА
(UNSCENTED KALMAN FILTER)

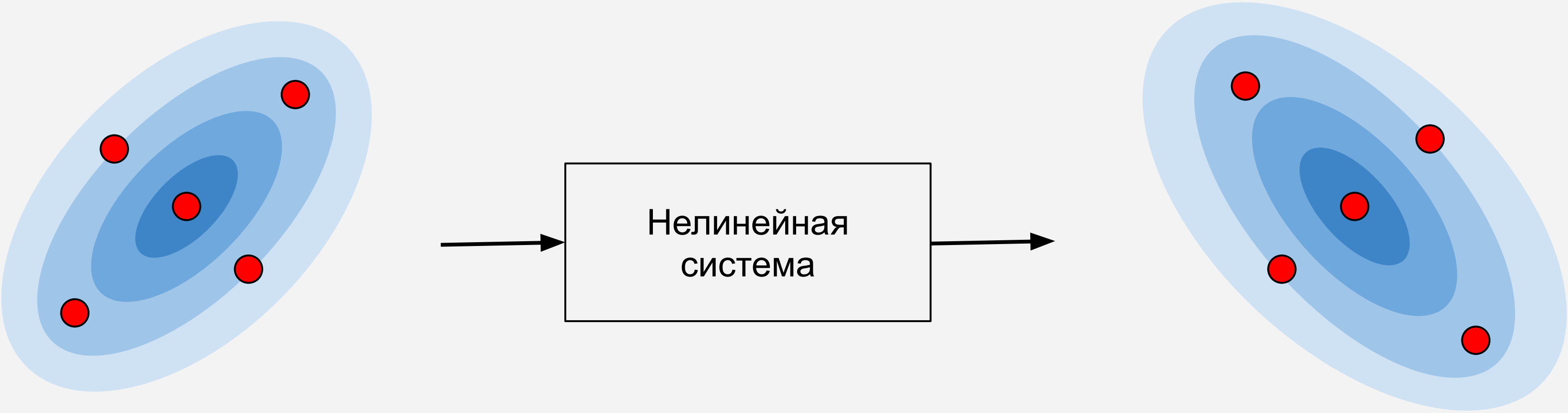


СИГМА-ТОЧЕЧНЫЙ ФИЛЬТР КАЛМАНА
(UNSCENTED KALMAN FILTER)



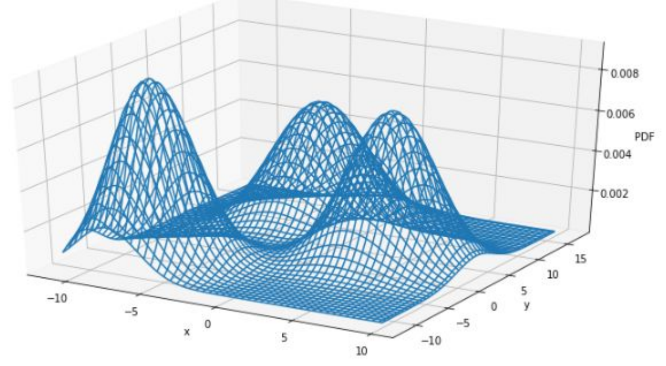
СИГМА-ТОЧЕЧНЫЙ ФИЛЬТР КАЛМАНА
(UNSCENTED KALMAN FILTER)

СИГМА-ТОЧЕЧНЫЙ ФИЛЬТР КАЛМАНА
(UNSCENTED KALMAN FILTER)



ФИЛЬТР ЧАСТИЦ
(МНОГОЧАСТИЧНЫЙ ФИЛЬТР, ЛОКАЛИЗАЦИЯ МОНТЕ-КАРЛО)

- Работает с любыми (в т.ч. нелинейными) моделями движения и измерения (наблюдения)
- Может описывать мультимодальное распределение
- Может решать задачу глобальной локализации — локализации без начального приближения



ФИЛЬТР ЧАСТИЦ

- Фильтр частиц позволяет оценить апостериорное распределение вектора состояния системы в скрытом Марковском процессе.

- x_t – состояние Марковского процесса в момент времени t.

- x_t зависит от предыдущего состояния x_{t-1} в соответствии с **моделью движения** $p(x_t|u_t, x_{t-1})$, где u_t – вектор управляющих сигналов, примененных в момент $t - 1$.

- x_t описывается вектором измерения z_t, генерируемым в соответствии с **моделью измерения** $p(z_t|x_t)$.

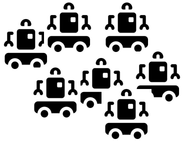
- Идея фильтра частиц – аппроксимация неизвестного апостериорного распределения набором гипотез $\{x^n_t\}$ и соответствующих им весов $\{w^n_t\}$,
где $n = 1, ..., N$ – количество гипотез / частиц.

- В задаче локализации каждая частица соответствует позе робота: $x_t = (x_t, y_t, \Theta_t)^T$.

Algorithm 1 Generic Monte-Carlo localization algorithm

procedure MCL($\mathbf{x_{t-1}}, m, \mathbf{u_t}, \mathbf{z_t}$)
    $\{\mathbf{x_t^n}\} = \{\widetilde{\mathbf{x_t^n}}\} = \emptyset$
    for $n = 1$ to N do
        sample $x_t^n \sim p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}^n})$
        $\widetilde{w_t^n} = p(\mathbf{z_t}|\mathbf{x_t^n}, map)$
        $\widetilde{\{\mathbf{x_t^n}\}} = \widetilde{\{\mathbf{x_t^n}\}} + \langle x_t^n, w_t^n \rangle$
    end for
    for $n = 1$ to N do
        draw i with probability $\propto \widetilde{w_t^i}$
        $\{\mathbf{x_t^n}\} = \{\mathbf{x_t^n}\} + \langle \widetilde{x_t^i}, \widetilde{w_t^i} \rangle$
    end for
    return $\{\mathbf{x_t^n}\}$
end procedure

ФИЛЬТР ЧАСТИЦ

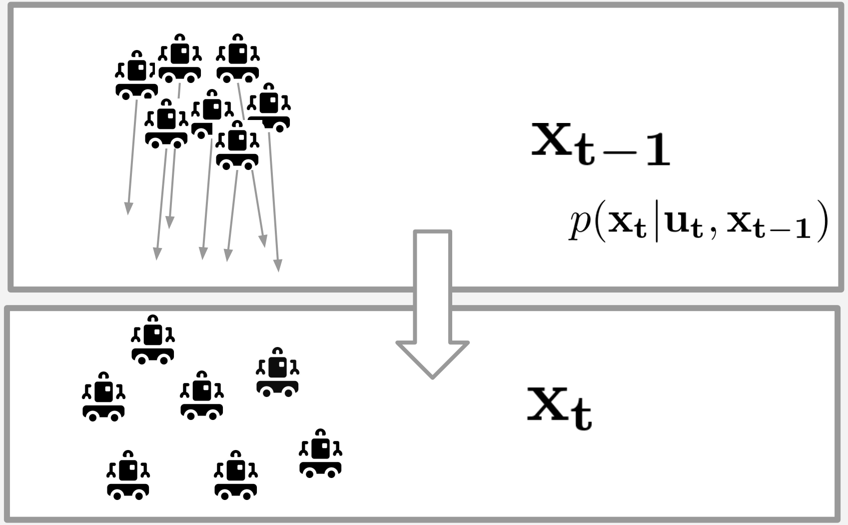


$$\mathbf{x_{t-1}}$$

Algorithm 1 Generic Monte-Carlo localization algorithm

```
procedure MCL($\mathbf{x_{t-1}}, m, \mathbf{u_t}, \mathbf{z_t}$)
    $\{\mathbf{x_t^n}\} = \widetilde{\{\mathbf{x_t^n}\}} = \emptyset$
    for $n = 1$ to $N$ do
        sample $x_t^n \sim p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}^n})$
        $w_t^n = p(\mathbf{z_t}|\mathbf{x_t^n}, map)$
        $\widetilde{\{\mathbf{x_t^n}\}} = \widetilde{\{\mathbf{x_t^n}\}} + \langle x_t^n, w_t^n \rangle$
    end for
    for $n = 1$ to $N$ do
        draw $i$ with probability $\propto \widetilde{w_t^i}$
        $\{\mathbf{x_t^n}\} = \{\mathbf{x_t^n}\} + \langle \widetilde{x_t^i}, \widetilde{w_t^i} \rangle$
    end for
    return $\{\mathbf{x_t^n}\}$
end procedure
```

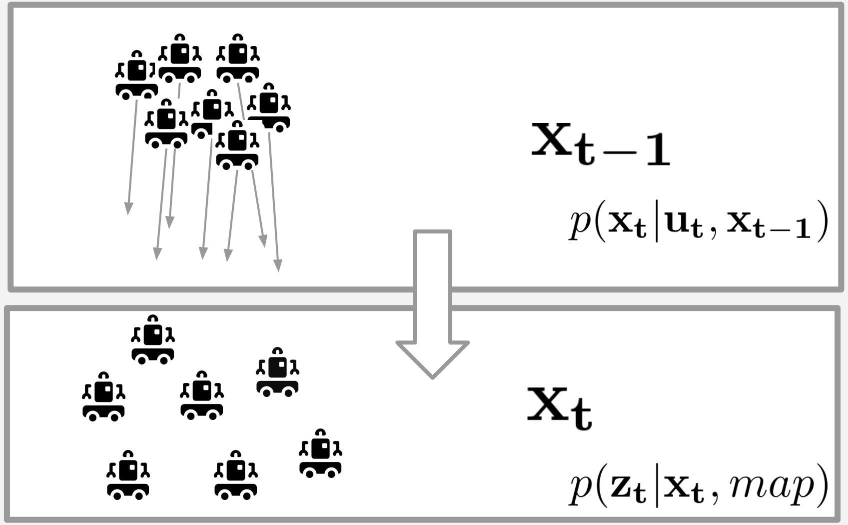
ФИЛЬТР ЧАСТИЦ



Algorithm 1 Generic Monte-Carlo localization algorithm

procedure MCL($\mathbf{x_{t-1}}, m, \mathbf{u_t}, \mathbf{z_t}$)
    $\{\mathbf{x_t^n}\} = \widetilde{\{\mathbf{x_t^n}\}} = \emptyset$
    for $n = 1$ to N do
        sample $x_t^n \sim p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}^n})$ Motion model
        $w_t^n = p(\mathbf{z_t}|\mathbf{x_t^n}, map)$
        $\widetilde{\{\mathbf{x_t^n}\}} = \widetilde{\{\mathbf{x_t^n}\}} + \langle x_t^n, w_t^n \rangle$
    end for
    for $n = 1$ to N do
        draw i with probability $\propto \widetilde{w_t^i}$
        $\{\mathbf{x_t^n}\} = \{\mathbf{x_t^n}\} + \langle \widetilde{x_t^i}, \widetilde{w_t^i} \rangle$
    end for
    return $\{\mathbf{x_t^n}\}$
end procedure

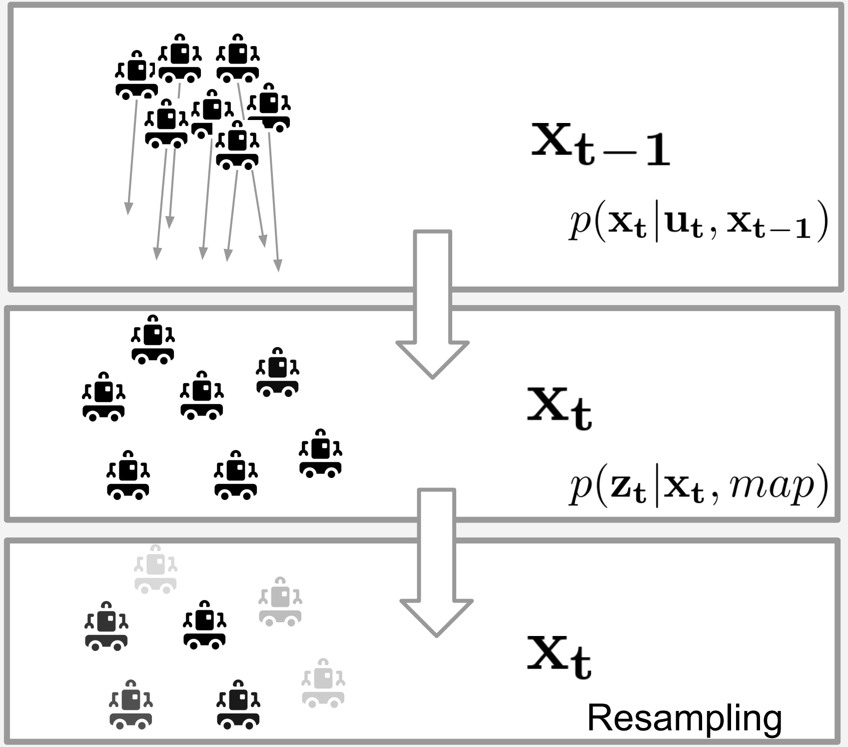
ФИЛЬТР ЧАСТИЦ



Algorithm 1 Generic Monte-Carlo localization algorithm

procedure MCL($\mathbf{x_{t-1}}, m, \mathbf{u_t}, \mathbf{z_t}$)
  $\{\mathbf{x_t^n}\} = \{\widetilde{\mathbf{x_t^n}}\} = \emptyset$
  for $n = 1$ to N do
    sample $x_t^n \sim p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}^n})$
    $w_t^n = p(\mathbf{z_t}|\mathbf{x_t^n}, map)$
    $\widetilde{\{\mathbf{x_t^n}\}} = \widetilde{\{\mathbf{x_t^n}\}} + \langle x_t^n, w_t^n \rangle$ Measurement model
  end for
  for $n = 1$ to N do
    draw i with probability $\propto \widetilde{w_t^i}$
    $\{\mathbf{x_t^n}\} = \{\mathbf{x_t^n}\} + \langle \widetilde{x_t^i}, \widetilde{w_t^i} \rangle$
  end for
  return $\{\mathbf{x_t^n}\}$
end procedure

ФИЛЬТР ЧАСТИЦ

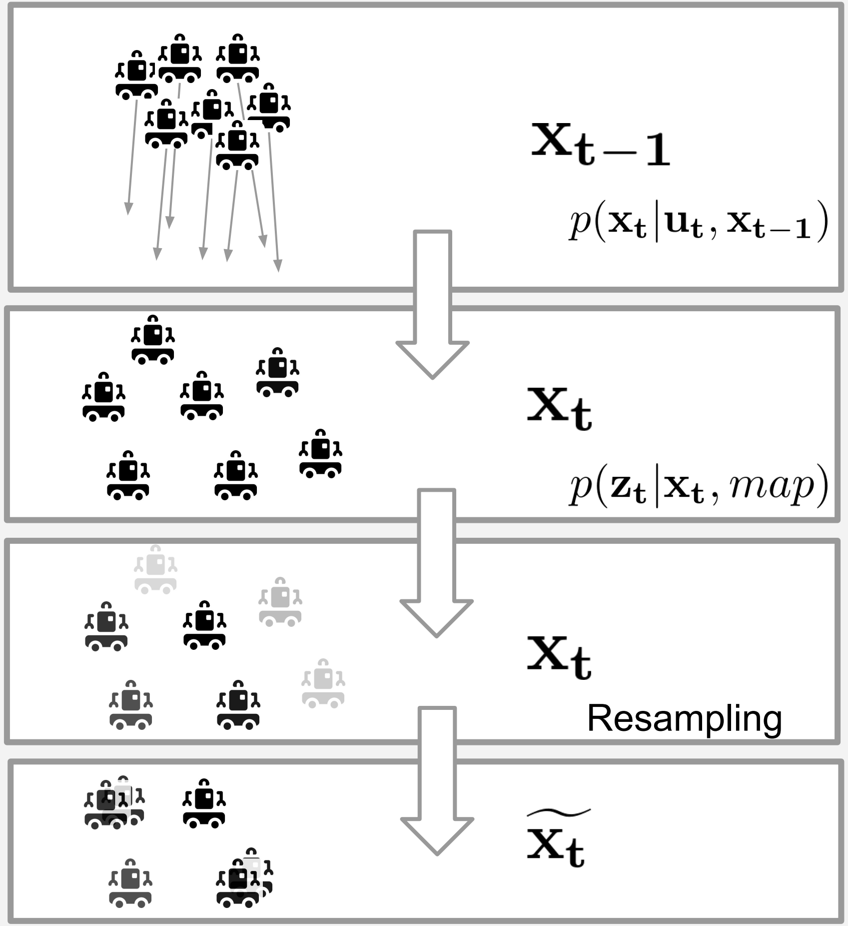


Algorithm 1 Generic Monte-Carlo localization algorithm

procedure MCL($\mathbf{x_{t-1}}, m, \mathbf{u_t}, \mathbf{z_t}$)
  $\{\mathbf{x_t^n}\} = \widetilde{\{\mathbf{x_t^n}\}} = \emptyset$
  for $n = 1$ to N do
    sample $x_t^n \sim p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}^n})$
    $w_t^n = p(\mathbf{z_t}|\mathbf{x_t^n}, map)$
    $\widetilde{\{\mathbf{x_t^n}\}} = \widetilde{\{\mathbf{x_t^n}\}} + \langle x_t^n, w_t^n \rangle$
  end for
  for $n = 1$ to N do
    draw i with probability $\propto \widetilde{w_t^i}$
    $\{\mathbf{x_t^n}\} = \{\mathbf{x_t^n}\} + \langle \widetilde{x_t^i}, \widetilde{w_t^i} \rangle$
  end for
  return $\{\mathbf{x_t^n}\}$
end procedure

Resampling

ФИЛЬТР ЧАСТИЦ



Algorithm 1 Generic Monte-Carlo localization algorithm

procedure MCL($\mathbf{x_{t-1}}, m, \mathbf{u_t}, \mathbf{z_t}$)

  $\{\mathbf{x_t^n}\} = \{\widetilde{\mathbf{x_t^n}}\} = \emptyset$

  for $n = 1$ to N **do**

    sample $x_t^n \sim p(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x_{t-1}^n})$ — Motion model

    $w_t^n = p(\mathbf{z_t}|\mathbf{x_t^n}, map)$ — Measurement model

    $\{\widetilde{\mathbf{x_t^n}}\} = \{\widetilde{\mathbf{x_t^n}}\} + \langle x_t^n, w_t^n \rangle$

  end for

  for $n = 1$ to N **do** — Resampling

    draw i with probability $\propto \widetilde{w_t^i}$

    $\{\mathbf{x_t^n}\} = \{\mathbf{x_t^n}\} + \langle \widetilde{x_t^i}, \widetilde{w_t^i} \rangle$

  end for

  return $\{\mathbf{x_t^n}\}$

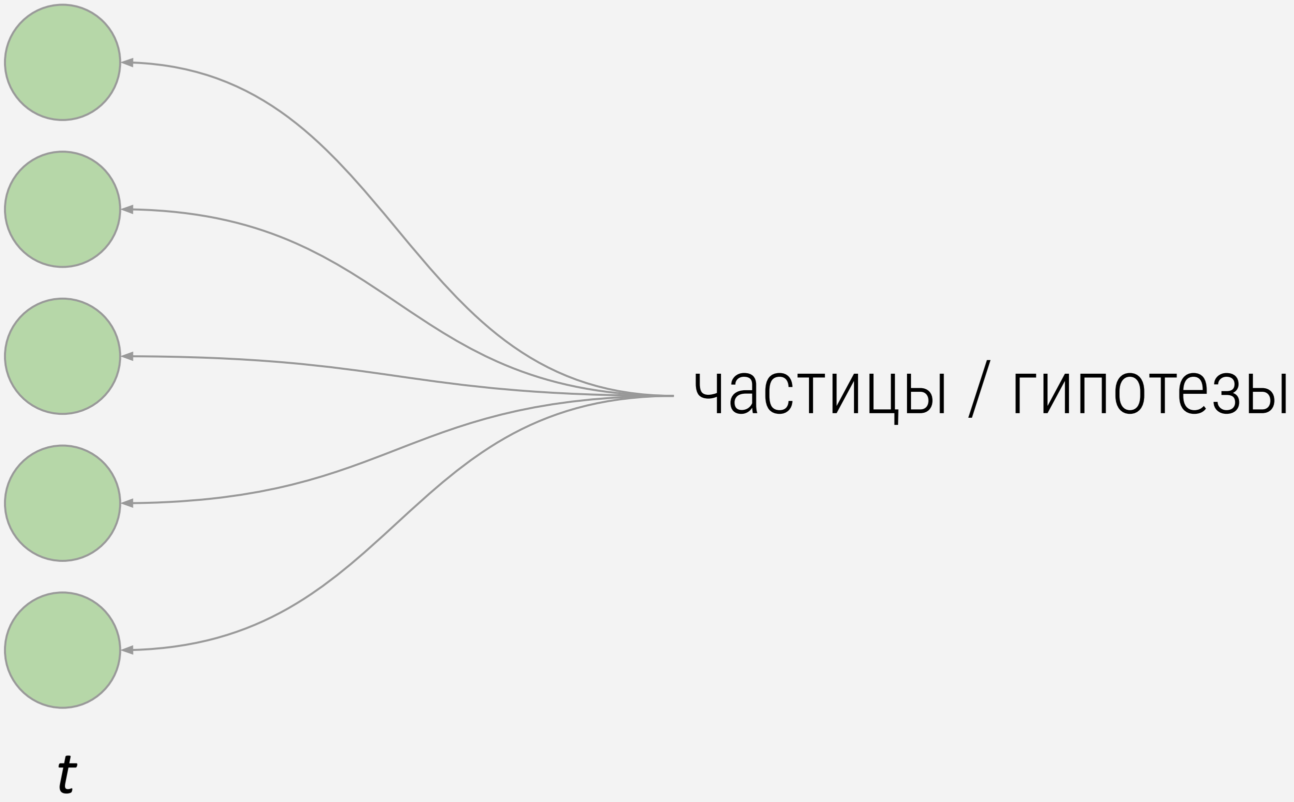
end procedure

В СЛЕДУЮЩЕЙ ЛЕКЦИИ

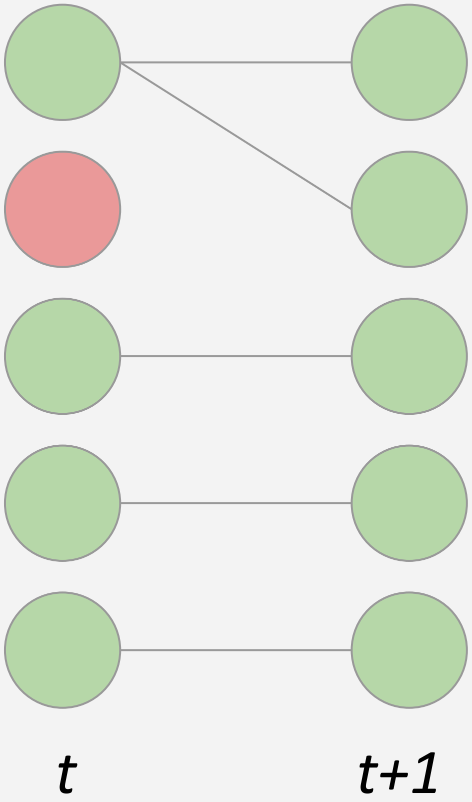
ФИЛЬТР ЧАСТИЦ
МОДЕЛЬ ИЗМЕРЕНИЯ

ЧЕРЕЗ ЛЕКЦИЮ

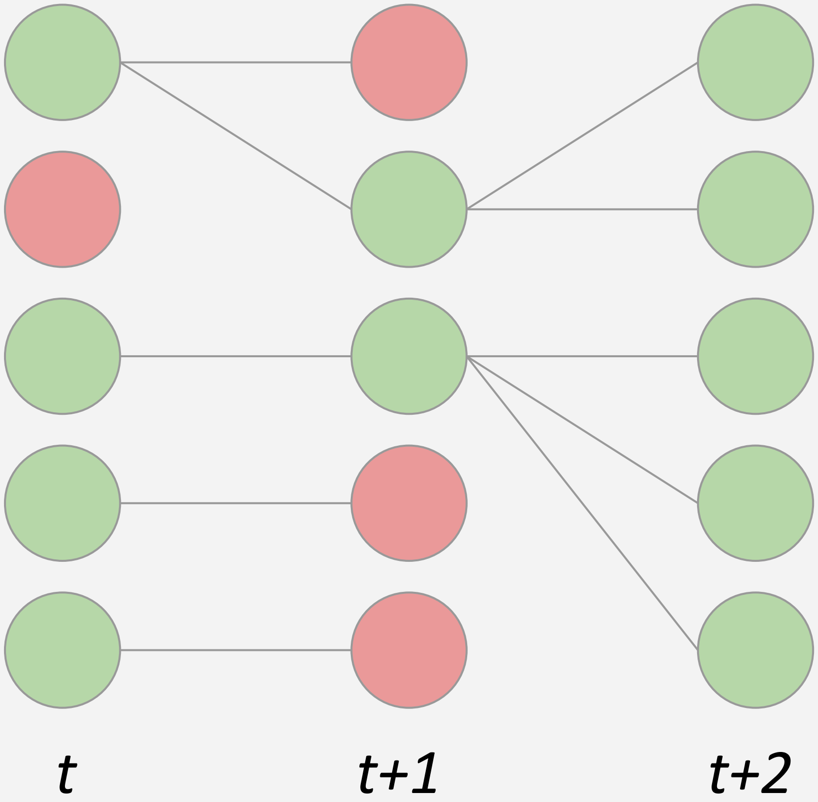
ФИЛЬТР ЧАСТИЦ
РЕСЕМПЛИНГ



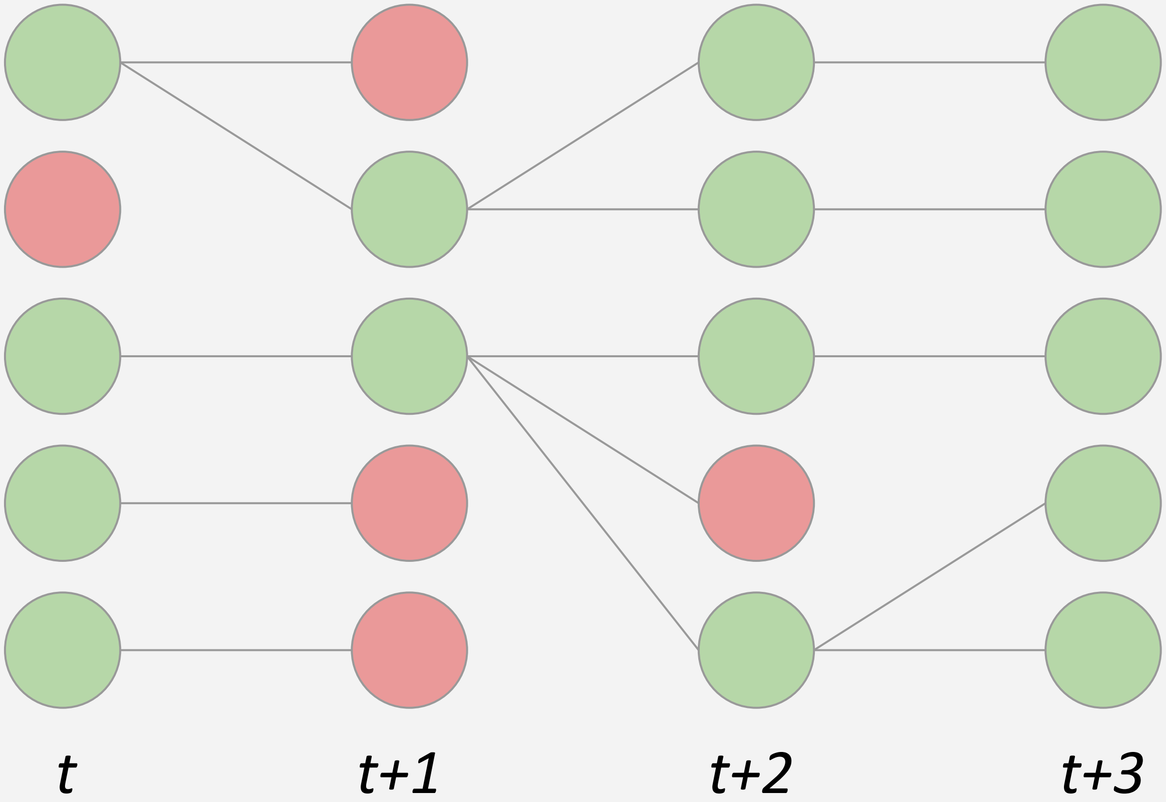
ФИЛЬТР ЧАСТИЦ
РЕСЕМПЛИНГ



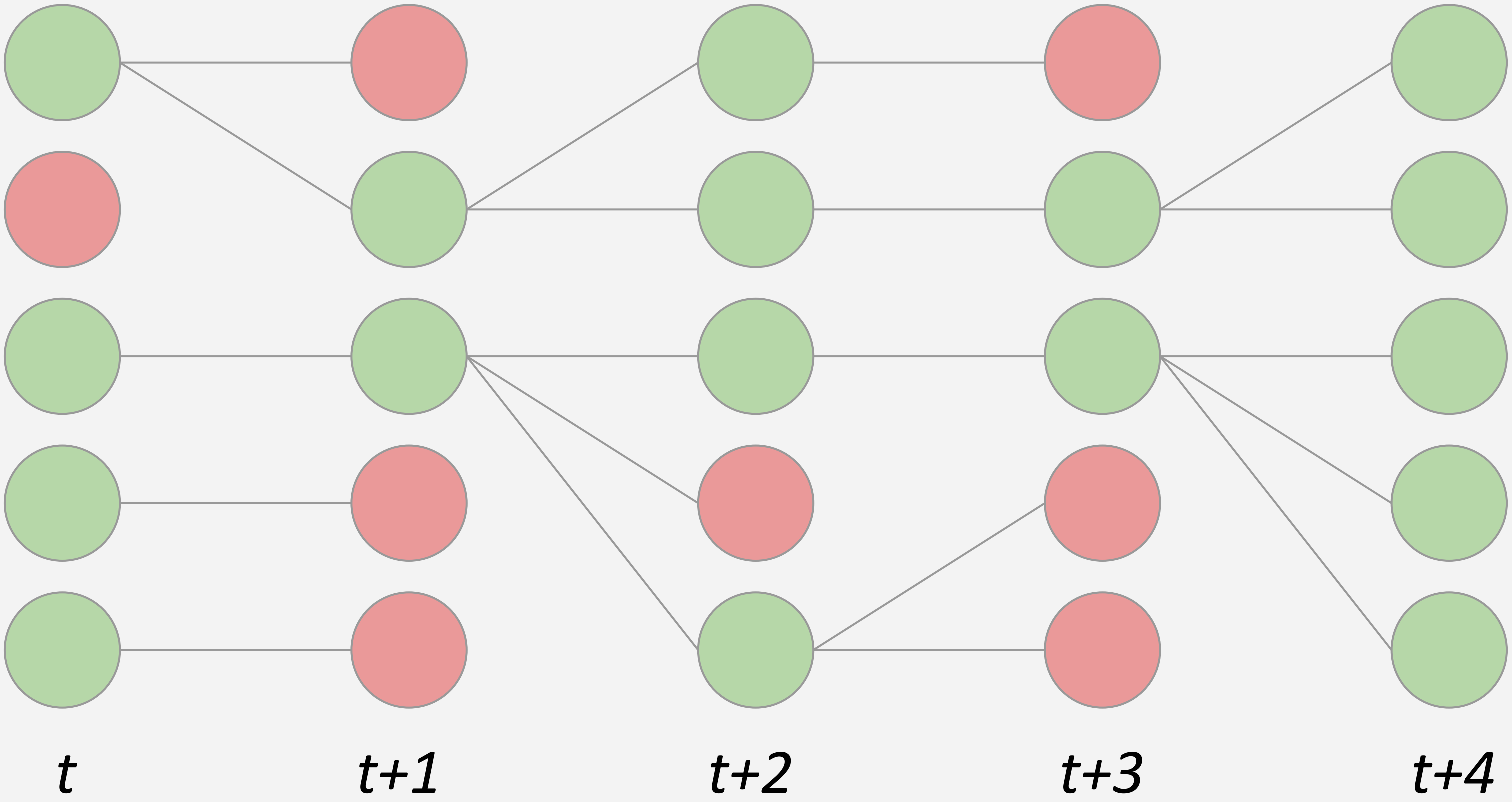
ФИЛЬТР ЧАСТИЦ
РЕСЕМПЛИНГ



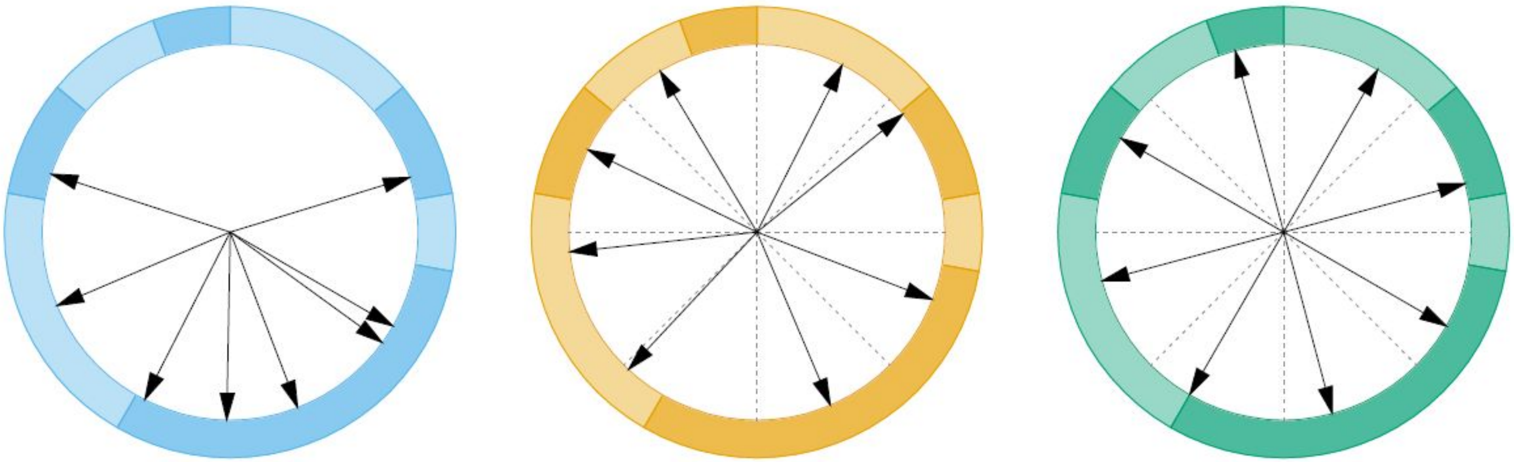
ФИЛЬТР ЧАСТИЦ
РЕСЕМПЛИНГ



ФИЛЬТР ЧАСТИЦ
РЕСЕМПЛИНГ

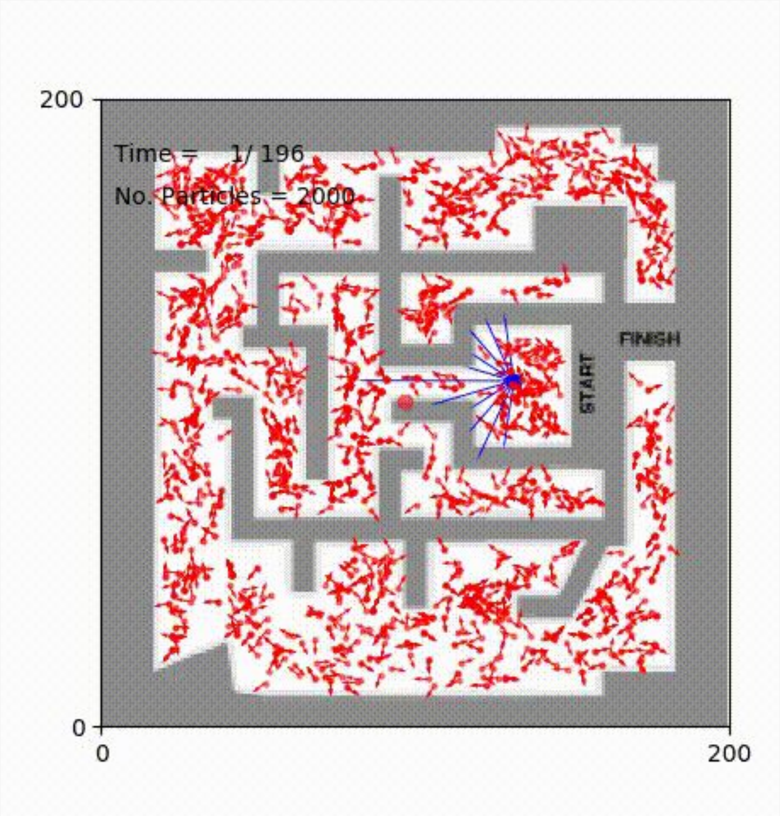


ФИЛЬТР ЧАСТИЦ
РЕСЕМПЛИНГ

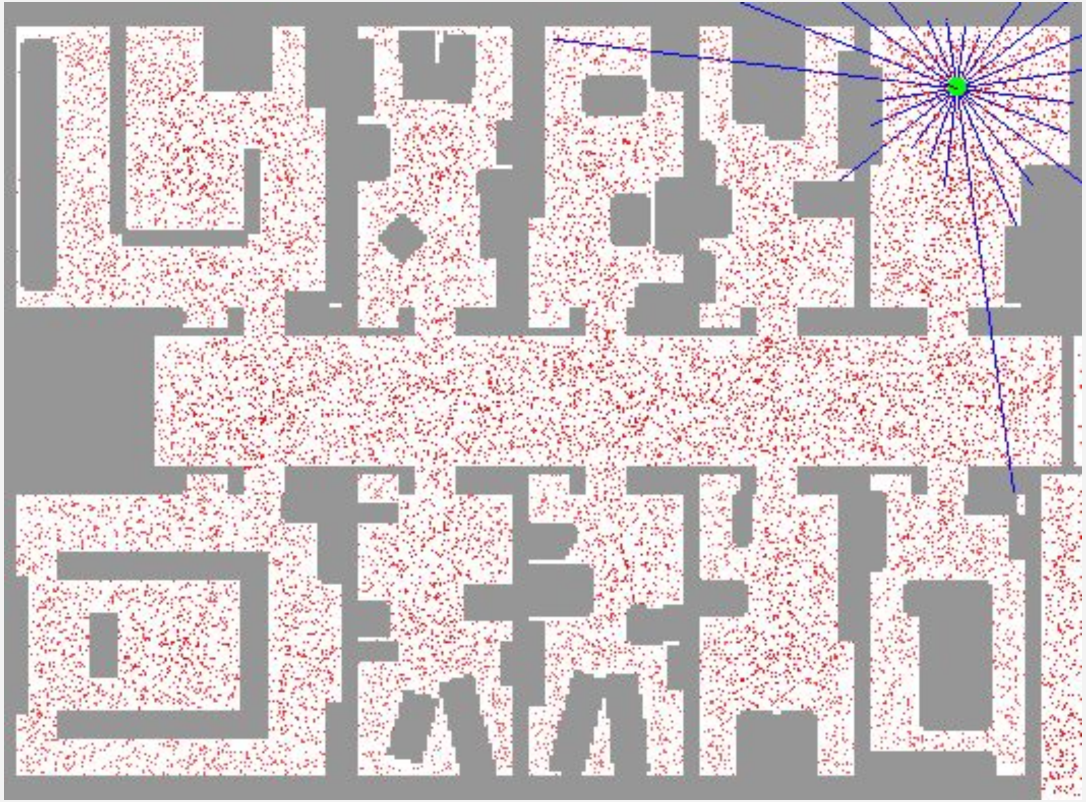


(a) Multinomial (b) Stratified (c) Systematic

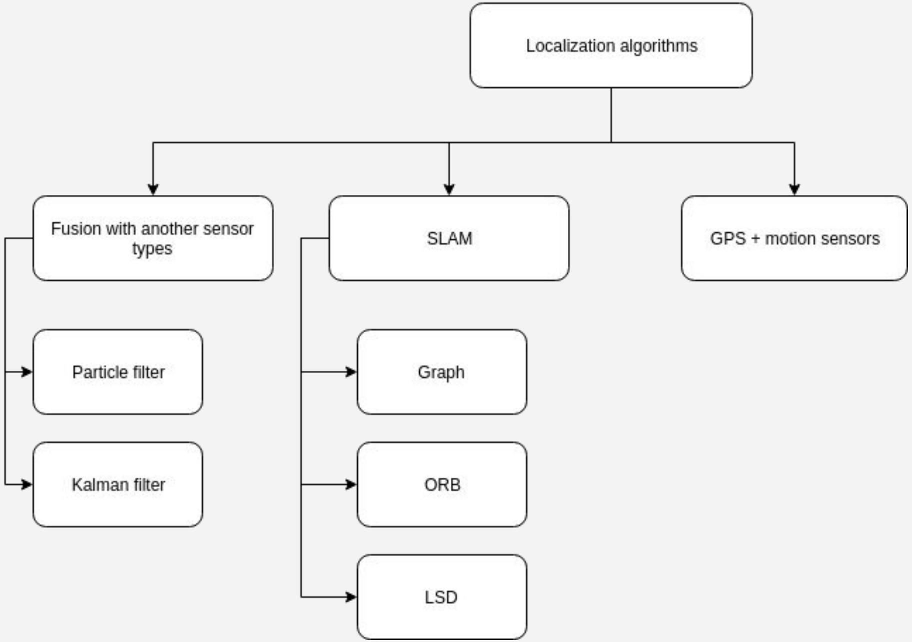
ФИЛЬТР ЧАСТИЦ
ПРИМЕР



ГЛОБАЛЬНАЯ ЛОКАЛИЗАЦИЯ
И KIDNAPPED ROBOT PROBLEM

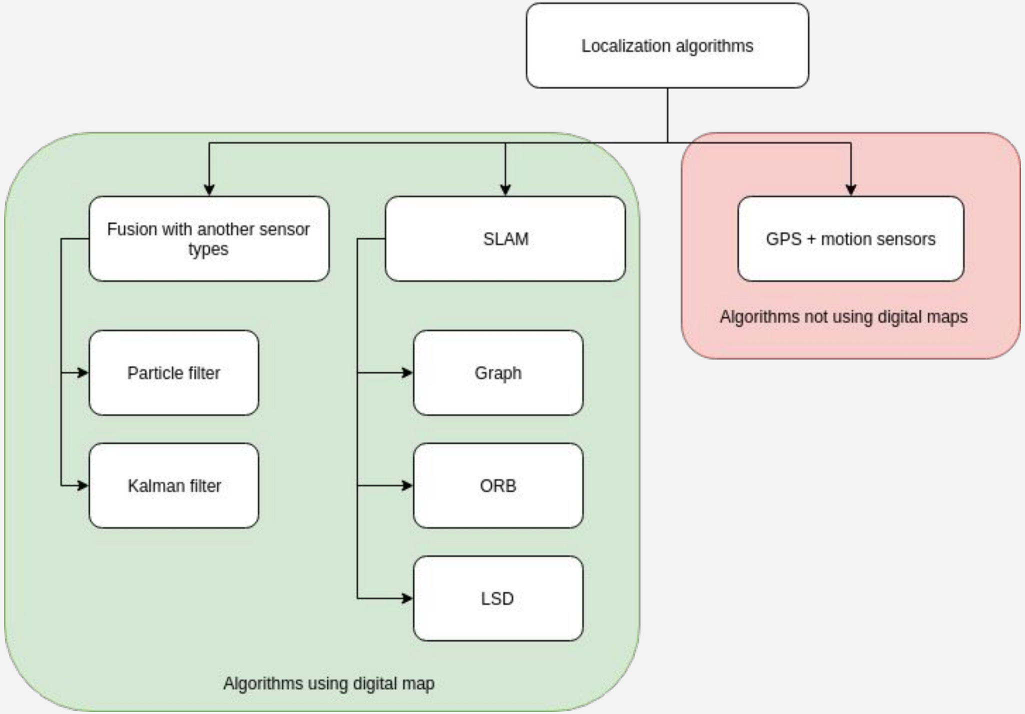


ЧТО ВЫБРАТЬ?



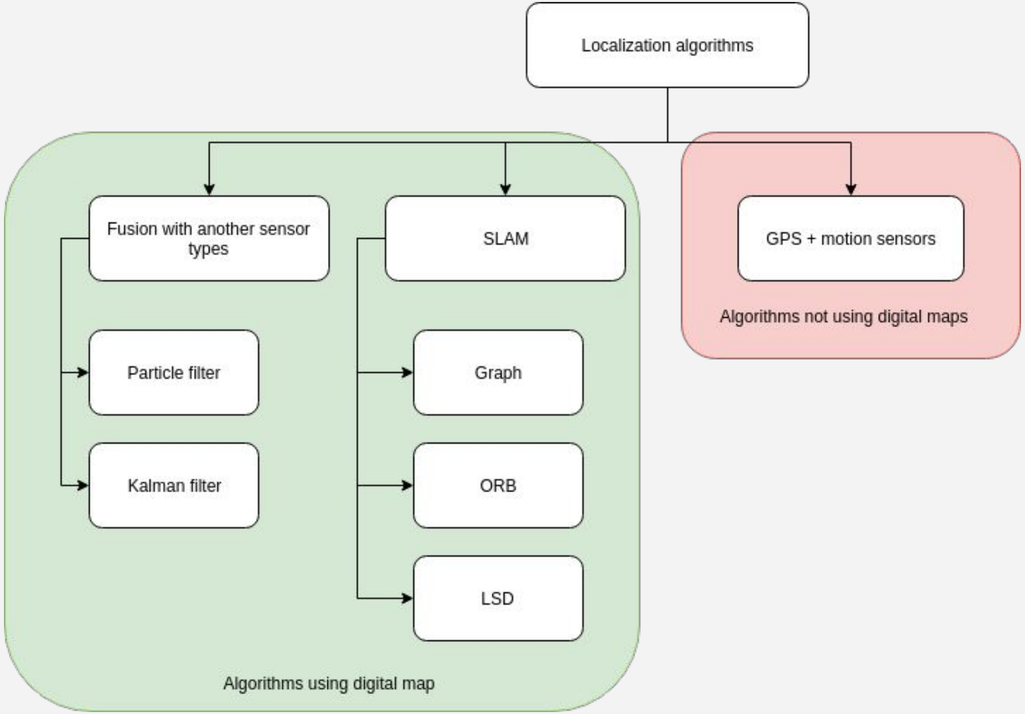
	Фильтр Калмана	**Фильтр частиц**
Тип шума	Гауссов, одномодальный	Любое распределение, мультимодальный
Решение	Оптимальное	Приближение
Реализация	Сложная (необходим вывод физической модели)	Простая
Устойчивость	Средняя	Высокая
Глобальная локализация	Нет	Да
Скорость вычислений	Быстрая	Зависит от реализации (как правило, медленнее)

ЧТО ВЫБРАТЬ?



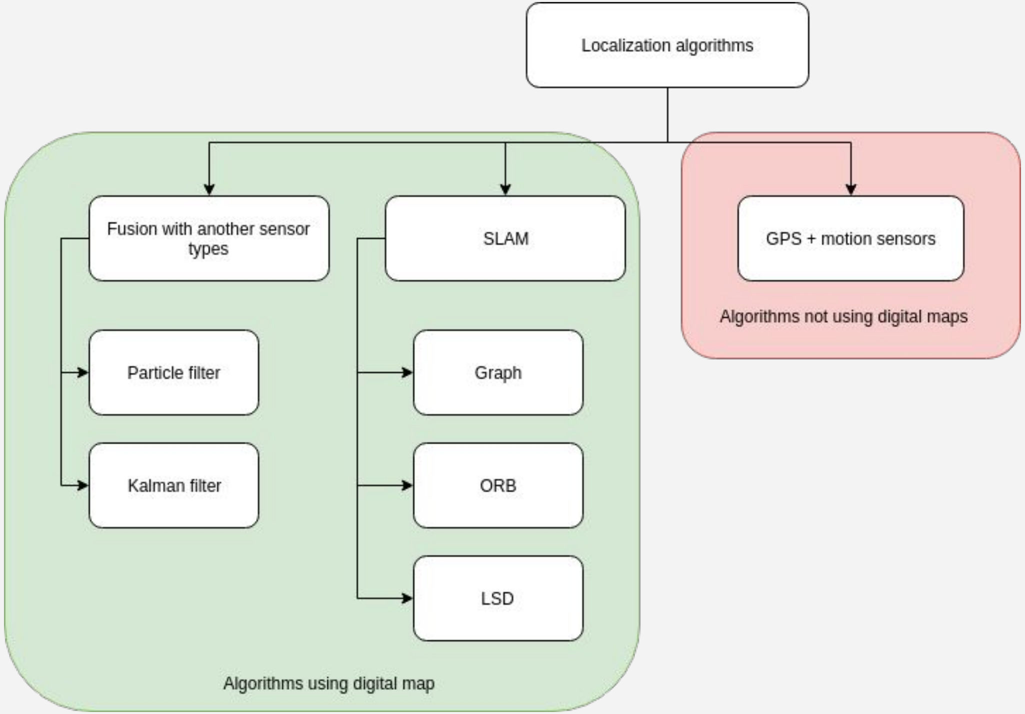
	Фильтр Калмана	**Фильтр частиц**
Тип шума	Гауссов, одномодальный	Любое распределение, мультимодальный
Решение	Оптимальное	Приближенное
Реализация	Сложная (необходим вывод физической модели)	Простая
Устойчивость	Средняя	Высокая
Глобальная локализация	Нет	Да
Скорость вычислений	Быстрая	Зависит от реализации (как правило, медленнее)

ЧТО ВЫБРАТЬ?



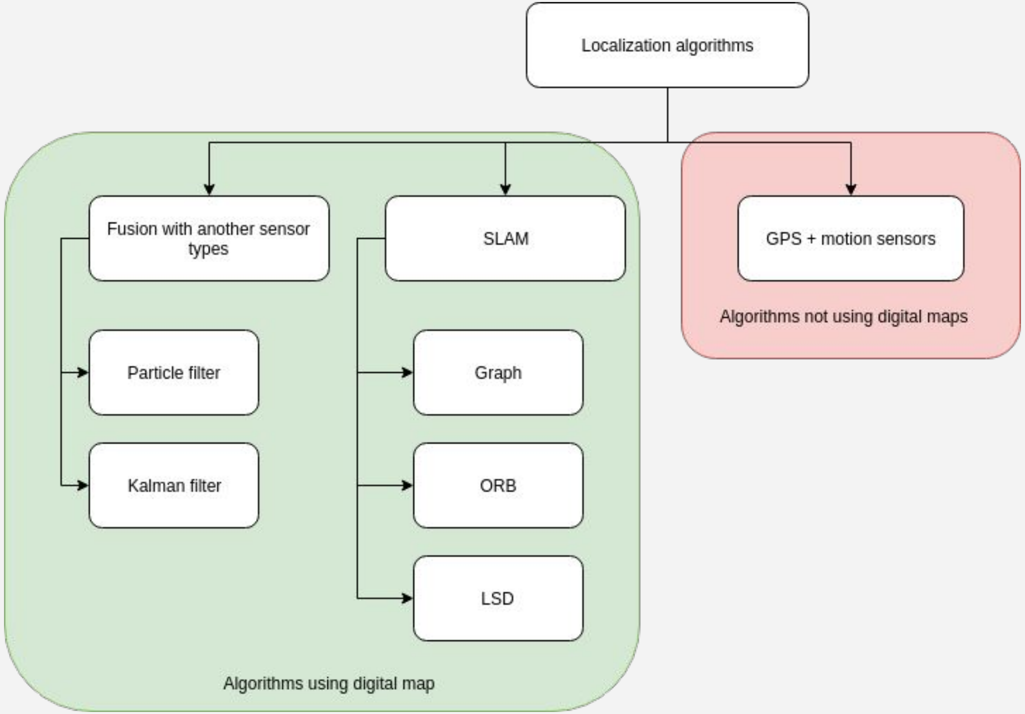
	Фильтр Калмана	**Фильтр частиц**
Тип шума	Гауссов, одномодальный	Любое распределение, мультимодальный
Решение	Оптимальное	Приближенное
Реализация	Сложная (необходим вывод физической модели)	Простая
Устойчивость	Средняя	Высокая
Глобальная локализация	Нет	Да
Скорость вычислений	Быстрая	Зависит от реализации (как правило, медленнее)

ЧТО ВЫБРАТЬ?



	РАСШИРЕННЫЙ **фильтр Калмана**	**Фильтр частиц**
Тип шума	Любое одномодальное	Любое распределение, мультимодальный
Решение	Приближенное	Приближенное
Реализация	Сложная (необходим вывод физической модели)	Простая
Устойчивость	Средняя	Высокая
Глобальная локализация	Нет	Да
Скорость вычислений	Быстрая	Зависит от реализации (как правило, медленнее)

ЧТО ВЫБРАТЬ?



	РАСШИРЕННЫЙ фильтр Калмана	Фильтр частиц
Тип шума	Любое одномодальное	Любое распределение, мультимодальный
Решение	Приближенное	Приближенное
Реализация	Сложная (необходим вывод физической модели)	Простая
Устойчивость	Средняя	Высокая
Глобальная локализация	Нет	Да
Скорость вычислений	Быстрая	Зависит от реализации (как правило, медленнее)

ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ

1. How a Kalman filter works, in pictures

2. Probabilistic Robotics. Главы 1, 2, 3, 4, 7, 8.

3. Cyrill Stachniss. SLAM -Course

ИНФОРМАЦИЯ О ПРЕЗЕНТАЦИИ

Эта презентация была подготовлена Олегом Шипитько в рамках курса “Моделирование колесных роботов” кафедры когнитивных технологий Московского физико-технического института (МФТИ). Автор выражает благодарность, авторам, чьи материалы были использованы в презентации. В случае, если вы обнаружили в презентации свои материалы, свяжитесь со мной, для включения в список авторов заимствованных материалов.

This presentation was prepared by Oleg Shipitko as part of the “Mobile Robotics” course at the Department of Cognitive Technologies, Moscow Institute of Physics and Technology. The author is grateful to the authors whose materials were used in the presentation. If you find your materials in a presentation, contact me to be included in the list of contributing authors.