

**Технічні науки**

**УДК 004.021, 004.896**

**Михалько Віталій Геннадійович**

студент

Національний технічний університет України  
«Київський політехнічний інститут»

**Михалько Виталий Геннадьевич**

студент

Национальный технический университет Украины  
«Киевский политехнический институт»

**Mykhalko V. H.**

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**Круш Ігор Володимирович**

студент

Національний технічний університет України  
«Київський політехнічний інститут»

**Круш Игорь Владимирович**

студент

Национальный технический университет Украины  
«Киевский политехнический институт»

**Krush I. V.**

student

National Technical University of Ukraine «Kyiv Polytechnic Institute»

**ЗАСТОСУВАННЯ ФІЛЬТРУ КАЛМАНА ДЛЯ ВИРІШЕННЯ  
ПРОБЛЕМИ ЛОКАЛІЗАЦІЇ РОБОТА  
ИСПОЛЬЗОВАНИЕ ФИЛЬТРА КАЛМАНА ДЛЯ РЕШЕНИЯ  
ПРОБЛЕМЫ ЛОКАЛИЗАЦИИ РОБОТА  
APPLICATION OF KALMAN FILTER TO SOLVE LOCALIZATION  
PROBLEM OF ROBOT**

**Анотація:**

Досліджено застосування фільтру Калмана для вирішення проблеми локалізації робота у середовищі. Описано алгоритм на прикладі руху робота з одним сенсором в одновимірному просторі. Наведено програмну реалізацію фільтру Калмана. Проведено експеримент для порівняння результатів застосування фільтру Калмана відносно результатів прямих вимірювань сенсорів. Зроблено висновки щодо доцільності застосування фільтру Калмана для задачі локалізації.

**Аннотация:**

Исследовано применение фильтра Калмана для решения проблемы локализации робота в среде. Описан алгоритм на примере движения робота с одним сенсором в одномерном пространстве. Приведена программная реализация фильтра Калмана. Проведен эксперимент для сравнения результатов использования фильтра Калмана относительно результатов прямых измерений сенсоров. Сделаны выводы относительно целесообразности применения фильтра Калмана для задачи локализации.

**Summary:**

Application of Kalman filter for solving localization problem. Description of algorithm on example of robot with single sensor in one-dimensional space. Programmatic implementation of Kalman filter. In order to compare results of Kalman filter and direct measurements experiment was performed. Conclusions were made regarding advantages of Kalman filter application in localization problem.

**Ключові слова:** задача локалізації, робототехніка, фільтр Калмана, нормальний розподіл, динамічні системи.

**Ключевые слова:** задача локализации, робототехника, фильтр Калмана, нормальное распределение, динамические системы.

**Key words:** localization problem, robotics, Kalman filter, normal distribution, dynamic systems.

## Вступ

В наші часи робототехніка набуває все більшої популярності. Автономні машини і роботи зараз активно застосовуються в промисловому виробництві, вони можуть замінити людей в умовах небезпечних ситуацій, а також допомагати людям в виконанні повсякденних завдань. Однією з основних задач ХХІ сторіччя у галузі робототехніки постала розробка автопілотів для автомобілів і інших рухомих платформ. Ця задача є дуже перспективною, бо зважаючи на те, що методи штучного інтелекту і машинного навчання в багатьох прикладних завданнях вже перевершують людські можливості, існує значна ймовірність зменшення аварійності на дорогах і збільшення безпеки пересування в цілому.

Основними задачами для побудови автоматизованих рухомих платформ є:

- локалізація
- контроль рухів
- пошук предметів
- одночасна локалізація і побудова карт

Ключовою з них є задача локалізації, бо якщо робот не знає, де він знаходиться, йому буде дуже складно визначити, які дії робити далі. Для того, щоб визначити місцезнаходження робота, можна використовувати різні датчики, такі як камери, ехолотатори, GPS та інші. Але основною проблемою є те, що дані з цих датчиків є часто неточними, через це потрібно застосовувати додаткові техніки, щоб дати більш ймовірну оцінку знаходження робота.

В цій статті буде досліджено застосування фільтру Калмана для вирішення задачі локалізації. Зокрема, буде змодельовано рухому платформу і проведено аналіз застосування фільтру Калмана для

вирішення задачі локалізації, а також будуть визначені його переваги в порівнянні з прямим використанням даних з сенсорів.

### **Фільтр Калмана**

Фільтр Калмана - це алгоритм, що будує оцінку значення невідомої змінної за допомогою поєднання поточних вимірів для цієї величини та її передбаченого значення. Цей алгоритм є більш точним за ті, що базуються тільки на вимірах із застосуванням формули Байеса. Фільтр названий на честь Рудольфа Калмана, одного з перших розробників цієї теорії.

Фільтр Калмана використовується у випадку, коли існує невизначена інформація про певну динамічну систему і генерує припущення про те, що система буде робити далі. Коректна робота алгоритму буде забезпечена навіть за умови дуже неоднорідних вхідних даних.

Фільтр Калмана ідеально підходить для систем, що неперервно змінюються. Серед його переваг є також те, що він використовує відносно невелику кількість пам'яті та обчислювальних потужностей, що робить алгоритм придатним для вирішення задач у режимі реального часу та у випадку вбудованих систем.

Алгоритм працює у два кроки. На кроці прогнозування фільтр Калмана генерує оцінку подальшого стану системи, використовуючи інформацію про попередні стани. На кроці вимірювань система виконує нові виміри параметрів. Після цього результати прогнозування і вимірювань поєднуються, щоб отримати більш точну оцінку.

Фільтр Калмана - це рекурсивний алгоритм. Його робота в режимі реального часу забезпечується попередньо обчисленими результатами, поточними вимірами та матрицею невизначеності.

Алгоритм не вимагає, щоб похибки були нормально розподілені. Однак, якщо всі похибки розподілені по Гаусу то ймовірнісна оцінка є точнішою.

Є також варіації цього алгоритму, такі як, розширений фільтр Калмана та беззапаховий фільтр Калмана, що працює у випадку нелінійних систем.

### **Застосування фільтру Калмана для задачі локалізації**

Фільтр Калмана можна застосувати для проблеми локалізації робота у просторі. Для цього треба описати робота як динамічну систему з набором параметрів. Зазвичай у робота є компоненти, які дозволяють йому рухатись, а також сенсори, за допомогою яких можна отримувати інформацію про навколишнє середовище.

Нехай у нас є рухома платформа з сенсором, яка може рухатись у двох напрямках(рис. 1).



Рис. 1. Графічна модель рухомої платформи

Сенсор використовується як пристрій, що може вимірювати відстань від об'єкта до перешкоди. В контексті даної ситуації його реалізація може варіюватись від ультразвукових або інфрачервоних датчиків до камер

різного типу. Як відомо, не існує ідеальних вимірювальних приладів і у будь-якому вимірюванні велика ймовірність похибки. Такі похибки зручно описувати за законом Гауса.

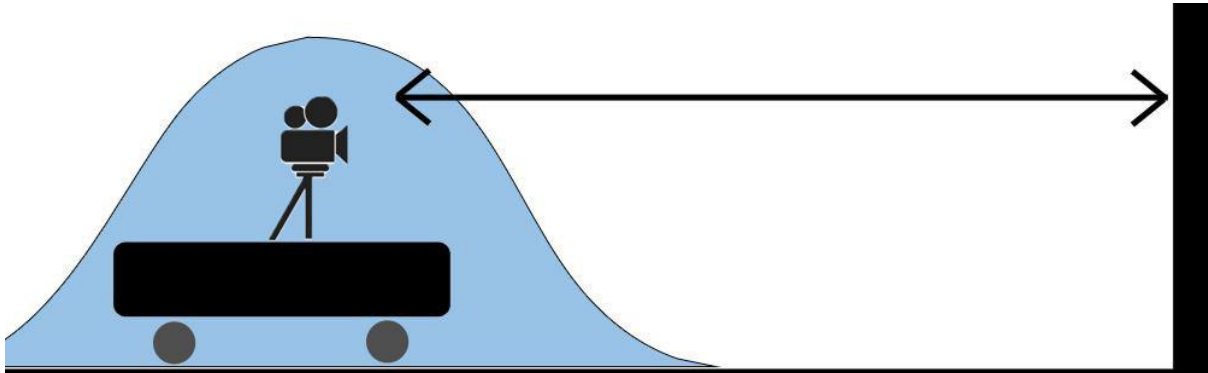


Рис. 2. Розподіл ймовірностей положення робота у просторі в залежності від показів сенсору

В кожен момент часу стан системи описується двома параметрами: положенням і швидкістю. Це можна записати у вигляді вектора:

$$\vec{s}_t = \begin{bmatrix} x_t \\ v_t \end{bmatrix}$$

В даному випадку розглянуто одновимірний рух, хоча вказані нижче техніки легко узагальнити і на багатовимірні випадки. Стан реальної системи, крім положення і швидкості, часто характеризується і іншими параметрами, такими як керуючі сигнали або прискорення. Але для спрощення моделі не будемо їх враховувати.

Використовуючи розподіл Гауса, стан системи в кожен момент часу характеризується вектором математичного сподівання і матрицею коваріацій:

$$\vec{\mu} = \begin{bmatrix} \mu_x \\ \mu_v \end{bmatrix}$$

$$C = \begin{bmatrix} \Sigma_x & \Sigma_{xv} \\ \Sigma_{vx} & \Sigma_v \end{bmatrix}$$

Для передбачення стану системи в наступний момент часу, можна використати рівняння з кінематики:

$$x_{t+1} = x_t + v_t * \Delta t$$

$$v_{t+1} = v_t$$

У матричному вигляді це виглядає таким чином:

$$\vec{s}_{t+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} * \vec{s}_t = F_t * \vec{s}_t$$

$$C_{t+1} = F_t * C_t * F_t^T$$

У вимірів, які робляться в якийсь момент часу, також є похибка, розподілена за нормальним законом, яку також можна описати у матричній формі:

$$\vec{m} = \begin{bmatrix} m_x \\ m_v \end{bmatrix}$$

$$L = \begin{bmatrix} \Sigma_x & \Sigma_{xv} \\ \Sigma_{vx} & \Sigma_v \end{bmatrix}$$

Отже, в певний момент часу у нас є дві оцінки положення робота: передбачена і виміряна. Основна ідея фільтру Калмана полягає в тому, що об'єднання цих двох оцінок значно підвищує точність результату. Так як обидві оцінки розподілені за законом Гаусом, то для отримання спільної

оцінки, два розподіла потрібно перемножити. Результат буде також розподілений за нормальним законом.

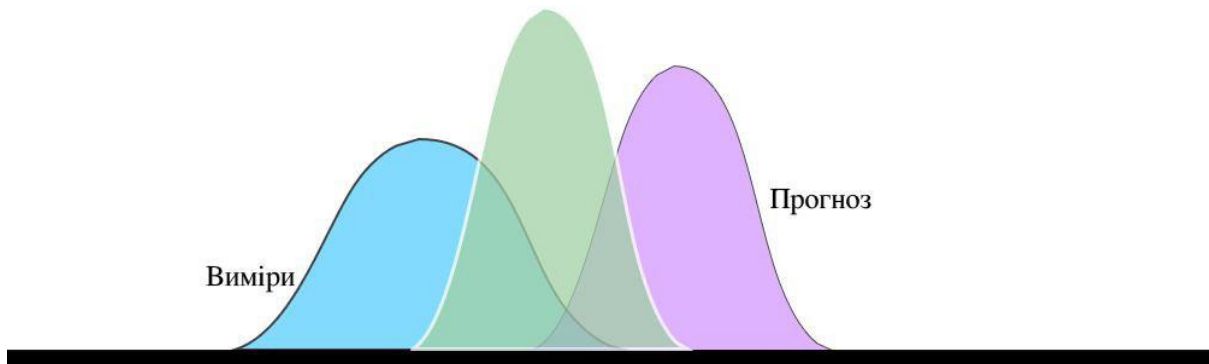


Рис. 3. Об'єднання оцінок

Як відомо, математичне сподівання і середньоквадратичне відхилення для добутку нормальних розподілів можна знайти за наступними формулами:

$$k = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

$$\mu' = \mu_1 + k(\mu_2 - \mu_1)$$

$$(\sigma^2)' = \sigma_1^2 - k\sigma_1^2$$

Для багатовимірного розподілу Гауса, формули добутку виглядають аналогічно:

$$K = \Sigma_1 * (\Sigma_1 + \Sigma_2)^{-1}$$

$$\vec{\mu}' = \vec{\mu}_1 + K * (\vec{\mu}_2 - \vec{\mu}_1)$$



$$\Sigma' = \Sigma_1 - K * \Sigma_1$$

Використовуючи наведену вище схему, можна реалізувати алгоритм фільтру Калмана. Псевдокод для нього виглядає таким чином:

```
function kalman_estimate(  
    prev_vector,  
    prev_cov_matrix,  
    transform_matrix,  
    measurements_vector,  
    measurements_cov  
) {  
    # передбачення нових параметрів  
    predicted_vector = transform_matrix * prev_vector  
    predicted_cov_matrix = transform_matrix * prev_cov_matrix  
                        * transpose(transform_matrix)  
    # поєднання вимірів із передбаченням  
    K = prev_cov_matrix * inverse(  
        prev_cov_matrix + measurements_cov_matrix  
    )  
    mu = predicted_vector + K * (measurements_vector - predicted_vector)  
    cov = predicted_cov_matrix - K * predicted_cov_matrix  
    return mu, cov  
}
```

## Експеримент

Для аналізу результатів вирішення задачі локалізації за допомогою фільтру Калмана і порівняння цих результатів зі звичайними вимірами із сенсорів була написана програма, яка моделює рух робота, а також виміри сенсора, помилка яких розподілена за нормальним законом.

Псевдокод програми для моделювання робота і проведення експерименту наведено нижче:

```
function experiment() {  
    # накопичені помилки  
    total_measurement_error = 0.0  
    total_kalman_error = 0.0
```

```
# початкова відстань
distance = 100.0

# математичне сподівання і середньоквадратичне відхилення швидкості
v_mean = 10.0
v_std_error = 0.5

# середньоквадратичне відхилення результатів вимірів
measurement_std_error = 5.0

# час
t = 0.0
delta_t = 1.0

# початкові значення вимірів
measurement = gauss(distance, measurement_std_error)
prev_mu = [measurement, v_mean]
prev_cov = [[measurement_std_dev**2, 0], [0, measurement_std_dev**2]]
# матриця трансформації
transform_matrix = [[1, -delta_t], [0, 1]]

for i in range(5) {
    # швидкість
    v = gauss(v_mean, v_std_error)

    t += delta_t
    distance -= v * delta_t

    # результати виміру відстані сенсором
    measured_distance = gauss(distance, measurement_std_dev)

    # оцінка швидкості
    velocity_estimate = (prev_mu[0] - measured_distance) / delta_t

    measurements_vector = [measured_distance, velocity_estimate]
    measurements_cov_matrix = [ [measurement_std_error**2, 0],
                                [0, measurement_std_error**2] ]
    mu, cov = kalman_estimate(
        prev_mu, prev_cov, transform_matrix,
        measurements_vector, measurements_cov
    )
}
```

```
# обчислення помилки
total_measurement_error +=abs(distance - measurement)
total_kalman_error +=abs(distance - mu[0])

prev_mu = mu
prev_cov = cov
}
}
```

Програма виконувалась 3 рази, щоб порівняти результати обох підходів для вирішення задачі локалізації при різній точності вимірювання сенсорів. Результати наведено нижче:

**1. Сенсор з високою точністю (measurement\_std\_error = 1.0)**

Момент часу	t1	t2	t3	t4	t5
Справжня відстань	90.23	80.29	70.64	61.35	51.58
Оцінка відстані сенсором	90.88	80.16	68.88	60.87	51.25
Оцінка відстані фільтром Калмана	90.05	80.36	70.26	60.49	50.76

Накопичені помилки:

1. Сумарна помилка оцінки відстані сенсором: **3.35**
2. Сумарна помилка оцінки відстані фільтром Калмана: **2.31**

**2. Сенсор з середньою точністю (measurement\_std\_error = 5.0)**

Момент часу	t1	t2	t3	t4	t5
Справжня	89.80	79.69	69.21	59.68	49.76

відстань					
Оцінка відстані сенсором	87.39	79.90	69.42	63.31	42.16
Оцінка відстані фільтром Калмана	86.85	78.04	68.94	60.46	50.05

Накопичені помилки:

1. Сумарна помилка оцінки відстані сенсором: **14.08**
2. Сумарна помилка оцінки відстані фільтром Калмана: **5.94**

3. Сенсор з низькою точністю (measurement\_std\_error = **10.0**)

Момент часу	t1	t2	t3	t4	t5
Справжня відстань	89.68	79.68	70.50	60.54	51.45
Оцінка відстані сенсором	75.74	83.76	83.74	45.07	50.40
Оцінка відстані фільтром Калмана	79.64	73.05	68.77	58.61	50.18

Накопичені помилки:

1. Сумарна помилка оцінки відстані сенсором: **47.77**
2. Сумарна помилка оцінки відстані фільтром Калмана: **21.61**

### Аналіз

Результати експерименту показують, що застосування фільтру Калмана дозволяє збільшити точність оцінювання відстані порівняно з прямими вимірами сенсорів. Також зазначимо, що ефективність використання фільтру Калмана особливо помітна при низькій точності сенсорів. Це показує доцільність використання цього алгоритму при

вирішення задачі локалізації роботів, так як в реальних умовах сенсори часто надають результати з вагомою похибкою.

### **Висновки**

У даній роботі було описано застосування фільтру Калмана у вирішенні задачі локалізації робота, наведено приклад програмної реалізації цього алгоритму, а також проведено експеримент, результати якого показали, що застосування фільтру Калмана підвищує точність визначення положення робота порівняно з прямими даними сенсорів. Різниця у коректності оцінки положення робота особливо помітна при використанні сенсорів з низькою точністю.

Варто також зазначити, що крім підвищення точності оцінки, фільтр Калмана характеризується невисокою алгоритмічною складністю, що дозволяє проводити обчислення в режимі реального часу, а також використовувати його на системах, які не надають великої обчислювальної потужності.

### **Література:**

1. R. E. Kalman, *A new approach to linear filtering and prediction problems*. J. Basic Eng., vol. 82, no. 1, Mar. 1960.
2. B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. New York: Dover, 2005.
3. S. Thrun, D. Fox, W. Burgard, *Probabilistic Robotics*. MIT Press, 2006.
4. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc., 1979.
5. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2002.