

# Analisis

## 1. Introduction to Localization and Filtering Theory

Teori lokalisasi dan penyaringan (filtering theory) merupakan dasar penting dalam robotika dan sistem otonom. Teori ini menyediakan alat matematis dan komputasi untuk memperkirakan posisi serta pergerakan robot di lingkungan yang penuh ketidakpastian. Program-program yang diberikan menunjukkan konsep utama dan teknik yang digunakan dalam lokalisasi, mulai dari Kalman Filter hingga Particle Filter, serta penggabungan data sensor. Berikut analisis menyeluruhnya:

### 1. Kalman Filter untuk Lokalisasi

- **Tujuan:** Kalman Filter digunakan pada sistem linear yang memiliki noise dengan distribusi Gaussian. Filter ini memprediksi keadaan sistem dan mengoreksinya berdasarkan pengukuran, dengan menggabungkan pengetahuan sebelumnya dan pengamatan yang mengandung noise.
- **Kelebihan:**
  - Sederhana dan efisien untuk sistem linear.
  - Memberikan estimasi posisi yang halus dengan memanfaatkan dinamika sistem dan pengukuran.
- **Contoh Program:** Program menunjukkan estimasi posisi robot dalam pergerakan linear. Hasilnya berupa jalur posisi sebenarnya, posisi yang diukur (dengan noise), dan posisi yang diestimasi menggunakan Kalman Filter.

### 2. Particle Filter untuk Lokalisasi

- **Tujuan:** Particle Filter digunakan untuk sistem non-linear dengan noise non-Gaussian. Metode ini mengestimasi posisi dengan merepresentasikan distribusi probabilitas menggunakan partikel-partikel.
- **Kelebihan:**
  - Dapat digunakan pada sistem yang kompleks dan non-linear.
  - Fleksibel terhadap berbagai jenis noise.
- **Contoh Program:** Program menunjukkan distribusi partikel yang merepresentasikan kemungkinan posisi robot. Setelah pembaruan bobot dan resampling, estimasi posisi akhir dihitung dari partikel yang paling relevan.

### 3. Extended Kalman Filter (EKF)

- **Tujuan:** EKF adalah pengembangan Kalman Filter untuk sistem non-linear. Ini dilakukan dengan melakukan linearisasi pada dinamika sistem.
- **Kelebihan:**
  - Efisien untuk sistem non-linear dengan noise Gaussian.
  - Cocok untuk robotika, seperti navigasi dan perencanaan jalur.
- **Contoh Program:** EKF memprediksi dan mengoreksi posisi robot berdasarkan model non-linear, menghasilkan estimasi jalur yang akurat.

### 4. Gabungan Sensor (IMU dan Lidar)

- **Tujuan:** Penggabungan data sensor (sensor fusion) dilakukan untuk meningkatkan akurasi dengan memanfaatkan kelebihan masing-masing sensor.
- **Kelebihan:**
  - IMU memberikan data pergerakan relatif, sedangkan Lidar memberikan data absolut seperti jarak.
  - Kombinasi ini menghasilkan estimasi posisi yang lebih andal.
- **Contoh Program:** Data IMU dan Lidar digabungkan menggunakan metode rata-rata berbobot (weighted average) untuk menghasilkan estimasi posisi.

## 5. Keseluruhan Implementasi

- Program-program ini mencakup spektrum metode penyaringan, dari teknik yang lebih sederhana seperti Kalman Filter hingga metode yang lebih kompleks seperti Particle Filter dan penggabungan sensor.
- **Perbandingan Utama:**
  - Kalman Filter cocok untuk sistem linear dengan noise Gaussian.
  - Extended Kalman Filter memperluas kemampuan Kalman Filter untuk sistem non-linear.
  - Particle Filter lebih fleksibel tetapi membutuhkan lebih banyak komputasi.
  - Sensor fusion menggabungkan data dari berbagai sensor untuk meningkatkan akurasi.

## 2. Implementasi Kalman Filter untuk Lokalitas Robot pada Webots

Kalman Filter adalah metode yang sangat berguna untuk meningkatkan estimasi posisi robot dengan menggabungkan prediksi dari model gerakan dan pembaruan berdasarkan pengukuran sensor. Berikut analisis dari kode yang diberikan:

### 1. Prediksi dan Pembaruan dalam Kalman Filter

- **Langkah Prediksi:**
  - Nilai encoder roda digunakan untuk menghitung estimasi pergerakan rata-rata robot (diberikan oleh variabel  $u$ ).
  - Posisi awal robot ( $x$ ) diperbarui dengan menambahkan nilai pergerakan ( $x_{pred} = x + u$ ).
  - Ketidakpastian posisi ( $P$ ) juga diperbarui dengan menambahkan noise proses ( $P_{pred} = P + Q$ ), di mana noise proses ( $Q$ ) bernilai 0.1 dalam kode.
- **Langkah Koreksi (Pembaruan):**
  - Data sensor jarak ( $z$ ) digunakan untuk memperbarui estimasi posisi. Kalman Gain ( $K$ ) dihitung untuk menentukan seberapa besar pengaruh pengukuran dibandingkan dengan prediksi.
  - Posisi robot diperbarui menggunakan rumus  $x = x_{pred} + K * (z - x_{pred})$ . Di sini, Kalman Gain ( $K$ ) memberikan bobot terhadap pengukuran baru.
  - Ketidakpastian ( $P$ ) diperbarui menjadi lebih kecil dengan rumus  $P = (1 - K) * P_{pred}$ , sehingga estimasi menjadi lebih akurat.

## 2. Input dan Pengukuran Sensor

- **Input dari Model Gerakan:**
  - Data dari encoder roda kiri dan kanan memberikan informasi tentang pergerakan robot, yaitu estimasi pergeseran posisi rata-rata (variabel  $u$ ).
- **Pengukuran dari Sensor:**
  - Data dari sensor jarak memberikan pengukuran posisi robot relatif terhadap objek atau batas tertentu. Namun, data pengukuran ini sering kali mengandung noise.

## 3. Manfaat Penggunaan Kalman Filter

- **Penggabungan Informasi:**
  - Kalman Filter secara optimal menggabungkan prediksi dari model gerakan dan data pengukuran sensor berdasarkan tingkat ketidakpastian masing-masing.
- **Mengurangi Ketidakpastian:**
  - Dengan setiap iterasi, ketidakpastian posisi ( $P$ ) menjadi lebih kecil, menghasilkan estimasi yang semakin akurat.
- **Mengatasi Noise pada Data Sensor:**
  - Kalman Filter dirancang untuk mengatasi noise atau kesalahan pada data pengukuran sensor, memberikan estimasi yang lebih andal.

## 4. Penerapan Real-Time

- Kode ini dirancang untuk dijalankan dalam lingkungan simulasi real-time menggunakan platform Webots. Pada setiap langkah waktu ( $TIME\_STEP$ ), posisi robot dihitung dan diperbarui berdasarkan data dari sensor jarak dan encoder roda.
- Estimasi posisi robot dicetak dalam waktu nyata, memungkinkan pemantauan posisi robot secara langsung.

## 5. Keterbatasan dan Potensi Peningkatan

- **Asumsi Linearitas:**
  - Model dalam kode mengasumsikan bahwa gerakan robot bersifat linear, yang mungkin kurang akurat untuk pergerakan melengkung atau lingkungan yang lebih kompleks.
- **Noise Tetap:**
  - Noise proses ( $Q$ ) dan noise pengukuran ( $R$ ) dalam kode diatur sebagai nilai tetap. Dalam aplikasi nyata, nilai ini dapat bervariasi dan memerlukan penyetelan lebih lanjut untuk meningkatkan akurasi.
- **Penggabungan Multi-Sensor:**
  - Kode ini hanya menggunakan data dari satu sensor jarak. Untuk meningkatkan akurasi pada aplikasi yang lebih kompleks, penggabungan data dari beberapa sensor, seperti IMU atau Lidar, dapat digunakan.

