Sultan Fa'iq Athallah
1103213020

Robotika Week 12

1. Implementasi *Filter Kalman* untuk Estimasi Posisi Robot

Deskripsi:
Filter Kalman adalah algoritma rekursif yang digunakan untuk memperkirakan keadaan sistem dinamis dengan mengurangi dampak *noise* pada pengukuran. Dalam hal ini, algoritma digunakan untuk memperkirakan posisi robot berdasarkan data pengukuran yang terpengaruh *noise*.

Hasil Analisis:

- Menghasilkan estimasi posisi yang lebih akurat dibandingkan data mentah dengan *noise*.
- Matriks *Kalman Gain* berfungsi menyesuaikan kontribusi pengukuran pada estimasi, sehingga mengurangi dampak *noise*.
- Visualisasi menunjukkan estimasi yang stabil dan mendekati posisi sebenarnya.

---

2. Implementasi *Filter Partikel* untuk Estimasi Posisi Robot

Deskripsi:
Filter Partikel adalah metode berbasis Monte Carlo untuk memperkirakan keadaan sistem non-linear atau non-Gaussian. Dalam simulasi ini, metode ini digunakan untuk memperkirakan posisi robot dengan memanfaatkan distribusi probabilitas partikel.

Hasil Analisis:

- Efektif menangani *noise* pengukuran yang kompleks dan distribusi non-linear.
- Proses *resampling* memastikan partikel relevan mendominasi estimasi, sementara partikel dengan bobot rendah dieliminasi.
- Grafik menunjukkan hasil estimasi yang akurat meskipun *noise* pada pengukuran tinggi.

---

3. Implementasi Lokalisasi dengan Sensor IMU dan Lidar

Deskripsi:
Sensor IMU (*Inertial Measurement Unit*) memberikan informasi tentang kecepatan dan orientasi, sementara sensor Lidar menyediakan data posisi melalui pengukuran jarak. Keduanya digabungkan untuk meningkatkan akurasi lokalisasi robot.

Hasil Analisis:

- Data IMU memiliki *noise* tinggi dan rentan terhadap *drift*, sedangkan Lidar lebih stabil.
- Fusi data, seperti rata-rata berbobot, menghasilkan estimasi posisi yang lebih akurat.
- Kombinasi IMU dan Lidar meningkatkan akurasi dibandingkan penggunaan sensor tunggal.

---

4. Implementasi Simulasi *Extended Kalman Filter* (EKF) untuk Navigasi

Deskripsi:
EKF adalah versi non-linear dari *Filter Kalman* yang digunakan untuk memperkirakan keadaan sistem dengan dinamika non-linear. Dalam simulasi, metode ini diterapkan untuk navigasi robot.

Hasil Analisis:

- EKF mampu memperkirakan posisi dan orientasi robot meskipun dinamika sistem non-linear.
- Matriks Jacobian digunakan untuk proses linearisasi, sehingga estimasi lebih akurat.
- Visualisasi hasil simulasi menunjukkan EKF mampu mengurangi *error* posisi selama navigasi.

---

5. Implementasi *Particle Filter* untuk Navigasi

Deskripsi:
Metode *Particle Filter* digunakan untuk navigasi robot dengan memperkirakan posisi robot berdasarkan distribusi probabilitas. Teknik ini sangat cocok untuk skenario dengan *noise* tinggi atau model non-linear.

Hasil Analisis:

- Memberikan estimasi posisi yang akurat meskipun data pengukuran memiliki *noise* tinggi.
- Proses *resampling* menjaga relevansi partikel dengan bobot tinggi untuk memperbaiki estimasi.
- Simulasi menunjukkan keunggulan metode ini dalam skenario navigasi yang kompleks.

Analisis Kode Webots

1. Inisialisasi Robot dan Komponen

Motor dan Robot:

- Motor kiri dan kanan diatur ke mode kecepatan dengan posisi tak terbatas (float('inf')).
- Kecepatan awal motor diatur ke 0.

Sensor Jarak:

- Delapan sensor jarak diaktifkan menggunakan .enable() dengan interval waktu tertentu (TIME_STEP).
- Data dari sensor digunakan untuk mendeteksi rintangan di sekitar robot.

2. Logika Deteksi dan Penghindaran Rintangan

- Sensor jarak mendeteksi rintangan di depan, kiri, atau kanan robot berdasarkan ambang batas (OBSTACLE_THRESHOLD).
- Robot memiliki tiga kondisi utama:
    - Rintangan di depan: Robot mundur dan berbelok.
    - Rintangan di kiri: Robot berbelok ke kanan.

    - Rintangan di kanan: Robot berbelok ke kiri.
    - Tidak ada rintangan: Robot berjalan lurus.
- Logika ini memungkinkan robot bergerak tanpa tabrakan di lingkungan dinamis.

3. *Kalman Filter* untuk Estimasi Posisi

- **Variabel Awal**:
    - x: estimasi posisi awal.
    - P: ketidakpastian estimasi awal.
- **Proses Filter**:
    - Prediksi: Posisi diperhitungkan dari keadaan sebelumnya (x_pred) dan *noise* proses.
    - Pembaruan: Estimasi posisi diperbarui dengan data sensor (z) menggunakan *Kalman Gain* (K).
    - *Kalman Gain* mengontrol pengaruh data sensor terhadap estimasi posisi.
- **Pengukuran Sensor**:
    - Nilai pengukuran (z) diambil dari jarak terdekat yang dilaporkan sensor.

4. *Output*

- Estimasi posisi (x) dan nilai sensor dicetak di setiap iterasi, memberikan gambaran posisi robot dan interaksinya dengan lingkungan.

Hasil Analisis

Keunggulan *Kalman Filter*

- Memberikan estimasi posisi robot yang lebih akurat dengan menggabungkan data sensor yang mengandung *noise* dengan model pergerakan robot.
- *Noise* proses dan pengukuran diminimalkan secara dinamis melalui mekanisme pembaruan *Kalman Gain*.

Efisiensi Deteksi Rintangan

- Logika berbasis sensor jarak memungkinkan robot secara adaptif menghindari rintangan.
- Kombinasi *Kalman Filter* dan sensor jarak membuat robot lebih responsif terhadap perubahan lingkungan.

Peningkatan yang Disarankan

- **Model Pergerakan**: Masukkan data pergerakan aktual (kecepatan atau orientasi) untuk meningkatkan akurasi estimasi posisi.
- **Ambang Sensor**: Sesuaikan ambang deteksi rintangan (OBSTACLE_THRESHOLD) dengan kondisi lingkungan simulasi.
- **Fusi Data Sensor**: Gabungkan data dari beberapa sensor untuk meningkatkan akurasi dan ketahanan estimasi posisi.