

ANALISIS TUGAS ROBOTIK WEEK 13

ANALISIS SIMULASI PARAMETRIC FILTERING DI GOOGLE COLAB

Extended Kalman Filter (EKF): Navigasi Robot dengan GPS dan IMU

Extended Kalman Filter (EKF) adalah algoritma yang digunakan untuk memperkirakan posisi dan arah robot dengan menggabungkan data dari sensor GPS dan IMU. EKF dirancang untuk menangani sistem yang tidak linier dengan cara menyederhanakannya menggunakan turunan matematika yang disebut *Jacobian*. Teknik ini memungkinkan EKF bekerja dengan baik meskipun model sistemnya rumit.

Salah satu keunggulan EKF adalah kemampuannya mengurangi gangguan atau *noise* dari sensor, sehingga menghasilkan estimasi posisi yang lebih stabil dan akurat. Namun, EKF juga memiliki kelemahan, seperti ketergantungannya pada proses penyederhanaan (*linierisasi*) yang bisa menyebabkan kesalahan jika model yang digunakan tidak cukup akurat.

Selain itu, performa EKF sangat bergantung pada pengaturan parameter *noise* (biasanya disebut Q dan R). Jika parameter ini tidak diatur dengan benar, kinerjanya bisa menurun, terutama dalam kondisi lingkungan yang berubah-ubah secara cepat.

Meski memiliki keterbatasan, EKF tetap menjadi pilihan populer dalam sistem navigasi robot yang menggunakan GPS dan IMU, terutama di lingkungan dengan banyak gangguan yang dapat mempengaruhi akurasi sensor.

Unscented Kalman Filter (UKF): Navigasi Robot dengan GPS dan IMU

Unscented Kalman Filter (UKF) adalah algoritma yang digunakan untuk memperkirakan posisi dan orientasi robot dengan menggabungkan data dari sensor GPS dan IMU. Berbeda dengan EKF yang memerlukan proses linierisasi, UKF menggunakan metode *Unscented Transform* untuk menangani sistem non-linear dengan lebih akurat.

Keunggulan utama UKF terletak pada kemampuannya dalam menangani sistem yang sangat non-linear dan perubahan yang cepat dalam dinamika sistem. Dengan metode ini, UKF mampu menghasilkan estimasi yang lebih baik meskipun sistem memiliki kompleksitas tinggi dan lingkungan penuh dengan gangguan.

Namun, UKF memiliki beberapa kelemahan. Algoritma ini memerlukan perhitungan yang lebih kompleks dibandingkan EKF, sehingga membutuhkan sumber daya komputasi yang lebih besar. Selain itu, pengaturan parameter yang optimal sangat penting agar UKF dapat bekerja dengan baik.

Meskipun demikian, UKF menjadi pilihan yang sangat baik untuk aplikasi navigasi robot yang membutuhkan estimasi posisi dan orientasi yang akurat dalam lingkungan yang dinamis dan penuh ketidakpastian. Oleh karena itu, UKF sering digunakan dalam robotika modern yang mengandalkan kombinasi data dari GPS dan IMU untuk menentukan lokasi dengan presisi tinggi.

Tracking Objek Bergerak dengan Kalman Filter

Kalman Filter adalah algoritma yang sering digunakan untuk melacak objek bergerak dengan pola gerakan yang dapat diprediksi, seperti kendaraan atau manusia yang bergerak dalam jalur linier. Keunggulan utama dari Kalman Filter adalah kemampuannya untuk menyaring noise dari data sensor, sehingga memberikan estimasi posisi yang lebih akurat dan stabil.

Selain itu, Kalman Filter memiliki efisiensi komputasi yang tinggi, membuatnya sangat cocok untuk aplikasi pelacakan secara real-time. Algoritma ini bekerja dengan memprediksi posisi objek berdasarkan model gerakan dan memperbarui prediksi tersebut menggunakan data sensor yang masuk.

Namun, Kalman Filter memiliki keterbatasan saat menangani gerakan objek yang bersifat non-linear atau ketika terdapat perubahan mendadak dalam arah dan kecepatan. Akibatnya, performa Kalman Filter dapat menurun dalam situasi di mana objek bergerak dengan pola yang tidak teratur atau sangat dinamis.

Meski begitu, Kalman Filter tetap menjadi pilihan yang populer dalam berbagai aplikasi pelacakan, terutama ketika pola pergerakan objek cukup konsisten dan dapat diprediksi dengan baik.

Tracking Drone dengan Gerakan Parabola

Pelacakan drone yang bergerak dengan pola parabola merupakan tantangan karena pola ini melibatkan perubahan akselerasi dan pengaruh gravitasi yang dinamis. Kalman Filter standar sering kali kesulitan dalam menangani perubahan akselerasi dan arah yang tiba-tiba dalam gerakan parabola.

Untuk mengatasi tantangan ini, digunakan varian seperti Extended Kalman Filter (EKF) atau Unscented Kalman Filter (UKF). Kedua algoritma ini lebih unggul dalam menangani sistem non-linear dibandingkan Kalman Filter standar. EKF menggunakan pendekatan linierisasi, sedangkan UKF menggunakan Unscented Transform untuk menghasilkan estimasi yang lebih akurat.

Selain itu, pemilihan sensor yang tepat, seperti kombinasi GPS, IMU, dan sensor tambahan lainnya, memainkan peran penting dalam meningkatkan akurasi pelacakan. Pengaturan parameter yang teliti juga sangat krusial untuk memastikan performa optimal dari algoritma yang digunakan.

Dengan pendekatan yang tepat dan algoritma yang sesuai, pelacakan drone dengan pola gerakan parabola dapat dilakukan dengan lebih akurat, bahkan dalam kondisi dinamis dan penuh ketidakpastian.

ANALISIS TUGAS SIMULASI WEBOTS YANG TERKAIT DENGAN MATERI PARAMETRIC FILTERING

Simulasi yang menggunakan teknik pemfilteran parametrik seperti **Kalman Filter** dan **Particle Filter** di Webots ini bertujuan untuk meningkatkan akurasi estimasi posisi robot dalam berbagai kondisi dan lingkungan yang mungkin menghadirkan noise dan ketidakpastian. Berikut ini adalah analisis terkait

dengan masing-masing repositori GitHub yang telah disebutkan.

1. Robot Positioning Estimation Using ML Techniques

Simulasi ini menggunakan **Webots 2020a**, **Python 3.7**, dan **Machine Learning Libraries** seperti **TensorFlow** dan **Keras** untuk estimasi posisi robot menggunakan **Neural Networks (NN)** dan **Particle Filter**. Particle Filter adalah salah satu metode pemfilteran parametrik yang sangat efektif dalam menangani masalah lokalisasi robot, terutama dalam lingkungan yang penuh dengan ketidakpastian dan noise.

Penjelasan Simulasi dan Kode Python:

Pada simulasi ini, robot menggunakan model neural network untuk mengestimasi posisi lokalnya. Berikut adalah beberapa elemen utama dalam kode Python yang digunakan:

- **Data Preprocessing:** Kode pertama mengumpulkan data dari sensor robot seperti sensor jarak, IMU (Inertial Measurement Unit), dan GPS. Data ini kemudian diproses untuk digunakan dalam model machine learning.

```
# Pengumpulan dan pre-processing data sensor
```

```
sensor_data = get_sensor_data()
```

```
processed_data = preprocess_data(sensor_data)
```

- **Pelatihan Neural Network:** Setelah data diproses, digunakan TensorFlow dan Keras untuk membangun model neural network yang akan memprediksi posisi robot berdasarkan data yang diberikan. Model ini dilatih menggunakan dataset yang berisi posisi yang diketahui dan data sensor yang dihasilkan oleh robot.

```
# Membuat model neural network dengan TensorFlow dan Keras
```

```
model = tf.keras.Sequential([
```

```
    tf.keras.layers.Dense(64, activation='relu', input_shape=(input_shape,)),
```

```
    tf.keras.layers.Dense(32, activation='relu'),
```

```
    tf.keras.layers.Dense(2) # Output untuk koordinat (x, y)
```

```
])
```

```
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
model.fit(train_data, train_labels, epochs=10)
```

- **Particle Filter:** Untuk lebih meningkatkan estimasi posisi robot, Particle Filter digunakan sebagai teknik pemfilteran parametrik untuk mengatasi ketidakpastian dan noise dalam estimasi posisi. Particle Filter bekerja dengan menggunakan sejumlah "partikel" yang mewakili kemungkinan posisi robot dan mengupdate bobotnya berdasarkan sensor dan hasil prediksi dari model neural network.

```
# Implementasi Particle Filter untuk estimasi posisi
```

```
particles = initialize_particles()
```

```
for particle in particles:
```

```
    particle.weight = update_weight(particle, sensor_data)
```

Simulasi ini menggabungkan teknik machine learning dan pemfilteran parametrik untuk menghasilkan estimasi posisi yang lebih akurat dan robust, meskipun terdapat noise dan ketidakpastian dalam lingkungan sekitar robot. Model ini dapat diterapkan pada robot yang bekerja dalam lingkungan dinamis dengan akurasi tinggi, asalkan data yang tersedia cukup representatif.

2. Four-Wheeled Robot Localization with Kalman Filter

Simulasi ini menggunakan **Webots** dengan robot 4 roda yang dilengkapi dengan **mekanisme kemudi Ackerman**, **wheel odometry**, dan **pengukuran inersia** untuk meningkatkan akurasi pemosisian. Kalman Filter digunakan untuk mengurangi kesalahan yang disebabkan oleh permukaan simulasi yang tidak rata dan tidak stabil. Dengan menggunakan **wheel odometry** dan **sensor inersia**, Kalman Filter memperbaiki estimasi posisi robot pada setiap langkah.

Simulasi dan Kode Python:

Dalam simulasi ini, Kalman Filter digunakan untuk menyaring noise dan meningkatkan akurasi posisi robot yang bergerak dengan mekanisme kemudi Ackerman, yang pada dasarnya adalah sistem non-holonomic. Berikut adalah beberapa bagian penting dari kode Python yang digunakan:

- **Wheel Odometry:** Robot menggunakan pengukuran jarak yang dihitung berdasarkan putaran roda untuk memperkirakan posisi robot. Odometri ini seringkali dipengaruhi oleh noise dan kesalahan kecil dalam perhitungan, yang dapat mengakumulasi dan menyebabkan estimasi posisi yang tidak akurat.

```
# Menghitung odometri dari roda
```

```
distance_left = get_left_wheel_distance()
```

```
distance_right = get_right_wheel_distance()
```

```
robot_position = update_position(distance_left, distance_right)
```

- **Kalman Filter:** Kalman Filter digunakan untuk memprediksi dan memperbarui posisi robot berdasarkan data odometri dan sensor inersia. Kalman Filter menggabungkan estimasi posisi sebelumnya dengan pengukuran baru dari sensor untuk menghasilkan estimasi posisi yang lebih akurat.

```
# Implementasi Kalman Filter
```

```
kalman_gain = calculate_kalman_gain(predicted_state, measurement_noise)
```

```
updated_state = update_kalman_state(predicted_state, kalman_gain, sensor_measurements)
```

- **Pengukuran Inersia:** Sensor inersia digunakan untuk mendeteksi perubahan kecepatan dan orientasi robot. Dengan memanfaatkan data dari sensor ini, Kalman Filter dapat memperbaiki estimasi posisi robot meskipun ada gangguan dari permukaan yang tidak rata.

```
# Menggunakan sensor inersia untuk memperbaiki estimasi posisi
```

```
acceleration = get_inertial_measurement()
```

```
velocity = update_velocity(acceleration)
```

Dengan menggunakan Kalman Filter, robot dapat memperkirakan posisi dengan lebih baik meskipun ada noise atau ketidakstabilan dari odometri dan sensor inersia. Meskipun kalman filter sangat efektif dalam mengatasi noise, kinerjanya sangat tergantung pada pemilihan parameter noise yang tepat, yang mempengaruhi hasil estimasi.

Kesimpulan

Simulasi pertama, **Robot Positioning Estimation Using ML Techniques**, mengintegrasikan pemfilteran parametrik (Particle Filter) dengan metode machine learning (Neural Networks) untuk mengatasi

ketidakpastian dalam estimasi posisi robot. Sementara itu, simulasi kedua, **Four-Wheeled Robot Localization with Kalman Filter**, mengandalkan Kalman Filter untuk meningkatkan akurasi pemosisian dengan menggunakan data dari odometri roda dan sensor inersia. Kedua pendekatan ini menunjukkan bagaimana pemfilteran parametrik dapat digunakan untuk meningkatkan estimasi posisi robot dalam berbagai kondisi yang penuh dengan noise dan ketidakpastian.