

ANALISIS TUGAS ROBOTIK WEEK 13

ANALISIS SIMULASI PARAMETRIC FILTERING DI GOOGLE COLAB

PENJELASAN CODE ADA DI LINK GOOGCLE COLAB

<https://colab.research.google.com/drive/1U2BbsRj03KkSnVzoR0i5k-V-oLFmKFZb?usp=sharing#scrollTo=ws9wsR3JU3vB>

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

Extended Kalman Filter (EKF) merupakan algoritma yang digunakan untuk estimasi posisi dan orientasi robot dengan menggabungkan data dari GPS dan IMU. EKF mampu menangani sistem non-linear dengan cara melakukan linierisasi menggunakan Jacobian, yang membuatnya dapat bekerja dengan baik pada model yang tidak linier. Keunggulannya adalah kemampuannya untuk menyaring noise yang ada pada sensor, menghasilkan estimasi posisi yang lebih stabil dan akurat. Namun, EKF memiliki kelemahan dalam hal ketergantungannya pada linierisasi, yang bisa mengarah pada kesalahan estimasi jika modelnya tidak akurat. Selain itu, kinerja EKF sangat bergantung pada pengaturan parameter noise QQ dan RR, dan dapat kurang efektif dalam situasi dinamis dengan perubahan yang signifikan. Meskipun demikian, EKF banyak digunakan dalam navigasi robot dengan sensor GPS dan IMU, di mana lingkungan bising dapat menurunkan akurasi data sensor.

2. Unscented Kalman Filter (UKF): Untuk Estimasi Navigasi Robot dengan Data GPS dan IMU

Unscented Kalman Filter (UKF) menawarkan keunggulan yang lebih signifikan dibandingkan EKF dalam menangani sistem non-linear, karena UKF tidak memerlukan linierisasi. Dengan menggunakan metode Unscented Transform, UKF dapat memberikan estimasi yang lebih akurat pada sistem yang kompleks dan sangat non-linear, seperti pergerakan robot yang dinamis. Kelebihan utama UKF adalah kemampuannya untuk menangani perubahan keadaan yang lebih besar dan lebih cepat, serta sistem yang lebih sulit diprediksi. Namun, UKF juga memiliki kelemahan, yaitu lebih kompleks dalam perhitungan dan membutuhkan lebih banyak sumber daya komputasi dibandingkan EKF. Pemilihan parameter yang sangat teliti juga diperlukan untuk mencapai performa yang optimal. Meskipun demikian, UKF sangat cocok digunakan dalam aplikasi robotik yang membutuhkan estimasi posisi dan orientasi yang akurat dalam lingkungan dinamis, seperti navigasi robot dengan data GPS dan IMU.

3. Tracking Objek Bergerak dengan Kalman Filter

Kalman Filter sering digunakan untuk pelacakan objek bergerak yang pergerakannya bisa diprediksi, seperti kendaraan atau manusia yang bergerak dalam pola linier. Keunggulan utama dari Kalman Filter adalah kemampuannya untuk menyaring noise dari data sensor, sehingga menghasilkan estimasi posisi yang lebih akurat dan stabil. Algoritma ini cukup efisien dalam hal komputasi, yang memungkinkan aplikasi pelacakan real-time. Namun, Kalman Filter memiliki keterbatasan dalam hal menangani objek yang bergerak secara non-linear, terutama ketika ada perubahan mendadak dalam arah atau kecepatan objek. Oleh karena itu, Kalman Filter cenderung kurang efektif untuk melacak objek dengan pola gerakan yang sangat kompleks atau tiba-tiba berubah. Meskipun demikian, Kalman Filter tetap menjadi pilihan utama dalam pelacakan objek bergerak yang memiliki pola pergerakan linier atau yang dapat diprediksi dengan baik.

4. Tracking Drone dengan Gerakan Parabola

Pelacakan drone yang bergerak dengan pola parabola adalah tantangan tersendiri, karena pola gerak parabola melibatkan perubahan akselerasi dan gravitasi yang dinamis. Untuk menghadapi tantangan ini, Kalman Filter dapat digunakan, namun dengan modifikasi, seperti pada Extended Kalman Filter (EKF) atau Unscented Kalman Filter (UKF). Kalman Filter standar kesulitan menangani perubahan akselerasi dan arah yang tiba-tiba dalam gerakan parabola, sehingga model gerak yang lebih akurat dan modifikasi algoritma diperlukan. Dalam hal ini, penggunaan EKF atau UKF dapat memberikan hasil estimasi posisi dan kecepatan yang lebih baik karena kemampuan mereka dalam menangani sistem non-linear. Selain itu, pemilihan sensor yang tepat dan pengaturan parameter yang teliti juga sangat krusial untuk meningkatkan akurasi pelacakan drone. Dengan demikian, pelacakan drone yang bergerak dalam pola parabola memerlukan modifikasi Kalman Filter yang lebih canggih untuk memberikan estimasi posisi dan kecepatan yang akurat, terutama ketika pergerakan drone tidak terduga.

Kesimpulan

Setiap metode Kalman Filter memiliki keunggulan dan kelemahan masing-masing, tergantung pada jenis sistem yang digunakan. EKF lebih sesuai untuk sistem non-linear yang dapat dilinierkan, sementara UKF lebih efektif untuk sistem dengan dinamika yang lebih kompleks dan non-linear. Untuk pelacakan objek bergerak yang lebih sederhana, Kalman Filter standar sangat berguna, namun ketika dihadapkan pada pola pergerakan yang lebih rumit, seperti pola parabola pada drone, diperlukan modifikasi yang lebih canggih, seperti EKF atau UKF, untuk mencapai estimasi yang lebih akurat.

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.