

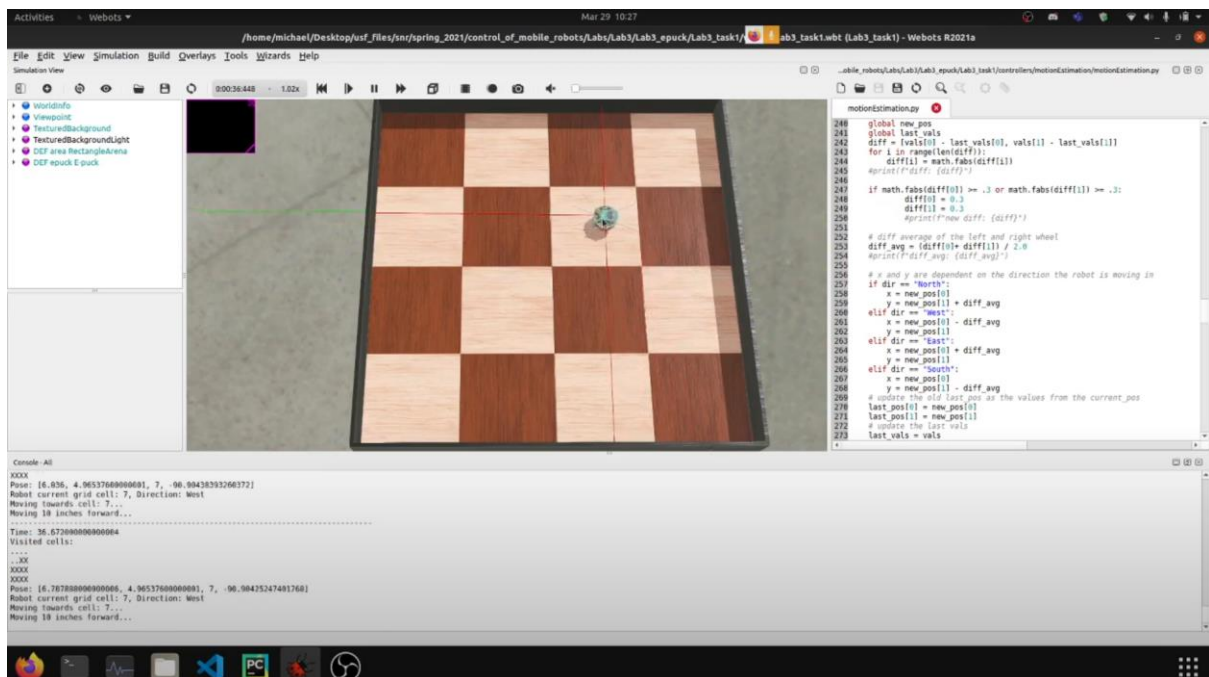
Webots Lab

Nama : Al Ghifary Akmal Nasheeri

NIM : 1103201242

Kelas : TK-44-06

Lab 3 Task 1: Webots motion estimation



Ringkasan:

- Tujuan: Menjelajahi teknik estimasi gerak di Webots, memungkinkan robot untuk melacak posisinya dalam suatu lingkungan tanpa mengandalkan sistem penentuan posisi eksternal.

Demonstrasi: Melibatkan pengembangan pengontrol robot yang secara efektif memperkirakan gerak menggunakan data sensor dan algoritma.

Komponen Utama:

- Lingkungan Simulasi Webots: Menyediakan platform virtual untuk kontrol robot dan pengujian.
- Model Robot: Robot spesifik yang digunakan dalam Task, kemungkinan dilengkapi dengan sensor untuk estimasi gerak.
- Sensor:
 - Sensor posisi (misalnya, encoder) untuk mengukur rotasi roda.
 - Unit Pengukuran Inersia (IMU) untuk melacak orientasi dan percepatan.

- Sensor potensial lainnya (misalnya, sensor jarak, kamera) untuk data tambahan.
- Pengontrol: Kode yang mengimplementasikan algoritma estimasi gerak.

Implementasi Task :

1. Inisialisasi:

- Atur sensor dan variabel.
- Tetapkan posisi dan orientasi awal untuk robot.

2. Akuisisi Data Sensor:

- Baca data sensor secara terus menerus pada interval reguler.

3. Algoritma Estimasi Gerak:

- Gunakan teknik seperti dead reckoning atau fusi sensor untuk menghitung perubahan posisi dan orientasi berdasarkan pembacaan sensor dan estimasi sebelumnya.

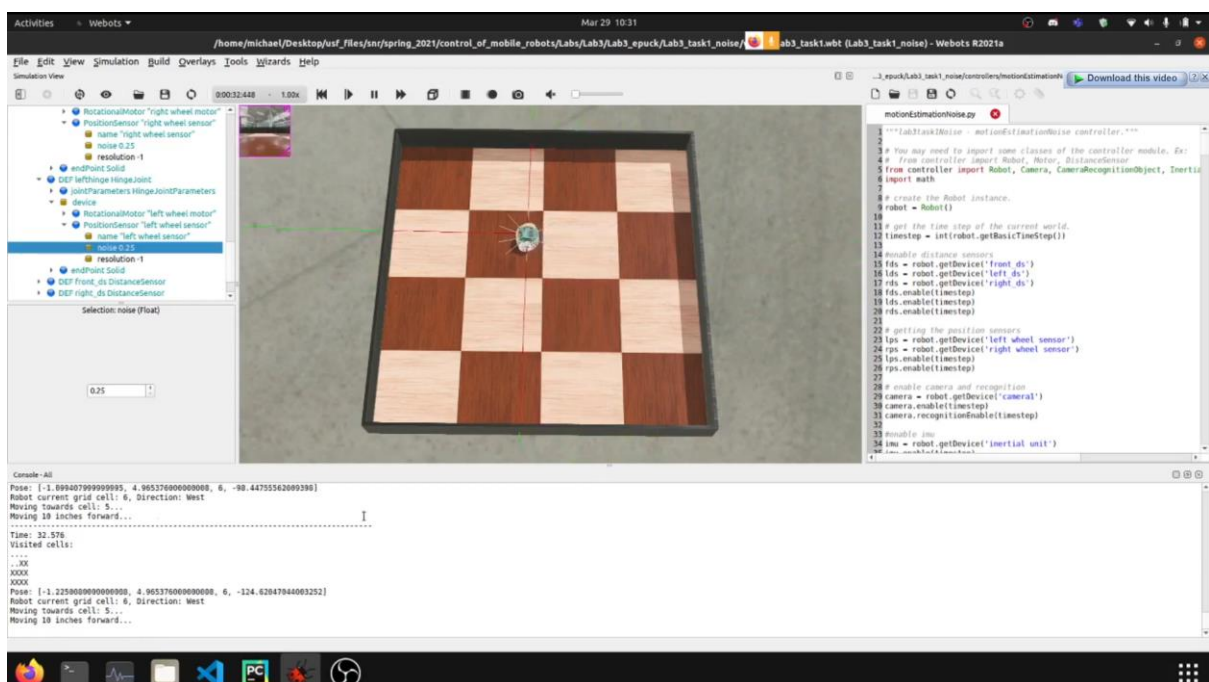
4. Pembaruan Posisi dan Orientasi:

- Integrasikan perubahan yang dihitung untuk memperbarui estimasi posisi dan orientasi robot.

5. Gerakan:

- Panduan tindakan robot berdasarkan posisi estimasi dan tujuan yang diinginkan.

Lab 3 Task 1 with noise: Webots motion estimation



Ringkasan:

- Tujuan: Menjelajahi teknik estimasi gerak di Webots di bawah kondisi noise, di mana pembacaan sensor mengandung kesalahan atau ketidakpastian.
- Demonstrasi: Task DevMike kemungkinan melibatkan pengembangan pengontrol robot yang secara efektif memperkirakan gerak meskipun ada noise sensor.

Komponen Utama:

- Lingkungan Simulasi Webots: Menyediakan platform virtual untuk mensimulasikan noise sensor dan menguji algoritma estimasi.
- Model Robot: Robot spesifik yang digunakan dalam Task, dilengkapi dengan sensor untuk estimasi gerak.
- Sensor:
 - Sensor posisi (misalnya, encoder) untuk mengukur rotasi roda (dengan noise).
 - Unit Pengukuran Inersia (IMU) untuk melacak orientasi dan percepatan (dengan noise).
 - Sensor potensial lainnya (misalnya, sensor jarak, kamera) dengan noise yang disimulasikan.
- Pengontrol: Kode yang mengimplementasikan algoritma estimasi gerak yang memperhitungkan noise.

Implementasi Task:

1. Inisialisasi:
 - Atur sensor dan variabel.
 - Tetapkan posisi dan orientasi awal untuk robot.
 - Definisikan model noise untuk sensor (misalnya, noise Gaussian dengan standar deviasi yang ditentukan).
2. Akuisisi Data Sensor:
 - Baca data sensor secara terus menerus pada interval reguler, dengan memasukkan noise yang disimulasikan.
3. Algoritma Estimasi Gerak yang Tangguh terhadap Noise:
 - Gunakan teknik yang mengurangi efek noise:
 - Penyaringan: Gunakan filter (misalnya, filter Kalman) untuk menghaluskan pembacaan sensor dan mengurangi noise.
 - Fusi data: Gabungkan data dari beberapa sensor untuk meningkatkan akurasi.
 - Deteksi outlier: Identifikasi dan hapus pembacaan sensor yang kemungkinan salah.

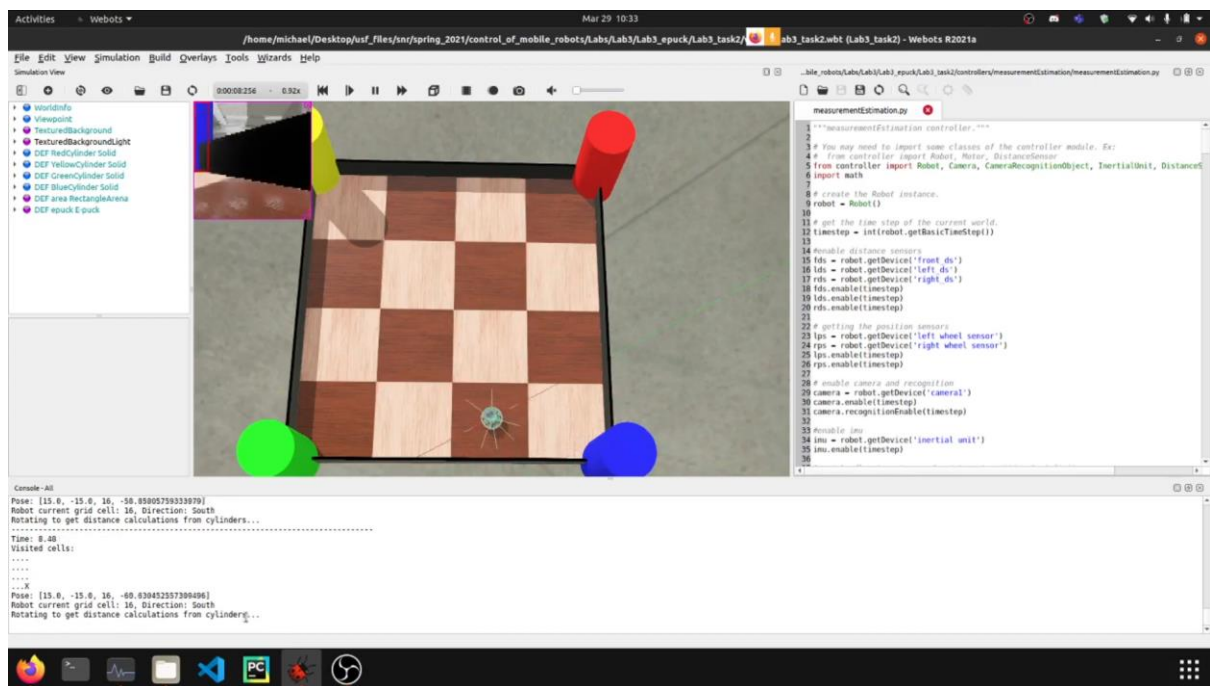
4. Pembaruan Posisi dan Orientasi:

- Integrasikan perubahan yang dikoreksi noise untuk memperbarui estimasi posisi dan orientasi robot.

5. Gerakan:

- Panduan tindakan robot berdasarkan posisi estimasi dan tujuan yang diinginkan,

Lab 3 Task 2: Webots measurement estimation



Ringkasan:

- Tujuan: Menginvestigasi teknik estimasi pengukuran di Webots, memungkinkan robot untuk melokalisasi dirinya dalam suatu lingkungan menggunakan pengukuran sensor dan landmark eksternal.
- Demonstrasi: Video DevMike kemungkinan menampilkan pengontrol robot yang secara efektif memperkirakan posisi menggunakan pembacaan sensor dan posisi landmark.

Komponen Utama:

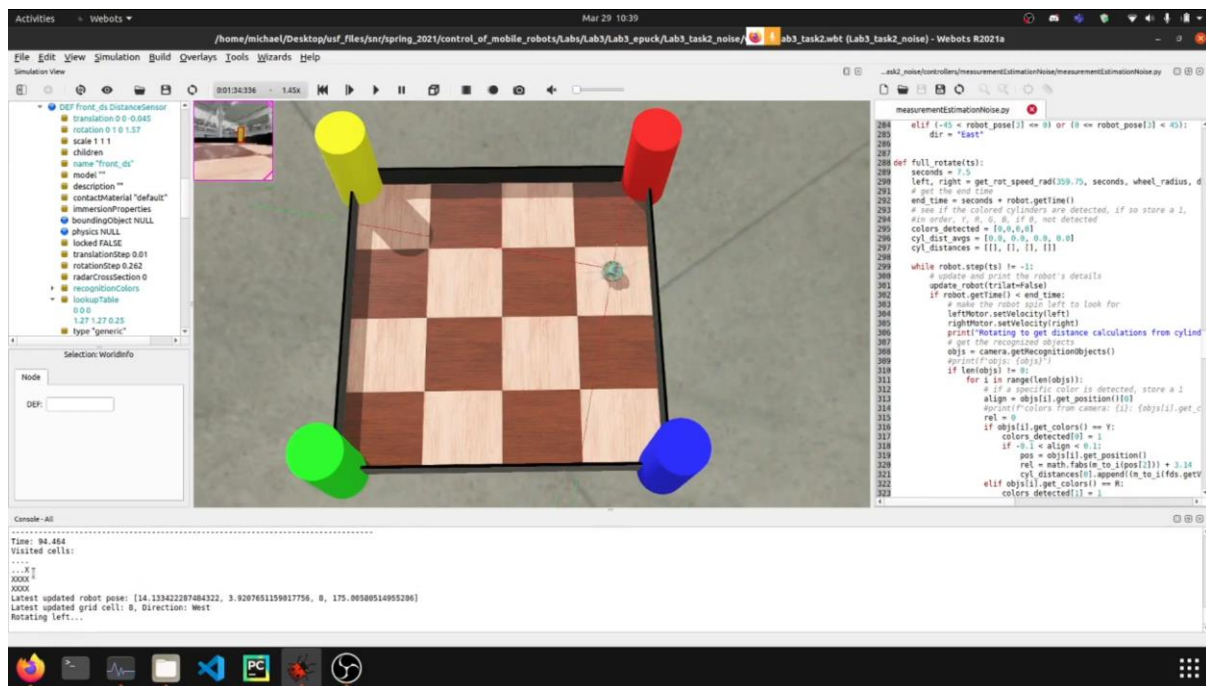
- Lingkungan Simulasi Webots: Menyediakan platform virtual dengan landmark untuk menguji algoritma estimasi.
- Model Robot: Robot spesifik yang digunakan dalam Task, dilengkapi dengan sensor untuk estimasi pengukuran.
- Sensor:

- Sensor jarak (misalnya, sonar, lidar, inframerah) untuk mengukur jarak ke landmark.
- Potensial sensor lain (misalnya, kamera) untuk identifikasi landmark visual.
- Landmark: Objek tetap dalam lingkungan dengan posisi yang diketahui, berfungsi sebagai titik referensi.
- Pengontrol: Kode yang mengimplementasikan algoritma estimasi pengukuran.

Implementasi Task :

1. Inisialisasi:
 - Atur sensor dan variabel.
 - Tetapkan posisi awal untuk robot (mungkin tidak diketahui).
 - Definisikan posisi landmark dalam lingkungan.
2. Akuisisi Data Sensor:
 - Baca data sensor secara terus menerus pada interval reguler, mengukur jarak ke landmark.
3. Identifikasi Landmark:
 - Cocokkan pembacaan sensor dengan posisi landmark yang diketahui.
4. Algoritma Estimasi Pengukuran:
 - Gunakan trilaterasi untuk memperkirakan posisi robot berdasarkan jarak ke tiga atau lebih landmark.
5. Pembaruan Posisi:
 - Integrasikan posisi yang diperkirakan dengan informasi gerakan (jika tersedia) untuk memperbarui posisi keseluruhan robot.
6. Gerakan:
 - Pandu tindakan robot berdasarkan posisi yang diperkirakan dan tujuan yang diinginkan.

Lab 3 Task 2 with noise: Webots measurement estimation



Ringkasan:

- Tujuan: Menginvestigasi teknik estimasi pengukuran di Webots di bawah kondisi noise, di mana pembacaan sensor mengandung kesalahan atau ketidakpastian.
- Tantangan: Mengembangkan pengontrol robot yang secara akurat memperkirakan posisi menggunakan data sensor noise dan posisi landmark.

Komponen Utama:

- Lingkungan Simulasi Webots: Mensimulasikan noise sensor dan menyediakan platform untuk menguji algoritma estimasi.
- Model Robot: Dilengkapi dengan sensor untuk estimasi pengukuran (misalnya, sensor jarak).
- Sensor: Tunduk pada noise, yang menyebabkan kesalahan dalam pengukuran jarak.
- Landmark: Objek tetap dengan posisi yang diketahui, berfungsi sebagai titik referensi.
- Pengontrol: Mengimplementasikan algoritma estimasi pengukuran yang tangguh terhadap noise.

Implementasi Task :

1. Inisialisasi:

- Atur sensor dan variabel.
- Tetapkan posisi awal untuk robot (mungkin tidak diketahui).
- Definisikan posisi landmark dalam lingkungan.
- Model karakteristik noise sensor (misalnya, noise Gaussian).

2. Akuisisi Data Sensor Noise:

- Baca data sensor secara terus menerus, memasukkan noise yang disimulasikan.

3. Identifikasi Landmark:

- Cocokkan pembacaan sensor noise dengan posisi landmark yang diketahui, berpotensi memperhitungkan noise.

4. Algoritma Estimasi yang Tangguh terhadap Noise:

- Gunakan teknik yang mengurangi efek noise:
 - Penyaringan: Filter Kalman, filter partikel, atau metode lain untuk menghaluskan pembacaan yang noise.
 - Fusi data: Gabungkan data dari beberapa sensor untuk mengurangi noise.
 - Deteksi outlier: Identifikasi dan hapus pembacaan yang salah.

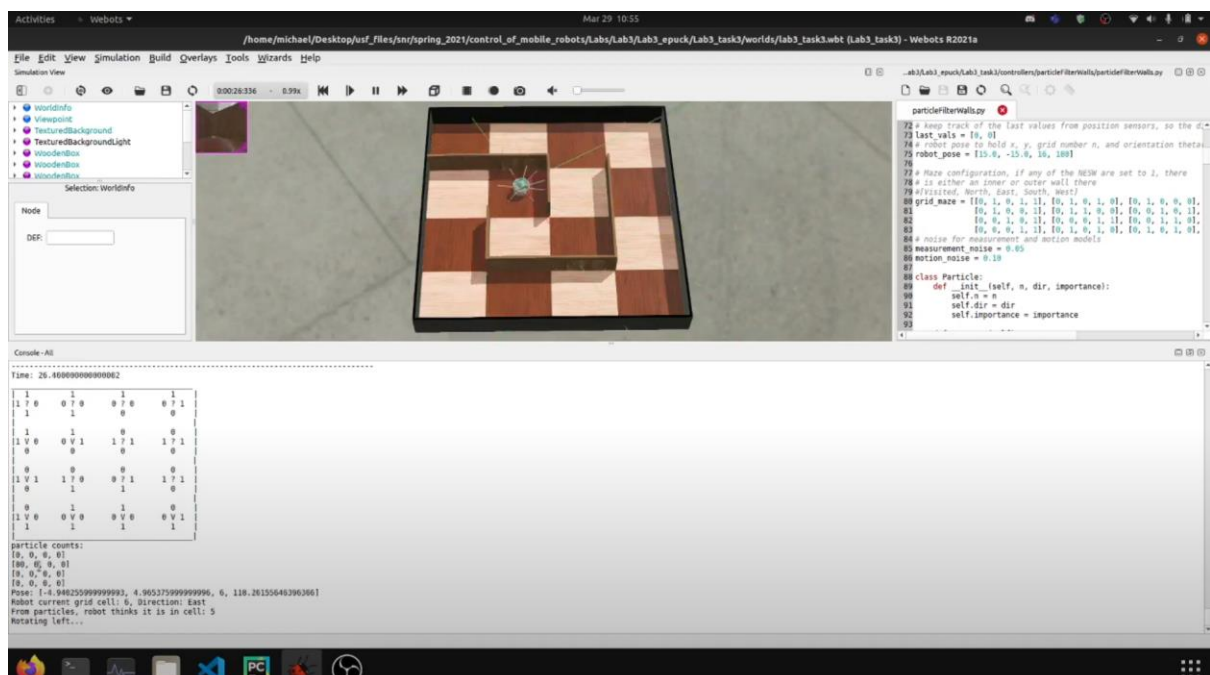
5. Pembaruan Posisi:

- Integrasikan perkiraan posisi yang dikoreksi noise dengan informasi gerakan (jika tersedia).

6. Gerakan:

- Pandu tindakan robot berdasarkan posisi yang diperkirakan dan tujuan yang diinginkan, dengan mempertimbangkan efek noise.

Lab 3 Task 3: Webots particle filters



Ringkasan:

- Tujuan: Menjelajahi filter partikel di Webots, teknik estimasi yang kuat untuk menangani sistem yang kompleks, non-linier, dan data sensor yang noise.
- Tantangan: Mengembangkan pengontrol robot yang secara akurat memperkirakan posisi dan orientasi menggunakan filter partikel.

Komponen Utama:

- Lingkungan Simulasi Webots: Menyediakan platform untuk mensimulasikan gerakan robot, noise sensor, dan algoritma filter partikel.
- Model Robot: Dilengkapi dengan sensor untuk estimasi keadaan (misalnya, sensor jarak, encoder, IMU).
- Algoritma Filter Partikel:
 - Mewakili keadaan robot yang mungkin dengan sekumpulan partikel.
 - Memberi bobot partikel berdasarkan data sensor dan model gerak.
 - Meresample partikel untuk fokus pada keadaan yang lebih mungkin.
- Penanganan Data Sensor: Memasukkan pembacaan sensor untuk memperbarui bobot partikel.
- Model Gerak: Memprediksi pergerakan partikel berdasarkan kontrol dan dinamika robot.

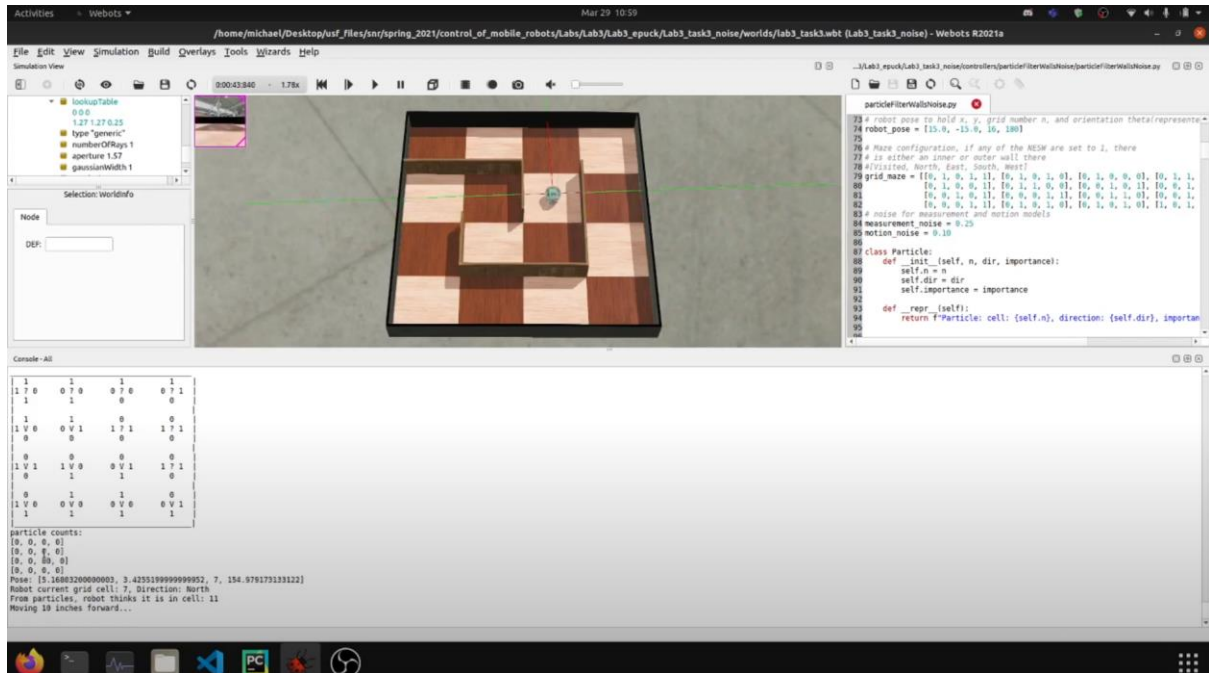
Implementasi Task :

1. Inisialisasi:
 - Atur sensor, variabel, dan parameter filter partikel (jumlah partikel, model noise).
 - Inisialisasi partikel dengan distribusi yang mewakili ketidakpastian awal.
2. Pembaruan Gerak:
 - Terapkan tindakan robot (misalnya, perintah motor) untuk menggerakkan partikel sesuai dengan model gerak.
3. Pembaruan Pengukuran Sensor:
 - Memasukkan pembacaan sensor untuk memperbarui bobot partikel, mencerminkan kemungkinan setiap keadaan.
4. Resampling:
 - Meresample partikel, memprioritaskan partikel dengan bobot yang lebih tinggi, untuk fokus pada keadaan yang lebih mungkin.
5. Estimasi Keadaan:
 - Estimasi posisi dan orientasi robot menggunakan rata-rata tertimbang partikel.

6. Gerakan:

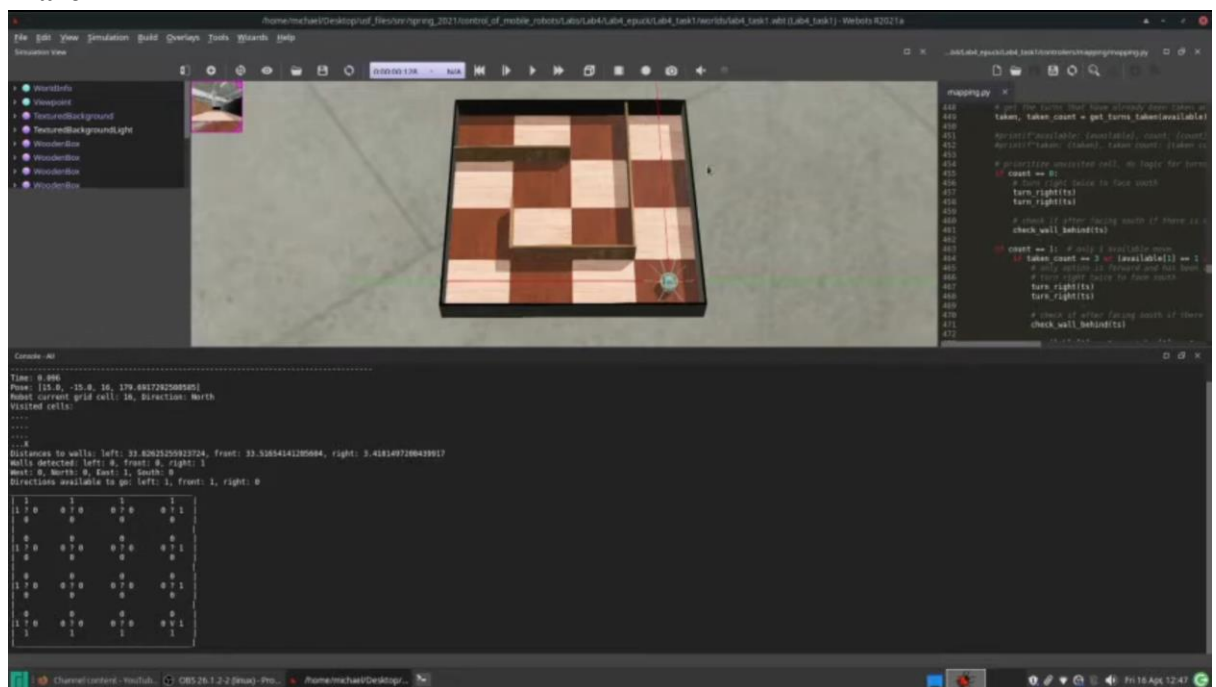
- Pandu tindakan robot berdasarkan keadaan yang diperkirakan dan tujuan yang diinginkan.

Lab 3 Task 3 with noise: Webots particle filters



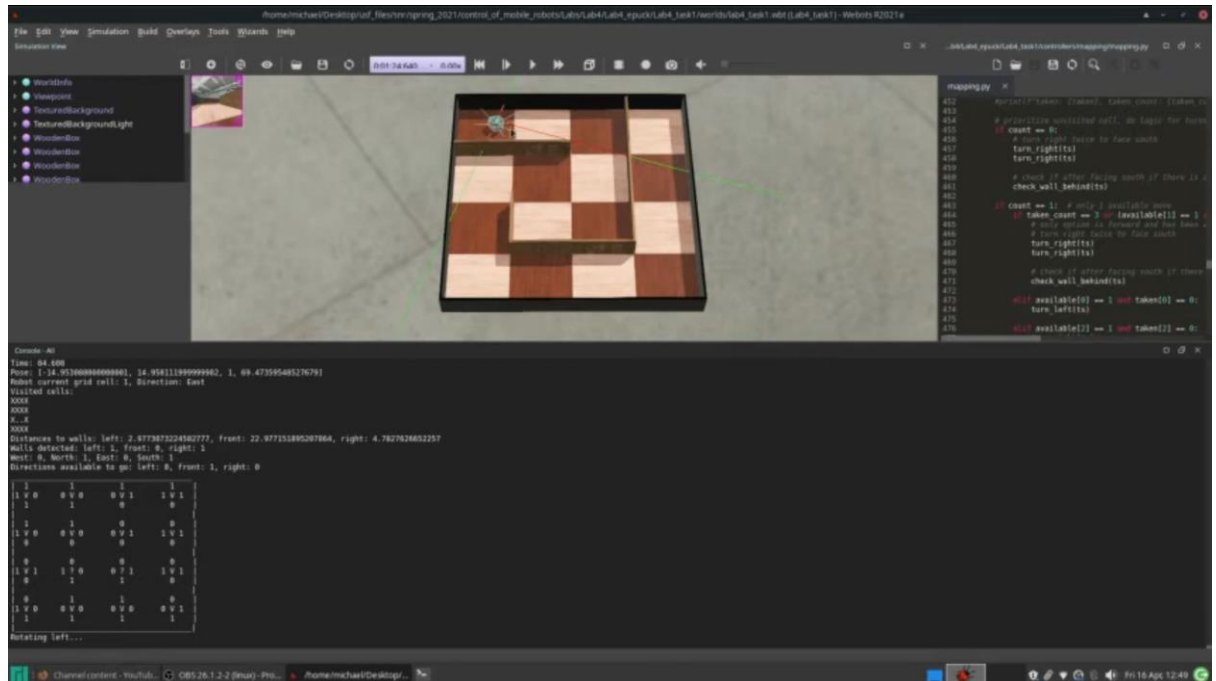
Lab 4 Task 1: Mapping a Webots world

1. Maze



3. Speed Mapping

4. Internal Walls



Ringkasan:

- Tujuan: Menjelajahi ketahanan filter partikel di Webots di bawah kondisi noise, di mana pembacaan sensor tidak sempurna.
- Tantangan: Mengembangkan pengontrol robot yang dapat memperkirakan posisi dan orientasi dengan akurat bahkan dengan data sensor yang noise.

Adaptasi Kunci untuk Noise:

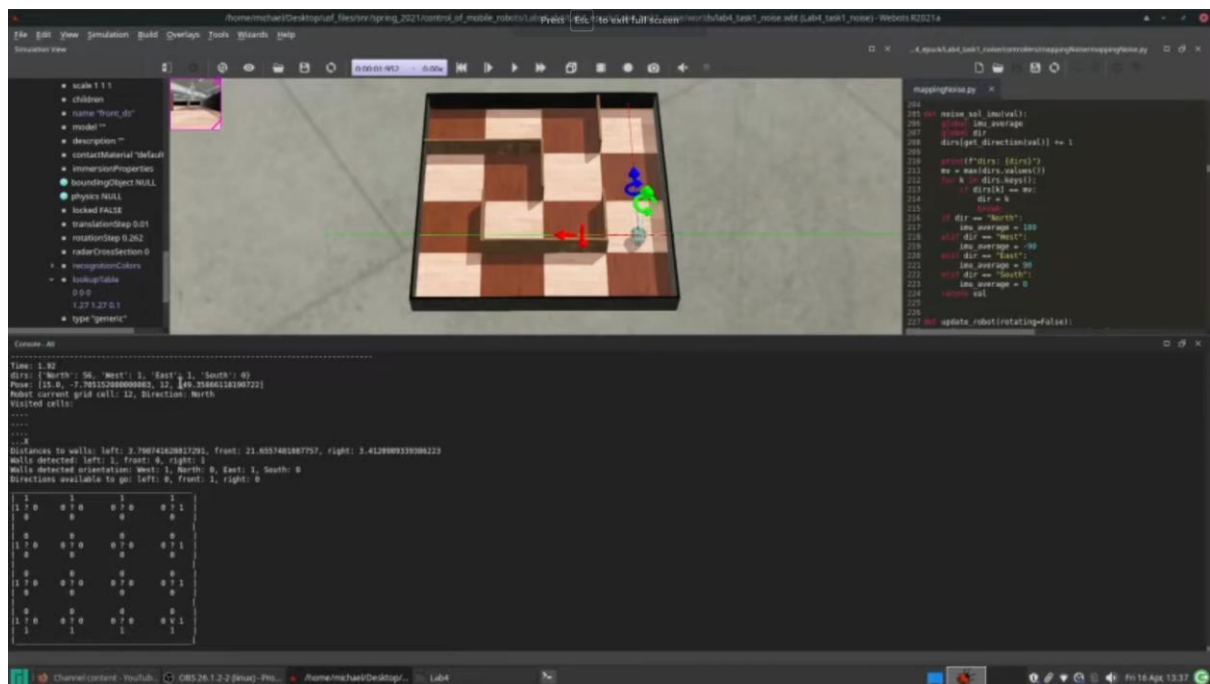
- Pemodelan Noise: Modelkan karakteristik noise (misalnya, noise Gaussian) dalam pembacaan sensor secara akurat.
- Teknik Penyaringan: Gunakan metode dalam algoritma filter partikel untuk mengurangi efek noise:
 - Fusi Data Sensor: Gabungkan data dari beberapa sensor untuk mengurangi ketidakpastian.
 - Deteksi dan Penolakan Outlier: Identifikasi dan hapus pembacaan yang salah.
 - Teknik Sampling Lanjutan: Jelajahi teknik seperti resampling bertahap atau sistematis untuk mempertahankan keragaman partikel.
- Penyesuaian Parameter: Sesuaikan parameter filter partikel (misalnya, jumlah partikel, ambang batas resampling) dengan hati-hati untuk menyeimbangkan akurasi dan efisiensi komputasi.

Implementasi Task :

1. Inisialisasi:

- Atur sensor, variabel, dan parameter filter partikel, dengan mempertimbangkan model noise.
 - Inisialisasi partikel, dengan memperhitungkan ketidakpastian awal dan karakteristik noise.
2. Pembaruan Gerak dengan Noise:
- Terapkan tindakan robot, dengan memasukkan potensi noise dalam masukan kontrol.
 - Prediksi pergerakan partikel berdasarkan model gerak yang noise.
3. Pembaruan Pengukuran Sensor dengan Noise:
- Memasukkan pembacaan sensor yang noise untuk memperbarui bobot partikel, mencerminkan kemungkinan setiap keadaan.
4. Resampling dengan Ketahanan Noise:
- Terapkan teknik resampling yang tahan terhadap noise untuk fokus pada keadaan yang lebih mungkin sambil mempertahankan keragaman.
5. Estimasi Keadaan dengan Noise:
- Estimasi posisi dan orientasi robot menggunakan rata-rata tertimbang partikel, dengan mempertimbangkan ketidakpastian yang disebabkan oleh noise.
6. Gerakan:
- Pandu tindakan robot berdasarkan keadaan yang diperkirakan dan tujuan yang diinginkan, dengan memperhitungkan implikasi noise.

Lab 4 Task 1 with Noise: Mapping a Webots world



Ringkasan:

- Tujuan: Menjelajahi teknik pemetaan dunia dalam Webots di bawah kondisi noise, di mana pembacaan sensor tidak sempurna.
- Tantangan: Mengembangkan pengontrol robot yang dapat memetakan dunia dengan akurat bahkan dengan data sensor yang noise.

Adaptasi Kunci untuk Noise:

- Pemodelan Noise: Modelkan karakteristik noise (misalnya, noise Gaussian) dalam pembacaan sensor secara akurat.
- Teknik Penyaringan: Gunakan metode dalam algoritma pemetaan untuk mengurangi efek noise:
 - Fusi Data Sensor: Gabungkan data dari beberapa sensor untuk mengurangi ketidakpastian.
 - Deteksi dan Penolakan Outlier: Identifikasi dan hapus pembacaan yang salah.
 - Teknik Sampling Lanjutan: Jelajahi teknik seperti resampling bertahap atau sistematis untuk mempertahankan keragaman partikel.
- Penyesuaian Parameter: Sesuaikan parameter algoritma pemetaan (misalnya, jumlah partikel, ambang batas resampling) dengan hati-hati untuk menyeimbangkan akurasi dan efisiensi komputasi.

Implementasi Task :

1. Inisialisasi:

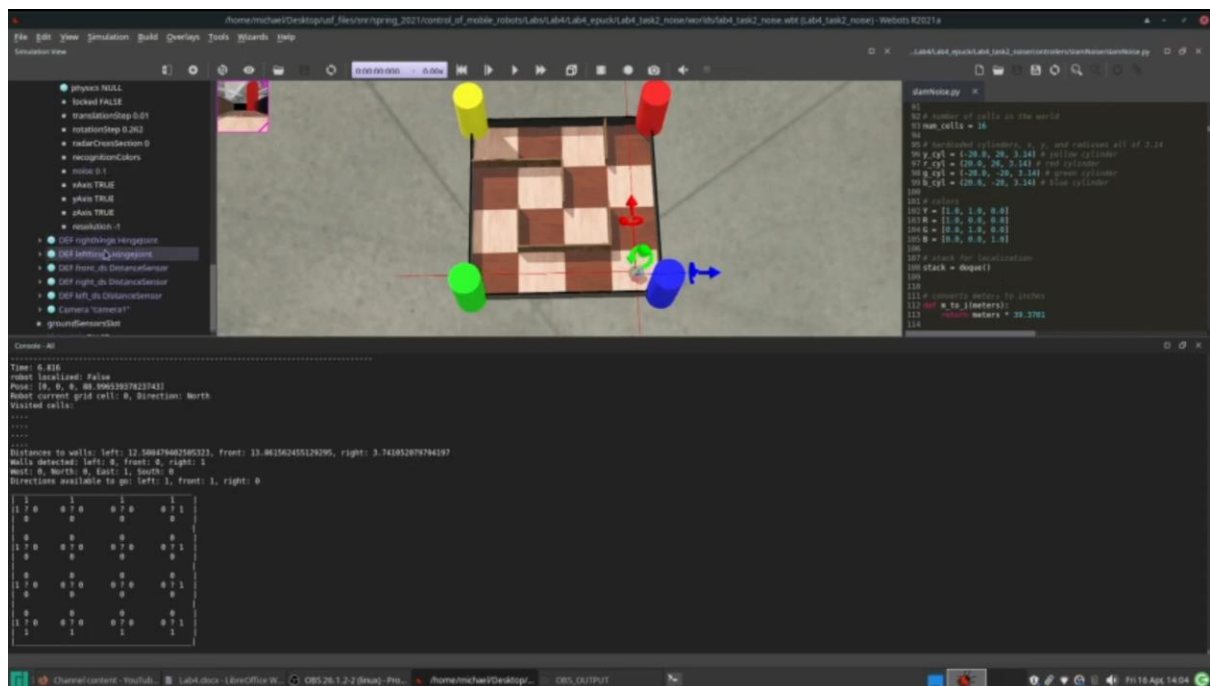
Ringkasan:

- Tujuan: Menjelajahi teknik Simultaneous Localization and Mapping (SLAM) dalam Webots.
- Tantangan: Mengembangkan pengontrol robot yang dapat melakukan SLAM dengan akurat dan efisien.

Implementasi Task :

1. Inisialisasi:
 - Atur sensor, variabel, dan parameter SLAM, dengan mempertimbangkan model noise.
 - Inisialisasi partikel, dengan memperhitungkan ketidakpastian awal dan karakteristik noise.
2. Pembaruan Gerak dengan Noise:
 - Terapkan tindakan robot, dengan memasukkan potensi noise dalam masukan kontrol.
 - Prediksi pergerakan partikel berdasarkan model gerak yang noise.
3. Pembaruan Pengukuran Sensor dengan Noise:
 - Memasukkan pembacaan sensor yang noise untuk memperbarui bobot partikel, mencerminkan kemungkinan setiap keadaan.
4. Resampling dengan Ketahanan Noise:
 - Terapkan teknik resampling yang tahan terhadap noise untuk fokus pada keadaan yang lebih mungkin sambil mempertahankan keragaman.
5. Pemetaan:
 - Estimasi posisi dan orientasi robot menggunakan rata-rata tertimbang partikel, dengan mempertimbangkan ketidakpastian yang disebabkan oleh noise.
6. Membuat peta:
 - Buat representasi grafis dari dunia berdasarkan estimasi posisi dan orientasi robot.
7. Pembaruan Peta:
 - Perbarui peta dengan informasi baru yang diperoleh dari pembacaan sensor.

Lab 4 Task 2 with Noise: Webots SLAM



Ringkasan:

- Tujuan: Menjelajahi ketahanan SLAM dalam Webots di bawah kondisi noise, di mana pembacaan sensor tidak sempurna.
- Tantangan: Mengembangkan pengontrol robot yang dapat melakukan SLAM secara akurat dan efisien bahkan dengan data sensor yang noise.

Adaptasi Kunci untuk Noise:

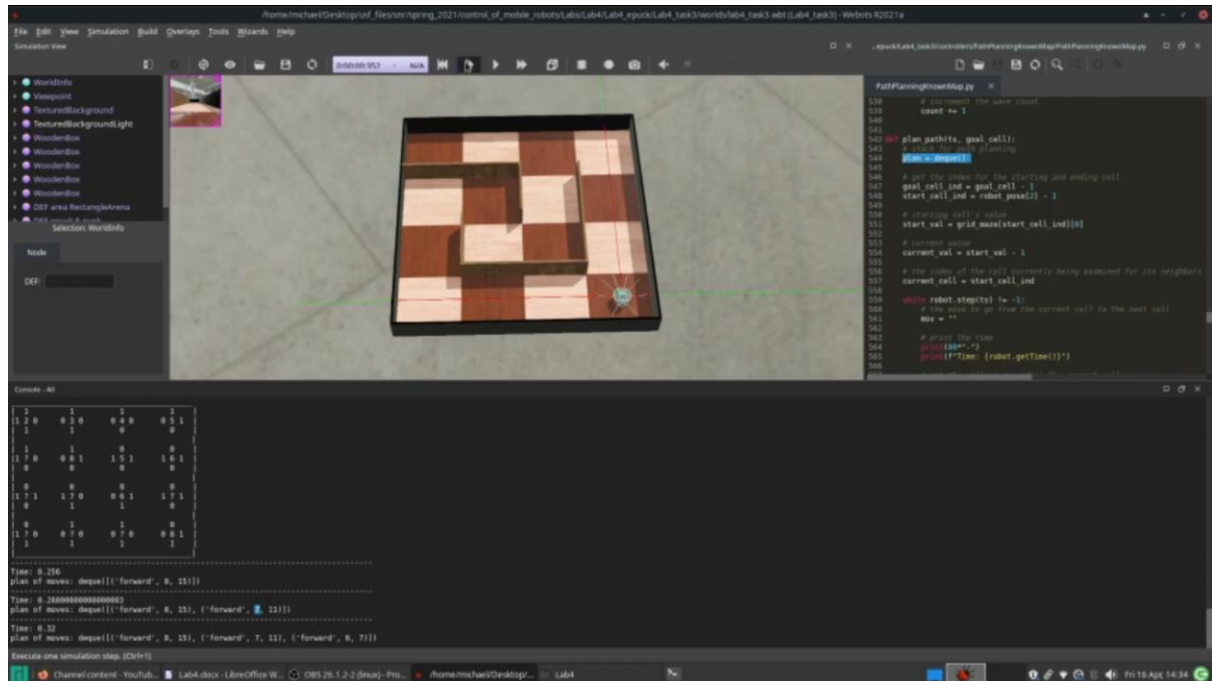
- Pemodelan Noise: Merepresentasikan karakteristik noise (misalnya, noise Gaussian) dalam pembacaan sensor secara akurat.
- Teknik Penyaringan: Menerapkan metode dalam algoritma SLAM untuk mengurangi efek noise:
 - Fusi Data Sensor: Menggabungkan data dari beberapa sensor untuk mengurangi ketidakpastian.
 - Deteksi dan Penolakan Outlier: Mengidentifikasi dan membuang pembacaan yang salah.
 - Teknik Sampling Lanjutan: Menggunakan metode seperti resampling stratifikasi atau sistematis untuk mempertahankan keragaman partikel.
- Penyesuaian Parameter: Menyesuaikan parameter SLAM (jumlah partikel, ambang batas resampling) dengan hati-hati untuk menyeimbangkan akurasi dan efisiensi komputasi.

Implementasi Task :

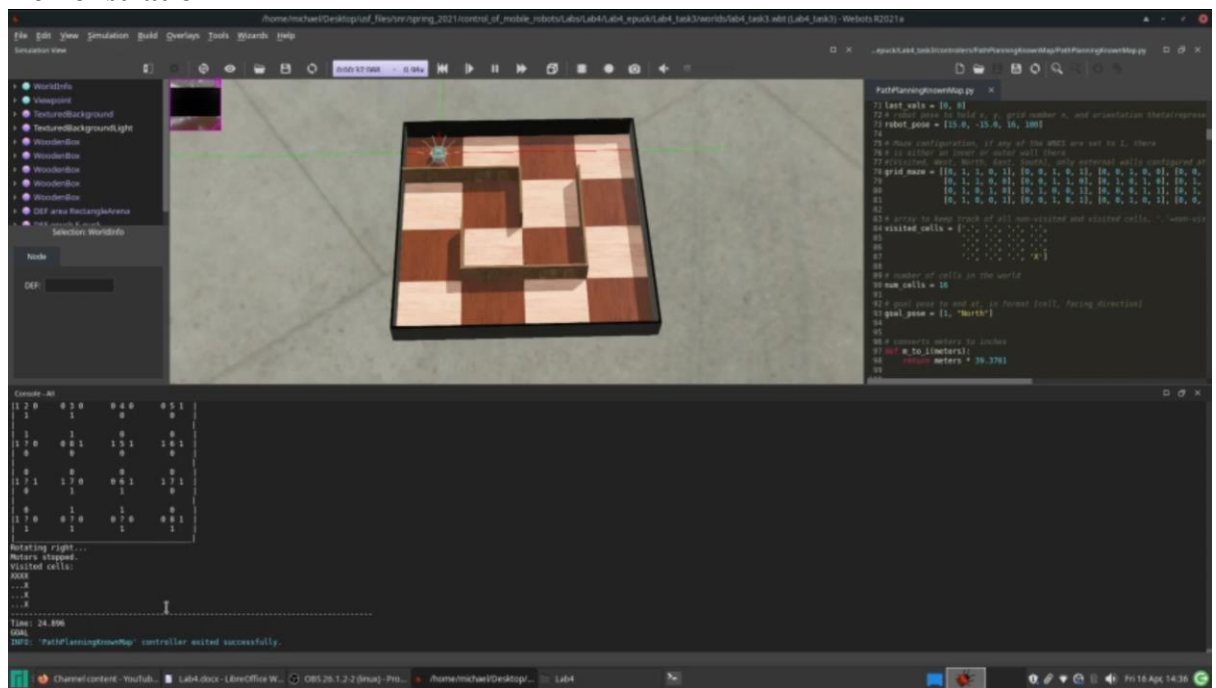
1. Inisialisasi:

- Atur sensor, variabel, dan parameter SLAM, dengan mempertimbangkan model noise.
 - Inisialisasi partikel, dengan memperhitungkan ketidakpastian awal dan karakteristik noise.
2. Pembaruan Gerak dengan Noise:
- Terapkan tindakan robot, dengan memasukkan potensi noise dalam masukan kontrol.
 - Prediksi pergerakan partikel berdasarkan model gerak yang noise.
3. Pembaruan Pengukuran Sensor dengan Noise:
- Memasukkan pembacaan sensor yang noise untuk memperbarui bobot partikel, mencerminkan kemungkinan setiap keadaan.
4. Resampling dengan Ketahanan Noise:
- Terapkan teknik resampling yang tahan terhadap noise untuk fokus pada keadaan yang lebih mungkin sambil mempertahankan keragaman.
5. Pemetaan dengan Noise:
- Estimasi posisi dan orientasi robot menggunakan rata-rata tertimbang partikel, dengan mempertimbangkan ketidakpastian yang disebabkan oleh noise.
6. Membuat peta:
- Buat representasi grafis dari dunia berdasarkan estimasi posisi dan orientasi robot.
7. Pembaruan Peta dengan Noise:
- Perbarui peta dengan informasi baru yang diperoleh dari pembacaan sensor, dengan mempertimbangkan ketidakpastian.

3. Execution



4. Demonstration



Ringkasan:

- Tujuan: Mengembangkan pengontrol robot untuk merencanakan jalur optimal dalam lingkungan dengan peta yang sudah diketahui.
- Tantangan: Mempertimbangkan batasan robot dan lingkungan untuk menemukan jalur terpendek, tercepat, atau paling aman.

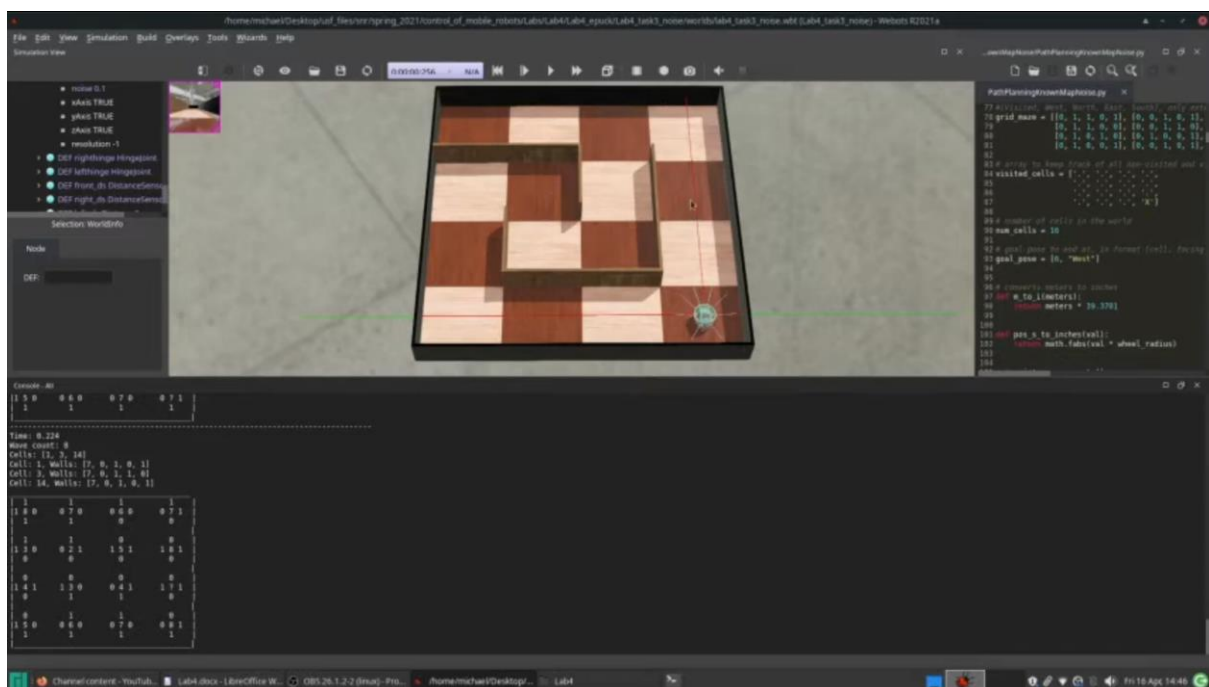
Algoritma Perencanaan Jalur (Contoh):

1. Representasi Peta: Konversikan peta menjadi struktur data yang dapat digunakan oleh algoritma perencanaan jalur (misalnya, grafik, grid).
2. Pencarian Titik Awal dan Akhir: Tentukan titik awal dan akhir robot dalam peta.
3. Eksplorasi Jalur: Gunakan algoritma pencarian jalur seperti:
 - Pencarian Dijkstra: Menemukan jalur terpendek dengan bobot jarak.
 - *Pencarian A**: Menemukan jalur tercepat dengan bobot jarak dan heuristik.
 - Pencarian RRT (Rapidly-exploring Random Tree): Menemukan jalur yang menghindari rintangan secara dinamis.
4. Optimalisasi Jalur: Haluskan jalur yang ditemukan untuk mengurangi kejanggalan dan mempertimbangkan batasan robot (kecepatan, manuver).
5. Eksekusi Jalur: Kirim perintah kontrol ke robot untuk mengikuti jalur yang direncanakan.

Faktor Pertimbangan:

- Jenis lingkungan: Apakah lingkungan statis atau dinamis? Apakah ada rintangan atau area terlarang?
- Kemampuan robot: Apa kecepatan maksimum, radius belok, dan sensor robot?
- Kriteria optimasi: Apakah prioritas terpendek, tercepat, atau teraman?
- Komputasi efisiensi: Pilih algoritma yang sesuai dengan keterbatasan komputasi robot.

Lab 4 Task 3 with Noise: Path Planning in a World with a Known Map



Ringkasan:

- Tujuan: Mengembangkan pengontrol robot yang merencanakan jalur optimal dalam lingkungan dengan peta yang diketahui, namun data sensor robot mengandung noise (ketidakakuratan).
- Tantangan: Menemukan jalur yang tetap optimal dan aman meskipun robot menerima informasi dari sensor yang noise.

Adaptasi untuk Noise:

- Pemodelan Noise: Akurat memperkirakan karakteristik noise pada data sensor (misalnya, noise Gaussian).
- Penanganan Sensor:
 - Fusi Data Sensor: Gabungkan data dari beberapa sensor untuk mengurangi ketidakpastian.
 - Deteksi dan Penolakan Outlier: Identifikasi dan buang pembacaan sensor yang salah.
 - Filtering: Gunakan teknik seperti Kalman filter untuk menyaring noise.
- Penyesuaian Jalur:
 - Perencanaan Jalur Robusta: Pilih algoritma seperti Pencarian A* dengan heuristik adaptif yang mempertimbangkan ketidakpastian.
 - Jalur Dinamis: Perbarui jalur secara online berdasarkan informasi sensor terbaru untuk menghindari rintangan tak terduga.
 - Kontrol Robusta: Gunakan pengontrol feedback untuk mengoreksi deviasi dari jalur akibat noise.

Implementasi Task (Contoh Langkah Umum):

1. Inisialisasi:
 - Atur sensor, variabel, dan parameter perencanaan jalur, dengan mempertimbangkan model noise.
 - Konversikan peta menjadi struktur data yang sesuai.
2. Pembaruan Sensor dengan Noise:
 - Kumpulkan data sensor dan terapkan teknik penanganan noise (filtering, outlier rejection).
3. Perencanaan Jalur Robusta:
 - Gunakan algoritma perencanaan jalur yang adaptif terhadap noise, memperbarui estimasi posisi robot dan peta dinamis.
 - Hitung jalur baru atau perbarui jalur yang ada berdasarkan informasi terbaru.
4. Optimalisasi Jalur:

- Haluskan jalur untuk mengurangi kejanggalan dan mempertimbangkan batasan robot serta ketidakpastian.

5. Eksekusi Jalur dengan Kontrol Robusta:

- Kirim perintah kontrol yang memperhitungkan noise dan deviasi dari jalur yang direncanakan.
- Gunakan feedback dari sensor untuk terus menyesuaikan kontrol dan menavigasi jalur secara akurat.

Faktor Tambahan:

- Tingkat noise sensor: Sesuaikan strategi penanganan noise dan algoritma perencanaan jalur berdasarkan tingkat ketidakakuratan data sensor.
- Komputasi efisiensi: Pertimbangkan algoritma yang seimbang antara efektivitas dan penggunaan sumber daya komputasi robot.
- Keamanan dan keandalan: Pastikan jalur yang direncanakan dan eksekusi kontrol tetap aman dan andal meskipun ada noise.