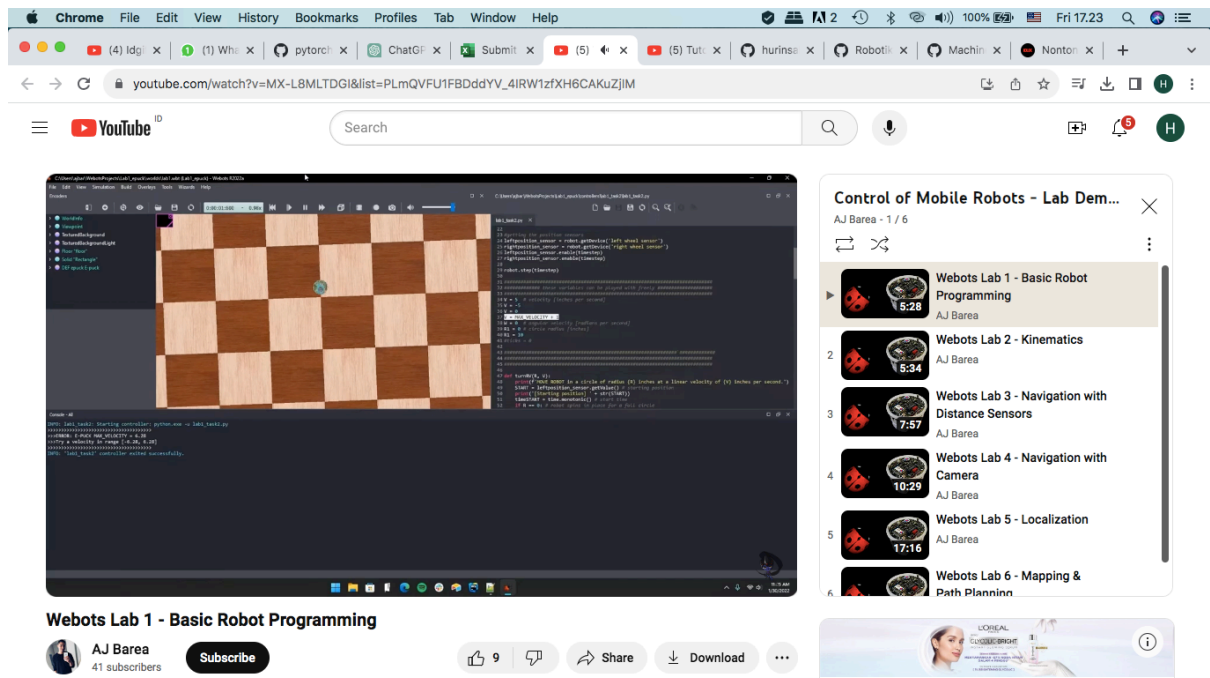


Nama : Hurin Salimah

NIM : 1103200021

Video 1



Pada lab 1 tugas 1, eksperimen dilakukan dengan mengemudikan robot sejauh x inci dengan kecepatan konstan v inci per detik. Terdapat empat kasus uji:

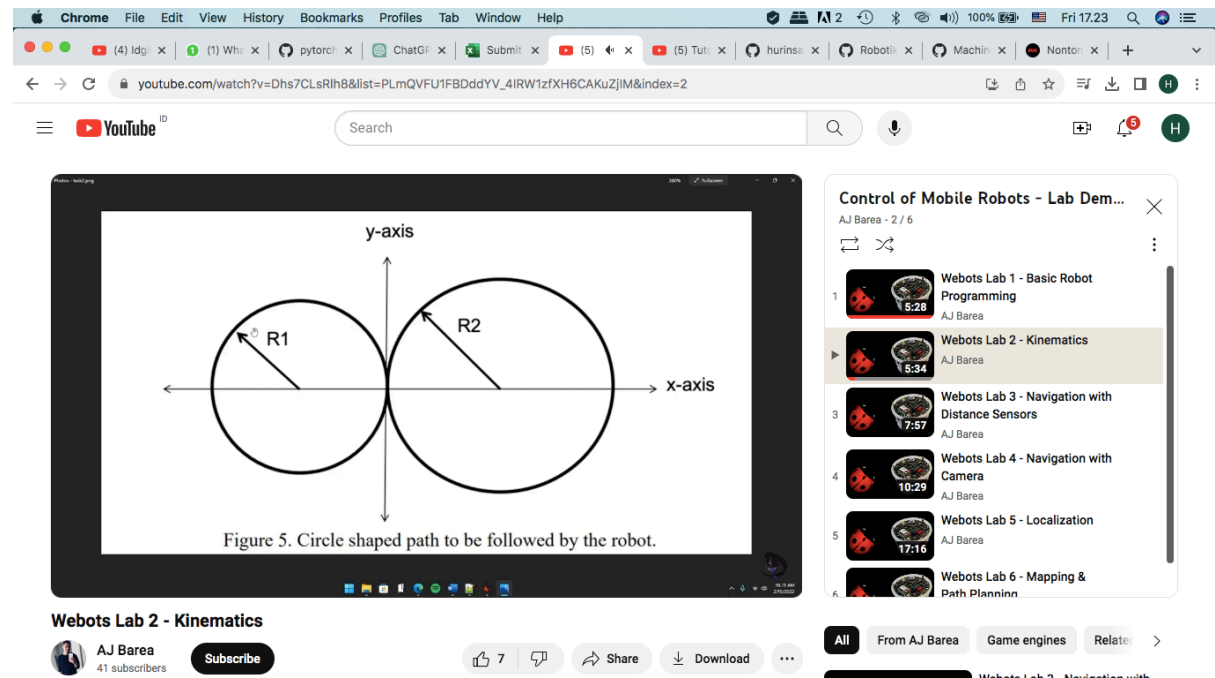
- Kasus kecepatan maksimum melebihi kecepatan motor ($2\pi + 1$), menyebabkan kesalahan karena jangkauan kecepatan yang terlalu besar.
- Pengujian kecepatan sebesar 6.28 inci per detik untuk membuat robot bergerak maju sejauh 30 inci. Dalam hal ini, jarak dan waktu perjalanan terhitung dengan benar.
- Ketika kecepatannya nol, robot tidak bergerak, dan output mencetak waktu, jarak, dan kecepatan yang sebenarnya nol.
- Kasus kecepatan negatif membuat robot bergerak mundur. Meskipun jarak tercetak, perhitungannya sedikit berbeda.

Pada lab 1 tugas 2, fokus pada pembacaan encoder pada simulator yang memutar robot dalam lingkaran dengan radius r_1 inci dan kecepatan v . Terdapat lima kasus uji:

- Melebihi kecepatan maksimal motor menyebabkan kesalahan dan robot tidak bergerak.
- Ketika kecepatannya nol, robot juga tidak bergerak, meskipun muncul pesan error yang berbeda.
- Percobaan dengan kecepatan negatif tetapi jari-jari lingkaran tetap 10 inci membuat robot bergerak berlawanan arah jarum jam dan menghitung perpindahannya.
- Kecepatan positif pada radius 10 inci membuat robot bergerak searah jarum jam, dan saat berhenti, kecepatan sudut dihitung setengah radian.
- Saat jari-jari menjadi nol, robot berputar di tempat dengan kecepatan yang disesuaikan pada masing-masing motor.

Semua perhitungan didasarkan pada kecepatan yang diberikan dan menggambarkan bagaimana robot bereaksi terhadap berbagai keadaan yang berbeda dalam eksperimen.

Video 2

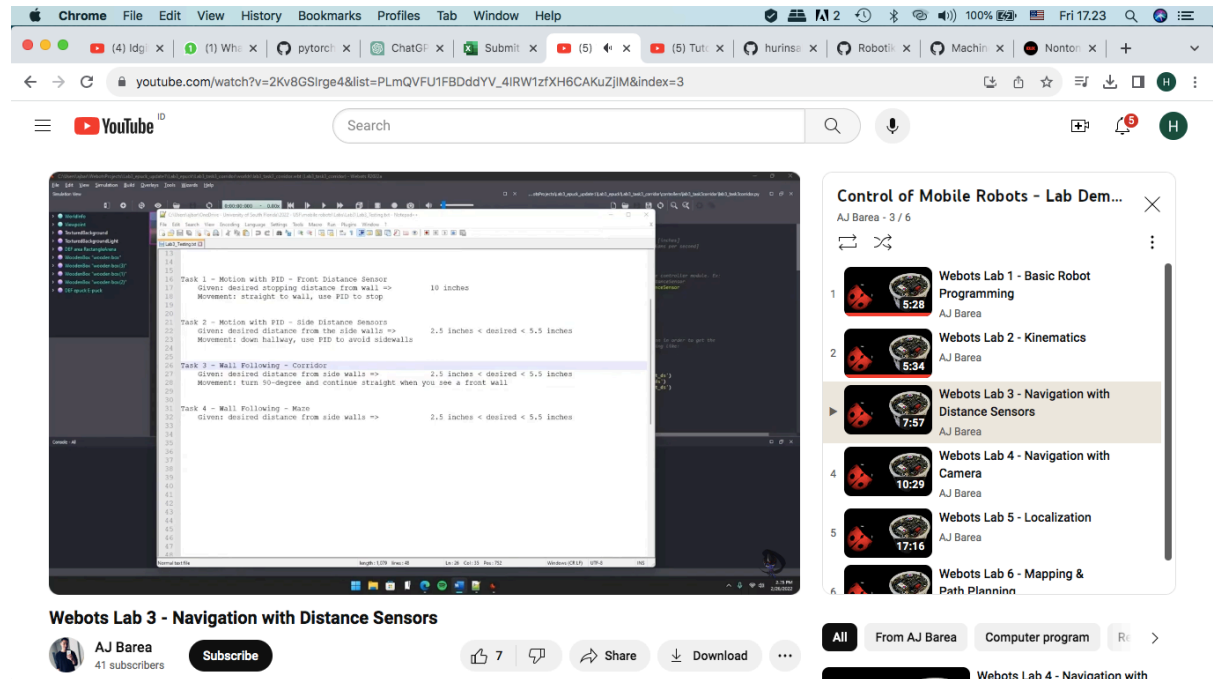


Lab 2 terdiri dari tiga tugas terpisah yang memerlukan implementasi pengontrol robot penggerak diferensial menggunakan Python. Setiap pengontrol memiliki peringatan kesalahan dalam dua skenario: ketika robot tidak menyelesaikan jalur dalam waktu yang ditentukan, dan ketika kecepatan melebihi kapasitas maksimal. Pada tugas pertama, pengontrol menggerakkan robot membentuk persegi panjang dengan perpindahan pada sumbu x, kemudian bergerak searah jarum jam. Pengontrol mencetak arah pergerakan robot dalam derajat, menghitung jarak berdasarkan tinggi, lebar, dan kecepatan, serta waktu.

Tugas kedua mewajibkan robot mengemudi dalam lingkaran dengan radius tertentu dan kecepatan konstan. Salah satu peringatan kesalahan adalah ketika waktu yang diberikan tidak mencukupi untuk pengujian. Robot harus bergerak searah jarum jam dalam lingkaran yang ditentukan.

Tugas terakhir, tugas ketiga, mengharuskan robot membentuk oval dengan radius, jarak, dan kecepatan konstan. Peringatan kesalahan muncul jika kecepatan maksimal terlampaui. Robot memulai pergerakan dari sumbu x dan mengikuti setengah lingkaran, kemudian bergerak searah jarum jam membentuk bentuk oval. Perhitungan jarak, waktu, dan kecepatan dicetak dalam konsol.

Video 3



Lab 3 terdiri dari empat tugas yang semuanya membutuhkan penggunaan sensor jarak pada robot, terutama sensor kiri, kanan, dan depan. Kecepatan motor dikendalikan menggunakan fungsi kontrol proporsional berdasarkan pembacaan sensor untuk menentukan kapan harus berhenti, berbelok, atau memperlambat.

Tugas pertama memanfaatkan PID controller untuk membuat robot bergerak secara lurus dengan memperlambat saat mendekati jarak 10 inci dari dinding. Pengendalian proporsional mengubah kecepatan robot berdasarkan kesalahan jarak dari nilai yang diinginkan.

Tugas kedua melibatkan kontrol proporsional yang juga menjaga jarak dari dinding samping, memungkinkan robot untuk bergerak sepanjang aula dengan menjaga jarak antara 2,5 hingga 5,5 inci dari dinding.

Tugas ketiga menantang robot untuk bermanuver di koridor dengan beberapa putaran menggunakan kontrol proporsional yang sama seperti sebelumnya, dengan penyesuaian kecil untuk kecepatan dan pergerakan zigzag.

Tugas terakhir, tugas keempat, melibatkan robot dalam sebuah labirin di mana kontrol proporsional digunakan untuk menghindari benturan dengan tembok.

Setiap tugas mencetak nilai pembacaan sensor dan memperlihatkan bagaimana kontrol proporsional beroperasi dengan mengubah kecepatan robot berdasarkan kesalahan jarak dari nilai yang diinginkan, memungkinkan robot untuk bergerak dengan memperlambat, berbelok, atau menjaga jarak dari dinding sekitarnya.

Video 4

The screenshot shows a YouTube video player interface. The video is titled "Webots Lab 4 - Navigation with Camera" by AJ Barea. The video content displays a simulation of a robot navigating a maze. The text on the screen reads: "BUG 0 ALGORITHM", "1. drive straight to goal", "2. if theres an obstacle, wall follow", "(Do not combine left and right turns in the same run)", and "SUCCESS". The video is 10:29 minutes long. The right sidebar shows a playlist of other Webots Lab videos, including "Webots Lab 2 - Kinematics", "Webots Lab 3 - Navigation with Distance Sensors", "Webots Lab 4 - Navigation with Camera", "Webots Lab 5 - Localization", and "Webots Lab 6 - Mapping & Path Planning".

Lab 4 memanfaatkan kamera pada robot untuk mendeteksi dan mengenali objek kuning, biasanya sebuah silinder, sebagai tujuan. Berbagai tugas dalam lab ini mengimplementasikan logika kontrol PID untuk menjaga robot pada kecepatan, orientasi, dan jarak yang diinginkan terhadap objek tersebut.

Pada tugas pertama, robot berusaha untuk menghadap objek kuning, mengandalkan kamera untuk mendeteksi posisinya. Ketika objek terdeteksi, robot berputar ke kiri atau kanan berdasarkan posisi objek dalam gambar kamera.

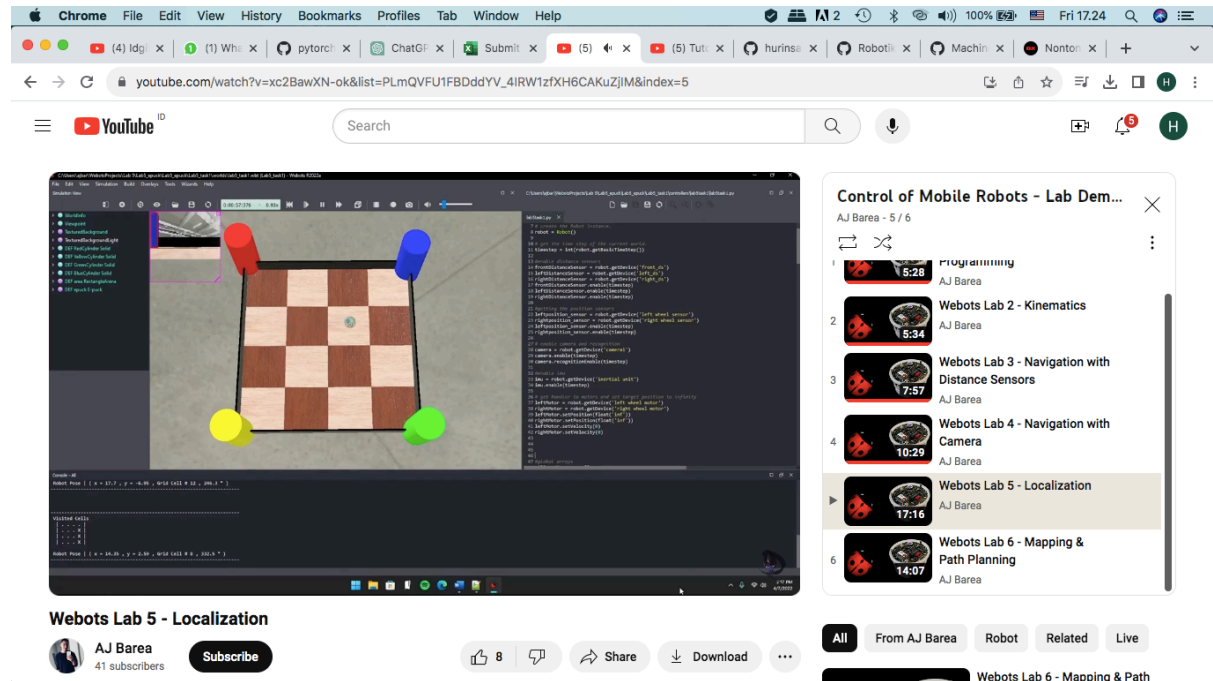
Tugas kedua mirip dengan tugas pertama, namun dengan tujuan untuk mendekati objek kuning hingga jarak lima inci. Pada tugas ini, fungsi PID ditambahkan untuk mengendalikan jarak robot dari objek kuning tersebut.

Tugas ketiga mengimplementasikan algoritma bug zero, yang memungkinkan robot untuk menuju tujuan tanpa terhalang oleh rintangan. Namun, jika terhalang, robot mengikuti rintangan tersebut ke sekitar tepinya untuk mencapai tujuan.

Tugas terakhir, yang belum selesai, mencoba menerapkan algoritma bug zero dengan beberapa perbaikan pada PID controller. Namun, ada kegagalan yang terjadi di mana robot terjebak dalam lingkaran berulang sambil mencoba mendekati tujuan karena permasalahan pada algoritma tersebut. Upaya dilakukan untuk memperbaiki kegagalan ini dengan memecahkan dinding di sudut-sudut tertentu.

Semua tugas dalam lab ini berfokus pada penggunaan kamera untuk mendeteksi objek kuning (atau silinder) sebagai tujuan, serta implementasi kontrol logika PID untuk menjaga robot pada orientasi, kecepatan, dan jarak yang diinginkan terhadap objek tersebut.

Video 5



Lab 5 memperkenalkan konsep lokalisasi, yang bertujuan untuk mengetahui posisi robot dalam peta pada waktu tertentu. Tugas pertama melibatkan pemetaan grid dan penggunaan algoritme trilaterasi untuk menemukan koordinat xy robot pada grid 4x4 dengan titik asal di tengahnya.

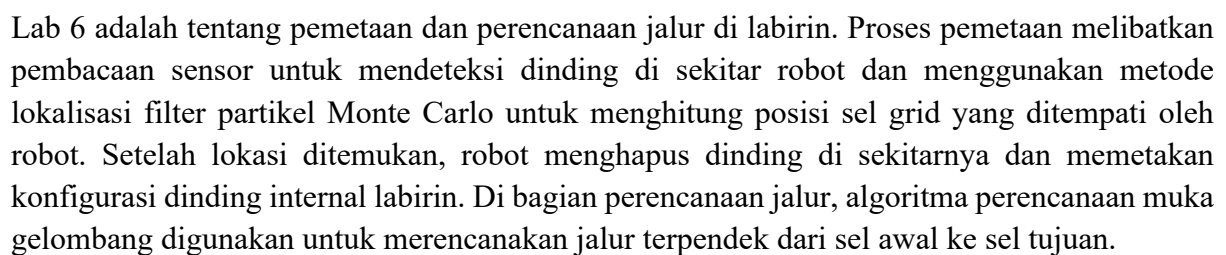
Trilaterasi adalah metode yang menggunakan jarak dari titik-titik yang diketahui untuk menemukan lokasi yang tepat dari suatu titik di ruang. Contohnya adalah menggunakan koordinat xy dari tiga satelit dan jarak dari robot ke masing-masing satelit untuk menentukan posisi robot.

Tugas kedua memfokuskan pada penggunaan sensor jarak dan arah kompas untuk menentukan lokasi robot berdasarkan apakah sensor jarak mendeteksi dinding atau tidak. Dengan memetakan dinding pada grid, robot dapat menentukan lokasinya dengan menggunakan informasi dari sensor jarak dan orientasi yang terbaca.

Pada tugas kedua, algoritma memungkinkan robot untuk bergerak di sekitar grid, mencatat sel mana yang telah dikunjungi berdasarkan pembacaan sensor jarak. Robot berhenti ketika semua sel grid telah dikunjungi atau ketika semua titik telah diubah menjadi tanda 'x'.

Kedua tugas melibatkan penggunaan sensor jarak dan arah kompas untuk menentukan posisi robot dalam grid yang telah ditentukan sebelumnya, dengan tujuan memahami konsep

Video 6



Untuk tugas 2, karena kita sudah mengetahui tata letak ruang kosong dan dinding di labirin sebelumnya, algoritma perencanaan muka gelombang digunakan untuk merencanakan jalur terpendek dari sel awal ke sel tujuan tanpa harus menjelajahi setiap sel dalam grid.

Pada akhirnya, robot dijalankan berdasarkan instruksi yang dihasilkan dari perencanaan jalur, dan jalur-jalur ini direncanakan dengan mempertimbangkan konektivitas antar-sel dalam labirin untuk menemukan jalur terpendek yang memungkinkan robot mencapai sel tujuan.