

Laporan Proyek Simulasi Robot Berpenggerak Diferensial di Webots

Nama Proyek: Simulasi Robot Berpenggerak Diferensial

Perangkat Lunak: Webots

Bahasa Pemrograman: Python

1. Latar Belakang

Robot berpenggerak diferensial banyak digunakan dalam aplikasi robotika, terutama dalam sistem navigasi sederhana yang memungkinkan robot bergerak maju, mundur, dan berbelok dengan menggunakan dua roda utama. Sistem penggerak ini menawarkan fleksibilitas untuk berbelok di tempat tanpa memerlukan roda kemudi tambahan, menjadikannya pilihan umum dalam robot otonom, seperti robot pembersih dan robot eksplorasi.

Proyek ini bertujuan untuk mensimulasikan sebuah robot berpenggerak diferensial dua roda di Webots, sebuah perangkat lunak simulasi robotik yang memungkinkan pengguna merancang, menguji, dan mengontrol berbagai jenis robot di lingkungan virtual.

2. Tujuan

Tujuan dari proyek ini adalah:

- Mendesain dan memprogram sebuah robot berpenggerak diferensial dua roda di Webots.
- Menggunakan bahasa Python untuk mengendalikan pergerakan robot secara real-time.
- Menganalisis hasil simulasi untuk memahami cara kerja penggerak diferensial dalam mengontrol pergerakan robot.

3. Metode

3.1 Perancangan Robot di Webots

Robot dirancang di Webots menggunakan fitur-fitur berikut:

- Badan Robot: Robot memiliki badan utama yang berfungsi sebagai tempat pemasangan komponen.
- Penggerak Diferensial: Robot menggunakan dua motor berjenis `RotationalMotor`, masing-masing terhubung ke roda kiri dan kanan. Motor diberi nama "Motor_1" dan "Motor_2".
- Pengaturan Motor: Kedua motor diset dalam mode rotasi bebas (`Position` disetel ke

`float("inf"))`), memungkinkan robot untuk terus berputar dan bergerak secara konstan.

3.2 Pemrograman Robot dengan Python

Kode untuk mengendalikan robot ditulis dalam Python. Struktur kode dasar terdiri dari:

1. Inisialisasi Robot dan Perangkat: Membuat instance robot, mengakses kedua motor, dan mengatur posisi rotasi motor.
2. Pengaturan Kecepatan Motor: Menetapkan kecepatan motor secara konstan atau dinamis untuk membuat robot bergerak maju, mundur, atau berbelok.
3. Loop Utama: Mengatur loop untuk mengontrol kecepatan motor secara berkelanjutan selama simulasi.

```
from controller import Robot

# Inisialisasi robot
robot = Robot()
timestep = int(robot.getBasicTimeStep())

# Mengakses motor kiri dan kanan
left_motor = robot.getDevice("Motor_1")
right_motor = robot.getDevice("Motor_2")
left_motor.setPosition(float("inf"))
right_motor.setPosition(float("inf"))

# Pengaturan kecepatan
default_speed = 1.0

# Loop utama
while robot.step(timestep) != -1:
    left_motor.setVelocity(default_speed)
    right_motor.setVelocity(default_speed)
```

4. Hasil

Setelah menjalankan simulasi, robot berhasil bergerak sesuai instruksi kecepatan yang diberikan dalam kode.

Berikut adalah hasil-hasil utama yang diamati:

1. Gerakan Lurus: Dengan mengatur kecepatan yang sama pada kedua motor, robot dapat bergerak lurus ke depan.

2. Belokan: Dengan mengatur kecepatan motor kiri atau kanan lebih rendah dari motor lainnya, robot berhasil berbelok ke arah yang diinginkan.
3. Stabilitas Gerakan: Robot bergerak dengan stabil pada kecepatan yang ditentukan dan bisa berhenti dengan mengatur kecepatan motor menjadi nol.

5. Pembahasan

Proyek ini berhasil menunjukkan bahwa sistem penggerak diferensial dapat digunakan untuk mengontrol pergerakan robot dengan fleksibel. Beberapa poin penting dalam pembahasan ini adalah:

1. Pengaturan Nama Perangkat: Pada awalnya, terdapat kendala karena nama perangkat motor dalam kode Python tidak sesuai dengan nama perangkat di file konfigurasi dunia Webots (Uts.wbt). Penyesuaian ini berhasil mengatasi masalah robot yang tidak merespons perintah kecepatan.
2. Penggunaan Mode Rotasi Bebas: Pengaturan posisi motor ke float("inf") memungkinkan robot bergerak tanpa batas rotasi. Ini penting untuk memastikan robot dapat berputar terus-menerus tanpa terbatas jarak putar tertentu.
3. Kecepatan Motor: Kecepatan motor dapat diatur secara dinamis untuk membuat robot berbelok atau bergerak lurus. Ini memberikan fleksibilitas dalam navigasi dan manuver di lingkungan simulasi.
4. Efek Fisika Webots: Webots memberikan efek fisika realistis, yang penting untuk simulasi. Dalam proyek ini, penyesuaian kecepatan juga mempertimbangkan interaksi robot dengan lantai simulasi untuk menghindari tergelincir atau ketidakstabilan.

6. Kesimpulan

Proyek simulasi robot berpenggerak diferensial dua roda ini berhasil mencapai tujuan utama, yaitu memahami dan mengaplikasikan sistem penggerak diferensial menggunakan Webots dan Python. Berikut adalah beberapa kesimpulan utama:

1. Implementasi Penggerak Diferensial: Sistem penggerak diferensial dapat dikendalikan dengan baik menggunakan dua motor independen. Hal ini memungkinkan robot untuk bermanuver dengan lincah, baik untuk bergerak maju, mundur, maupun berbelok.
2. Fleksibilitas dan Stabilitas: Dengan mengatur kecepatan motor secara tepat, robot mampu bergerak dengan stabil di lingkungan simulasi. Webots memberikan dukungan simulasi yang baik untuk memahami pengaruh fisika terhadap gerakan robot.
3. Potensi Pengembangan: Proyek ini bisa dikembangkan lebih lanjut dengan

menambahkan sensor atau perangkat lainnya, sehingga robot bisa berfungsi sebagai robot otonom yang mampu mendeteksi dan menghindari rintangan.

Proyek ini merupakan dasar yang baik untuk aplikasi yang lebih kompleks di bidang robotika. Dengan menggunakan simulasi, kita dapat memahami cara kerja penggerak diferensial tanpa risiko kerusakan pada perangkat keras. Webots menyediakan platform yang sangat berguna untuk eksplorasi robotik sebelum implementasi di dunia nyata.