

```
from controller import Robot
```

```
TIME_STEP = 32
```

```
# Fungsi Kalman Filter
```

```
def kalman_filter(z, u, x, P):
```

```
    # Prediksi langkah
```

```
    x_pred = x + u
```

```
    P_pred = P + 0.1 # Noise proses
```

```
    # Koreksi langkah
```

```
    K = P_pred / (P_pred + 1) # Gain Kalman
```

```
    x = x_pred + K * (z - x_pred) # Pembaruan posisi
```

```
    P = (1 - K) * P_pred # Pembaruan ketidakpastian
```

```
    return x, P
```

```
# Inisialisasi robot
```

```
robot = Robot()
```

```
# Motor roda
```

```
left_motor = robot.getDevice("left wheel motor")
```

```
right_motor = robot.getDevice("right wheel motor")
```

```
left_motor.setPosition(float('inf')) # Mode kecepatan
```

```
right_motor.setPosition(float('inf')) # Mode kecepatan
```

```
left_motor.setVelocity(50.0)
```

```
right_motor.setVelocity(50.0)
```

```

# Encoder roda

left_encoder = robot.getDevice("left wheel sensor")
right_encoder = robot.getDevice("right wheel sensor")
left_encoder.enable(TIME_STEP)
right_encoder.enable(TIME_STEP)


# Sensor jarak

distance_sensor = robot.getDevice("distance sensor")
distance_sensor.enable(TIME_STEP)


# Variabel untuk Kalman Filter

x = 0.0 # Posisi awal
P = 1.0 # Ketidakpastian awal


# Loop utama

while robot.step(TIME_STEP) != -1:

    # Ambil nilai encoder

    left_distance = left_encoder.getValue()
    right_distance = right_encoder.getValue()


    # Estimasi pergerakan robot (input u)

    u = (left_distance + right_distance) / 2.0


    # Ambil pengukuran sensor jarak (z)

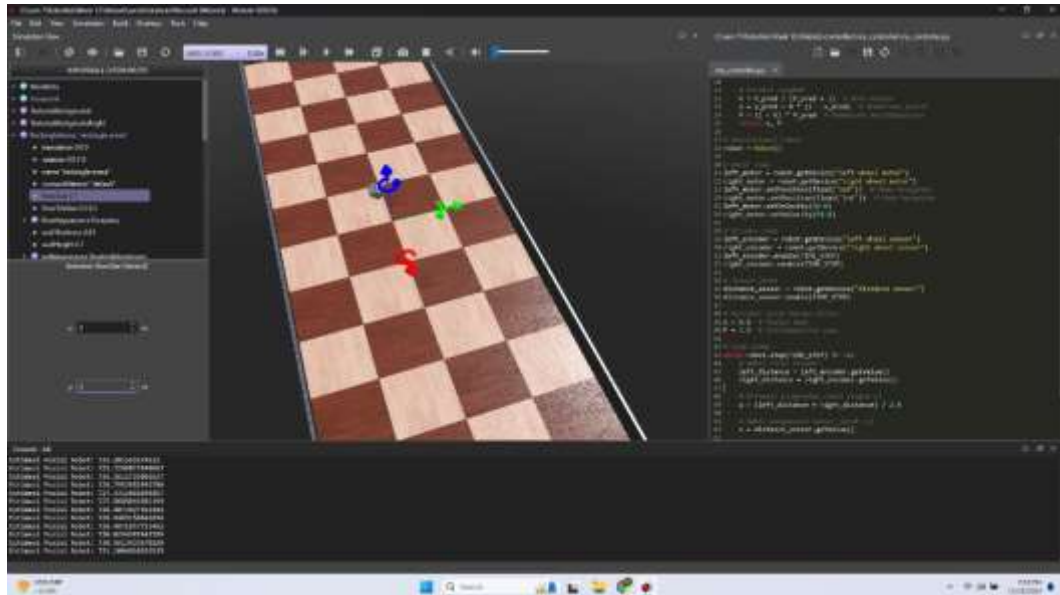
    z = distance_sensor.getValue()


    # Terapkan Kalman Filter

```

```
x, P = kalman_filter(z, u, x, P)
```

```
print(f"Estimasi Posisi Robot: {x}")
```



Penjelasan

1. Kalman Filter

Fungsi `kalman_filter`:

Fungsi ini digunakan untuk memproses data dari sensor dengan Kalman Filter, yang menggabungkan prediksi posisi robot dari encoder roda dan pengukuran jarak dari sensor untuk memperbaiki estimasi posisi.

Parameter:

`z`: Pengukuran sensor jarak.

`u`: Estimasi pergerakan (dari encoder roda).

`x`: Estimasi posisi sebelumnya.

`P`: Ketidakpastian posisi sebelumnya.

Proses Kalman Filter:

Langkah Prediksi:

`x_pred`: Prediksi posisi berdasarkan pergerakan.

P_pred: Prediksi ketidakpastian dengan menambahkan noise proses.

Langkah Koreksi:

K: Kalman Gain, menghitung seberapa besar kepercayaan pada pengukuran sensor.

x: Pembaruan estimasi posisi berdasarkan prediksi dan pengukuran sensor.

P: Pembaruan ketidakpastian.

2. Inisialisasi Robot

Robot Controller:

robot = Robot() menginisialisasi kontrol robot di Webots.

Motor Roda:

left_motor dan right_motor mengendalikan roda kiri dan kanan.

setPosition(float('inf')) mengatur motor ke mode kecepatan.

setVelocity(50.0) mengatur kecepatan awal motor ke 50.0 (nilai arbitrer).

Encoder Roda:

left_encoder dan right_encoder digunakan untuk membaca jarak yang ditempuh masing-masing roda.

Fungsi enable(TIME_STEP) mengaktifkan encoder dengan pembaruan data setiap 32 ms.

Sensor Jarak:

distance_sensor digunakan untuk membaca jarak ke objek di depan robot.

Fungsi enable(TIME_STEP) mengaktifkan sensor dengan pembaruan data setiap 32 ms.

3. Variabel untuk Kalman Filter

x: Posisi awal robot, diinisialisasi ke 0.0.

P: Ketidakpastian awal posisi, diinisialisasi ke 1.0 (nilai arbitrer).

4. Loop Utama

Fungsi `robot.step(TIME_STEP)` digunakan untuk menjalankan simulasi dalam langkah waktu 32 ms. Loop ini akan terus berjalan hingga simulasi dihentikan.

Proses Utama:

Baca Data Sensor:

Encoder roda (`left_distance` dan `right_distance`) digunakan untuk memperkirakan jarak rata-rata yang ditempuh robot.

Sensor jarak (`z`) memberikan pengukuran posisi dari lingkungan.

Hitung Estimasi Pergerakan (Input `u`):

`u` dihitung sebagai rata-rata jarak yang ditempuh oleh roda kiri dan kanan.

Terapkan Kalman Filter:

Posisi robot (`x`) diperbarui menggunakan hasil dari Kalman Filter, yang menggabungkan estimasi pergerakan (`u`) dan pengukuran sensor (`z`).

Cetak Posisi:

Estimasi posisi robot dicetak setiap langkah waktu dengan:

python

Copy code

```
print(f"Estimasi Posisi Robot: {x}")
```

Fungsi Utama Kode

Kode ini mensimulasikan robot bergerak dan menggunakan Kalman Filter untuk memperkirakan posisi robot secara lebih akurat dengan:

Menggabungkan data pergerakan dari encoder roda.

Memanfaatkan pengukuran jarak dari sensor.

Kalman Filter membantu mengurangi pengaruh noise pada data sensor, sehingga estimasi posisi robot menjadi lebih stabil dan dapat diandalkan.