

Analisis Kontrol Robot e-Puck Menggunakan Webots

1. Pendahuluan

Robot otonom seperti e-puck sering digunakan dalam simulasi untuk eksperimen kontrol gerak. Pada eksperimen ini, terdapat dua jenis kontrol yang digunakan, yaitu **kontrol open-loop** dan **kontrol closed-loop**. Kedua pendekatan ini berbeda dalam cara robot merespons lingkungannya. Dengan menggunakan Webots, platform simulasi yang mendukung berbagai model robot, eksperimen ini berfokus pada penggunaan sensor dan kontrol motor pada robot e-puck untuk menjalankan berbagai skenario gerak.

2. Kontrol Open-Loop

Pada program pertama (e-puck_open_loop.py) dan kedua (e-puck_circle_motion.py), digunakan **kontrol open-loop**. Kontrol open-loop adalah sistem di mana robot tidak menerima feedback dari lingkungan, dan perintah gerak diberikan secara langsung tanpa memperhatikan kondisi sekitar. Dalam program pertama, robot bergerak maju secara konstan dengan kecepatan yang sama pada kedua motor. Sedangkan, pada program kedua, robot bergerak melingkar dengan memberikan kecepatan yang berbeda pada motor kiri dan kanan.

Dalam kedua program tersebut, kontrol open-loop memiliki beberapa keuntungan dan keterbatasan:

- **Keuntungan:** Implementasi kontrol open-loop sangat sederhana karena hanya membutuhkan pengaturan kecepatan tanpa perlu memproses informasi dari lingkungan atau sensor. Hal ini membuat sistem lebih mudah diprogram dan dijalankan, terutama dalam simulasi yang membutuhkan kecepatan pengujian.
- **Keterbatasan:** Keterbatasan utama dari kontrol open-loop adalah kurangnya adaptasi terhadap lingkungan. Jika terdapat hambatan di depan robot, ia tidak akan menghentikan gerakannya karena tidak ada mekanisme untuk menerima feedback. Dalam konteks dunia nyata, hal ini dapat menyebabkan tabrakan atau kegagalan dalam menyelesaikan tugas, sehingga kontrol open-loop hanya cocok untuk skenario yang sangat terkendali atau yang tidak memerlukan interaksi dengan lingkungan.

Contoh sederhana dari kontrol open-loop di dunia nyata adalah penggunaan conveyor belt pada pabrik yang berjalan dengan kecepatan tetap. Conveyor belt tersebut tidak akan berhenti atau memperlambat kecuali diatur oleh operator manual, tidak peduli apakah ada barang yang mungkin jatuh dari belt atau tidak.

3. Kontrol Closed-Loop dengan Sensor Proximity

Pada program ketiga (e-puck_proximity_stop.py), dilakukan peningkatan dengan menggunakan **kontrol closed-loop**. Kontrol closed-loop memungkinkan robot untuk berinteraksi dengan lingkungannya melalui penggunaan sensor proximity. Pada simulasi ini, e-puck dilengkapi dengan delapan sensor proximity yang ditempatkan di berbagai titik di sekitar robot, khususnya pada bagian depan dan belakang. Sensor ini memungkinkan robot untuk mendeteksi keberadaan objek di sekitarnya.

Proses kerja sistem ini adalah sebagai berikut:

- **Pembacaan Sensor:** Pada setiap langkah simulasi, sensor proximity membaca jarak objek di depan robot.
- **Keputusan Kontrol:** Jika sensor mendeteksi adanya objek dalam jarak tertentu (threshold), robot akan menghentikan pergerakannya. Jika tidak ada objek, robot akan terus bergerak maju.

Keuntungan dari kontrol closed-loop ini cukup signifikan:

- **Adaptasi terhadap Lingkungan:** Robot menjadi lebih "cerdas" karena dapat merespons lingkungan di sekitarnya. Dalam hal ini, robot akan berhenti saat mendeteksi objek di depan, yang mencegah terjadinya tabrakan.
- **Keandalan dan Keamanan:** Dalam situasi nyata, kontrol closed-loop lebih aman dan andal karena mampu bereaksi terhadap kejadian tak terduga, seperti rintangan yang tiba-tiba muncul.

Namun, penggunaan kontrol closed-loop juga memiliki tantangan. Misalnya, sistem harus diatur dengan hati-hati agar threshold sensor yang digunakan sesuai dengan kondisi sebenarnya. Jika threshold terlalu rendah, robot bisa berhenti terlalu sering, bahkan jika tidak ada hambatan nyata. Sebaliknya, jika threshold terlalu tinggi, robot mungkin gagal mendeteksi objek kecil di depannya.

4. Aplikasi di Dunia Nyata

Kontrol open-loop dan closed-loop memiliki aplikasi yang luas dalam dunia nyata. Sebagai contoh, sistem kontrol open-loop sering ditemukan pada perangkat yang tidak memerlukan interaksi dengan lingkungannya, seperti pemanas ruangan yang beroperasi dengan pengaturan waktu tetap. Sebaliknya, kontrol closed-loop sering diterapkan dalam sistem yang memerlukan respons real-time, seperti drone atau mobil otonom yang harus mendeteksi dan menghindari hambatan selama operasi.

Pada mobil otonom, penggunaan sensor proximity atau kamera untuk mendeteksi rintangan di jalan adalah contoh dari kontrol closed-loop. Mobil-mobil ini dilengkapi dengan berbagai sensor yang secara konstan memantau lingkungan sekitar dan menyesuaikan kecepatan atau arah untuk menghindari kecelakaan.

5. Kesimpulan

Eksperimen ini menunjukkan perbedaan antara kontrol open-loop dan closed-loop dalam pengendalian robot. Kontrol open-loop, meskipun sederhana, memiliki keterbatasan dalam beradaptasi dengan lingkungan yang berubah. Sebaliknya, kontrol closed-loop menawarkan keandalan dan keamanan yang lebih baik dengan memanfaatkan feedback dari sensor untuk mengatur gerakan robot secara dinamis.

Dalam penerapannya, memilih antara kontrol open-loop dan closed-loop tergantung pada kebutuhan sistem. Jika lingkungan statis dan prediktif, kontrol open-loop dapat menjadi

pilihan yang tepat. Namun, jika interaksi dengan lingkungan diperlukan, seperti dalam aplikasi robotika modern atau kendaraan otonom, kontrol closed-loop adalah pendekatan yang lebih efektif.