

SOAL 4 BAGIAN B

Valentino Daniel Kusumo – 19624107

Algoritma adalah sekumpulan aturan-aturan berhingga yang memberikan sederetan operasi-operasi untuk menyelesaikan masalah tertentu. Algoritma adalah upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu. Ada beberapa algoritma yang digunakan, yaitu kinematics, ADRC, PID, dan A*. Berikut penjelasan masing masing algoritma.

Kinematics

Pergerakan objek dan posisi relatif mereka dalam ruang 3D tanpa memperhatikan gaya yang menyebabkan pergerakan tersebut. Tiga aplikasi utamanya adalah

1. **Object Detection** : Algoritma deteksi objek bertujuan mengidentifikasi dan melokalisasi objek di dalam gambar atau video. Kinematika dalam deteksi objek digunakan untuk memahami bagaimana objek bergerak dalam suatu adegan berdasarkan data visual (seperti gambar atau video).
2. **Pose Estimation** : Estimasi pose bertujuan untuk menentukan posisi dan orientasi objek atau manusia dalam ruang 3D berdasarkan gambar 2D. Teknik ini memanfaatkan **kinematika invers** atau *inverse kinematics* (ruang 3D) yang adalah penggunaan persamaan kinematik untuk menentukan gerak robot untuk mencapai posisi yang diinginkan, diberikan posisi end-effector, maka yang akan dicari adalah berapa besar sudut yang harus diubah untuk tiap joint untuk dapat mencapai posisi end-effector tersebut. Dalam *inverse kinematics*, tujuannya adalah menemukan sudut-sudut sendi yang diperlukan untuk mencapai posisi dan orientasi tertentu.
3. **Camera Calibration** : Model ini menjelaskan bagaimana titik-titik di dunia dipetakan ke koordinat ruang gambar. Hal ini dilakukan dengan terlebih dahulu mengubah titik-titik yang diberikan dalam koordinat dunia ke dalam bingkai referensi kamera, lalu memproyeksikan titik-titik tersebut ke bidang gambar. Untuk menggunakan model ini, kita perlu mengetahui parameternya, seperti panjang fokus f dan miring S kamera (parameter intrinsik), dan posenya relatif terhadap bingkai dunia atau robot (parameter ekstrinsik). Kalibrasi mengacu pada proses estimasi parameter ini. Kalibrasi kamera memungkinkan kita mengubah deteksi ini menjadi kerangka acuan robot. Hasilnya kemudian dapat digunakan untuk memahami geometri lajur dan di mana robot berada relatif terhadap lajur.

ADRC

Teknik kontrol bebas model yang digunakan untuk merancang pengontrol untuk sistem dengan dinamika yang tidak diketahui dan gangguan eksternal. Pendekatan ini hanya memerlukan representasi estimasi perilaku sistem untuk merancang pengontrol yang secara efektif menangkal gangguan tanpa menyebabkan overshooting. ADRC terdiri dari pembeda pelacakan (TD), pengamat keadaan yang diperluas (ESO), hukum kontrol umpan balik kesalahan keadaan nonlinier (NLSEF), dan perangkat kompensasi gangguan (DCD).

- **Tracking Differentiator (TD)** : Bagian ini digunakan untuk menghasilkan sinyal referensi dengan melakukan filtrasi sinyal input untuk mencegah gangguan yang tiba-tiba. Hal ini dilakukan dengan mendeteksi dinamika sinyal masukan.
- **Extended State Observer (ESO)** : ESO digunakan untuk mengestimasi keadaan sebenarnya dari sistem dan gangguan eksternal yang tidak diketahui. ESO menggabungkan pengamatan dari keluaran sistem dan estimasi model untuk terus-menerus memantau perubahan pada keadaan sistem.
- **Nonlinear State Error Feedback (NLSEF)** : Bagian ini mengatur bagaimana kontrol diberikan pada sistem berdasarkan error (kesalahan) yang diperkirakan oleh ESO. NLSEF memastikan bahwa gangguan eksternal yang diprediksi tidak mengganggu performa sistem.
- **Disturbance Compensation Device (DCD)** : sebuah mekanisme yang digunakan dalam sistem kontrol untuk mengatasi gangguan (disturbances) yang dapat mempengaruhi performa sistem. DCD bekerja dengan cara memperkirakan dan mengompensasi gangguan-gangguan yang tidak diketahui atau sulit dimodelkan secara langsung. Ini biasanya diterapkan dalam sistem kontrol yang memerlukan tingkat kestabilan dan ketepatan yang tinggi

PID (Proportional-Integral-Derivative) Control Algorithm

Instrumen yang menerima data masukan dari sensor, menghitung perbedaan antara nilai aktual dan titik setel yang diinginkan, dan menyesuaikan keluaran untuk mengontrol variabel seperti suhu, laju aliran, kecepatan, tekanan, dan tegangan

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

dengan K_p , K_i , and K_d , semuanya positif, menandakan koefisien untuk term proporsional, integral, dan derivatif secara berurutan (atau P , I , dan D). Pada model ini

- P bertanggung jawab untuk nilai kesalahan saat ini. Contohnya, jika nilai kesalahan besar dan positif, maka keluaran kontrol juga besar dan positif.
- I bertanggung jawab untuk nilai kesalahan sebelumnya. Contoh, jika keluaran saat ini kurang besar, maka kesalahan akan terakumulasi terus menerus, dan kontroler akan merespon dengan keluaran lebih tinggi.
- D bertanggung jawab untuk kemungkinan nilai kesalahan mendatang, berdasarkan pada rate perubahan tiap waktu.

PID adalah algoritma kontrol yang digunakan secara luas dalam berbagai aplikasi industri untuk mengontrol proses seperti suhu, kecepatan, dan posisi

A* Algorithm

Algoritma yang digunakan untuk mencari lintasan terpendek antara titik awal dan titik akhir. Algoritma ini sering digunakan untuk penjelajahan peta guna menemukan jalur terpendek yang akan diambil. Aspek lain yang membuat A* begitu powerful adalah penggunaan graf berbobot dalam penerapannya. Graph berbobot menggunakan angka untuk mewakili biaya pengambilan setiap jalur atau tindakan. Ini berarti bahwa algoritma dapat mengambil jalur dengan biaya paling sedikit, dan menemukan rute terbaik dari segi jarak dan waktu

Adapun kelemahan utama dari algoritma ini adalah kompleksitas ruang dan waktunya. Algoritma A* membutuhkan banyak ruang untuk menyimpan semua kemungkinan jalur dan banyak waktu untuk menemukannya

A* menggunakan Best First Search (BFS) dan menemukan jalur dengan biaya terkecil (least-cost path) dari node awal (initial node) yang diberikan ke node tujuan (goal node).

Algoritma ini menggunakan fungsi heuristik jarak ditambah biaya (biasa dinotasikan dengan $f(x)$) untuk menentukan urutan di mana search-nya melalui node-node yang ada pada tree.

Notasi yang dipakai oleh algoritma A* adalah sebagai berikut:

$$f(n) = g(n) + h(n)$$

dimana

- $f(n)$ = biaya estimasi terendah
- $g(n)$ = biaya dari node awal ke node n
- $h(n)$ = perkiraan biaya dari node n ke node akhir

Cara kerja A* :

1. Mulai dari node awal, tambahkan ke daftar open (node yang akan dievaluasi).
2. Pilih node dengan nilai $f(n)$ terkecil.
3. Jika node tersebut adalah tujuan, jalur ditemukan.

4. Jika tidak, pindahkan node tersebut ke daftar closed (sudah dievaluasi) dan periksa semua tetangganya.
5. Untuk setiap tetangga, hitung nilai $f(n)$, dan tambahkan tetangga tersebut ke daftar open jika belum ada.
6. Ulangi proses hingga tujuan ditemukan atau daftar open kosong