

Kinematics

Algoritma kinematics memanfaatkan konsep kinematika di Fisika. Berbeda dengan dinamika, kinematika digunakan untuk menggambarkan pergerakan benda tanpa memerhatikan gaya-gaya penyebab dan gaya-gaya yang memengaruhi benda. Konsep algoritma ini digunakan dalam robotika untuk mendeteksi objek, mengestimasi orientasi/pose, dan mengkalibrasikan kamera.

- Object Detection

Umumnya, deteksi objek adalah identifikasi objek dalam suatu gambar atau video. Hal ini dapat dicapai dengan pendekatan berbasis fitur atau deep learning. Pendekatan berbasis fitur dilakukan dengan mengidentifikasi fitur-fitur khas objek dengan suatu algoritma. Di sisi lainnya, deep learning dilakukan dengan melatih suatu AI dengan suatu database berisi gambar atau video objek sehingga AI mampu mengidentifikasi objek melalui pola dan fitur objek.

- Pose Estimation

Estimasi pose adalah estimasi posisi dan orientasi suatu objek. Konsep ini kerap dihubungkan dengan computer vision dan AI. Estimasi pose suatu objek dapat dicapai dengan mendeteksi, menghubungkan, dan melacak pergerakan titik-titik semantik. Pada suatu robot, estimasi pose dapat dilakukan menggunakan sensor, seperti kamera atau LiDAR, untuk mendapatkan data dan informasi, seperti mengenai kedalaman objek, titik-titik semantik, dll. Estimasi pose dapat memanfaatkan algoritma Kalman Filters.

- Camera Calibration

Kalibrasi kamera adalah proses untuk mengatur dan menyesuaikan kamera ketika kamera tersebut digunakan untuk mengidentifikasi suatu objek. Kalibrasi ini dilakukan untuk meminimalisasikan kesalahan dalam deteksi akibat distorsi kamera, seperti distorsi lensa, fokus yang kurang tepat, dll. Umumnya, terdapat dua jenis distorsi, yakni distorsi radial dan distorsi tangensial. Distorsi radial menyebabkan garis-garis yang seharusnya lurus tampak melengkung, sedangkan distorsi tangensial merujuk pada jarak antara objek dengan kamera. Untuk mengatasi ini, kalibrasi kamera dilakukan dengan mengubah parameter intrinsik dan ekstrinsik kamera. Parameter intrinsik adalah parameter-parameter yang spesifik hanya pada kamera, seperti jarak fokus lensa. Di sisi lainnya, parameter ekstrinsik berhubungan dengan rotasi dan translasi vektor-vektor koordinat objek. Kalibrasi kamera umumnya menggunakan library dari OpenCV.

ADRC (Active Disturbance Rejection Control)

ADRC adalah sebuah teknik kontrol yang model-free. ADRC digunakan untuk merancang kontroler yang mengendalikan plant yang dinamika pergerakannya dan gangguannya tidak diketahui. Algoritma ini mampu membuat kontroler hanya dengan memperkirakan dinamika dan mengkompensasikan gangguan secara real-time. Cara kerja ADRC adalah dengan menolak gangguan-gangguan aktif dengan memperkirakan ketidakpastian dan gangguan yang dapat mengganggu sistem. ADRC menggunakan model aproksimasi ordo satu atau ordo dua. ADRC dapat dibagi menjadi 3 komponen, yakni:

- Tracking differentiator
Komponen ini berfungsi untuk menyaring input dan output untuk menghindari efek perubahan yang mendadak. Hal ini dicapai dengan mengambil setpoint dan menghasilkan respons yang halus/lemah sehingga output merespon secara bertahap.
- Extended State Observer
Komponen ini berfungsi untuk melacak kondisi sistem dan gangguan-gangguan eksternalnya. Selain itu, ESO juga berfungsi untuk memantau output sistem dan memperkirakan gangguan yang tidak diketahui. Hal ini dicapai dengan menganggap gangguan sebagai variabel tambahan dari sistem dan secara aktif memperkirakan perubahan variabel tersebut.
- Nonlinear State Error Feedback
Komponen ini berfungsi untuk memanfaatkan feedback akibat gangguan atau kesalahan untuk mengendalikan sistem. Setelah itu, komponen mengoreksi output sistem sehingga gangguan terminimalisir.

PID (Proportional-Integral-Derivative) control algorithms

Algoritma kontrol PID adalah suatu algoritma yang menghitung kesalahan antara nilai yang diinginkan dan output yang dihasilkan. Kesalahan yang digunakan adalah kesalahan lalu, sekarang, dan masa depan (past, present, future) dan algoritma akan memperbaikinya terus-menerus dengan tiga komponen utama sesuai dengan namanya, Proportional, Integral, dan Derivative.

- Proportional
Komponen proportional menghasilkan sinyal kontrol sebesar kesalahan yang dialami. Sinyal kontrol tersebut dirumuskan dengan mengalikan besarnya kesalahan dengan suatu konstanta, seperti konstanta proportional. Akan tetapi, kontrol ini dapat menghasilkan offset atau kesalahan kecil antara setpoint dan output, meskipun output sudah stabil. Untuk menghilangkan offset tersebut, digunakan komponen integral dan derivative.
- Integral
Komponen ini menghitung kesalahan kumulatif berdasarkan waktu dan mengoreksi kesalahan-kesalahan kecil yang masih ada. Seperti yang disebutkan, komponen ini dapat digunakan untuk menghilangkan offset oleh kontrol proportional. Kontrol integral dirumuskan dengan mengalikan integral dari kesalahan waktu ke waktu dengan konstanta integral.
- Derivative
Komponen derivative berfungsi untuk memperkirakan tren kesalahan berdasarkan laju perubahan kesalahan (turunan waktu dari kesalahan). Komponen ini juga berfungsi untuk mencegah terjadinya overshoot dan memperlambat kontrol jika laju tersebut terlalu tinggi. Komponen derivative dirumuskan sebagai perkalian antara konstanta derivative dengan turunan kesalahan terhadap waktu.

Gabungan dari ketiga komponen tersebut merupakan suatu algoritma yang menghasilkan sinyal kontrol yang kemudian dikirimkan ke plant. Adanya komponen I dan D mencegah komponen P untuk mengirimkan sinyal yang terlalu besar sehingga sistem mampu memperbaiki kesalahan tanpa menimbulkan osilasi atau overshoot yang terlalu besar.

A* (A star) algorithm

Algoritma A* adalah sebuah algoritma pencarian yang digunakan untuk mencari jalan terpendek untuk mencapai suatu titik akhir dari suatu titik awal. Algoritma ini umumnya menggunakan teknik map traversal dan memang awalnya dirancang sebagai soal graph traversal untuk membantu robot mencari jalan tercepat. A* umumnya digunakan pada suatu grid atau graf. Cara kerja algoritma ini adalah dengan menghitung dua nilai fungsi terlebih dahulu, yakni nilai dari $g(n)$ dan $h(n)$. Fungsi $g(n)$ adalah jarak antara titik awal hingga mencapai suatu kotak/node tertentu pada grid dengan mengikuti jalan yang telah dibuat. Di sisi lainnya, fungsi $h(n)$ adalah teknik heuristik untuk mengestimasi jarak dari node yang sekarang ke titik akhir. Fungsi akhir adalah penjumlahan antara kedua fungsi tersebut sehingga dapat dirumuskan sebagai berikut.

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

Algoritma akan mencari berbagai kemungkinan yang akhirnya akan menghasilkan nilai $f(n)$ yang terkecil (nilai dengan jarak total terkecil). Algoritma A* dianggap memiliki algoritma yang efisien, optimal, dan lengkap, serta mampu memecahkan berbagai persoalan yang membutuhkan pencarian jalur. Meskipun begitu, kelemahan A* terletak pada butuhnya memori besar untuk mengalokasikan kemungkinan-kemungkinan jalur dan kompleksitas waktu dalam mencari jalur yang bergantung pada heuristik, dengan kemungkinan terburuknya sebesar $O(b^d)$.