

1. Kinematics (Object Detection, Pose Estimation, Camera Calibration)

a. Object Detection

Object detection adalah proses identifikasi dan lokalisasi objek dalam gambar atau video. Algoritma ini bertujuan untuk menentukan keberadaan, jenis, serta posisi objek dalam suatu gambar. Beberapa metode yang digunakan untuk object detection mencakup teknik tradisional seperti metode Haar Cascade atau Histogram of Oriented Gradients (HOG), hingga metode yang lebih modern berbasis Convolutional Neural Networks (CNN) seperti YOLO (You Only Look Once) dan SSD (Single Shot Detector).

b. Pose Estimation

Pose estimation bertujuan untuk menentukan posisi dan orientasi (pose) objek atau manusia dalam ruang 2D atau 3D. Salah satu contoh yang umum adalah estimasi pose manusia dari gambar atau video, di mana algoritma ini mendeteksi titik-titik kunci tubuh seperti sendi-sendi. Teknik ini dapat dilakukan melalui pendekatan berbasis Deep Learning (seperti OpenPose) atau dengan metode geometris berbasis kamera ganda (stereo vision). Pose estimation sangat penting dalam augmented reality, robotics, dan animasi.

c. Camera Calibration

Camera calibration adalah proses untuk menentukan parameter intrinsik dan ekstrinsik kamera. Parameter intrinsik mengacu pada karakteristik kamera itu sendiri, seperti focal length, ukuran pixel, dan distorsi lensa. Parameter ekstrinsik mengacu pada posisi dan orientasi kamera terhadap objek yang sedang difoto. Proses ini dapat digunakan untuk menghitung hubungan antara koordinat gambar (2D) dan koordinat dunia nyata (3D), yang sangat penting untuk aplikasi, seperti robotika dan augmented reality.

2. ADRC (Active Disturbance Rejection Control)

ADRC adalah teknik kontrol yang digunakan untuk menangani gangguan atau ketidakpastian dalam suatu sistem tanpa memerlukan model matematis yang sangat presisi. Ide utama ADRC adalah dengan menganggap semua ketidakpastian sistem (baik dari model atau gangguan eksternal) sebagai suatu gangguan yang dapat ditolak secara aktif. ADRC terdiri dari tiga bagian utama:

- Extended State Observer (ESO), memprediksi dan memperkirakan gangguan total pada sistem, baik internal maupun eksternal, serta memperkirakan status sistem itu sendiri.
- State Feedback Control (SFC), menggunakan perkiraan dari ESO, pengontrol ADRC menghasilkan sinyal kontrol yang meminimalkan efek gangguan dan mencapai kontrol yang diinginkan.
- Nonlinear State Error Feedback (NLSEF), algoritma ini memastikan respons kontrol yang halus dan sesuai dengan kondisi aktual sistem.

Keunggulan ADRC dibandingkan metode tradisional seperti PID adalah kemampuannya untuk menangani dinamika sistem yang kompleks dan tidak linear dengan lebih efektif tanpa memerlukan model sistem yang detail.

3. PID (Proportional-Integral-Derivative) Control Algorithms

PID adalah algoritma kontrol yang paling banyak digunakan dalam sistem kontrol industri. PID mengendalikan suatu sistem dengan menyesuaikan tiga parameter utama:

- Proportional (P), menghasilkan sinyal kontrol yang sebanding dengan kesalahan saat ini (error antara setpoint dan output aktual). Semakin besar error, semakin besar output kontrol.
- Integral (I), menghitung jumlah kesalahan sebelumnya dan mengintegrasikannya untuk menghilangkan error yang terus-menerus atau steady-state error. Bagian integral meningkatkan respon kontrol berdasarkan akumulasi kesalahan.
- Derivative (D), berfokus pada laju perubahan kesalahan, yang membantu untuk meredam osilasi dan mengurangi respon yang terlalu agresif.

PID berfungsi dengan cara menggabungkan ketiga aspek ini untuk mencapai keseimbangan antara respon cepat (dari P), perbaikan steady-state (dari I), dan stabilitas sistem (dari D).

4. A* (A Star) Algorithm

A* adalah algoritma pencarian jalur yang digunakan dalam pemetaan, permainan, dan robotika untuk menemukan jalur terpendek antara dua titik (misalnya dari posisi awal ke tujuan) di lingkungan yang berisi halangan. A* adalah pengembangan dari algoritma Dijkstra yang menggabungkan efisiensi pencarian terpendek dengan heuristik yang memandu proses pencarian, sehingga lebih efisien.

Algoritma A* menggunakan dua fungsi utama:

- $g(n)$: Biaya dari titik awal hingga titik n (biaya aktual yang telah dilalui).
- $h(n)$: Estimasi biaya terpendek dari titik n ke tujuan (heuristik, sering dihitung sebagai jarak Euclidean atau Manhattan).

Fungsi total yang digunakan A* adalah: $f(n) = g(n) + h(n)$

Algoritma akan memilih jalur dengan nilai $f(n)$ terkecil, meminimalkan jarak aktual (g) sekaligus mempertimbangkan estimasi jarak tersisa (h). Penggunaan heuristik yang tepat memungkinkan A* menjadi algoritma yang optimal dan efisien untuk menemukan jalur terpendek dalam banyak kasus.

Masing-masing algoritma ini memiliki kegunaan yang luas dalam berbagai aplikasi, mulai dari robotika, kontrol otomatis, pengolahan citra, hingga perencanaan jalur.