URO TEST SOAL 4 NOMOR 2

Jelaskan lebih jauh masing-masing algoritma di bawah dengan detail :
- Kinematics (object detection, pose estimation, camera calibration)
- ADRC (Active Disturbance Rejection Control)
- PID (Proportional-Integral-Derivative) control algorithms
- A* (A star) algorithm

Jawaban:

1. Kinematics (object detection, pose estimation, camera calibration)
   a. Object detection

      Biasanya untuk object detection, model yang pernah saya gunakan dan sebagian besar orang gunakan adalah YOLO (You Only Look Once). YOLO sendiri memiliki banyak versi dan sering kali diimplementasikan oleh project-project yang cukup besar. Selain YOLO, model yang sering digunakan dan lebih konvensional adalah R-CNN (Region-based Convolutional Neural Networks). R-CNN sendiri menghasilkan beberapa model seperti Fast R-CNN, Faster R-CNN, dan Mask R-CNN.

      - YOLO

        YOLO adalah sebuah model object detection (bisa juga untuk image classification) yang menggunakan CNN (convolutional neural networks) untuk memprediksi bounding box dan melakukan klasifikasi terhadap objek. Ada 6 tahap pada algoritma YOLO secara general:
        - Image dimasukkan kedalam CNN agar dilakukan extract features
        - Features ini diteruskan melewati beberapa layer, yang akan memprediksi objek dan membuat prediksi lokasi bounding box.
        - Image dibagi menjadi banyak bagian berbentuk grid, dimana pada masing-masing grid di prediksi bounding box dan probability class.
        - Output dari network ini adalah berbagai bounding box dan probability class untuk setiap sel.
        - Bounding box kemudian akan di filter menggunakan Non-Maximum Suppression untuk menghapus box yang tumpang tindih.
        - Terakhir, akan muncul output berupa bounding boxes dan class labelnya pada image.
      - R-CNN

        R-CNN adalah sebuah model deep learning yang menggabungkan CNN (convolutional neural networks) dan model region-based, hence, the name. Cara kerja model ini secara umum dibagi 3 tahap, yaitu: extract region candidates/region proposal network, feature extraction, object classification.
        - Region proposal: langkah ini menggunakan Region Proposal Network (RPN) untuk membagi image (sebelumnya telah di input) menjadi beberapa region, yang disebut dengan region candidates.
        - Feature extraction: sekitar 2000 region yang telah di ekstrak, masuk ke dalam CNN model, biasanya harus di process agar semuanya jadi size yang sama, misalnya 224x224 atau 512x512 pixels.

- Object classification: feature yang sudah di ekstrak akan dimasukkan ke dalam classifier, yang akan memberi tahu setiap region proposal mengandung object yang di cari atau tidak. Kemudian, pada object detection, terdapat bounding box yang akan menandai lokasi objek yang dicari. Bounding box akan di lengkapi dengan Non-Maximum Suppression (NMS) yang akan mengeliminasi box yang tumpang tindih.

R-CNN lebih akurat secara general daripada YOLO, karena R-CNN menggunakan model dengan dua langkah (two-stage detector), sedangkan YOLO menggunakan model dalam satu percobaan (single-stage detector). Oleh karena itu, YOLO dikenal karena kecepatannya, sehingga cocok digunakan untuk *real-time* projects. Sebaliknya, R-CNN lebih lambat karena melewati two-stage process. Selain itu, R-CNN juga dikenal karena kemampuannya dalam mendeteksi objek yang kecil, salah satu aspek dimana YOLO kesulitan untuk bersaing. Terakhir, YOLO memiliki arsitektur yang lebih simpel dan mudah digunakan, karena sifatnya end-to-end. Sedangkan, R-CNN mempunyai beberapa komponen (RPN, classifier), yang membuatnya lebih sulit untuk di train.

Sumber: https://blog.roboflow.com/what-is-r-cnn/, https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z#yolo-algorithm--how-does-it-works?, https://blog.roboflow.com/guide-to-yolo-models/

b. Pose estimation

Pose estimation adalah sebuah pekerjaan computer vision, dimana tujuan akhirnya adalah untuk mendeteksi posisi dan orientasi dari seseorang atau sebuah objek. Biasanya, model ini mengidentifikasi keypoints seperti sendi dan bagian tubuh lainnya. Lokasi keypoints ini direpresentasikan dalam bentuk koordinat 2D [x, y] atau 3D [x, y, visible].

Model pose estimation ini ada bermacam-macam, misalnya OpenPose, HRNet, AlphaPose, bahkan YOLO. Namun, secara general, model-model ini adalah two-step frameworks yang mendeteksi bounding boxes dan memperkirakan pose objek pada masing-masing box. Selain itu, ada dua tipe pose estimation, yaitu multi-pose dan single-pose. Sesuai namanya, single-pose digunakan untuk satu objek pada suatu image/input, sedangkan multi-pose dapat memprediksi beberapa objek pada suatu image/input.

Sumber: https://viso.ai/deep-learning/pose-estimation-ultimate-overview/, https://paperswithcode.com/task/pose-estimation

c. Camera calibration

Geometric camera calibration, atau yang biasa disebut camera resectioning, merupakan proses prediksi parameter lensa dan sensor gambar dari sebuah input kamera, baik itu foto maupun video. Misalkan, bentuk video menggelembung karena lensa kamera, disebut juga distorsi, model ini dapat mengoreksi distorsi, mengukur ukuran asli suatu objek, atau mengidentifikasi lokasi dari kamera pada sebuah scene. Distorsi yang sering terjadi adalah radial distortion dan tangential distortion.

Algoritma secara general dari camera calibration yaitu pengumpulan gambar kalibrasi, keypoints detection, optimasi parameter intrinsik (focal length, optical center, skew

coefficient, distortion coefficient), estimasi parameter ekstrinsik (rotation, translation), mengoreksi distorsi kamera (radial dan tangensial), dan terakhir meminimalisir error dengan optimasi non-linear.

Sumber: https://www.mathworks.com/help/vision/ug/camera-calibration.html, https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

2. Active Disturbance Rejection Control (ADRC)

ADRC pada dasarnya adalah kemampuan untuk memperkirakan gangguan eksternal yang tidak diketahui. Teknik ini berfungsi untuk merancang kontroler untuk plant/alat dengan dinamika yang diketahui. Namun, algoritma ini memerlukan perkiraan dinamika plant agar dapat mendesain kontroller yang melakukan penolakan gangguan yang kuat. Ada beberapa komponen utama dalam ADRC:

- Tracking differentiator: menghasilkan lintasan referensi yang diinginkan untuk sistem.
- Extended state observer: mengawasi dan memperkirakan keadaan sistem serta gangguan yang tidak diketahui.
- Non-linear state error feedback: memberikan feedback terhadap error antara nilai output dan referensi untuk mengatur sinyal kontrol yang diberikan ke plant.

Sumber: https://www.mathworks.com/help/slcontrol/ug/active-disturbance-rejection-control.html, https://www.sciencedirect.com/topics/engineering/active-disturbance-rejection-control

3. PID (Proportional-Integral-Derivative) control algorithms

Seperti namanya, algoritma PID memiliki tiga bagian utama, yaitu: proportional, integral, dan derivative. PID sendiri merupakan salah satu contoh sistem loop tertutup atau closed loop system. Ketika loop PID sedang berlangsung, algoritma ini menghitung error value dari output yang dikeluarkan dengan nilai referensi. Kemudian, kontroler akan meminimalisir error dengan mengubah variabel kontrol. Dalam PID kontrol, ada tiga komponen utama yang bekerja bersama: proportional (P), integral (I), dan derivative (D).

Saat ini, ada tiga algoritma PID kontrol sederhana, yaitu incremental algorithm, position type algorithm, differential algorithm.

- Incremental algorithm menghitung perubahan pada sinyal kontrol berdasarkan perubahan pada error (selisih nilai referensi dan nilai output).
- Position type algorithm menghasilkan sinyal kontrol yang merepresentasikan posisi sinyal kontrol, berbeda seperti perubahan yang dicatat pada incremental algorithm. Pada algoritma ini, sinyal kontrol dihitung sebagai proportional terhadap error, integral dari error sebelumnya, dan turunan dari perubahan error.
- Differential algortihm lebih berfokus pada komponen D (derivative), yaitu menghitung turunan error untuk mendapatkan informasi tentang perubahan cepat dalam error.

PID control dapat diterapkan pada berbagai sistem, mulai dari cruise control mobil hingga aplikasi kontrol proses yang kompleks. Tantangan utama dalam implementasi PID controller adalah memilih nilai yang tepat untuk nilai proporsional, integral, dan derivatif, serta memastikan stabilitas loop sistem tertutup.

Sumber: https://www.wevolver.com/article/pid-loops-a-comprehensive-guide-to-understanding-and-implementation, https://www.veichi.com/knowledge/what-is-pid-control-algorithm.html

4. A* algorithm

Algoritma ini adalah algoritma search atau algoritma pencarian yang digunakan untuk mencari jalan terpendek dari titik awal ke titik akhir. Awalnya, A* di desain sebagai graph tranversal problem untuk membantu dalam pembuatan robot yang dapat mencari jalannya sendiri. Sampai sekarang, A* algorithm ini merupakan salah satu algoritma yang paling baik dan populer untuk path-finding. A* menggabungkan breadth-first search dan greedy best-first search untuk mendapatkan solusi yang paling efisien.

Semua grafik memiliki node atau titik yang harus dilewati oleh algoritma untuk mencapai tujuan. Setiap node ini memiliki nilai numerik. Jumlah total nilai semua jalur yang dilewati disebut sebagai cost dari rute tersebut.

Jadi, algoritma A* memilih node berdasarkan nilai 'f', yang merupakan parameter bernilai jumlah dari dua parameter lainnya – 'g' dan 'h'. Pada setiap langkah, algoritma memilih node dengan nilai 'f' terendah, kemudian memproses node atau sel tersebut. Definisi 'g' dan 'h' adalah:

- g : movement cost untuk berpindah dari titik awal ke sebuah node tertentu pada grid, mengikuti jalur yang telah dihasilkan.
- h : perkiraan cost pergerakan dari node tersebut ke tujuan akhir. Ini sering disebut sebagai heuristic, yang pada dasarnya merupakan tebakan pintar. Kita tidak tahu jarak sebenarnya sampai kita menemukan jalurnya, karena ada berbagai hal yang mungkin menghalangi.

Ada dua cara kita untuk mencari nilai h, yaitu mencari nilai h sebenarnya (yang akan memakan waktu), dan melakukan pendekatan nilai h (lebih hemat waktu. Metode ini melibatkan pre-komputasi jarak antara setiap pasangan node sebelum algoritma A*, atau menggunakan rumus jarak euclidean jika tidak ada rintangan untuk menghitung jarak lurus antara titik awal dan tujuan. Sedangkan pada pendekatan nilai h, kita dapat menggunakan manhattan distance, diagonal distance, dan euclidian distance.

Namun, A* sebagai algoritma path-finding paling baik, algoritma ini tidak selalu menghasilkan path terpendek, karena algoritma ini bergantung kepada heuristics.

Sumber: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm, https://www.geeksforgeeks.org/a-search-algorithm/