Nama : Dominikus David
Fakultas : STEI-R
NiM : 16524255

1. **Kinematics (object detection, pose estimation, camera calibration)**
    a. Object Detection
        - Merupakan proses untuk menemukan dan mengenali suatu objek melalui sebuah gambar atau video.
        - Algoritma ini akan mencari suatu pola yang spesifik dari suatu objek yang dideteksi. Teknik yang digunakan seperti convolutional neural networks (CNN). Jadi nanti akan diekstraksi polanya dari gambar, baru dideteksi objeknya.
        - Cara kerja detailnya :
            - Input suatu gambar (dapat berupa gambar statis ata video).
            - Algoritma akan menganalisis inputan untuk mengenali pola spesifiknya.
            - Setelah itu, nanti ia akan menentukan lokasi objek dengan menggambar suatu kotak penanda di sekitar objek target.
            - Setelahnya ia akan labeling si objeknya (kaya di kasi nama).
    b. Pose Estimation
        - Pose estimation adalah proses atau algoritma untuk menentukan posisi dan arah suatu objek dalam ruang 2D atau 3D.
        - Algoritma akan menggunakan key points atau titik kunci seperti sendi objek untuk menghitung sudut dan posisi relatif dari bagian-bagian tersebut.
        - Cara kerja detailnya :
            - Ia akan mencari key points seperti sendi, lutut, tangan pada objek di suatu gambar.
            - Ia akan menghubungkan beberapa key points tadi untuk menjadi suatu kerangka yang mewakili posisi objek.
            - Ia akan menghitung sudut dan arah setiap bagian untuk tau posisi detail objek.
    c. Camera Calibration
        - Camera calibration adalah algoritma untuk memahami bagaimana kamera menangkap gambar dan melakukan penyesuaian atau koreksi agar gambar lebih akurat.
        - Karena kamera memiliki distorsi bawaan, si algoritma inilah yang mencoba memperbaiki distorsi tersebut.
        - Cara kerja detailnya :
            - Algoritma akan membuat suatu pustaka dengan beberapa gambar dari berbagai sudut pandang dengan kamera yang sama.
            - Algoritma akan melakukan pattern recognition dari foto untuk mendapatkan bagaimana kamera melihat objek.

        - Setelah itu, algoritma akan menghitung parameter kamera seperti faktor distorsi, focal length, dll.
        - Kemudian, menggunakan parameter tersebut, algoritma akan melakukan koreksi atau perbaikan agar gambar lebih akurat.

2. **ADRC (Active Disturbance Rejection Control)**
    - ADRC merupakan metode kontrol untuk mengendalikan suatu sistem dan mengatasi gangguan dan kesalahan secara aktif. Bedanya dengan metode kontrol tradisional adalah, kalau yang tradisional ia perlu model yang sangat akurat dari suatu sistem, kalau ADRC ia akan bekerja tanpa butuh model sistem yang jelas.
    - ADRC akan memperlakukan semua ketidakpastian, gangguan eksternal, atau perubahan dalam suatu sistem sebagai 1 gangguan yang mencakup semuanya. Nanti ia akan menghilangkan pengaruh tersebut dan menolaknya secara aktif.
    - Komponen ADRC :
        - Extended State Observer (ESO)
            - Fungsinya untuk mengamati keadaan sistem dan mengestimasi probabilitas gangguan total.
            - ESO sebagai suatu CCTV atau mata untuk memantau tak hanya sistem tapi gangguan dari eksternal juga. Karena itu sistemnya jadi secara real-time.
        - Nonlinear State Error Feedback (NLSEF)
            - Fungsinya untuk menyesuaikan sinyal kontrol berdasarkan kesalahan kondisi aktual dan kondisi seharusnya.
            - Jadi nanti ketika ada kesalahan yang terjadi dan perbandingan jadi tak sesuai antara kondisi teoritis dan aktual, NLSEF akan menghasilkan suatu sinyal kontrol untuk memperbaiki kesalahan tersebut.
        - State Feedback Controller (SFC)
            - Untuk mengendalikan sistem dari data yang diperoleh dari ESO dan NLSEF.
            - Nanti SFC akan memberikan sinyal kontrol ayng tepat pada aktuator atau komponen kontrol lainnya untuk memastikan sistem stabil dan sesuai plan.
    - Cara kerja detailnya :
        - ESO akan mengobservasi dan melakukan estimasi keadaan sistem dan gangguan yang mungkin akan terjadi.
        - Berdasarkan hasil estimasi ESO, ADRC akan merespons dengan sinyal kontrol yang gunanya untuk reject dampak dari gangguan tersebut.
        - NLSEF dan SFC akan melakukan pengecekan kembali terhadap sistem agar tetap stabil dan sesuai dengan plan.

3. **PID (Proportional-Integral-Derivative) control algorithms**
    - PID merupakan suatu algoritma kontrol yang digunakan untuk menjaga output dari suatu sistem tetap stabil dan mencapai target value yang diinginkan

(setpoint) dengan mempertimabngkan Propotional (P), Integral (I), dan Derivative (D).

- Cara kerja PID adalah dengan melakukan penyesuaian sinyal kontrol dengan 3 faktor yang tadi dijelaskan dan memperhitungkan errornya (selisih nilai teoritis dan aktual).
- Tiga Faktor Utama PID :
    - Proportional (P)
        - Fungsinya untuk menghasilkan sinyal kontrol yang proporsional dengan error yang ada.
        - Jadi ketika error besar, maka sinyal kontrolnya juga besar. Fungsinya adalah membuat sistem kita lebih responsif untuk melakukan perbaikan terhadap errornya.
    - Integral (I)
        - Fungsinya untuk hilangkan sisa error atau steady-state-error dengan cara menjumlahkan errornya dari waktu ke waktu.
        - Gunanya Integral adalah mempertimbangkan total error yang terjadi dari waktu ke waktu. Ketika error tersebut terjadi terus, maka integral akan memperbaiki kontrol dan menghilangkan sisa error tersebut.
    - Derivative (D)
        - Fungsinya adalah memperkirakan dan menghitung kecepatan perubahan error.
        - Cara kerja derivative adalah dengan mengurangi lonjakan respon yang berlebihan atau overshoot. Jadi nanti ia akan membantu supaya sistem tidak melakukan respons berlebih ketika terjadi error tiba-tiba.
- Penulisan secara matematis

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

dengan

$$K_i = K_p \times \frac{1}{T_i} \text{ dan } K_d = K_p \times Td$$

Kp : koef Proporsional
Ki : koef Integral
Kd : koef derivatif
e(t) : selisih value dari teoritis dan value aktual saat t

- Cara kerja detailnya :

- PID akan menghitung error antara setpoint (nilai teoritis) dan output (nilai aktual).
- PID akan menghitung respons tadi berdasarkan komponen-komponenya.
- PID akan menggabungkan hasil dari komponen-komponennya untuk menghasilkan suatu sinyal kontrol yang akan dikirimkan ke sistem.
- Sistem akan merespons sinyal kontrol dan memperbaiki outputnya agar semakin mendekati setpoint.

4. A* (A star) algorithm

- A* algorithm adalah algoritma pencarian yang digunakan untuk menemukan rute terpendek dari titik awal ke titik tujuan melalui sebuah graf atau grid. Algortima ini kerap digunakan dalam pengembanagn AI/ML untuk masalah pencarian jalan seperti dalam video games atau robotika.
- Keuntungan A* algorithm dibandingkan algoritma pencarian yang lain adalah karena ia akan melakukan pencarian dengan metode heuristik, yakni metode untuk mempersempit pencarian (gausa cek satu-satu kaya brute-force).
- A star menggunakan 2 pendekatan utama :
  - Greedy Search : memilih langkah yang tampak paling dekat dengan tujuan.
  - Uniform Cost Search : memastikan ulang bahwa rute yang dipilih sejauh ini memiliki biaya yang minimum.
- Cara kerja algoritma :
  - Inisialisasi OPEN LIST
  - Letakkan simpul awal pada OPEN LIST
  - Inisialisasi CLOSE LIST
  - Ikuti langkah-langkah berikut sampai OPEN LIST tidak kosong :
    - Temukan simpul dengan f terkecil pada OPEN LIST dan beri nama "Q".
    - Hapus Q dari OPEN LIST.
    - Generate delapan turunan Q dan tetapkan Q sebagai induknya.
    - Untuk setiap keturunan :
      - Jika menemukan penerus adalah tujuannya, pencarian dihentikan
      - Jika tidak, hitung g dan h untuk penerusnya. penerus.g = q.g + jarak yang dihitung antara penerus dan q. suksesor.h = jarak terhitung antara suksesor dan tujuan. penerus.f = penerus.g ditambah penerus.h
      - Lewati penerus ini jika node dalam daftar OPEN dengan lokasi yang sama tetapi nilai f lebih rendah dari penggantinya.
      - Lewati penerusnya jika ada simpul dalam CLOSE LIST dengan posisi yang sama dengan penerusnya tetapi nilai

f lebih rendah; jika tidak, tambahkan simpul ke ujung OPEN LIST (untuk loop).

- Push Q ke dalam CLOSE LIST dan akhiri loop sementara.