

Buatlah sebuah Blok Diagram pada draw.io dari 2 jenis sistem kontrol dan detail penjelasan skemanya!

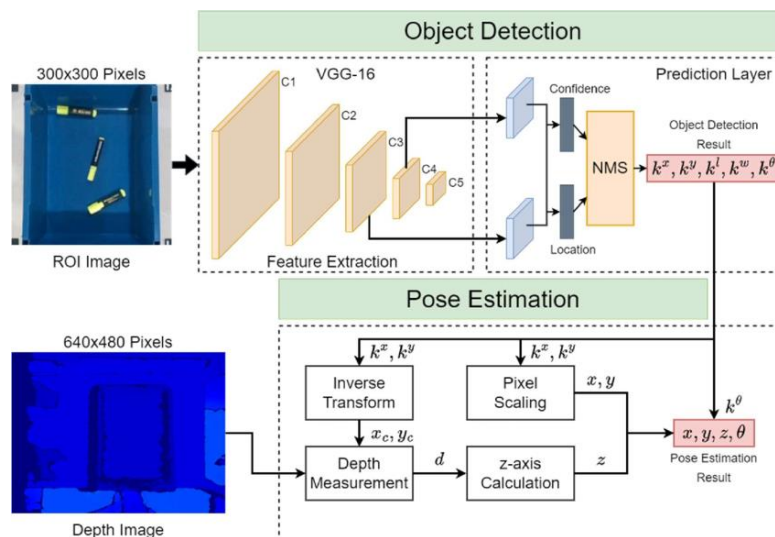
Nama: Dietrich Pepalem tarigan

NIM: 10223037

Kinematics (Object Detection, Pose Estimation, Camera Calibration)

Object Detection dan Pose Estimation

Pose Estimation bertujuan untuk menentukan posisi dan orientasi objek dalam ruang tiga dimensi (6 derajat kebebasan - 6DoF) relatif terhadap kamera. Estimasi ini melibatkan pendeteksian fitur objek (misalnya, tepi, sudut, atau pola) pada gambar, mengekstraksi data terkait, dan memetakan fitur tersebut ke dalam sistem koordinat 3D.



Salah satu penerapan *Pose Estimation* adalah dalam navigasi robot, augmented reality, dan sistem otomatisasi industri. Beberapa algoritma modern menggunakan *Convolutional Neural Networks* (CNN) untuk meningkatkan akurasi deteksi objek dengan memanfaatkan data gambar RGB-D (kombinasi gambar dan kedalaman).

Camera Calibration:

Camera Calibration adalah proses untuk menentukan parameter internal kamera (misalnya, panjang fokus, distorsi lensa) dan parameter eksternal (posisi dan orientasi kamera relatif terhadap objek dunia nyata). Kalibrasi ini penting untuk memastikan akurasi dalam deteksi objek dan estimasi pose.

Metode umum untuk kalibrasi melibatkan penggunaan pola objek kalibrasi yang diketahui, seperti pola papan catur, yang digunakan untuk menemukan korespondensi antara titik 2D di gambar dan titik 3D di dunia nyata.

ADRC (Active Disturbance Rejection Control)

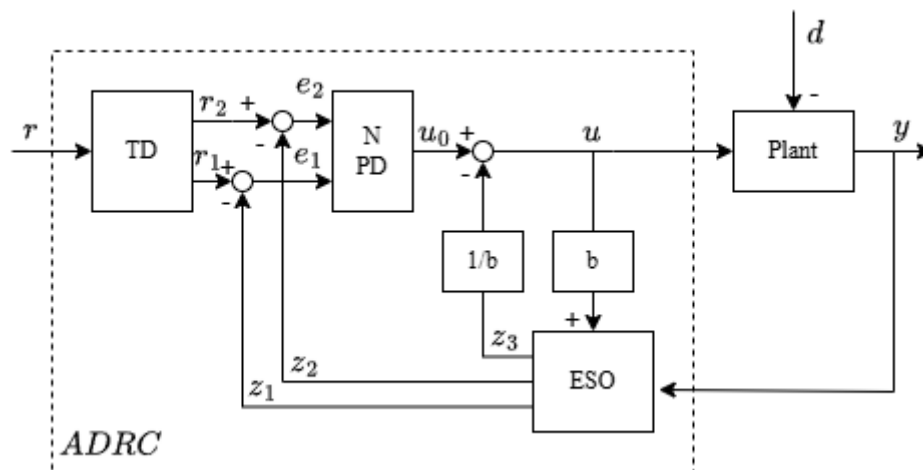


Figure 1. Skema ADRC

ADRC (Active Disturbance Rejection Control) adalah metode kontrol yang efektif untuk menangani gangguan eksternal dan ketidakpastian dalam sistem. Pada gambar yang saya buat, terdapat beberapa elemen utama dalam ADRC yang berfungsi untuk mengestimasi gangguan secara real-time dan mengkompensasi ketidakpastian sistem.

Tracking Differentiator (TD): TD bertugas menghaluskan sinyal masukan referensi r menjadi dua komponen r_1 (output halus dari referensi) dan r_2 (turunan dari r_1). Fungsi ini berguna untuk mengurangi lonjakan yang tiba-tiba pada sinyal referensi, yang bisa menyebabkan osilasi atau kesalahan besar pada sistem kontrol.

Error Signals (e_1, e_2): e_1 adalah selisih antara sinyal r_1 dan sinyal keluaran dari estimator z_1 , yang mewakili output sistem. Sedangkan e_2 adalah selisih antara turunan dari referensi r_2 dengan

estimasi derivatif output sistem z_2 . Kedua sinyal ini digunakan oleh kontroler untuk menentukan besarnya koreksi yang perlu diterapkan.

Nonlinear PD (NPD) Controller: Bagian ini menggantikan Proportional-Derivative (PD) klasik dengan versi nonlinier untuk meningkatkan respons terhadap perubahan yang tidak diinginkan. PD ini memberikan sinyal kontrol awal u_0 berdasarkan kombinasi kesalahan e_1 dan e_2 .

Extended State Observer (ESO): ESO berfungsi sebagai inti dari ADRC. ESO mengestimasi keadaan sistem, termasuk keadaan tersembunyi atau gangguan yang tidak terukur. Tiga variabel output ESO, yaitu z_1 (estimasi output sistem), z_2 (estimasi kecepatan output), dan z_3 (estimasi gangguan), digunakan untuk memperbaiki sinyal kontrol. Dengan ESO, ADRC dapat memperkirakan gangguan dalam sistem dan secara langsung mengkompensasinya, sehingga meningkatkan kinerja sistem secara keseluruhan

Plant: Ini adalah sistem yang ingin dikontrol. Output plant y dipengaruhi oleh sinyal kontrol u yang telah diperbaiki oleh ADRC serta gangguan eksternal d . Tujuan ADRC adalah untuk memastikan bahwa output y mengikuti referensi r meskipun ada gangguan d .

PID (Proportional-Integral-Derivative) Controller

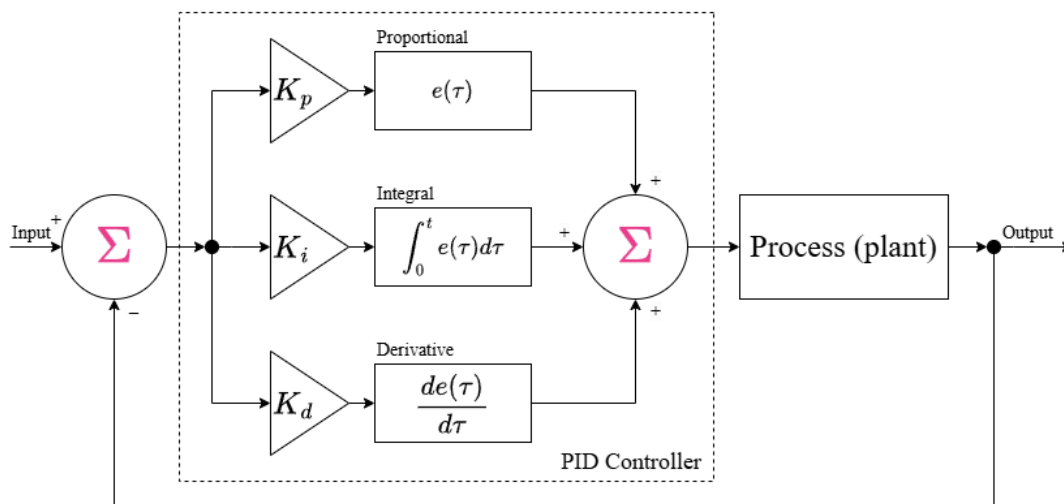


Diagram pengontrol PID yang saya buat diatas mengilustrasikan sistem kontrol umpan balik yang banyak digunakan dalam sistem otomasi dan kontrol industri. PID adalah singkatan dari Proportional, Integral, dan Derivative, mewakili tiga komponen yang menyesuaikan output pengontrol untuk meminimalkan kesalahan antara setpoint yang diinginkan (input) dan output aktual dari proses (plant).

Sinyal Kesalahan (Error Signal): Di awal skema, sinyal input dibandingkan dengan sinyal keluaran sistem (output). Hasil dari perbandingan ini menghasilkan sinyal kesalahan $e(\tau)$, yang merupakan selisih antara input (yang diinginkan) dan output (yang aktual). Sinyal kesalahan ini adalah dasar dari

kontrol PID, karena akan digunakan untuk mengatur output agar sesuai dengan input yang diinginkan.

Kontrol Proportional (P): Bagian proportional memberikan respons yang sebanding dengan besarnya kesalahan. Penguat proportional K_p akan menghasilkan sinyal kontrol yang besarnya sebanding dengan nilai kesalahan. Jika kesalahan besar, kontribusi proportional juga besar. Namun, P-controller sendirian tidak bisa menghilangkan kesalahan steady-state secara keseluruhan

Kontrol Integral (I): Kontrol integral menambahkan aksi yang sebanding dengan integral dari kesalahan selama waktu. Artinya, kontribusi integral bergantung pada akumulasi kesalahan dari waktu ke waktu. Ini membantu menghilangkan kesalahan steady-state yang mungkin tidak bisa dihilangkan oleh kontrol proportional saja. Penguatan integral K_i menentukan seberapa cepat kontrol integral bereaksi terhadap kesalahan yang terakumulasi.

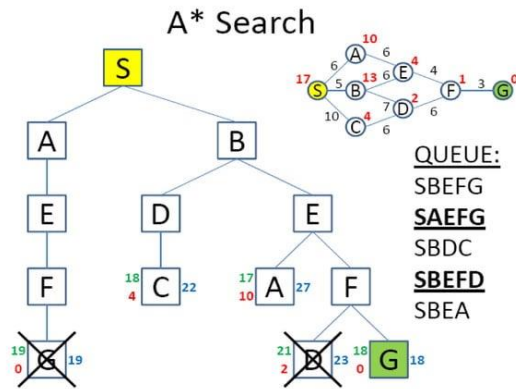
Kontrol Derivatif (D): Kontrol derivatif memberikan sinyal kontrol berdasarkan kecepatan perubahan kesalahan. Bagian ini bertindak untuk memperkirakan perubahan yang akan datang dari kesalahan, sehingga memberikan respons yang lebih halus dan mengurangi osilasi yang berlebihan. Penguatan derivatif K_d menentukan kontribusi dari kecepatan perubahan kesalahan terhadap sinyal kontrol.

Penjumlahan Sinyal (Σ): Setelah setiap komponen (P, I, D) menghasilkan sinyal kontrolnya masing-masing, mereka dijumlahkan untuk menghasilkan sinyal kontrol gabungan yang kemudian digunakan untuk mengontrol proses atau plant. Dengan cara ini, PID menggabungkan ketiga aksi kontrol untuk memperbaiki kesalahan secara efisien, baik dalam waktu nyata maupun dalam jangka waktu yang lebih lama.

Proses (Plant): Plant merupakan sistem yang dikendalikan oleh PID. Output plant diukur dan dibandingkan dengan input yang diinginkan. PID berusaha untuk mengatur plant agar output-nya mengikuti input set point secara akurat meskipun ada gangguan atau perubahan dalam dinamika plant.

Algoritma A* (A-star)

Algoritma A* adalah algoritma pencarian *shortest path* yang sangat populer dan efisien. Algoritma ini menggunakan pendekatan *heuristic* untuk mencari jalur yang paling menjanjikan berdasarkan kombinasi antara biaya yang telah diketahui dan estimasi biaya ke tujuan.



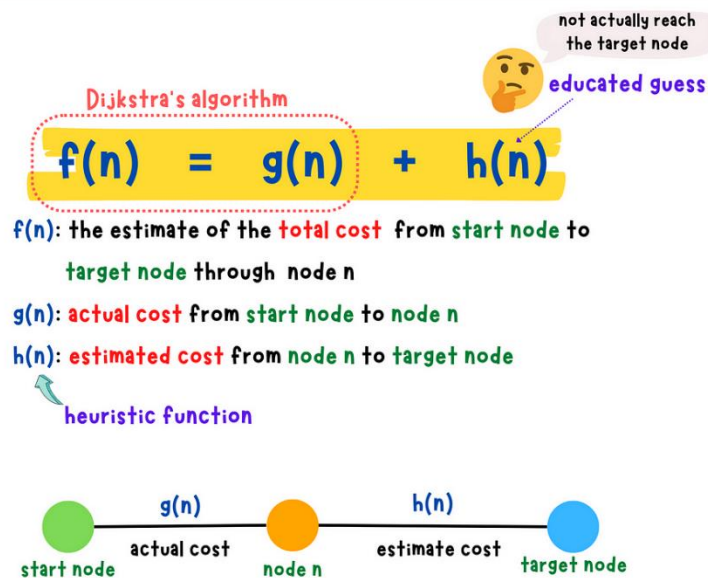
A* menggunakan dua komponen penting untuk mengevaluasi setiap node:

$g(n)$: Biaya aktual dari titik awal ke node saat ini.

$h(n)$: Estimasi biaya dari node saat ini ke tujuan, juga dikenal sebagai *heuristic*.

Fungsi total biaya adalah **$f(n) = g(n) + h(n)$** . A* memprioritaskan node dengan nilai $f(n)$ terkecil, yang menunjukkan jalur yang paling efisien. Heuristik yang dipilih harus *admissible*, artinya estimasi biaya tidak boleh melebihi biaya sebenarnya, untuk menjamin bahwa A* menemukan jalur terpendek.

A* Search Algorithm



Heuristik dan Performa: Algoritma ini bekerja paling baik ketika heuristik yang digunakan akurat. Misalnya, dalam grid 2D, jarak Euclidean atau Manhattan sering digunakan tergantung pada apakah pergerakan diagonal diizinkan. Jika heuristik yang digunakan *admissible* dan konsisten (monoton), A* akan memberikan hasil optimal dan efisien

A* banyak digunakan dalam navigasi robot untuk mencari jalur optimal dalam lingkungan yang tidak sepenuhnya dikenal. Algoritma ini juga sangat penting dalam pengembangan kecerdasan buatan (AI) untuk pergerakan karakter dalam gim serta dalam protokol routing dinamis pada jaringan komputer.

Refrensi:

1. Eko Rudiawan Jamzuri, Riska Analia, & Susanto, S. (2023). Object Detection and Pose Estimation with RGB-D Camera for Supporting Robotic Bin-Picking. *Elkomika*, 11(1), 128–128.
<https://doi.org/10.26760/elkomika.v11i1.128>
2. [Active Disturbance Rejection Control - Design controller for plants with unknown dynamics and disturbances - Simulink \(mathworks.com\)](#)
3. seri, C. (2020, December). *Memahami PID Controller (seri PID Controller part1)*. YouTube.
<https://youtu.be/aaMA-v509QQ?si=SpyL-sCdMNMvVJcN>
4. [A* Search Algorithm. A* search algorithm is a path finding... | by Claire Lee | Medium](#)