

## 1. PID Control (Proportional-Integral-Derivative)

Penjelasan PID:

PID adalah algoritma kontrol berbasis umpan balik yang sangat populer dan banyak digunakan di berbagai sistem kontrol, seperti robotika, otomasi industri, dan mesin-mesin canggih. Tujuan utama dari PID adalah untuk meminimalkan kesalahan antara setpoint (nilai yang diinginkan) dan output (nilai yang dicapai oleh sistem).

PID bekerja dengan tiga parameter:

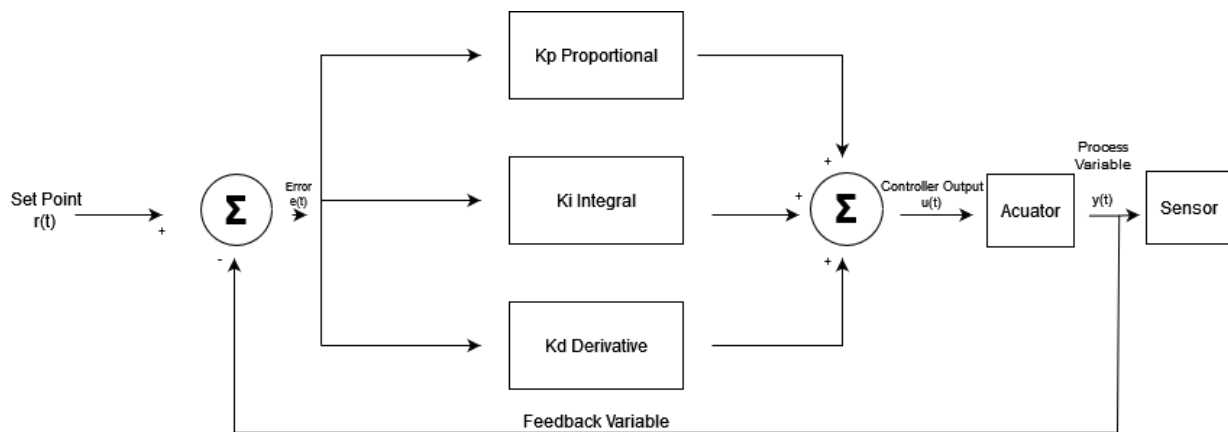
- Proportional (P): Merespons proporsional terhadap error saat ini. Semakin besar error, semakin besar sinyal kontrol yang dikirim. Namun, jika hanya menggunakan P, sistem mungkin tidak mencapai nilai setpoint dan bisa menyebabkan steady-state error.
- Integral (I): Mengintegrasikan error dari waktu ke waktu untuk menghilangkan steady-state error. Jika error kecil tetapi berlangsung lama, elemen integral akan terus menambah sinyal kontrol untuk mengoreksi kesalahan tersebut.
- Derivative (D): Berfungsi untuk mencegah perubahan tiba-tiba dalam sistem dengan merespons tingkat perubahan error. Bagian ini membantu mengurangi osilasi dan membuat sistem lebih stabil.

Blok Diagram PID:

Diagram dasar dari sistem kontrol PID melibatkan beberapa elemen utama:

1. Setpoint: Nilai yang diinginkan dari sistem.
2. Error: Selisih antara setpoint dan output sistem.
3. PID Controller: Mengambil error sebagai input dan menghitung sinyal kontrol berdasarkan parameter P, I, dan D.
4. Actuator: Komponen yang menerima sinyal dari kontroler dan menggerakkan sistem fisik (misalnya, motor atau robot).
5. Process (Plant): Sistem fisik yang dikendalikan.
6. Feedback: Sensor mengukur output aktual dan mengirimkannya kembali ke kontroler untuk perhitungan error baru.

Blok Diagram PID:



Keuntungan PID:

- Mudah diimplementasikan.
- Cocok untuk berbagai jenis sistem (linier dan non-linier).
- Dapat mencapai kontrol yang cukup baik jika parameternya disetel dengan benar.

Kekurangan PID:

- Perlu tuning yang tepat untuk mendapatkan kinerja optimal.
- Sulit menangani sistem dengan gangguan yang sering berubah-ubah.
- Tidak terlalu efektif dalam menangani ketidakpastian dan non-linieritas yang besar pada sistem.

## 2. ADRC (Active Disturbance Rejection Control)

Penjelasan ADRC:

ADRC adalah pendekatan kontrol yang lebih canggih dibandingkan PID. Tujuannya adalah untuk menolak gangguan aktif dalam sistem. Alih-alih bergantung pada model presisi sistem, ADRC menggunakan teknik pengamatan untuk mengestimasi gangguan secara real-time dan melakukan kontrol adaptif untuk mengkompensasi gangguan tersebut. Ini membuat ADRC lebih cocok untuk sistem non-linier atau sistem dengan ketidakpastian tinggi.

ADRC terdiri dari tiga bagian utama:

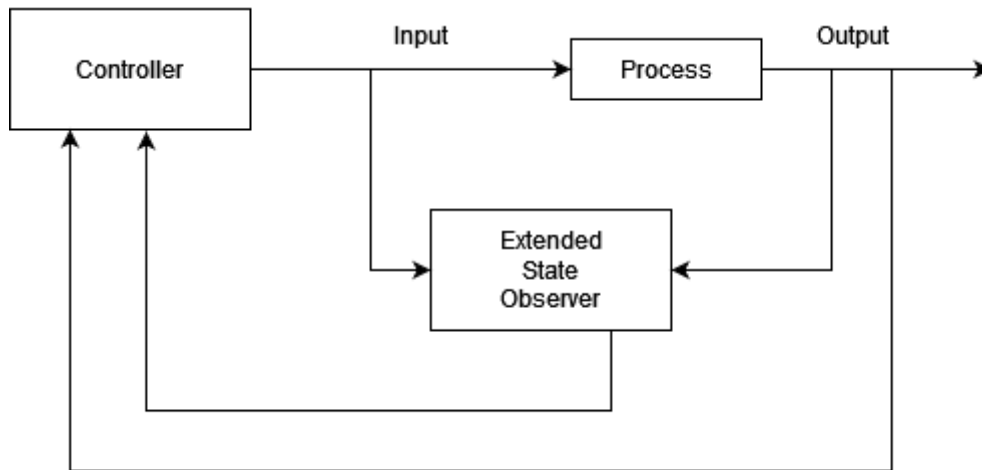
1. Tracking Differentiator (TD): Membantu menyaring noise dari sinyal input dan memperkirakan laju perubahan.
2. Nonlinear State Observer (ESO): Memperkirakan status sistem, termasuk gangguan eksternal dan dinamis sistem.
3. Control Law: Mengambil estimasi dari ESO dan menghasilkan sinyal kontrol untuk mengoreksi gangguan serta memandu sistem ke nilai yang diinginkan.

Blok Diagram ADRC:

Diagram dasar dari sistem kontrol ADRC melibatkan beberapa elemen:

1. Setpoint: Nilai yang diinginkan untuk sistem.
2. Tracking Differentiator: Menghasilkan sinyal referensi yang halus.
3. ESO (Extended State Observer): Memperkirakan gangguan pada sistem.
4. Control Law: Menghitung sinyal kontrol berdasarkan estimasi gangguan.
5. Actuator: Menjalankan sinyal kontrol untuk menggerakkan sistem fisik.
6. Process (Plant): Sistem fisik yang dikendalikan.
7. Feedback: Sensor yang mengukur output dan memberikannya ke ESO untuk memperbarui estimasi gangguan.

Blok Diagram ADRC:



Keuntungan ADRC:

- Lebih baik dalam menangani ketidakpastian sistem.
- Memiliki kemampuan adaptif yang baik terhadap gangguan eksternal.
- Tidak terlalu bergantung pada model matematis sistem, membuatnya fleksibel.

Kekurangan ADRC:

- Implementasi lebih kompleks dibandingkan PID.
- Memerlukan tuning yang lebih sulit.
- Membutuhkan estimasi status yang akurat untuk kinerja yang baik.

**Kesimpulan:**

- PID Control: Lebih sederhana dan cocok untuk sistem linier dengan sedikit gangguan atau perubahan lingkungan. Cocok untuk banyak aplikasi industri dan robotika sederhana.
- ADRC: Lebih kompleks tetapi jauh lebih kuat dalam menghadapi ketidakpastian dan gangguan yang tidak terduga. Cocok untuk sistem non-linier atau aplikasi robotik canggih yang beroperasi di lingkungan dinamis.