# PERANCANGAN KENDALI PID DENGAN MATLAB

Sri Sukamta

**ABSTRAK**

Perancangan PID Controller selama ini menggunakan metoda trial and error dengan perhitungan yang memakan waktu lama. MatLab yang dilengkapi Control Toolbox, membantu perancang untuk melihat respon berbagai kombinasi konstanta dengan variasi input yang berbeda. Penggunaan MatLab ini sangat membantu perancang dalam menentukan kombinasi di antara P, I, dan D Controller untuk menghasilkan sistem pengaturan yang baik dan sederhana.

Kata kunci : Perancangan, Kendali PID

## A. Pendahuluan

Kontrol otomatis saat ini memberikan peran yang sangat besar dalam kehidupan manusia, terutama dalam bidang ilmu pengetahuan dan industri. Beberapa di antaranya adalah robot, pengaturan suhu sebuah tangki, pengaturan kelembaban udara dalam sebuah ruangan, tekanan udara dalam suatu pipa tertutup yang dijaga tetap, dll. Penguasaan sistem kontrol baik dalam hal teori maupun praktek akan menghasilkan suatu sistem dengan respon yang sesuai dengan kebutuhan. Sebuah sistem kontrol yang dirancang, perlu dianalisa terlebih dahulu untuk mendapatkan gambaran respon sistemnya. Gambaran tersebut meliputi :

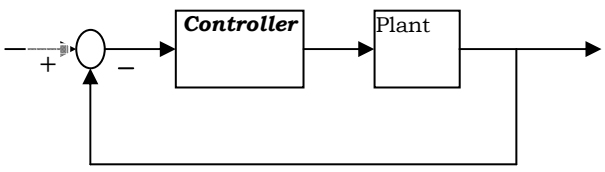
1. Respon sistem terhadap berbagai macam input (step function, ramp function, dan impulse function, dll), termasuk jika adanya gangguan dari luar.
2. Kestabilan sistem (metode : root locus, frekuensi respon, state space).
3. Respon sistem terhadap berbagai macam jenis kontroler (P, I, D, dan/atau kombinasinya).

Namun demikian, ada paling tidak 2 kendala dalam menganalisa sistem yang akan dibuat, yaitu bagaimana mendapatkan transfer function dari sistem tersebut kemudian menganalisanya secara cepat dan mudah.

Sampai saat ini, para perancang sistem kontrol masih mendapatkan kesulitan untuk mendapatkan transfer function suatu sistem. Namun jika persamaan tersebut sudah didapatkan, maka proses analisanya tidak terlalu sulit. Sebab saat ini sudah dikembangkan software-software yang akan membantu proses di atas. Salah satu software tersebut adalah MatLab dari Mathworks, Inc.

## B. PID Controller

PID Controller merupakan salah satu jenis pengatur yang banyak digunakan. Selain itu sistem ini mudah digabungkan dengan metoda pengaturan yang lain seperti Fuzzy dan Robust. Sehingga akan menjadi suatu sistem pengatur yang semakin baik Tulisan ini dibatasi pada sistem dengan Unity Feedback System, yang gambarnya sebagai berikut gambar 1:




**Gambar 1**
Blok diagram untuk *Unity Feedback Systems* [1]

PID Controller memiliki transfer function sebagai sebagai berikut :

$$H(s) = \frac{K_D s^2 + K_p s + K_I}{s^3 + K_D s^2 + K_p s + K_I} \quad \text{.....(1)}$$

PID Controller sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu P (Proportional) Controller, D (Derivative) Controller, dan I (Integral) Controller. Masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang disebut sebagai konstanta. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel 1di bawah ini :

Tabel 1
Respon PID Controller Terhadap Perubahan Konstanta[1]

| Closed-Loop Response | Rise Time | Overshoot | Settling Time | SS Error |
|---|---|---|---|---|
| Kp | Decrease | Increase | Small change | Decrease |
| Ki | Decrease | Increase | Increase | Eliminate |
| Kd | Small change | Decrease | Decrease | Small change |

Parameter-parameter tersebut, tidak bersifat independen, sehingga pada saat salah satu nilai konstantanya diubah, maka mungkin sistem tidak akan bereaksi seperti yang diinginkan. Tabel di atas hanya dipergunakan sebagai pedoman jika akan melakukan perubahan konstanta. Untuk merancang suatu PID Controller, biasanya dipergunakan metoda trial & error. Sehingga perancang harus mencoba kombinasi pengatur beserta konstantanya untuk mendapatkan hasil terbaik yang paling sederhana.

**C. Metode Konvensional**

Desain sebuah sistem kontrol, dimulai dengan membuat blok diagram sistem. Blok diagram (yang berisi transfer function) tersebut selanjutnya akan dianalisa dengan menggunakan aksi pengontrolan yang berbeda. Dengan perubahan sinyal input sehingga perancang dapat melihat respon sistem jika mendapat input sinyal tertentu. Kombinasi antara sinyal input dan jenis aksi pengontrolan ini akan menghasilkan respon yang berbeda-beda.

Dahulu untuk melihat respon suatu sistem dengan berbagai macam kombinasi sinyal input dan aksi pengontrolan merupakan hal yang sulit dan membosankan. Adapun prosedur yang harus dilalui adalah sebagai berikut [2]:

1. Mendapatkan transfer function sistem (dalam s-domain) dengan Laplace Transform.
2. Menentukan jenis aksi pengontrolan beserta dengan konstantanya.
3. Menggabungkan transfer function yang sudah didapatkan dengan jenis aksi pengontrolan.
4. Menentukan sinyal input yang akan dimasukkan (biasanya fungsi step, fungsi ramp dan pulse) dan menggabungannya ke dalam transfer function yang baru.
5. Melakukan perhitungan invers Laplace Transform untuk mendapatkan fungsi dalam t-domain.
6. Menggambar respon berdasarkan fungsi dalam t-domain.

Untuk melakukan langkah-langkah di atas diperlukan ketelitian yang tinggi dan hasil penggambarannya sering kali kurang (tidak) akurat. Selain itu, jika perancang ingin mengamati respon sistem terhadap sinyal input yang lain, maka proses-proses tersebut sebagian besar akan diulang kembali. Hal ini bertambah kompleks jika perubahan yang dilakukan tidak

terbatas pada sinyal input, tetapi juga pada jenis aksi pengontrolannya. Sehingga untuk mendapatkan respon dari berbagai macam kombinasi, membutuhkan waktu yang relatif lama. Selain itu, perancang juga melakukan proses perhitungan yang rumit dan membosankan.

**D. Metode Simulasi Menggunakan Komputer**

Perkembangan teori kontrol juga diikuti oleh software pendukungnya. Mulai dari software untuk pemrograman sistem, sampai dengan software untuk proses simulasinya. Salah satu software yang dapat dipergunakan untuk simulasi tersebut adalah MatLab dari Mathworks, Inc.

Software ini dilengkapi dengan berbagai toolbox yang memudahkan pemakai untuk melakukan perhitungan-perhitungan tertentu. Bahkan saat ini sudah dikembangkan toolbox khusus untuk simulasi yang diberi nama Simulink. [3]

Aplikasi MatLab dalam bidang pengaturan dilengkapi Control Toolbox. Toolbox ini sudah dilengkapi dengan berbagai macam fungsi pendukung yang dipergunakan dalam analisa sistem kontrol. Beberapa fungsi pendukung yang sering dipergunakan untuk menganalisa suatu sistem adalah : feedback, step, rlocus, series, dll. Untuk menganalisa suatu sistem, software hanya memerlukan masukan berupa transfer function yang ditulis dalam Laplace Transform (dalam s-domain) atau matriks. Untuk selanjutnya, pemakai tinggal memilih analisa yang akan dipergunakan. Tulisan ini akan membahas penggunaannya secara khusus untuk merancang PID Controller pada suatu sistem.

Sebagai contoh, suatu sistem kontrol memiliki transfer function sebagai berikut :

$$H(s) = \frac{1}{s^2 + 10s + 20} \quad \text{.....(2)}$$

Dengan kriteria perancangan sebagai berikut :

1. memiliki rise time yang cepat
2. overshoot sekecil mungkin
3. tidak memiliki steady state error.

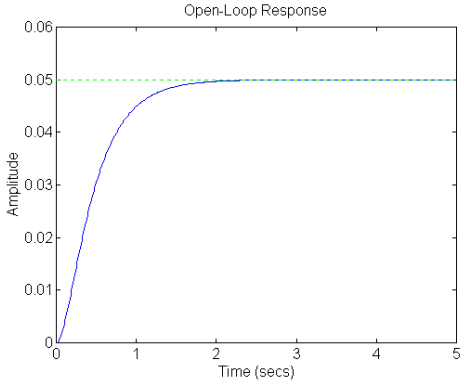
Dari fungsi di atas, maka parameter-parameter yang dimasukkan berupa koefisien pembilang dan penyebutnya. Biasanya dipergunakan variabel num untuk pembilang dan den untuk penyebut. Kedua nama variabel tersebut tidak mutlak, jadi penggunaan nama variabel yang lain juga diperbolehkan. Setelah itu komputer sudah siap untuk menganalisa sistem kontrol.

Langkah kedua yang perlu dilakukan adalah memilih jenis input yang akan dimasukkan ke dalam sistem. Input ini bisa berupa step, pulse, ramp, sinus, dan sebagainya. Sebagai dasar analisa akan diperlunakan fungsi step. Sedang penggunaan jenis input yang lain akan dibicarakan pada bagian akhir tulisan ini.

Fungsi dasar yang akan sering dipergunakan adalah step, dengan syntax : step(num,den,t) untuk s-domain atau step(A,B,C,D) untuk state space. Fungsi ini menghasilkan gambar respon sistem bila diberi input unit step dalam t-domain.

```
num = [1];
den = [1 10 20];
step(num,den)
title('Open Loop Response')
```

Respon sistem terbuka (open loop response) dapat dilihat pada gambar di bawah ini :




Gambar 2

Respon Sistem Dengan Unit Step Input

Sistem di atas memiliki steady state error yang tinggi, yaitu 0,95. Sebab respon tertinggi hanya didapatkan pada amplitudo 0,05. Selain itu, sistem tersebut memiliki rise time yang cukup besar (sekitar 1,5 detik). Hal tersebut jelas tidak menguntungkan.

Untuk menghasilkan sistem kontrol yang baik, diperlukan sistem yang tertutup (close loop system). Sistem ini memiliki feedback, yang akan membandingkan kondisi sesungguhnya dengan seting poin yang diberikan

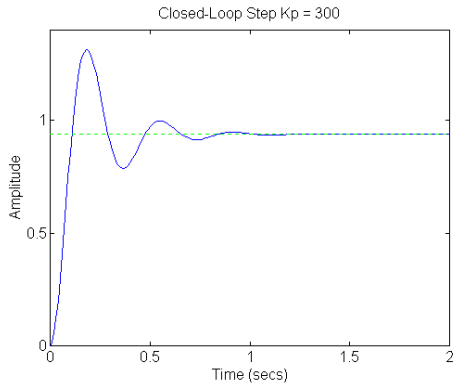
**E. Proportional Controller**

Dari tabel 1 diketahui bahwa P Controller dapat mengurangi rise time, menambah overshoot, dan mengurangi steady state error. Closed-loop transfer function sistem di atas dengan menggunakan P Controller adalah sebagai berikut :

$$H(s) = \frac{K_p}{s^2 + 10s + (20 + K_p)} \quad \text{..... (3)}$$

Misal, diambil konstanta Kp = 300, maka :

```
Kp = 300;
num = [Kp];
den = [1 10 20+Kp];
t = 0 : 0.01 : 2;
step(num,den)
title('Closed-Loop Step Kp = 300')
```




Gambar 3
Respon Sistem Tertutup Menggunakan P Controller

Penambahan variabel t=0:0.01:2 dimaksudkan untuk melihat respon sistem dari t=0 s/d t=2, dengan step 0,01 detik.

Dari gambar 2 di atas, dapat dilihat bahwa penambahan P Controller mengurangi rise time dan steady state error, tetapi menambah overshoot. Namun, overshoot yang terjadi masih terlalu besar. Jika konstanta Kp diperbesar, maka overshoot yang terjadi juga semakin besar, settling time juga semakin besar, tetapi rise timenya menjadi kecil. Kebalikan dari keadaan itu terjadi jika konstanta Kp diperkecil.

**F. Proportional-Derivative Controller**

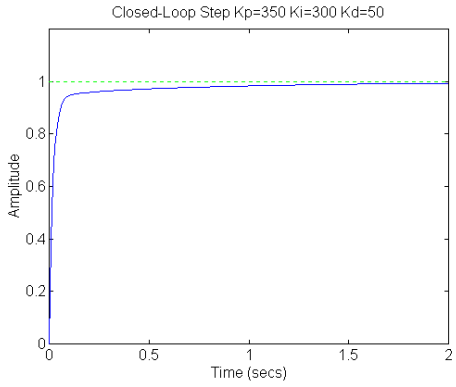
Closed-Loop transfer function sistem di atas dengan PD Controller adalah :

$$H(s) = \frac{K_D s + K_p}{s^2 + (10 + K_D)s + (20 + K_p)} \quad (4)$$

Misal, Kp = 300 dan Kd = 10, maka :

```
Kp = 300;
Kd = 10;
num = [Kd Kp];
den = [1 10+Kd 20+Kp];
t = 0 : 0.01 : 2;
step(num,den)
title('Closed-Loop Step Kp=300 Kd=10')
```

Respon sistem tergambar seperti di bawah ini.




Gambar 4
Respon Sistem Tertutup Menggunakan PD Controller

Berdasarkan gambar di atas, penggunaan PD Controller dapat mengurangi overshoot dan settling time, tetapi tidak memberikan dampat apa pun terhadap steady state error.

**G. Proportional-Integral Controller**

Closed-Loop transfer function sistem di atas dengan PD Controller adalah :

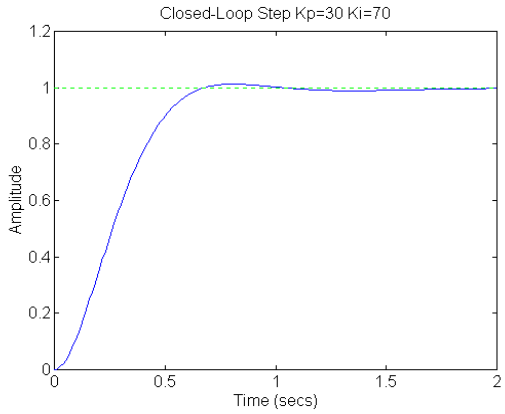
$$H(s) = \frac{K_p s + K_I}{s^3 + 10s^2 + (20 + K_p) + K_I} \quad .(5)$$

Integral Controller memiliki karakteristik menguraangi rise time, menambah overshoot dan setling time, serta menghilangkan steady state error (karakteristik ini tidak dimiliki oleh jenis yang lain).

Misal, Kp = 30 dan Ki = 70, maka :

```
Kp = 300;
Ki = 70;
num = [Kp Ki];
den = [1 10 20+Kp Ki];
t = 0 : 0.01 : 2;
step(num,den)
title('Closed-Loop Step Kp=30 Ki=70')
```

P dan I Controller memiliki karakteristik yang sama dalam hal rise time dan overshoot. Oleh karena itu, nilai Kp harus dikurangi untuk menghindari overshoot yang berlebihan. Nilai Ki diambil lebih besar dari Kp, karena diinginkan untuk meniadakan steady state error. Jika Kp > Ki, maka steady state errornya tidak dapat dihilangkan. Gambar di bawah ini memperlihatkan respon sistem dengan PI Controller




Gambar 5
Respon Sistem Tertutup Menggunakan PI Controller

Dari gambar di atas terlihat bahwa rise time sistem menurun, dengan overshoot yang kecil, serta steady state errornya dapat dihilangkan.

**H. Proportional-Integral-Derivative Controller**

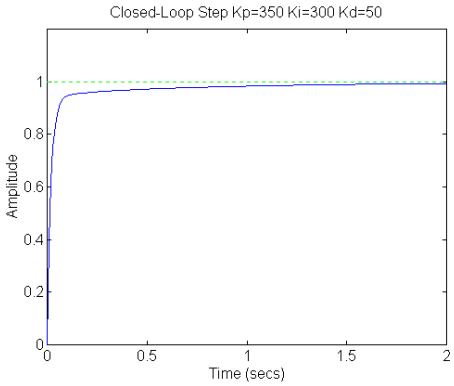
Bagian akhir dari simulasi ini adalah PID Controller, yang memiliki transfer function untuk sistem di atas adalah :

$$\frac{K_D s^2 + K_p s + K_I}{s^3 + (10 + K_D)s^2 + (20 + K_p)s + K_I} \quad ...(6)$$

Misal :

```
Kp = 350;
Ki = 300;
Kd = 50;
num = [Kp Ki Kd];
den = [1 10+kd 20+Kp Ki];
t = 0 : 0.01 : 2;
step(num,den)
title('Closed-Loop Step Kp=350 Ki=300 Kd=50')
```

Respon sistem tergambar di bawah ini




Gambar 6
Respon Sistem Tertutup Menggunakan PID Controller

Dari gambar di atas terlihat bahwa kriteria sistem yang diinginkan sudah terpenuhi, yaitu tidak memiliki overshoot, rise time yang cepat, dan tidak memiliki steady state error.

Nilai-nilai konstanta yang terdapat pada tulisan ini diperoleh dari percobaan (trial and error). Sehingga perancang yang berbeda akan mendapatkan nilai yang berlainan untuk memenuhi kriteria di atas. Hal itu terjadi karena perubahan pada salah satu konstanta akan berpengaruh pada konstanta yang lain. Artinya tidak akan didapatkan hasil sesuai dengan tabel 1. Tabel tersebut hanya dipergunakan sebagai pedoman.

Berikut ini beberapa tips yang dapat dipergunakan untuk mendapatkan respon yang diinginkan

1. Dapatkan respon sistem terbuka (open-loop) untuk menentukan bagian mana yang harus diperbaiki (rise time, settling time, overshoot, steady state error).
2. Tambahkan P Controller untuk memperbaiki rise time.
3. Tambahkan D Controller untuk memperbaiki overshoot.
4. Tambahkan I Controller untuk menghilangkan steady state error.
5. Kombinasikan konstanta yang ada untuk mendapatkan respon yang diinginkan.

Dalam mengimplementasikan sistem kontrol, sebenarnya tidak perlu menggunakan PID Controller. Jika sistem sudah memberikan responyang cukup baik hanya dengan PI Controller, maka tidak perlu menambahkan D Controller ke dalamnya. Sehingga sistem menjadi lebih sederhana (kombinasi yang main banyak membuat sistem menjadi makin kompleks)

Analisa pada contoh di atas, dilakukan dengan input unit step. Apabila diinginkan analisa dengan input yang berbeda, maka harus dilakukan modifikasi transfer function.

Untuk menganalisa sistem dengan input impulse function, maka transfer function dikalikan dengan faktor s. Demikian juga untuk input ramp function, perlu dikalikan dengan faktor 1/s.

## I. Kesimpulan

Merancang suatu PID Controller memang tidak mudah, sebab penentuan konstantanya dilakukan secara trial and error. Selain itu perubahan salah satu konstanta akan berpengaruh terhadap parameter yang lain. Namun demikian, kesulitan tersebut sudah sedikit teratasi dengan bantuan komputer. Sebab respon dapat langsung dilihat tanpa melakukan perhitungan yang rumit dan memakan waktu panjang.

Penggunaan MatLab tidak terbatas pada perancangan PID Controller saja, melainkan ada berbagai macam analisa sistem kontrol seperti Bode Diagram, analisa kestabilan Nyquist, Root Locus, State Space, dll. Fungsi-fungsi untuk keperluan di atas sudah tersedia di dalam Control Toolbox. Jika tidak tersedia, pemakai dapat membuat m-file yang sesuai dengan kebutuhan.

**DAFTAR PUSTAKA**

MatLab High Performance Numeric Computation and Visualization Software. The Mathworks, Inc. 1992

Messner, William and Dawn Tilbury. Control Tutorials for MatLab and Simulink. A Web Based Approach. Addisson Wesley, Inc. 1999.

Ogata, Katsuhiko. Modern Control Engineering. 3rd ed. Prentice Hall International. 1997

Ogata, Katsuhiko. Solving Control Engineering Problems with MatLab. Englewood Cliffs, New Jersey : Prentice Hall Inc. 1994

**BIOGRAFI**

Drs.Sri Sukamta, M.Si , Dosen Teknik Elektro UNNES. Lahir di Klaten tahun 1965. Lulus sarjana tahun 1989 di FPTK IKIP Semarang pada jurusan Pendidikan Teknik Elektro. Meminati bidang kendali dalam STL
Email : srisukamtote@yahoo.com