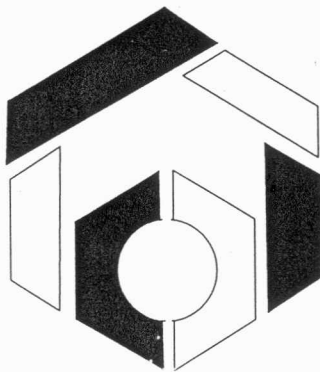


09/0014

**PEMBUATAN PERANGKAT LUNAK MENGGUNAKAN
MATLAB/SIMULINK UNTUK PENGENDALIAN
KECEPATAN MOTOR DC DENGAN METODE
CASCADE**

Hasil Penelitian / Pemikiran yang tidak dipublikasikan
Disusun sebagai salah satu syarat untuk
Kenaikan Angka Kredit Jabatan Fungsional Lektor

Oleh
Pipit Anggraeni
197908242005012001



JURUSAN TEKNIK OTOMASI MANUFAKTUR
DAN MEKATRONIKA
POLITEKNIK MANUFAKTUR NEGERI BANDUNG BANDUNG
2009

Pembuatan Perangkat Lunak Menggunakan MATLAB/Simulink untuk Pengendalian Kecepatan Motor DC dengan Metode Cascade

Pipit Anggraeni

Jurusan Teknik Otomasi Manufaktur dan Mekanika – POLMAN Bandung, Jl. Kanayakan 21 Bandung.

Email : pipit_anggraeni@polman-bandung.ac.id

Abstrak

Beberapa tahun terakhir, mulai dikembangkan simulasi sistem kendali dengan menggunakan perangkat lunak MATLAB/Simulink. Perangkat lunak tersebut mempunyai banyak sumber daya yang memungkinkan berbagai simulasi pemodelan, visualisasi, dan model plant kendali. Suatu sistem kendali yang baik harus mempunyai respon yang cepat dan akurat.

Pada Tugas Akhir ini dibahas mengenai aplikasi MATLAB / Simulink untuk kendali kecepatan motor DC dengan metode Cascade. Pengendalian dengan metoda Cascade dilakukan untuk memperbaiki respon sistem dari suatu sistem yang telah diperoleh sebelumnya. Simulasi akan menunjukkan pengaruh perubahan nilai parameter kendali terhadap pengaruh respon sistem. Mikrokontroler AVR ATmega8535 digunakan sebagai interface yang menghubungkan PC dengan perangkat keras.

Respon sistem ditampilkan dalam bentuk kurva. Hasil pengujian menunjukkan bahwa dengan metoda Cascade, sistem memberikan respon yang baik, dilihat dari setting time dan steady state yang lebih baik dari respon sistem konvensional dan sistem tanpa pengendali.

Abstract

Last few years, began development of control system simulation using software MATLAB / Simulink. The software has many resources that enable a variety of simulation modeling, visualization, and control plant model. A good control system should have a rapid response and accurate.

At this final task is discussed about the application MATLAB / Simulink for DC motor speed control with Cascade method. Cascade control method is to improve the system response of a system that has been obtained previously. The simulation will show how change the value of control parameters influence the system response. ATmega8535 AVR microcontroller is used as an interface that connects the PC with the hardware.

System response is shown in the form of curves. Test results show that the Cascade method, the system provides a good response, seen from the setting time and the steady state is better than the conventional system response and control system without

1. Pendahuluan

Sistem kendali merupakan suatu cara bagaimana memperoleh suatu nilai yang diinginkan dengan cara mengatur parameter-parameter yang terdapat didalam suatu sistem. Pada umumnya tujuan perancangan suatu sistem kendali adalah untuk mendapatkan aksi kontrol sedemikian rupa sehingga sistem yang dikendalikan menjadi stabil. Suatu sistem dikatakan stabil jika memiliki ketahanan terhadap gangguan serta mempunyai respon perubahan yang cepat dan tepat.

Di era teknologi ini, sistem kendali merupakan suatu hal yang sangat penting,

sehingga diciptakanlah perangkat lunak yang digunakan untuk mensimulasikannya. Salah satu perangkat lunak yang umumnya digunakan untuk simulasi tersebut adalah MATLAB/Simulink.

Banyak sumber daya yang dapat dikembangkan seperti pemodelan, visualisasi, dan model plant kendali yang dibangun memiliki kemampuan *reutilisasi* dan parameter kendali MATLAB/Simulink dapat dikonversi oleh blok Simulink dan diagram diagram alir ke dalam PC. Pada tugas akhir ini dibahas mengenai aplikasi MATLAB / Simulink untuk kendali

kecepatan motor DC dengan metode Cascade. Simulasi waktu sebenarnya akan menunjukkan pengaruh perubahan nilai parameter kendali terhadap perubahan kecepatan motor DC, bagaimana mengendalikan kecepatan motor DC dengan perangkat lunak Simulink, bagaimana membuktikan bahwa perubahan nilai parameter kendali yang berpengaruh terhadap respon system, dan bagaimana perbandingan hasil simulasi pada Simulink dengan kondisi *plant*.

Dalam keseharian motor DC merupakan motor yang sering digunakan dalam sistem kendali. Hal ini dikarenakan motor arus searah berbeda dengan jenis motor-motor lain, terutama dalam hal torsi mula dan momen inersia. Motor arus searah atau yang memiliki torsi mula yang besar dan momen inersia yang kecil, karena motor jenis ini dirancang agar menghasilkan percepatan yang besar pada keadaan diam atau hampir diam, motor-motor biasa tidak bersifat demikian.

2. Sistem Kendali

Untuk keperluan pengendalian, pengaturan dan *supervisi* dari suatu peralatan teknik biasanya diperlukan alat pendeteksi berupa alat ukur sinyal listrik, dimana pada awalnya sinyal ini biasanya mempunyai besaran fisika yang bisa diukur sesuai dengan harga besarnya. Besaran sinyal fisika ini kemudian diubah menjadi sinyal listrik oleh sensor/detektor melalui pengukuran.

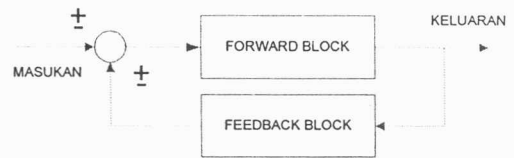
Pengendalian biasanya menggunakan sensor untuk mendeteksi / mengukur keluaran yang akan dikembalikan sebagai umpan balik untuk dibandingkan dengan masukan selaku referensi atau titik setyelan. Sistem pengendalian ini disebut sebagai sistem kendali lingkaran tertutup. Pengendalian yang tidak memanfaatkan keluarannya sebagai umpan balik untuk dibandingkan dengan masukan selaku referensi, disebut sebagai sistem kendali loop terbuka.



Gambar 2.1 Sistem kendali ideal, keluaran : $f(\text{masukan})$



Gambar 2.2 Diagram Blok Sistem kendali loop terbuka



Gambar 2.3 Diagram Blok Sistem kendali loop tertutup

2.1 Pengaruh kendali

Pembahasan pengaruh dari kendali diberikan dengan contoh pada pengaplikasian sistem pada kendali motor untuk kecepatan motor. Kendali yang akan digunakan adalah Proporsional, Integral & Derivative /diferensial.

a. Proporsional

Pengaruh komponen proporsional terhadap kecepatan motor dijelaskan sebagai berikut. *Error* positif, yang dihasilkan ketika kecepatan motor kurang dari *setpoint*, diperkuat oleh pengendali dengan nilai penguatan tertentu (umumnya dinotasikan K_p) untuk menghasilkan sinyal kendali yang lebih besar, sehingga kecepatan motor bertambah. Ketika kecepatan motor bertambah maka sinyal kesalahan akan bertambah kecil yang berarti sinyal kendali juga bertambah kecil. Pada akhirnya, kecepatan motor akan stabil pada kecepatan tertentu di bawah *setpoint* dimana sinyal kendali seimbang dengan beban yang diberikan pada motor. Jika penguatan pengandali sangat tinggi maka kemungkinan terjadi osilasi yang disebabkan oleh koreksi berlebihan terhadap sinyal *error* secara terus menerus. Sinyal kontrol yang terlalu besar mengakibatkan motor berputar di atas *setpoint*, yang artinya dihasilkan sinyal *error* negatif. Sinyal *error* ini diperkuat

oleh pengandali sehingga dihasilkan sinyal kendali negatif yang berarti ada usaha untuk menghentikan putaran motor. Kecepatan motor akan berkurang drastis dan dihasilkan sinyal kesalahan positif yang memerlukan koreksi lebih lanjut. Kesalahan positif diperkuat oleh pengandali menghasilkan sinyal kendali berlebihan yang membuat motor berputar pada kecepatan di atas *setpoint*. Demikian seterusnya sehingga terjadi osilasi dan dikatakan bahwa sistem tidak stabil. Komponen proporsional memiliki kegunaan terbatas sebab tidak dapat membuat motor untuk berputar tepat (mendekati) *setpoint*, namun mampu menghasilkan respon yang cepat terhadap sinyal *error*.

b. Integral

Pengembangan lebih lanjut dilakukan dengan menambahkan atau mengurangi nilai tertentu pada sinyal kendali sampai motor mencapai *setpoint*, dimana tidak terjadi perubahan lebih lanjut. Nilai ini dapat diperoleh dengan mengakumulasi sinyal kesalahan. Secara efektif nilai ini adalah integral dari sinyal kesalahan. Nilai ini diperkuat dengan nilai penguatan tertentu (umumnya dinotasikan K_i) membentuk suku integral, sebelum ditambahkan pada sinyal kendali. Suku integral bekerja lebih lambat untuk mengoreksi kesalahan, namun mampu menghilangkan *steady state error*.

c. Derivative / diferensial

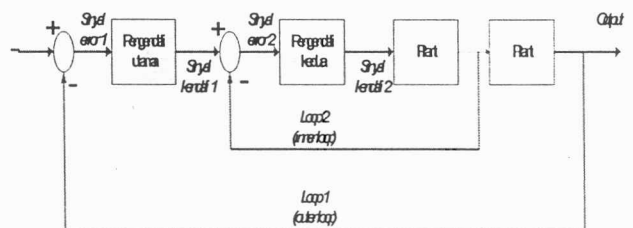
Perbaikan selanjutnya adalah dengan menggunakan laju perubahan sinyal *error* untuk ditambahkan pada sinyal kendali. Ketika motor mengalami perlambatan yang sangat cepat, misalnya akibat penambahan beban secara tiba-tiba, sinyal kesalahan juga akan meningkat secara cepat. Laju perubahan (diferensial) sinyal kesalahan diperkuat dengan nilai penguatan tertentu (umumnya dinotasikan K_d) membentuk suku diferensial, sebelum ditambahkan pada sinyal kendali. Suku diferensial bersifat antisipatif karena merespon terhadap laju perubahan sinyal kesalahan. Artinya semakin besar laju perubahan sinyal kesalahan, semakin

besar pula suku diferensial. Akibatnya sinyal kendali juga meningkat.

2.2 Metode Kendali Cascade

Cascade loop control digolongkan sebagai *Advanced Regulatory Control* bersama *feedforward control*, *ratio control*, dan *override control*. Cascade dan *feedforward* mempunyai fungsi yang sama yaitu untuk meredam gangguan dari sebuah *existing feedback loop control* yang sulit mencapai kestabilan akibat gangguan pada process, bukan ketidakstabilan karena salah memperlakukan PID.

Kebanyakan dalam suatu sistem kendali, suatu kendali mengatur *setpoint* daripada yang lainnya. Akan tetapi jika dalam suatu sistem kendali analog, kendali utama (*primary controller*) pada dasarnya mengolah perbaikan *setpoint* untuk kendali kedua (*secondary controller*) dalam alur serial yang sama sebagaimana jika hasil perbaikan telah dibuat untuk hasil terakhir. Sistem kendali tersebut adalah kendali cascade (*cascade control*).



Gambar 2.4 Diagram Blok Sistem Kontrol metode cascade sederhana

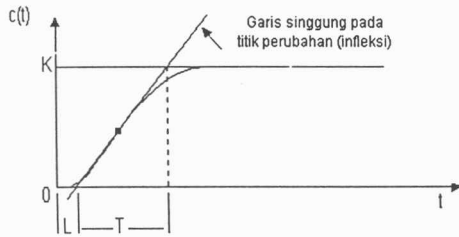
Jadi pada intinya pengontrolan dengan metoda cascade adalah pengendalian suatu sistem kendali yang memiliki pengontrol lebih dari satu yang dipasang secara seri. Dalam sistem kendali metode cascade tradisional/sederhana, sistem memiliki dua buah pengendali yaitu kendali utama (*primary controller*) dan kendali kedua (*secondary controller*). Sistem ini menggunakan sinyal keluaran dari pengendali utama untuk mempengaruhi *setpoint* dari pengendali kedua yang

seolah-olah sinyal tersebut adalah sinyal dari bagian akhir sistem kendali.

3. Pengujian Motor Untuk Menentukan Fungsi Transfer Plant

Tujuan dari pengujian ini dilakukan adalah untuk memperoleh fungsi transfer yang akan kita gunakan untuk proses pengendalian.

Pengujian motor untuk penentuan fungsi transfer *plant* ini dilakukan dengan cara memberikan masukan *step* pada sistem, maksudnya kita nyalakan dan matikan masukan tegangan dari rangkaian penggerak yang menuju ke motor. Pada perubahan nyala dan padamnya tegangan dilakukan pengamatan sinyal yang keluar dari Tachogenerator, dari hasil pengamatan yang dilakukan didapat bentuk kurva karakteristik motor. Dari kurva respon *plant* yang didapat maka akan dapat dicari nilai fungsi *transfer* dari *plant* (motor + rangkaian penggerak).



Gambar 3.1 Gambar kurva acuan

Dari kurva respon yang dibaca melalui *digital oscilloscope*, dapat ditentukan nilai penguatan/gain (K) dan konstanta waktu (τ). Nilai penguatan ditentukan dengan rumus :

$$K = \frac{V_{out}}{V_{in}}$$

Nilai V_{in} yang diberikan pada *plant* adalah 12 V DC. Nilai V_{out} *steady state* dari kurva ditentukan dengan menggunakan fasilitas *cursor* pada *digital oscilloscope*. Sehingga, perhitungan nilai penguatan (K) dan konstanta waktu (τ) adalah sebagai berikut :

$$K = \frac{10.8V}{12V} = 0.9$$

$$\tau = 152.86ms$$

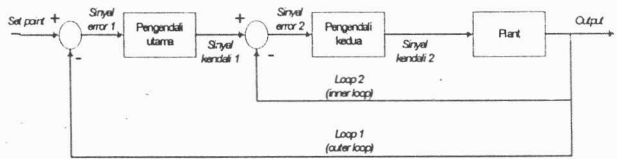
$$\tau = 0.153s$$

Maka, fungsi transfer *plant* adalah :

$$FT = \frac{K}{\tau s + 1}$$

$$FT = \frac{0.9}{0.153 s + 1}$$

3.1 Menentukan nilai parameter kendali



Gambar 3.2 Diagram Blok Sistem Kontrol metode observasi cascade

perhitungan nilai parameter kendali untuk kendali kedua dengan menggunakan data hasil pengujian *tuning loop* terbuka menghasilkan nilai berikut :

diperoleh nilai rerata $L = 18,57$

dan nilai $T = 134,29$

maka

$$K_p = 1.2 \cdot [(rerata T) / (rerata L)]$$

$$= 1.2 \cdot (134,29 / 18,57)$$

$$K_p = 8,68$$

$$T_i = 2 \cdot rerata L$$

$$= 2 \cdot 18,57 ms$$

$$K_i = 26,93$$

$$T_d = 0,5 \cdot rerata L$$

$$= 0,5 \cdot 18,57 ms$$

$$K_d = 0,0093$$

Metode penalaan yang digunakan adalah metode *Root locus*. Berikut adalah metode perhitungan nilai parameter kendali :

Nilai parameter Z_o didapat dengan persamaan sebagai berikut:

$$Z_o = \frac{K_i}{K_p}$$

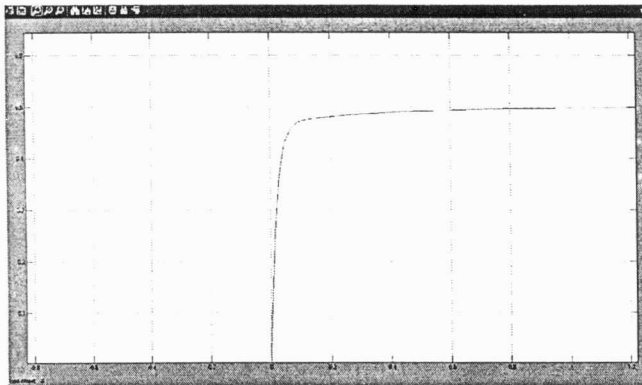
Untuk menentukan nilai konstanta integral dan konstanta proporsional dengan metode *root locus* dimana nilai rasio redaman (ζ) dan waktu penetapan (*settling time*, t_s) ditentukan terlebih dahulu. Maka sistem harus mempunyai *pole* kalang terbuka sebesar,

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2}$$

Dengan

$$\omega_n = \frac{4}{\xi t_s}$$

ω_n adalah frekuensi alami (*natural frequency*).



Gambar 3.3 Kurva Acuan penalaan Root Locus

settling time (t_s) = 1s

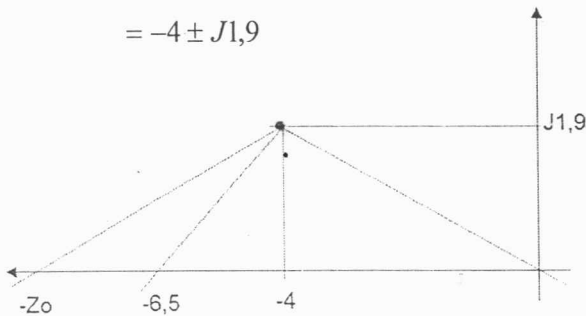
rasio redaman (ζ) = 0.9s

$$\omega_n = \frac{4}{\xi t_s} = \frac{4}{0,9 \cdot 1} = 4,44$$

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2}$$

$$= -0,9 \cdot 4,44 \pm j4,44\sqrt{1-0,81}$$

$$= -4 \pm j1,9$$



$$\theta Z_0 - (38,6^\circ + 153,4^\circ) = -180^\circ$$

$$\tan 12^\circ = \frac{4}{x}$$

$$0,21 = \frac{4}{x}$$

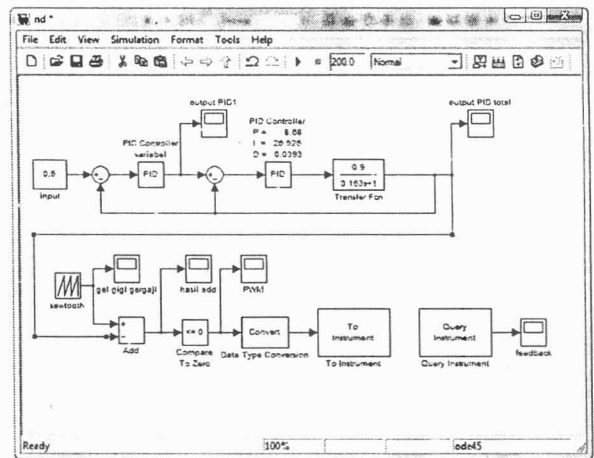
$$x = 18,8$$

$$Z_0 = 18,8 + 4$$

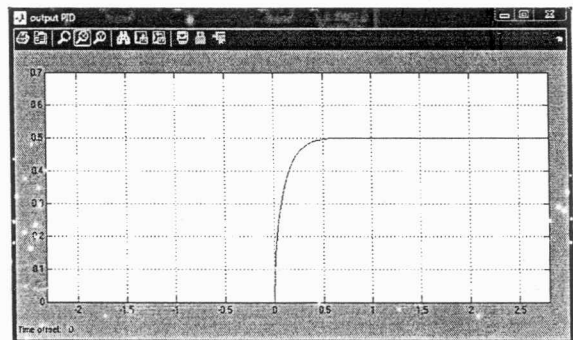
$$= 22,8$$

$$K_i = 22,8 \quad K_p$$

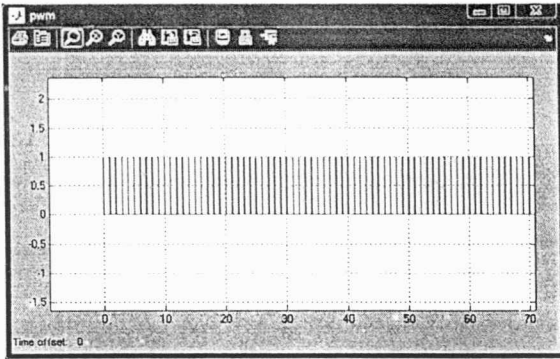
4. Hasil



Gambar 4.1 Model Sistem kendali Metode Cascade



Gambar 4.2 Respon PID total pada in = 0.5



Gambar 4.3 Keluaran PWM pada $V_{in} = 0.5$

5. Kesimpulan

Perangkat lunak Simulink pada MATLAB mampu mengendalikan kecepatan putaran motor DC.

Metode kendali yang diimplementasikan pada sistem pengendalian kecepatan putaran motor DC adalah metode kendali cascade. Metode cascade digunakan untuk memperbaiki respon dari suatu sistem kendali yang mana kendali yang dimiliki saat itu tidak memungkinkan untuk dilakukan perubahan untuk parameter kendalinya. Karena dengan metode cascade kita bisa menambahkan kendali dalam suatu sistem yang dipasang secara seri dengan kendali sebelumnya yang telah ada.

Daftar acuan

1. Away, Gunadi A. 2004. *The Shortcut of MATLAB Programming*. Bandung : Informatika.
2. Franklin, Gene F, dkk. 2006. *Feedback Control of Dynamic Systems, Fifth Ed.* New Jersey : Prentice Hall International.
3. Indarto, Purnomo W. (Editor). 1996. *Teknik Kontrol Automatik, Edisi kedua*. Jakarta : Erlangga.
4. Ogata, Katsuhiko. 2002. *Modern Control Engineering, Fourth Ed.*

New Jersey : Prentice Hall International.

5. Prasimax. 23 Mei 2009. *Pengendalian Motor DC PWM*. [Pengendalian-Motor-DC-PWM.html](http://www.mathworks.com/PWM.html).
6. Pretitl, Zsuzsa. Peter Bauer. 1 Juli 2009. *Cascade Control Solution for Traction Motor for Hybrid Electric Vehicle*. Budapest University of Technology and Economics. http://www.bmf.hu/journal/Preitl_Bauer_Bokor_11.pdf.
7. 3 Mei 2009. *Cascade Control*. http://www.bgu.ac.il/chem_eng/pages/Courses/oren%20courses/Chapter_10.pdf.
8. 13 Mei 2009. *Simulink Dynamic System Simulation for MATLAB*(pdf). www.mathworks.com.
9. 20 Mei 2009. *Instrument Control Toolbox 2.8*. www.mathworks.com.
10. 3 Juli 2009. *Performance Assesment of Cascade Control Systems*. Journal of The Institution of Engineers, Singapore. http://www.ies.org.sg/journal/current/v45i6/v45i6_3.pdf.