

PERANCANGAN PI KONTROLER PADA KONTROL KECEPATAN MOTOR DC DENGAN KOMBINASI POLE PLACEMENT DAN SYMMETRICAL OPTIMUM

Christopher Imantaka¹, Moch. Rusli², Erni Yudaningtyas².

¹Mahasiswa Teknik Elektro Univ. Brawijaya, ²Dosen Teknik Elektro Univ. Brawijaya

Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya

Jalan MT. Haryono 167, Malang 65145, Indonesia

Email: christopherimantaka20@gmail.com

Abstrak— Terdapat banyak metode dalam perancangan sistem kontrol, salah satunya adalah perancangan sistem kontrol menggunakan metode penempatan kutub (*pole placement*). Penempatan *pole*/kutub pada metode *pole placement* masih bersifat *trial* dan *error*, oleh karena itu dengan menggunakan kombinasi *pole placement* dan *symmetrical optimum* akan dimungkinkan perancangan *pole placement* agar semakin terarah untuk penentuan *pole* sistem pada *plant*. Didapatkan nilai $K_p=0.0508$ dan $K_i=2.6595$ menggunakan *symmetrical optimum* untuk mencari nilai fungsi alih keseluruhan kontroler. Dari fungsi alih keseluruhan didapatkan nilai *pole* -471 dan -23 yang nantinya akan digunakan sebagai *pole* referensi dalam metode *pole placement*.

Kata Kunci— *Pole Placement, Symmetrical Optimum, Kontroler.*

I. PENDAHULUAN

Motor Direct Current (DC) adalah penggerak yang sering digunakan dalam teknologi kontrol baik dalam industri maupun rumah tangga. Motor DC memudahkan pekerjaan sehingga proses industri dapat berjalan efisien. Motor DC memiliki respon yang cepat, namun masih memiliki *error steady state*. Oleh karena itu, dibutuhkan suatu kontroler yang tepat dan sesuai dengan *plant* sistem (Faisol, 2014).

Dalam suatu sistem kontrol ada beberapa macam kontroler, salah satunya adalah kontroler *proporsional* (P) dan kontroler *integral* (I). Pengontrolan *proporsional* (P) memiliki keluaran yang sebanding atau *proporsional* dengan besarnya sinyal kesalahan selisih antara besaran yang diinginkan dengan harga aktualnya. Pengontrolan *integral* (I) berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan *steady state*. Masing-masing kontroler ini mempunyai keunggulannya tertentu, dimana aksi kontrol *proporsional* (P) mempunyai keunggulan *rise time* yang cepat dan aksi kontrol *integral* mempunyai keunggulan untuk memperkecil *error*. Kontroler *proporsional integral* (PI) adalah penggabungan antara kontrol *proporsional* (P) dengan kontrol *integral* (I). Kontrol PI sendiri berfungsi untuk mempercepat output dan mengurangi *offset*. Sehingga dengan menggunakan kontrol PI dalam kontrol kecepatan motor DC dapat memberi keuntungan dalam memperbaiki respon transien, mengurangi *error steady state* dan memberi efek redaman.

Pada desain sistem kontrol yang baik harus memenuhi persyaratan-persyaratan tertentu yang telah ditetapkan. Persyaratan yang harus dipenuhi sistem kontrol disebut sebagai indeks untuk kerja (*performance index*). Indeks ini berkaitan dengan ketelitian, kestabilan dan waktu *output* mencapai *steady state*. Terdapat banyak metode dalam perancangan sistem kontrol. Terdapat beberapa cara untuk mendapatkan nilai-nilai dalam memenuhi kontroler *proporsional* (P) dan kontroler *integral* (I). Salah satu metode perancangan sistem kontrol adalah menggunakan metode *pole placement*. Dari hasil desain kriteria performansi yang diinginkan, ditentukan lokasi *pole* yang diperlukan. Kestabilan suatu sistem kontrol lup tertutup, baik waktu kontinu maupun waktu diskrit ditentukan oleh letak *pole*. Meskipun suatu sistem sudah stabil, belum tentu *pole* dari sistem tersebut sesuai dengan yang diinginkan karena hal ini menentukan tingkat kecepatan mencapai *steady state* (Nugroho, 2013). Dalam perancangan motor DC, terdapat pula metode optimum. Salah satunya metode optimum adalah *symmetrical optimum*. Metode *symmetrical optimum* (SO) diusulkan oleh Kessler pada tahun 1958 dan dimodifikasi lebih lanjut oleh Voda dan Landau pada tahun 1955 untuk memastikan bahwa kontroler dapat mencapai fasa maksimum dalam sistem loop tertutup. Metode *symmetrical optimum* dirancang untuk memaksimalkan *phase margin* dari sistem kontrol yang telah dirancang (Martin Macaba, 2012).

Penempatan *pole* pada metode *pole placement* masih bersifat *trial* dan *error*. Oleh karena itu dengan kombinasi *pole placement* dan *symmetrical optimum* akan dimungkinkan perancangan sistem kontrol pada motor DC mendapatkan hasil seperti yang diinginkan.

II. TINJAUAN PUSTAKA

A. Motor DC

Motor yang digunakan dalam penelitian ini merupakan motor DC magnet permanen. Merubah kecepatan di motor ini dengan cara mengatur tegangan yang masuk ke dalam motor. Motor DC yang digunakan dalam penelitian ini dapat dilihat pada Gambar 1.

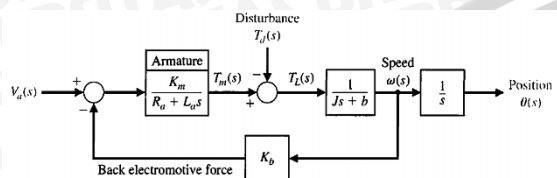


Gambar 1. Motor DC

(Sumber : http://mabuchi-motor.co.jp/rf_130ch)

B. Fungsi Alih Motor DC

Motor *Direct Current* (DC) merupakan aktuator yang banyak digunakan dalam teknologi kontrol. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Satu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan dalam satu lilitan disebut *armature* (jangkar). *Armature* adalah sebatang untuk komponen yang berputar di antara medan magnet. Diagram blok pengontrolan kecepatan motor DC magnet permanen dapat dilihat dalam Gambar 2.



Gambar 2. Diagram blok pengontrolan kecepatan motor DC magnet permanen

Sumber: Dorf, R. C. & Bishop, R. H. (2011)

Apabila *disturbance* (gangguan) $T_d(s)=0$, maka fungsi alih motor DC adalah

$$\frac{\omega(s)}{V_a(s)} = \frac{K_m}{(R_a + L_a s)(J s + b) + (K_b K_m)}$$

dengan:

K_m = konstanta motor

K_b = konstanta *back electromotive force*

R_a = resistansi *armature*

L_a = induktansi *armature*

J = inersia

b = gesekan

B. Rotary Encoder

Rotary Encoder adalah elektromekanik yang dapat mendeteksi atau memonitor gerakan dan posisi. *Rotary encoder* biasanya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat dijadikan gerakan, posisi, dan arah. Adapun *rotary encoder* ditunjukkan oleh gambar 3.



Gambar 3. Rotary Encoder

(Sumber : <http://indo-ware.com/produk-2587-di-rotary-encoder.html>)

C. Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke

komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke aUno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial. Adapun bentuk dari arduino uno dapat dilihat dalam gambar 4.



Gambar 4. Arduino Uno

(Sumber : http://geetech.com/wiki/index.php/Arduino_Uno)

D. Kontroler

Sistem pengontrolan dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengontrolan adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K. 1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon *input* dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *output* berosilasi terus menerus ketika dikenai suatu gangguan. Suatu sistem pengontrolan biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak tapi menunjukkan respon transien berupa osilasi teredam sebelum mencapai *steady state*.

E. Kontroler Proportional Integral (PI)

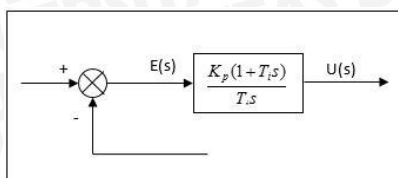
Kontroler *proporsional integral* (PI) memiliki kemampuan untuk mempercepat output dan mengurangi *offset*. Persamaan kontroler *proporsional integral* (PI) adalah

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad \dots \dots \dots (1)$$

Adapun fungsi alihnya adalah

$$\begin{aligned} \frac{U(s)}{E(s)} &= K_p \left(1 + \frac{1}{T_i s} \right) \\ &= \frac{K_p (1 + T_i s)}{T_i s} \end{aligned}$$

dengan K_p penguatan proporsional dan T_i disebut waktu integral, yang keduanya dapat ditentukan. Waktu integral mengatur aksi kontrol internal sedangkan perubahan nilai K_p berakibat pada bagian aksi kontrol proporsional maupun integral. Diagram blok Kontroler PI dapat dilihat dalam Gambar 5.



Gambar 5. Diagram Blok Kontroler PI

F. Symmetrical Optimum (SO)

Symmetrical Optimum (SO) adalah metode desain yang didasarkan pada optimasi. Tuning parameter dengan *Symmetrical Optimum* pertama dikemukakan oleh Kessler pada 1985. Metode ini memaksimalkan *phase margin* dari sistem kontrol dan mengarahkan ke fasa yang simetri dan karakteristik amplitudo. Persamaan fungsi alih *symmetrical optimum* pada persamaan (10).

$$F_{ol}(s) = \underbrace{G_c \left(\frac{T_i s + 1}{T_i s} \right)}_{\text{Kontroler PI}} \underbrace{\frac{G_m}{T_m s + 1}}_{\text{Plant}} \frac{1}{T_{mn} s} \dots\dots(2)$$

Dengan :

F_{ol} = Fungsi open loop

G_m = Nilai gain motor

T_m = Time konstan motor terkecil

G. Pole Placement atau Penempatan Kutub

Pole placement adalah suatu metode desain kontrol dimana menentukan letak dari *pole* sistem loop tertutup pada bidang kompleks dengan mengatur *gain* kontroler. Pada pendekatan secara konvensional, kita mengasumsikan bahwa pengaruh respon *pole* loop tertutup tak domain dapat diabaikan. Pada pendekatan dengan penempatan *pole* (*pole placement*), pendekatan penempatan *pole* yang diberikan menetapkan semua loop tertutup. Terdapat persyaratan pada bagian sistem untuk *pole-pole* loop tertutup yang akan ditempatkan pada lokasi yang dipilih secara sembarang yaitu bahwa sistem harus berupa keadaan lengkap yang dapat dikontrol.

Keterkontrolan keadaan secara sempurna dapat dilihat dari sistem waktu diskrit. Sistem diskrit yang dinyatakan oleh

$$x((k+1)T) = Gx(kT) + Hu(kT) \dots\dots\dots(3)$$

Dimana :

$x(kT)$ = vektor keadaan (vektor n dimensi)

$u(kT)$ = sinyal kontrol

G = matriks nonsingular n x n

H = matriks n x 1

T = periode cacah

Kondisi keterkontrolan sempurna diturunkan berdasarkan kenyataan bahwa jika suatu sistem terkontrol sempurna, maka ada sinyal kontrol kontinyu sepotong-sepotong yang akan memindahkan setiap keadaan awal ke titik asal dalam sejumlah periode cacah yang terhingga.

Observability atau keteramatan sistem linier dapat dinyatakan oleh persamaan berikut:

$$\dot{x}(t) = Ax(t)$$

$$y = Cx(t)$$

dimana :

x = vektor keadaan (vektor n-dimensi)

y = vektor keluaran (vektor m-dimensi)

A = matriks n x n

C = matriks m x n

Keteramatan sempurna dari sistem waktu diskrit. Tinjau sistem yang dinyatakan oleh :

$$x((k+1)T) = Gx(kT)$$

$$y(kT) = Cx(kT)$$

dimana:

$x(kT)$ = vektor keadaan (vektor n-dimensi)

$y(kT)$ = vektor keluaran (vektor m-dimensi)

G = matriks n x n

C = matriks m x n

T = periode cacah

Persamaan ruang keadaan waktu berbentuk :

$$\dot{x}(t) = Ax(t) + Bu(t) \dots\dots\dots(4)$$

Dan sinyal kontrol diberikan oleh :

$$u = -kx \dots\dots\dots(5)$$

Dan substitusi persamaan (4) ke persamaan (5) sehingga didapatkan

$$\dot{x}(t) = (A - BK)x(t) \dots\dots\dots(6)$$

Solusi dari persamaan (6)

$$x(t) = e^{(A-BK)t} x(0)$$

Kestabilan dari sistem (6) ditentukan oleh nilai eigen dari $(A-BK)$, artinya jika matriks K dapat dipilih secara tepat, maka bagian riil dari nilai eigen matriks $(A-BK)$ terletak di sebelah kiri sumbu imajiner bidang s , hal itu berarti, untuk semua $x(0) \neq 0$, $x(t) \rightarrow 0$ untuk $t \rightarrow \infty$. Syarat perlu dan cukup untuk penempatan pole dinyatakan dalam teorema berikut :

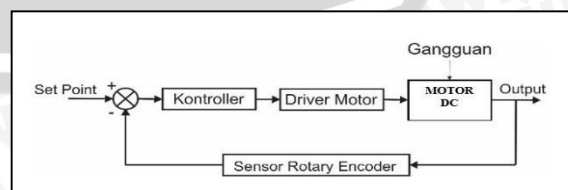
Teorema [Ogata Katsuhiko, 1997]: “Jika diberikan suatu sistem, maka syarat perlu dan cukup untuk penempatan sembarang pole yang diinginkan adalah bahwa sistem tersebut terkontrol secara lengkap”.

III. PERANCANGAN

Perancangan dalam pembuatan skripsi ini bertujuan untuk merancang beberapa perangkat maupun alat secara keseluruhan. Perancangan perangkat tersebut meliputi perancangan perangkat keras maupun perancangan perangkat lunak.

A. Diagram Blok Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana. Diagram blok sistem dapat dilihat pada gambar 6.



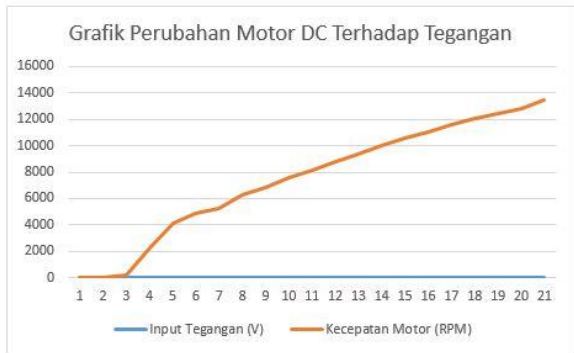
Gambar 6. Diagram Blok Sistem Loop Tertutup

Setpoint berupa nilai kecepatan yang diberikan oleh *user*. Nilai *setpoint* akan disimpan di dalam arduino uno. Nilai *error* yaitu deviasi antara pembacaan aktual kecepatan motor dari *rotary encoder* dan nilai *set point*.

Nilai – nilai tersebut akan dikalkulasikan dan diolah oleh arduino uno.

B. Karakteristik Motor DC

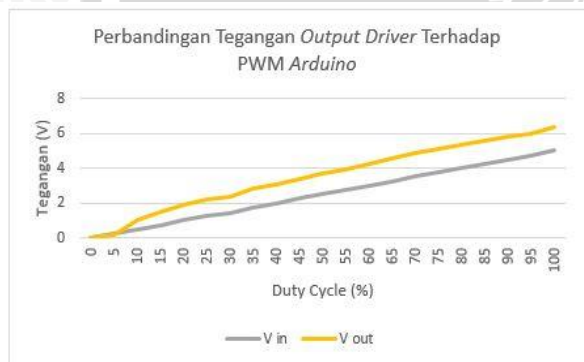
Karakteristik motor DC dilakukan untuk mengetahui karakter atau gain motor DC. Hasil diperoleh dengan mengamati kecepatan motor DC terhadap perubahan tegangan motor DC. Adapun grafik perubahan motor DC terhadap tegangan ditunjukkan pada Gambar 7.



Gambar 7. Grafik Perubahan Motor DC Terhadap Tegangan

C. Karakteristik Driver Motor

Pengujian karakteristik driver motor bertujuan untuk mengetahui masukan dutycycle PWM dari mikrokontroler dengan besarnya nilai perubahan V_{rms} . Adapun grafik perbandingan tegangan output driver terhadap PWM arduino ditunjukkan pada gambar 8.



Gambar 8. Grafik Perbandingan Tegangan Output Driver Terhadap PWM Arduino

D. Fungsi Alih Motor

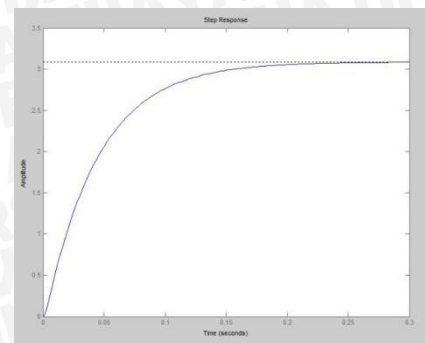
Penentuan fungsi alih dilakukan dengan cara mencari nilai-nilai sebagai berikut :

$$\begin{aligned} R &= 2.8 \, \Omega \\ L &= 0.0057 \, H \\ K_t &= 0.3200 \, Nm/A \\ K_e &= 0.2831 \, Vs/rad \\ b &= 0.0047 \, Nms \\ J &= 0.0017 \, kgm^2 \end{aligned}$$

Kemudian parameter yang ada dimasukkan dalam persamaan fungsi alih motor pada persamaan (2) dan mendapatkan hasil sebagai berikut:

$$Fm(s) = \frac{3,347 \times 10^4}{s^2 + 494s + 1,084 \times 10^4} \dots\dots\dots(7)$$

Dengan memberi masukan sinyal unit *step* pada persamaan diatas maka didapatkan grafik respon *open loop* yang ditunjukkan dalam Gambar 9.



Gambar 9. Respon Plan *Open Loop*

E. Penentuan Konstanta Kp dan Ki Dengan Metode *Symmetrical Optimum*

Berdasarkan persamaan (2), fungsi alih motor perlu diubah. G_m adalah nilai gain dari motor yang telah dimodifikasi dan T_m adalah *Time Konstan* motor terkecil. Pada persamaan (7) harus diubah terlebih dahulu ke bentuk

$$F(s) = \frac{G}{(T_1s + 1)(T_2s + 1)} \dots\dots\dots(8)$$

Mulanya bagian penyebut diakarkan menjadi

$$F(s) = \frac{3.347 \times 10^4}{s^2 + 494s + 1.084 \times 10^4}$$

$$F(s) = \frac{3.347 \times 10^4}{(s + 471)(s + 23)}$$

$$F(s) = \frac{3.347 \times 10^4}{(471)(23)\left(\frac{1}{471}s + 1\right)\left(\frac{1}{23}s + 1\right)}$$

$$F(s) = \frac{3.0896}{\left(\frac{1}{471}s + 1\right)\left(\frac{1}{23}s + 1\right)} \dots\dots\dots(9)$$

$$T_1 = \frac{1}{471} = 2.1231 \times 10^{-3}$$

$$T_2 = \frac{1}{23} = 0.04348$$

$$T_1 < T_2$$

Dengan :

$F(s)$ = Fungsi alih motor

G_m = Nilai gain motor

T_m = *Time Konstan* motor terkecil

Maka nilai G_m adalah 3.0896 dan $T_m = T_1$. Nilai T_{mn} ditetapkan samadengan 0.001. Dengan mengubah nilai faktor redaman (D) maka dapat ditentukan nilai G_c dan T_i melalui persamaan (20).

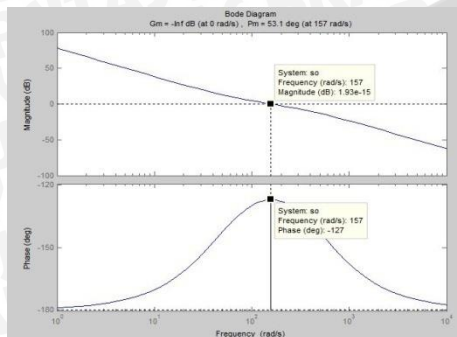
$$G_c = \frac{1}{a} \frac{T_{mn}}{G_m T_m} \dots\dots\dots(10)$$

$$T_i = a^2 T_m \quad a > 1 \dots\dots\dots(11)$$

$$a = 2D + 1 \dots\dots\dots(12)$$

Nilai-nilai dari variabel diatas dimasukkan kedalam Matlab dan ditampilkan dalam bodeplot untuk mengetahui *frequency crossover* pada margin terletak pada fasa maksimumnya.

Dengan nilai D=1 didapatkan nilai Kp dan Ki dengan metode *symmetrical optimum* didapatkan nilai Kp=0,0508 dan Ki=2,6595. Hasil bodeplot dengan metode *symmetrical optimum* terdapat pada gambar 10.



Gambar 10. Hasil Bodeplot Dengan Metode *Symmetrical Optimum*

F. Desain Kontroler Struktur *Output Feedback Control*

Dalam mendesain kontroler struktur *Output Feedback Control* diperlukan nilai dari matriks A,B,C,D,K dan M. Dimana matriks A adalah matriks keadaan orde $n \times n$, B adalah matriks masukan orde $n \times m$, C adalah matriks keluaran orde $p \times n$, D adalah matriks transmisi langsung orde $p \times n$, K adalah matriks umpan balik dan M adalah matriks pra filter. Langkah – langkah yang dilakukan untuk mencari matriks – matriks tersebut adalah sebagai berikut :

1. Fungsi alih dari motor DC yang didapatkan dari pemodelan digunakan pertama – tama untuk mencari matriks A,B,C dan D dengan menggunakan *state space*.

$$\frac{Y(s)}{U(s)} = \frac{33470}{s^2 + 494s + 10840}$$

Fungsi alih dari motor DC tersebut dimasukkan kedalam matlab dengan menggunakan persamaan yaitu :

```
>> num=[33470];
>> den=[1 494 (1.084*(10^4))];
>> [A,B,C,D]=tf2ss(num,den)
```

Setelah itu didapatkan matriks:

$$A = \begin{bmatrix} -494 & 10840 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = [1 \quad 33470]$$

$$D = [0]$$

2. Setelah diketahui matriks A, B, C dan D dari plant selanjutnya adalah menentukan pole dari kontroler untuk mendapatkan nilai matriks umpan balik K. Matriks umpan balik K bisa

didapatkan dengan menggunakan persamaan pada matlab yaitu:

K=place(A,B,p) ;

dimana:

A = matriks A, matriks orde 2×2

B = matriks B, matriks orde 2×1

p = pole kontroler yang nilainya bebas, matriks orde 2×1

3. Setelah mendapatkan nilai matriks K, langkah selanjutnya adalah mencari matriks *pra filter* M menggunakan persamaan pada matlab yaitu:

m=C*inv(-Ad)*B

dimana:

m = matriks M

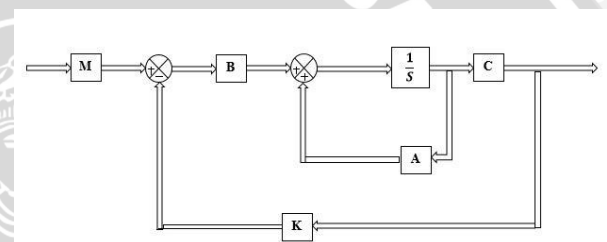
C = matriks C

Ad = matriks hasil dari $(A-BK)x(t)$

B = matriks B, matriks orde 2×1

G. Diagram Blok Perancangan Desain

Setelah dilakukan perancangan, didapatkan blok perancangan desain ditunjukkan pada gambar 11.



Gambar 11. Diagram Blok Perancangan Desain

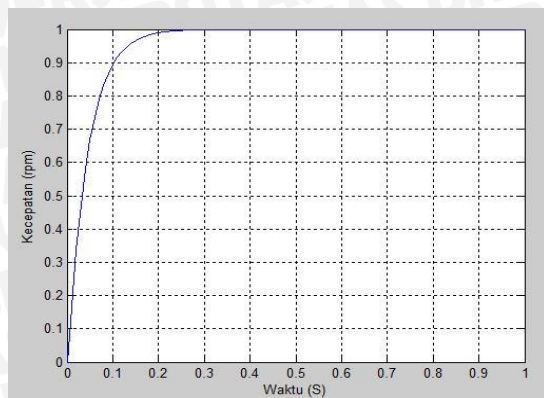
IV. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dilakukan dengan melakukan simulasi dan menganalisis sistem serta melakukan pengujian keseluruhan sistem. Simulasi dan analisis sistem dilakukan untuk mengetahui apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan skema pembuatan. Pengujian keseluruhan sistem dilakukan untuk menguji apakah parameter yang sudah didapatkan dapat diaplikasikan pada alat. Pengujian dibagi menjadi beberapa bagian, yaitu:

1. Simulasi *Pole Placement* menggunakan MATLAB.
2. Pengujian keseluruhan sistem.

A. Simulasi *Pole Placement* Menggunakan MATLAB

Simulasi *pole placemenet* pada matlab diperlukan agar dapat mengetahui hasil dari *state space* fungsi alih motor yang telah diaplikasikan kedalam metode *pole placement*. Hasil dari simulasi *pole placement* pada matlab ditunjukkan pada Gambar 12.

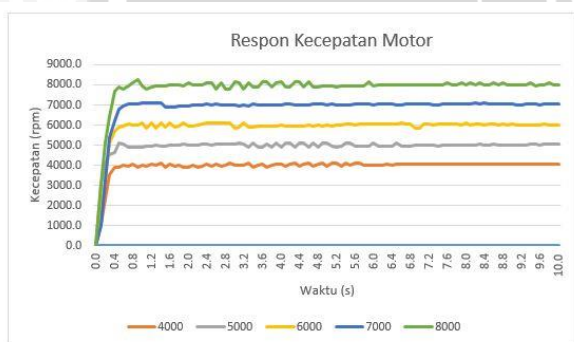


Gambar 12. Hasil Simulasi Pole Placement Matlab

B. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem bertujuan untuk mengetahui respon motor ketika diberikan metode *pole placement*. Data dengan nilai *pole* -471, -23 dan dengan nilai $M = 3.0896$ akan diimplementasikan pada alat. Implementasi dengan mengubah – ubah setpoint ini dilakukan supaya kita mengetahui bagaimana respon *output* kecepatan sistem apabila diberikan dengan nilai yang berubah – ubah.

Setpoint diberikan pada kecepatan 4000, 5000, 6000, 7000 dan 8000. Grafik hasil dari pengujian keseluruhan sistem tanpa gangguan ditunjukkan pada Gambar 13.



Gambar 13. Respon Kecepatan Motor

Dari grafik *output* respon kecepatan motor DC dapat dilihat bahwa kontroler *pole placement* mampu mengendalikan kecepatan motor pada setiap *setpoint*. Nilai *error steady state* pada *setpoint* 4000 sampai 8000 sebesar 1,9% sampai 2,28%. Untuk hasil pada Gambar 4.2 dirangkum pada tabel 1.

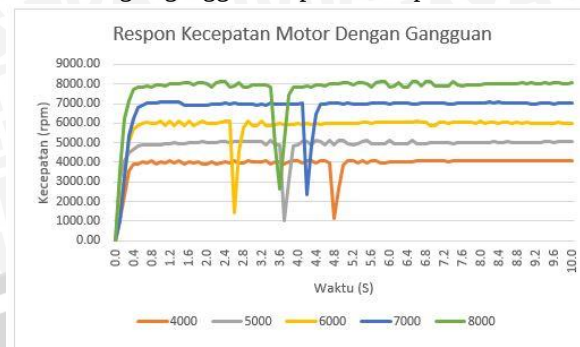
Tabel 1. Rekapitulasi Analisis Grafik Respon Kecepatan Motor

Setpoint	ts	td	tr	tp	mp
4000	0,8 s	0,2 s	0,3 s	0,5 s	5,80%
5000	1 s	0,2 s	0,4 s	0,6 s	2,60%
6000	0,6 s	0,3 s	0,5 s	1 s	2,50%
7000	0,7 s	0,3 s	0,4 s	1 s	1,45%
8000	1,1 s	0,2 s	0,6 s	0,9 s	3,02%

C. Pengujian Keseluruhan Sistem Dengan Gangguan

Pengujian keseluruhan sistem dengan gangguan bertujuan untuk mengetahui bagaimana respon motor ketika diberikan gangguan ketika sedang bekerja. Setpoint diberikan pada kecepatan 4000, 5000, 6000,

7000 dan 8000. Grafik hasil pengujian keseluruhan sistem dengan gangguan dapat dilihat pada Gambar 14.



Gambar 14. Respon Kecepatan Motor Dengan Gangguan

Dari grafik respon kecepatan motor dengan gangguan dapat dilihat, bahwa waktu yang dibutuhkan sistem untuk mencapai *steady state* adalah 0,3 dan 0,4 detik tanpa *overshoot* dengan nilai *error steady state* sebesar 0,5% sampai 2,44%.

D. Pengujian Sistem Dengan Beban Cakram Padat

Salah satu aplikasi dari motor DC tipe RF-130CH adalah sebagai pemutar disk pada CD/DVD player. Beban berupa cakram padat dengan ukuran diameter 120 mm dan berat 15-20 mg. Hasil pengujian dengan setpoint 4000 rpm diberikan oleh gambar 15.



Gambar 15. Respon Kecepatan Motor

Berdasarkan grafik respon yang diberikan maka dapat diketahui waktu yang dibutuhkan sistem untuk mencapai *steady state* adalah 1.1 detik dengan *overshoot* 4.1 % serta nilai *error steady state* sebesar 2,28%.

V. KESIMPULAN DAN SARAN

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada penelitian sistem pengontrolan kecepatan motor DC diambil kesimpulan sebagai berikut :

- 1) Parameter PI yang didapatkan menggunakan metode *symmetrical optimum* digunakan untuk mendapatkan nilai pole dari motor DC. Langkah awal yang dilakukan dengan mengubah nilai fungsi alih dari motor DC ke bentuk

$$F(s) = \frac{G}{(T_1s+1)(T_2s+1)}$$
 yang dimana akan didapatkan nilai Gm. Variabel-variabel tersebut dimasukkan kedalam matlab dan ditampilkan dalam *bodeplot* untuk mengetahui *frequency crossover* pada margin terletak pada fasa maksimumnya. Dengan metode *symmetrical optimum*, didapatkan nilai Kp = 0,0508 dan nilai Ki = 2,6595.

- 2) Simulasi output motor DC setelah diberi kontroler *pole placement* mempunyai *settling time* 0,3 detik tanpa *error steady state* dan tanpa *overshoot*. Hasil dari implementasi tanpa gangguan mempunyai *settling time* sebesar 0,6 detik sampai 1 detik dengan *error steady state* sebesar 1,9% sampai 2,28%. Sedangkan implementasi dengan gangguan, waktu yang dibutuhkan untuk mencapai *steady state* kembali setelah diberi gangguan bernilai 0,3 detik sampai 0,4 detik tanpa *overshoot* dan *error steady state* sebesar 0,5% sampai 2,44%.

B. SARAN

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan :

1. Motor DC yang digunakan dapat diganti dengan yang lain seperti BLDC motor atau motor DC *shunt*.
2. Rotary encoder yang digunakan bisa diganti dengan hall sensor untuk lebih baik dalam sensing kecepatan.
3. Sinyal PWM yang berasal dari Arduino lebih baik menggunakan mode 16-bit daripada 8-bit untuk memperkecil *error steady state* karena lebih teliti dalam pengaturan *dutycycle*.

DAFTAR PUSTAKA

- [1] Ogata, K. 1997. Teknik Kontrol Automatik. Jakarta: Erlangga.
- [2] Levine, William S. 1996. The Control Handbook. US : CRC Press and IEEE Press
- [3] Soemarwanto. 1993. Dasar Konversi Energi Elektrik Jilid II. Malang : Fakultas Teknik Universitas Brawijaya.
- [4] Faisol, M. Arif. 2015. Sistem Kontrol Kecepatan Motor DC D-6759 Berbasis Arduino Mega 2560. Malang : Skripsi Teknik Elektro Universitas Brawijaya Malang.
- [5] Nugroho, Ari. 2013. Penstabilan Sistem Melalui Umpan Balik Keadaan Proporsional-Integral. Semarang : Skripsi Fakultas Sains dan Matematika Universitas Diponegoro.
- [6] Mizera, Roman. 2005. Modification Of Symmetric Optimum Methode. Seminar, Instruments and Control. XXX: 319-320.
- [7] Siswoyo, Bambang. <https://bsiswoyo.lecture.ub.ac.id/2012/01/motor-dc-fungsi-alih/> (diakses pada 7 November 2015).