Jurnal Teknik Elektro Vol. 8 No. 2 ISSN 1411 - 0059
Juli - Desember 2016

Perancangan Kendali Kecepatan Motor Arus Searah Menggunakan Metode *Root Locus*

Khoirudin Fathoni[1] dan Alfa Faridh Suni[2]

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang
Kampus UNNES, Sekaran, Gunungpati, Semarang, Jawa Tengah, 50229, Indonesia
khoirudinfathoni@mail.unnes.ac.id[1], alfafs@mail.unnes.ac.id[2]

Abstrak— Motor arus searah (DC) menjadi populer karena fleksibilitas dan kemudahan dalam pengoperasiannya. Namun motor DC perlu diatur agar mempunyai kecepatan serta respon transien yang diinginkan terlebih ketika diberi beban. Penelitian ini bertujuan untuk merancang dan mensimulasikan alternatif pengendalian motor DC. Pengendalian yang digunakan adalah dengan metode *Proporsional Integral Derivatif* (PID) melalui *root locus*. Pada artikel ini akan dipaparkan pemodelan motor DC yang akan diperoleh fungsi alih antara kecepatan sudut dan input tegangan jangkar. Kemudian akan dilihat dan dirancang skema pengendalian dengan respon transien dan steady state yang diinginkan melalui *root locus* dengan menambahkan *pole* dan *zero*. Setelah disimulasikan, hasil perancangan kendali PID yang diperoleh telah sesuai spesifikasi yang diinginkan serta stabil dengan uji kestabilan Routh Hurwitz.

Kata kunci— kecepatan motor arus searah, PID, *root locus*

I. PENDAHULUAN

Motor listrik semakin banyak digunakan sebagai penggerak pada skala industri, rumahtangga, robotika dan lain seba*gain*ya. Terlebih motor listrik arus searah (motor DC) yang pengoperasiannya cukup mudah. Meskipun demikian, kecepatan motor DC harus diatur supaya mempunyai kecepatan *steady state* dan respon *transient* sesuai dengan yang diinginkan terlebih lagi ketika motor DC tersebut digunakan pada sistem yang membutuhkan ketepatan, respon yang cepat dan diberi beban.

Metode yang banyak diterapkan untuk mengatur kecepatan motor arus searah adalah kontrol PID (*Proporsional Integral Derivatif*) diantaranya oleh [1]-[4]. Pada perancangan kontrol PID, harus dicari parameter konstanta proporsional (P), konstanta integral (τ_i), dan konstanta derivatif (τ_d) yang menentukan respon *transient* dan *steady state* sistem.

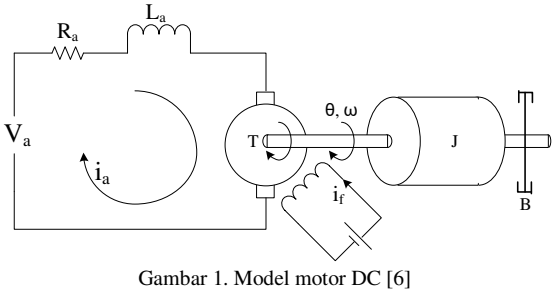
Ada beberapa cara dalam penetuan parameter konstanta PID. Pertama penentuan parameter tanpa harus mengetahui model matematis sistem. Dengan penalaan, dilakukan *tuning* nilai P, τ_i, τ_d kemudian diamati respon sistem sebagaimana [1]. Kemudian dengan metode Ziegler-Nichols dengan melihat tanggapan *transient open loop* sistem [4]. Penggabungan metode fuzzy logic dan PID dimana konstanta PID didapatkan dari proses logika fuzzy [5]. Kedua penentuan konstanta PID dengan mengetahui model sistem menggunakan metode *root locus* yaitu dengan mengetahui letak *pole* dan *zero* dari *open loop* sistem, kemudian dengan menambahkan *pole* dan *zero* agar *pole* dan *zero closed loop* sistem terletak pada titik/jalur yang baru sehingga respon *transient* dan *steady state* sistem yang diinginkan terpenuhi. Penelitian tentang ini telah dilakukan oleh [2] yang mengimplementasikan kendali PID dengan *root locus* pada mikrokontroler ATMega16 untuk mengatur kecepatan motor DC namun langkah dalam penentuan kriteria respon dan perancangan *root locus* langsung menggunakan Matlab tidak dijelaskan lebih detail. [6] merancang dan mensimulasikan kendali posisi motor arus searah pada Matlab menggunakan kendali PID dengan *root locus* namun bukan kecepatannya.

Perancangan PID menggunakan *root locus* dari model sistem mempunyai parameter respon (*settling time, rise time, maximum oveshoot, dumping ratio,* dsb) dan analisis kestabilan yang lebih jelas daripada penentuan parameter PID tanpa model sistem.

Agar perancangan kendali mempunyai parameter respon, analisis kestabilan yang jelas, maka artikel ini merancang skema pengendalian kecepatan motor arus searah dengan metode PID melalui *root locus* sistem. Pertama akan dibahas pemodelan motor arus searah. Kemudian akan dilihat *root locus* dari *pole* dan *zero* sistem mula-mula, kemudian dirancang penambahan *pole* dan *zero* agar *root locus* yang baru sesuai dengan respon yang diinginkan. Kemudian dilakukan simulasi analisis kestabilan dari sistem *closed loop* yang dirancang.

II. PEMODELAN MOTOR ARUS SEARAH

Untuk melakukan perancangan kendali dengan *root locus* kecepatan motor DC perlu dimodelkan berupa fungsi alih antara output kecepatan sudut dan input tegangan motor DC, ditunjukkan pada Gambar 1.



Gambar 1. Model motor DC [6]

Dimana:

L_a= induktansi kumparan jangkar
R_a= resistansi kumparan jangkar
i_a= arus kumparan jangkar (armature)
i_f = arus medan (field)
θ = perpindahan sudut dari poros ke motor
ω = kecepatan sudut dari poros ke motor
V_a = tegangan kumparan jangkar
V_{ggl} = tegangan gaya gerak listrik balik
J = momen inersia dari motor dan beban pada poros motor
B = koefisien gesek viskos ekivalen dari motor dan beban pada poros motor

Persamaan differensial pada rangkaian kumparan jangkar adalah:

$$V_a(t) - V_{ggl}(t) = L\frac{di_a(t)}{dt} + R_a i_a(t) \quad (1)$$

Ubah (1) dalam bentuk Laplace ($\mathcal{L}$) menjadi,

$$V_a(s) - V_{ggl}(s) = (Ls + R_a)I_a(s) \quad (2)$$

$$I_a(s) = \frac{1}{(L_a s + R_a)}\left[V_a(s) - V_{ggl}(s)\right] \quad (3)$$

Dari sisi mekanik, torsi yang dihasilkan motor berbanding lurus dengan hasil kali arus kumparan jangkar dan medan serta medan magnet yang dihasilkan penguat medan (K) yang dapat dituliskan,

$$T = Ki_f(t)i_a(t) \quad (4)$$

Karena arus medan i_f konstan, maka dalam Laplace i_f dapat dianggap bergabung dalam K_{TM} (Konstanta Torsi Motor),

$$T(s) = K_{TM} I_a(s) \quad (5)$$

Torsi yang dihasilkan motor bekerja terhadap inersia dan gesekan viskos, sehingga

$$T = J\frac{d\omega(t)}{dt} + B\omega(t) \quad (6)$$

Atau dalam Laplace,

$$T(s) = Js\Omega(s) + B\Omega(s) \quad (7)$$

$$\Omega(s) = \frac{1}{(Js + B)}T(s) \quad (8)$$

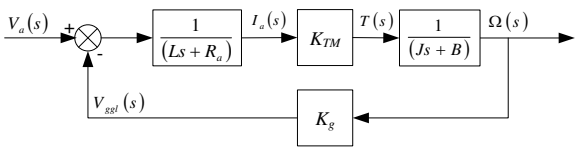
Besarnya tegangan gaya gerak listrik V_{ggl} adalah berbanding lurus dengan hasil kali dari arus medan dan kecepatan sudut motor. Sehingga,

$$V_{ggl} = K.i_f(t)\omega(t) \quad (9)$$

Karena arus medan i_f konstan, maka dalam Laplace i_f dapat dianggap bergabung dalam K_g, sehingga

$$V_{ggl} = K_g\Omega(s) \quad (10)$$

Dari (3), (5), (8), (10) dapat digabungkan menjadi sebuah diagram blok sebagaimana pada Gambar 2.



Gambar 2. Blok Diagram Model Motor DC

Dengan menyederhanakan blok diagram Gambar 2, maka diperoleh model fungsi alih orde-2 dengan output kecepatan sudut dan input tegangan jangkar,

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_{TM}}{L_a Js^2 + (L_a B + R_a J)s + (R_a B + K_{TM}K_g)} \quad (11)$$

Motor yang disimulasikan mempunyai paramater L_a= 0,5 H, R_a= 1 Ω, K_{TM}= 0,01 N.m/A, K_g= 0,01 V/rad/dtk, B = 0,1 N.m.s, J = 0,01 kg.m^2 . Sehingga model motor DC,

$$\frac{\Omega(s)}{V_a(s)} = \frac{0{,}01}{0{,}005s^2 + 0{,}06s + 0{,}1001} = \frac{2}{(s+10)(s+2)} \quad (12)$$

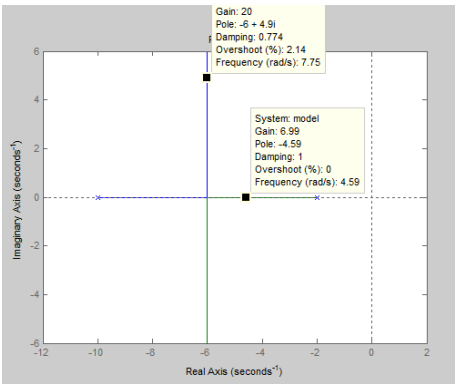
Nampak dari model bahwa semua *pole* terletak pada sebelah kiri sumbu imajiner, sehingga sistem stabil.

III. PERANCANGAN KENDALI *ROOT LOCUS*

A. Perancangan Kendali PID dengan *Root locus*

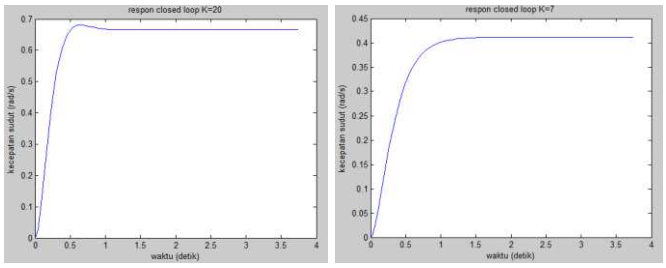
Motor DC dengan model (12) disimulasikan open dan closed loop dengan simulink Matlab mempunyai respon *open loop* untuk masukan *unit step* (1 Volt) diperoleh bahwa sistem mempunyai karakteristik *time constant* (τ)=0,61 dtk, *settling time* t_s 2% sebelum steady state pada 2,09 dtk (kriteria t_s ini yang akan digunakan dalam simulasi dan perancangan), ζ=1 (tanpa *overshoot*).

Pada problem *tracking*, agar sistem mengikuti *setpoint* kecepatan yang diinginkan, maka sistem dibuat *closed loop* dimana output diumpanbalikkan negatif. *Root locus* memberikan gambaran lintasan *pole-pole closed loop* sistem untuk tiap *gain* pengali K yang dikalikan pada sistem. *Root locus* dari sistem dengan persamaan (12) nampak pada Gambar 3.



Gambar 3. Respon *Closed Loop* Sistem Untuk Masukan Tangga Satuan

Untuk *gain* pengali K=7 dan K=20 respon closed loop sistem dengan masukan tangga kecepatan sudut yang diinginkan 1 rad/dtk dapat dilihat pada Gambar 4. Pada K=7 sistem mempunyai ω_n=4,59 rad/dtk, K_p=0,7 dengan karakteristik respon τ=0,22 dtk, t_s=0,87 dtk, tanpa *overshoot*, *error steady state* 0,588. Sedangkan untuk K=20, sistem mempunyai ω_n=7,75 rad/dtk, K_p=2, dengan karakteristik respon τ=0,17 dtk, t_s=0,66 dtk, ζ=0,774 *overshoot* =2,14%, *error steady state* 0,33.



Gambar 4. Respon closed loop sistem dengan perubahan *gain*

Nampak bahwa dengan memperbesar *gain K* dapat mengurangi *steady state error* (namun memperbesar *overshoot*) dan meningkatkan tanggapan transien (dengan ikut berubahnya ω_n). Namun hal ini tetap tidak dapat menghilangkan *steady state error*, dan apabila diinginkan tanggapan transien yang tidak dilewati *root locus*, dengan hanya memperbesar *K*, hal ini juga tidak akan tercapai.

Pada penelitian ini akan dirancang agar sistem mempunyai respon transien yang lebih cepat dari sistem mula-mula dengan t_s=0,8 dtk, *steady state error* = 0 dan maksimal *% overshoot* kurang dari 3%. Dari spesifikasi ini diperoleh,

$$\zeta = \sqrt{\frac{\ln^2\{os\}}{\pi^2+\ln^2\{os\}}} = \sqrt{\frac{\ln(0.03)}{\pi^2+\ln(0.03)}} = 0.748 \approx 0.75$$

$$t_s = \frac{4}{\zeta\omega_n} \quad \Rightarrow \quad \omega_n = \frac{4}{0{,}75.0{,}8} = 6{,}67 \text{ rad/detik}$$

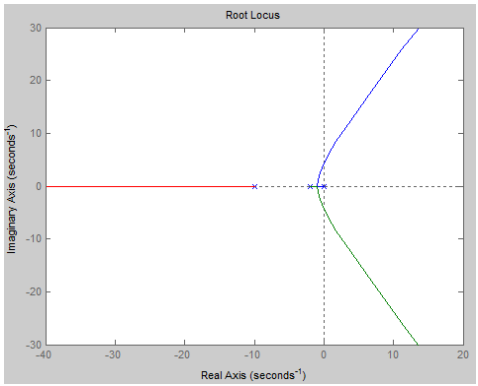
$$\omega_d = \omega_n\sqrt{1-\zeta^2} = 4{,}41 \text{ rad/detik}$$

Sehingga *pole* yang diinginkan harus melewati $s_{1,2}$=-5±*j*4,41.

Pertama untuk menghilangkan *steady state error*, ditambahkan integrator pada sistem. Sehingga model sistem berubah menjadi,

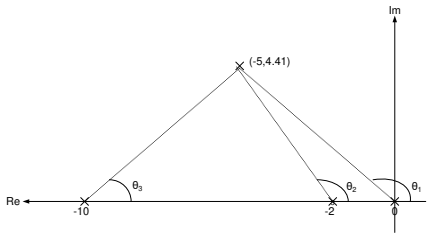
$$\frac{\Omega(s)}{V_a(s)} = \frac{2}{s(s+10)(s+2)} = \frac{0{,}01}{0{,}005\,s^3+0{,}06\,s^2+0{,}1s} \qquad (13)$$

Root locus sistem yang baru nampak pada Gambar 5.



Gambar 5. *Root locus* sistem dengan penambahan integrator

Agar *root locus* sistem dapat melewati *pole* yang diinginkan ($s_{1,2}$=-5±*j*4,41), perlu ditambahkan *zero* dengan nilai yang tepat.

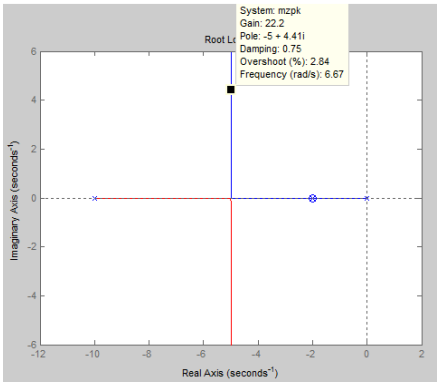


Gambar 6. Kedudukan *pole-pole* sistem terhadap *pole* yang diinginkan

Jika sudut yang dibentuk oleh *zero* adalah θ_4, maka jumlah sudut-sudut yang dibentuk oleh keempatnya adalah kelipatan 180^0.[7]

$$\theta_4 - \theta_3 - \theta_2 - \theta_1 = (2k+1)180^0 \qquad (14)$$

θ_4-41,41^0-124,23^0-138,59^0=(2*k*+1)180^0. θ_4=124,23^0. Sehingga *zero* harus diletakkan di -2. Sehingga *root locus* sistem dengan kompensator PID dengan menambahkan *pole* (integrator pada *origin*) dan *zero* (diferensiator) pada -2 adalah,



Gambar 7. *Root locus* sistem dengan PID

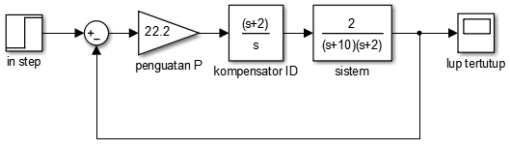
Nampak bahwa dengan penambahan *pole* dan *zero* tersebut, pada *gain* 22,2 *root locus* sistem akan berada pada *pole* yang diinginkan -5±*j*4,41, ζ=0,75, ω_n=6,67 rad/dtk.

Sehingga sistem keseluruhan dengan penambahan PID menjadi,

$$\frac{\Omega(s)}{V_a(s)} = \frac{44{,}4(s+2)}{s(s+10)(s+2)} = \frac{0{,}222\,s+0{,}444}{0{,}005\,s^3+0{,}06\,s^2+0{,}1s} \qquad (15)$$

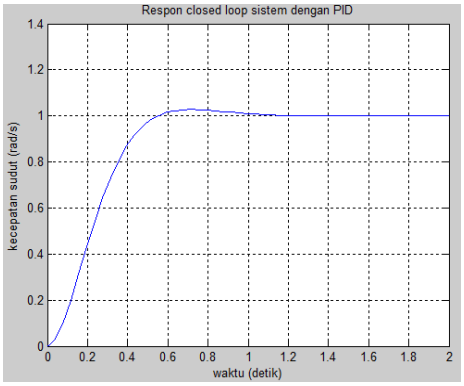
IV. HASIL DAN PEMBAHASAN

Hasil perancangan melalui *root locus* ini akan disimulasikan dengan simulink Matlab dengan blok diagram sebagaimana Gambar 8.



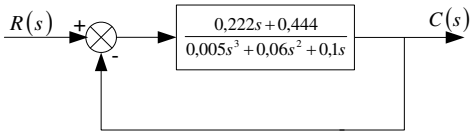
Gambar 8. Blok diagram simulasi PID pada Simulink

Hasil simulasi nampak pada Gambar 9. Dari simulasi dengan masukan tangga satuan, pada problem *tracking* diperoleh bahwa sistem yang diberi kompensasi PID mempunyai karakteristik tanggapan transien τ=0,27 dtk, t_s=0,835 dtk, maksimal *% overshoot* 2,83%, dan *steady state error*=0. Hasil ini sesuai dengan perancangan.



Gambar 9. Respon *step closed loop* sistem dengan PID

Selanjutnya perlu dilakukan uji kestabilan untuk mengetahui apakah sistem yang diberi PID tetap stabil. Untuk pengujiannya digunakan kriteria Routh Hurwitz. Sistem keseluruhan mempunyai blok diagram sebagaimana Gambar 10.



Gambar 10. Blok diagram sistem dengan PID

Dari blok diagram di atas diperoleh,

$$\frac{C(s)}{R(s)} = \frac{0{,}222\,s + 0{,}444}{0{,}005\,s^3 + 0{,}06\,s^2 + 0{,}322\,s + 0{,}444} \qquad (16)$$

Persamaan karakteristiknya,

$$0{,}005s^3+0{,}06s^2+0{,}322s+0{,}444$$

Akar-akar Routh

s^3	0,005	0,322
s^2	0,06	0,444
s^1	0,285	
s^0	0,444	

Dari akar-akar Routh nampak bahwa pada kolom pertama tidak terjadi perubahan tanda. Sehingga dapat disimpulkan bahwa sistem dengan kompensasi PID tetap stabil.

V. SIMPULAN DAN SARAN

Dari penelitian yang dilakukan dapat disimpulkan bahwa perancangan kendali kecepatan motor DC PID melalui metode *root locus* telah berhasil dilakukan dan disimulasikan melalui Simulink Matlab agar respon transien dan *steady state* sistem sesuai dengan yang diinginkan. Dan melalui analisis kestabilan Routh Hurwitz dapat dilihat bahwa sistem yang telah dikompensasi denggn PID menghasilkan sistem yang stabil.

Untuk penelitian selanjutnya, metode ini dapat diimplementasikan pada plant dan aktuator sesungguhnya. Untuk pengendali digital, dapat dilakukan analisis *root locus* pada ranah diskrit.

REFERENSI

[1] Waluyo, A. Fitriansyah, Syahrial, "Analisis Penalaan Kontrol PID pada Simulasi Kendali Kecepatan Putaran Motor DC Berbeban Menggunakan Metode Heuristik", *Jurnal Elkomnika Itenas*., *vol.*1 *no.*2, pp. 79-92, Des. 2013.

[2] M. R. Setiawan, M. A. Muslim, G. D. Nusantoro, "Kontrol Kecepatan Motor DC dengan Metode PID Menggunakan Visual Basic 6.0 dan Mikrokontroler ATmega 16", *Jurnal EECCIS., vol.*6 *no.*2, pp. 1-6, Des 2012.

[3] D. A. Surya,"Perancangan Kontroler PI *Anti-Windup* Berbasis Mikrokontroler ATmega 32 Pada Kontrol Kecepatan Motor DC", *Pros. Seminar Tugas Akhir Jurusan Teknik Elektro FTI-ITS*, 2011.

[4] Z. Jamal, "Implementasi Kendali PID Penalaan Ziegler-Nichols Menggunakan Mikrokontroler", *Jurnal Informatika., vol.*15 *no.*1, pp.81-88, Juni 2015.

[5] H. Hibban, J. Pramudijanto, N. Gamayanti, "Desain Kontroler *Fuzzy* PID *Gain Schedulling* Untuk Pengaturan Kecepatan Motor DC Tanpa Sikat", *Jurnal Teknik ITS vol.*4 *no.*2, pp 37-40, 2015.

[6] K. Sao, Mr. D. K. Singh, Mr. A Agrawal, "Study of DC Motor Position Control Using *Root locus* and PID Controller in MATLAB", *International Journal for Scientific Research & Development vol.*3 *issue* 5, pp 183-190, 2015.

[7] N. S. Nise, *Control System Engineering*, 6th ed., John Willey & Sons, Inc., 2011.