Seminar Nasional Instrumentasi, Kontrol dan Otomasi (SNIKO) 2018
Bandung, Indonesia, 10-11 Desember 2018

Sistem Pemantauan Waktu Nyata Menggunakan Labview Pada Pengendalian Kecepatan Motor BLDC Berbasis PID-Root Locus

[1]Ihsan Auditia Akhinov, [2]Robith Urwatal Wusko & [1]Nurul Fahmi Arief Hakim

[1]Teknik Elektronika, Politeknik Gajah Tunggal.

Komplek Industri Gajah Tunggal, Jalan Gatot Subroto KM. 7, Pasir Jaya, Pasir Juga, Gandasari, Jatiuwung, Kota Tangerang, Banten,15001.

[2]Direktorat Metrologi, Kementerian Perdagangan Republik Indonesia.

Jl. Pasteur No.27, Pasir Kaliki, Cicendo, Kota Bandung, Jawa Barat 40171

ihsan@poltek-GT.ac.id, robith.u.w@gmail.com, nurulfahmiarief@gmail.com

Abstrak

Penelitian ini membahas tentang teknologi BLDC yang banyak digunakan pada kendaraan listrik seperti motor listrik, sepeda listrik dan juga mobil listrik. Hal ini tidak lepas dari perawatan yang murah, torsi yang besar dan tingkat kebisingan yang dihasilkan oleh motor BLDC sangat kecil. Untuk implementasi BLDC pada bidang transportasi diperlukan pengendalian kecepatan putaran motor. Pengendalian kecepatan pada paper ini didekati dengan metode PID . Parameter metode PID didapatkan berdasarkan analisis root locus. Hasil dari parameter tersebut ditanamkan pada atmega2560 dan dipantau menggunakan Labview. Hasil penelitian didapatkan bahwa pengendalian kecepatan motor BLDC mencapai peaktime 0.22 detik dan overshoot 2.2%.
Kata Kunci: BLDC,Root Locus, PID,Labview.

1 Pendahuluan

Motor Brushless Direct Current (BLDC) telah banyak digunakan pada kendaraan berbasis elektrik, seperti sepeda listrik, sepeda motor listrik. Hal ini tidak lepas dari kemampuan BLDC dalam menghasilkan torsi yang besar serta perawatan yang murah ketimbang dengan motor DC biasa dengan sikat[1][2][3].

Untuk pengoperasian BLDC dibutuhkan rangkaian inverter 3 fasa dan microcontroller yang mengatur *timing* komutasi gelombang input. *Timing* komutasi ini dilakukan berdasarkan posisi kutub yang akan dibaca oleh 3 sensor hall, dengan adanya sistem *timing* komutasi yang tepat, maka sinyal masukan pada BLDC mempunyai beda fasa 60 derajat. Proses inilah yang menjadi pembangkit sinyal trapezoid 3 fasa untuk motor BLDC.

Untuk menjaga kecepatan pada motor BLDC sesuai dengan referensi masukan diperlukan penambahan sistem kendali. Dalam hal ini penulis menggunakan metoda PID yang didapatkan menggunakan *root locus*.

Pengamatan dilakukan secara realtime pada software Labview. Untuk proses persamaan sistem didapatkan menggunakan sistem identifikasi yang disediakan oleh MATLAB dan pengaturan *timing* komutasi dilakukan oleh atmega2560.

2 Diskusi

2.1 Dasar Teori

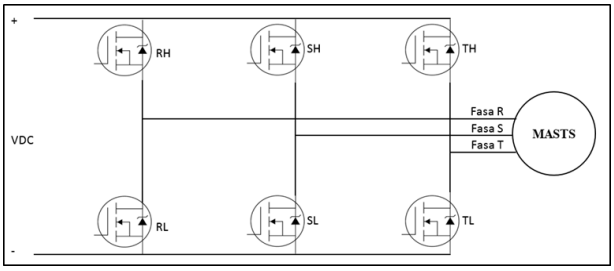
Untuk teori yang digunakan pada perancangan sistem kendali PID pada motor BLDC ini terdiri dari BLDC, PID dan Root Locus.

2.1.2 Motor BLDC

Pada BLDC tidak terdapat sikat seperti pada motor DC biasa, oleh karena itu pada motor ini untuk menentukan *timing* komutasi dengan bantuan 3 sensor hall. Pada sensor Hall, *timing* komutasi ditentukan dengan cara mendeteksi medan magnet rotor dengan menggunakan 3 buah sensor hall untuk mendapatkan 6 kombinasi yang berbeda. Untuk tabel komutasinya dapat dilihat pada Tabel 1 dan gambar rangkaian dapat dilihat pada Gambar 1 [1][2][3][4].

Tabel 1 Kombinasi pensaklaran dengan 2 saklar terhubung untuk motor dan sensor hall

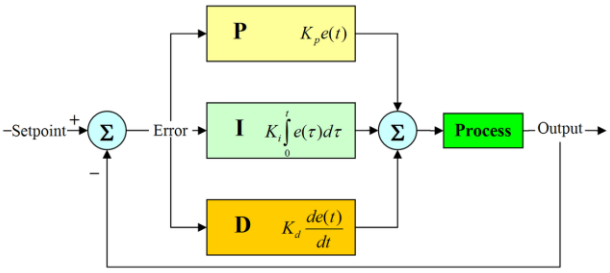
No	ha	hb	hc	Keluaran Mosfet					
				RL	RH	SL	SH	TL	TH
0	0	1	0	1	0	0	1	0	0
1	1	1	0	1	0	0	0	0	1
2	1	0	0	0	0	1	0	0	1
3	1	0	1	0	1	1	0	0	0
4	0	0	1	0	1	0	0	1	0
5	0	1	1	0	0	0	1	1	0



Gambar 1 Saklar Inverter 3 fasa.

2.1.2 PID

Sistem PID terdiri dari Pengendali proposional, integral dan derivatif. Hal ini dapat dilihat pada Gambar 2.



Gambar 2 Kontrol PID.

Parameter nilai Kp, Ki dan Kd berpengaruh terhadap respon sistem. Hal ini dapat dilihat pada Tabel 2 [4][5].

Tabel 2 Respon Sistem Terhadap Perubahan Parameter

Parameter	Waktu Naik	Overshoot	Settiling time	Error keadaan tunak
Propotional (Kp)	-	+	Perubahan sedikit	-
Integral (Ki)	-	+	+	hilang
Derrivative (Kd)	Perubahan sedikit	-	-	Perubahan sedikit

Note : + = bertambah

- = berkurang

Untuk persamaan PID pada sinyal analog dapat dilihat pada [6][7] :

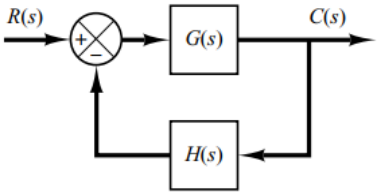
$$C(t) = K[e(t) + \frac{1}{T_i}\int e(t)\,dt + T_d\frac{de(t)}{dt}] \quad (1)$$

Untuk sinyal digital :

$$C(z) = K.e(z)[1 - \frac{T}{2T_i} + \frac{T}{T_i}.\frac{1}{1-z^{-1}} + \frac{T_d}{T}.(1-z^{-1})] \quad (2)$$

2.1.3 Root Locus

Jika sistem closed loop pada Gambar 3 [6] .

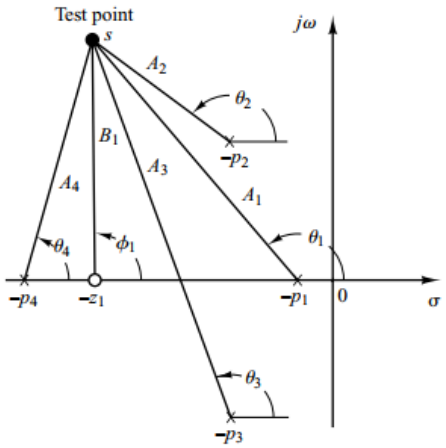


Gambar 2 Closed Loop sistem.

Mempunyai fungsi alih :

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)} \quad (3)$$

Jika dilihat persamaan karakteristiknya maka jumlah sudutnya terhadap satu titik uji adalah



Gambar 3 S-Plane Root Locus.

$$\angle G(s)H(s) = \phi_1 - \theta_1 - \theta_2 - \theta_3 - \theta_4 \quad (4)$$

Sedangkan fungsi alih yang akan digunakan pada motor BLDC didekati untuk Orde 2.

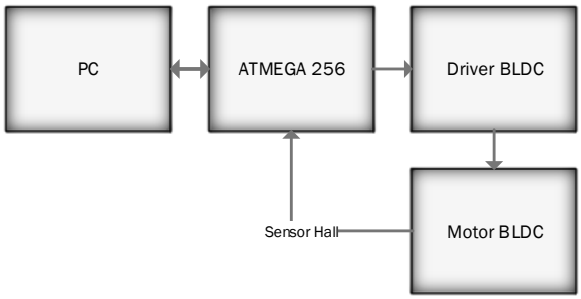
$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5)$$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} \quad (6)$$

2.2 Perancangan

2.2.1 Hadware

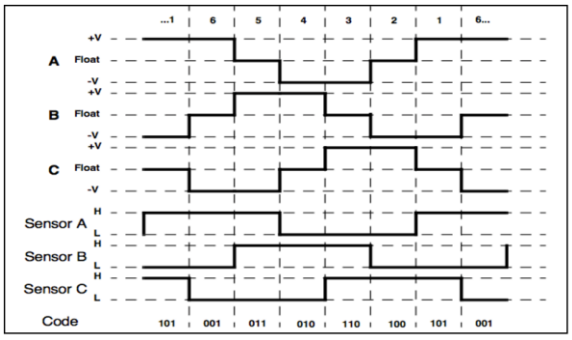
Untuk pengoperasian motor BLDC melalui perangkat keras, dapat dilihat pada Gambar 4 tentang *block diagram* pengoperasian motor BLDC.



Gambar 4 Diagram Pengoperasian Motor BLDC.

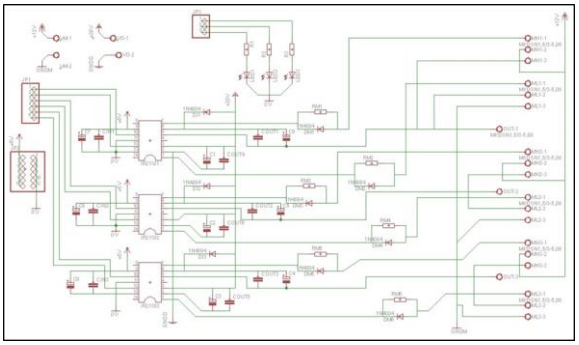
Pada PC akan terdapat software Labview yang berguna sebagai monitoring dan perekaman kecepatan motor BLDC, dan juga masukan kecepatan referensi. Sistem Komunikasi serial digunakan sebagai penghubung antara Labview dan atmega 2560, baud rate yang digunakan adalah 2.000.000.

Atmega2560 sebagai pengatur *timing* komutasi sehingga sinyal yang dihasilkan menggerakkan motor BLDC.



Gambar 5 Kombinasi komutasi berdasarkan kondisi sensor hall dan output tegangan yang dihasilkan.

Untuk driver motor BLDC dapat dilihat pada Gambar 6. Pada rangkaian ini menggunakan komponen utama mosfet dan optocoupler.



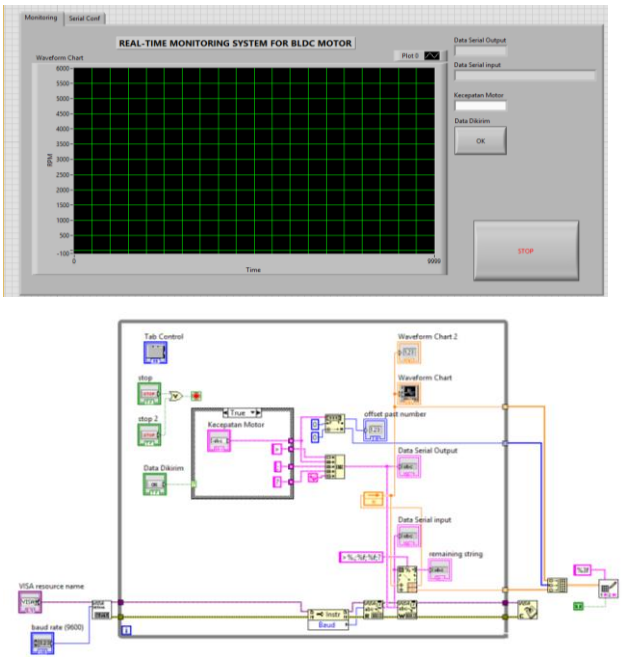
Gambar 6 Driver Motor BLDC.

2.2.2 Perangkat Lunak

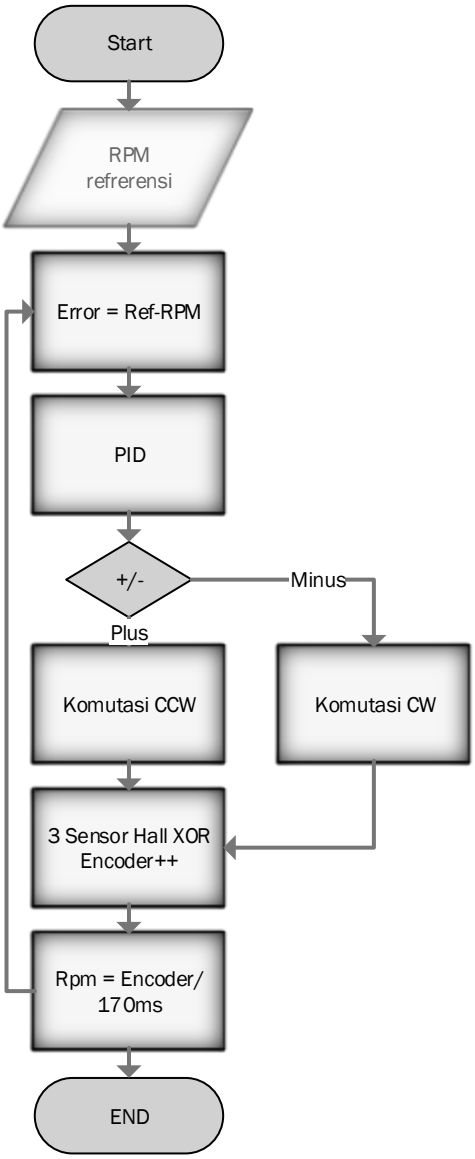
Implementasi yang pernah dilakukan menggunakan Labview adalah untuk mengendalikan kecepatan putaran motor DC menggunakan metode Fuzzy Inference System pada mesin cuci industri tekstil. Sistem tersebut memanfaatkan protocol VISA Labview sebagai penghubung sistem akuisisi data dan Komputer [8]. Konfigurasi inilah yang akan diterapkan untuk penilitian ini.

Untuk Tampilan Labview dapat dilihat pada Gambar 7. Sedangkan flowchart pemograman pengendali dapat dilihat pada Gambar 8.

Pada pegendalian PID akan terdapat Komutasi CCW dan CW hal ini menandakan ketika sinyal pengendalian bernilai positif komutasi BLDC untuk orientasi putar berlawanan arah jarum jam dan ketika sinyal pengendali bernilai negatif maka komutasi yang dilakukan kebalikannya.



Gambar 7 Labview Program.



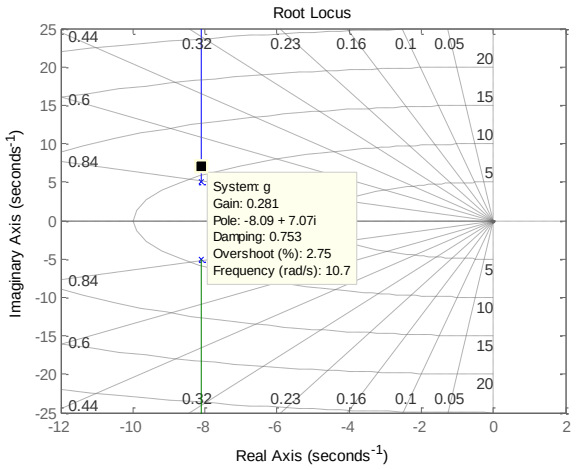
Gambar 8 Flowchart Program Pengendali.

2.2.3 Perancangan PID dengan Root Locus

Pertama-tama hal yang dilakukan dalam perancangan pengendali PID adalah menetukan persamaan matematis dari sistem motor BLDC. Pada penelitian ini persamaan tersebut didapatkan dari sistem identifikasi signal yang terdapat pada MATLAB. Jadi data signal input dan respn output motor BLDC diolah menjadi persamaan matematis dengan bantuan toolbox MATLAB. Sedangkan persamaan didekati dengan sistem orde dua setara dengan motor DC biasa. Persamaanya adalah

$$\frac{V_o(s)}{V_i(s)} = \frac{89.81}{s^2 + 16.18s + 90.24} \quad (7)$$

Setelah itu dilakuan pengamatan rootlocusnya



Gambar 9 Root Locus.

Selanjutnya dilakukan pemilihan konstanta damping. Untuk damping 0.753 didapatkan posisi pole -8.09+7.07i dengan nilai *gain* 0.281 serta besar θ adalah 138.849. Jika dilakukan analisa respon *closed loop*, Maka didapatkan parameter peaktimenya adalah 0.4441 detik. Jika dirancang sebuah PID dengan nilai *peaktime* 2/3 dari nilai sebelumnya kita akan mendapatkan persamaan Kp, Ki dan Kd. Sebagai berikut:

$$G(s) = \frac{K.(s+a)(s+b)}{s} * \frac{89.81}{s^2 + 16.18s + 90.24}$$

pole imajiner/ ω_d :

$$= \frac{\pi}{\frac{2}{3}.0.0441} = -10.6075$$

Pole realnya :

$$= \frac{\omega_d}{tan\ \theta} = -12.137$$

Jika Pole yang pasti terdiri dari :

$$(s)(s + 8.09 - 4.98i)(s + 8.09 + 4.98i)$$

Maka jika digunakan persamaan 4 didapatkan nilai

$$\phi_1 + \phi_2 = 189.12$$

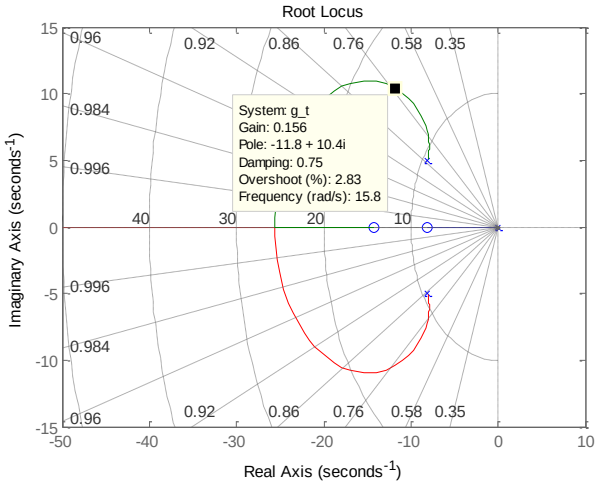
Jika ϕ_2 adaah 110 maka ϕ_1 = 79.1215 dengan menggunakan persaman phytagras untuk mencari zero yang tersisa akan didapatkan nilai :

$$(s + 14.175)(s + 8.139)$$

Sehingga persamaan G(s):

$$G(s) = \frac{K.(s + 14.175)(s + 8.139)}{s} * \frac{89.81}{s^2 + 16.18s + 90.24}$$

Untuk mencari nilai Gain setelah penambahan cntroler PID dapat menggunakan rot locus kembali :



Gambar 10 Root Locus dengan PID.

Maka akan didapatkan nilai K= 0.16. Selanjut persaman tersebut diubah menjadi fungsi continiu

$$C(t) = 3.57[e(t) + \frac{1}{0.193}\int e(t)\,dt + 0.00448\frac{de(t)}{dt}]$$

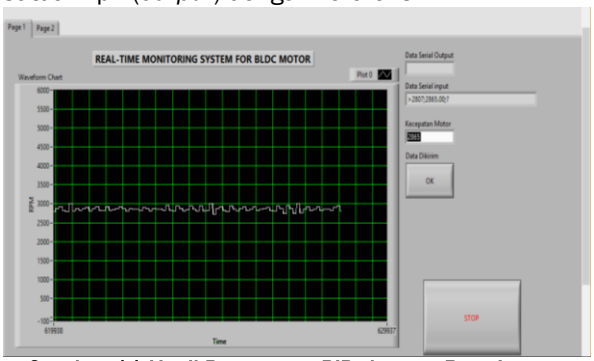
Dibuah menjadi persamaan diskrit dengan time sampling 1.4 ms :

$$C(z) = 3.57.e(z)[0.996 + \frac{0.025}{1-z^{-1}} + 11.424.(1-z^{-1})]$$

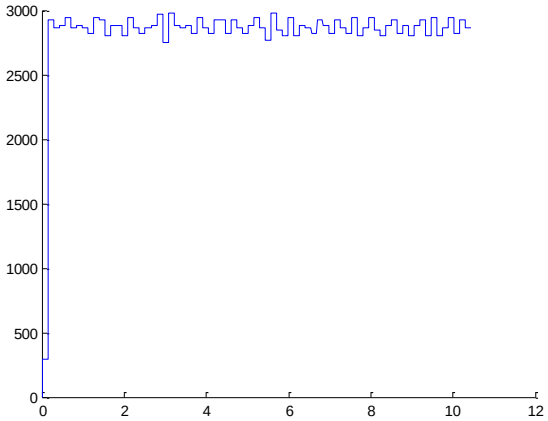
Persamaan disktir inilah yang akan ditanamkan kedalam atmega 2560 yang mana untuk:
- Proposinal : error dikalikan 3.55
- Integral : jumlah error sekarang dan sebelumnya dikalikan dengan 0.025
- Derivative : Selisih error sekarang dan error sebelumnya dikalikan 11.424.

2.3. Hasil

Untuk hasil penerapan PID dapat dilihat pada Gambar 11 ,yang terdiri dari sinyal keluaran dalam satuan rpm(*output*) dengan referensi.



Gambar 11 Hasil Penerapan PID dengan Root Lccus dengan referensi 2865 rpm.

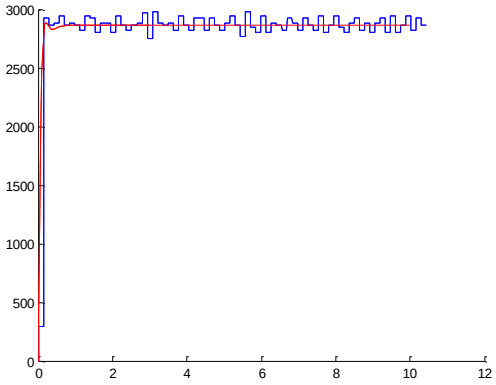


Gambar 12 Grafik Hasil Penerapan PID dengan Root Locus dengan referensi 2865 rpm.

Tabel 3 Respon sistem dengan PID

Parameter	Angka	Satuan
Peaktime	0.222	Detik
Overshoot	2.024	%

2.4. Analisa



Gambar 13 Perbandingan Sinyal Respon Simulasi dengan Aktual.

Hasil simulasi menggunakan matlab didapatkan nilai *peaktime* 0.211 . Jika dibandingkan dengan nilai *peaktime* pada perancangan seharusnya adalah 0.296. Hal ini terjadi dikarena pembulatan angka dalam perhitungan. Sehingga ada pergeseran sebesar 0.085 detik. Akan tetapi nilai yang aktual *peaktime* dibandingkan dengan simulasi mempunya selisih 0.01 detik. Hal ini bisa disebabkan waktu siklus suatu program pada atmega2560 tidak bersifat konstan. Sehingga dapat dapat menyebabkan pergeseran hasil dari seharusnya. Waktu siklus yang diambil dalam penentua PID-diskritnya adalah waktu siklus rata-rata.

Tabel 4 Hasil Simulasi

Parameter	Angka	Satuan
Peaktime	0.211	Detik
Overshoot	0.97	%

Untuk overshoot yang terjadi sebesar 2.024%. Lebih besar 1.1% dari simulasi. Hal ini dikarenakan resolusi kecepatan(rpm) yang besar, karena hasil jumlah encoder dalam satu putaran yang didapatkan dari logika XOR berjumlah 12. Terdiri dari 6 kondisi *on*, dan 6 kondisi *off*. Sehingga resolusi yang dihasilkan sangat besar. Besarnya nilai resolusi tersebut akan berpengaruh ke nilai error yang cukup besar ketika kecepatan mengalami osilasi disekitar referensi acuan. Nilai error yang cukup besar tersebut dikalikan dan diakumulasikan oleh algoritma PID yang menyebabkan sinyal pengendali yang dihasilkan cukup besar dalam rentang referensi acuan.

3 Kesimpulan

Hasil dari penerapan PID dengan Root locus adalah 0.222 detik untuk mencapai puncak dan 2.024% untuk nilai *overshoot*-nya. Perancangan PID dengan rootlocus dapat mengacu pada *peaktime* dikarena jika menginginkan respon sinyal yang cepat, dapat didekati dengan membuat nilai *peaktime*-nya menjadi lebih kecil sehingga nilai *rise time* dapat menjadi lebih cepat juga.

Semakin cepat perancangan peaktime yang dibuat maka akan berpengaruh pada nilai *overshoot* dan juga *settling time*.

Untuk banyak encoder dalam satu putaran akan berpengaruh terhadap resolusi kecepatan putar yang dihasilkan. Nilai resolusi yang kecil akan membuat keakuratan nilai error sehingga pengendalian menjadi lebih baik.

4 Nomenklatur

K = Gain

Tp = Peaktime

Vi = Tegangan input

Vo = Tegangan output

ϕ = Sudut Zero

θ = Sudut Pole

ω_d = Damped natural frequency

ω_n = Undamped natural frequency

ξ damped

5 Daftar Pustaka

[1] R. Urwatal Wusko, I. Munawar, “Desain dan Impementasi Extended PC-104 Sebagai Modul Pendukung Perancangan Sistem Pengendalian Pengoperasian Motor Arus Tanpa Sikat Tanpa Sensor Menggunakan Frekuensi Tangga” Seminar Nasional IPTEK Penerbangan dan Antariksa XX-2016.

[2] M.P. Chavhan, S.M.Shinde, “Modeling and Simulation of a Controller of Brushless DC Motor for Electric Vehicle Application” National Conference on Innovative Trends in Science and Engineering (NC-ITSE'16) Volume: 4 Issue: 7,2016.

[3] Z.Y. Pan, F.L. Luo, “Steady state reference current determination technique for brushless DC motor drive system”, IEE Proc.-Elec. Pow.

[4] M. Nasri, H. Nezamabadi-pour, M. Maghfoori, “A PSO-Based Optimum Design of PID Controller for Linear Brushless DC motor”, World Academy of Science, Engineering and Technology International Journal of Electrical and Information Engineering Vol:1, No:2, 2007.

[5] M.Ali, “Pembelajaran Perancangan Sistem Kontrol PID dengan Software Matlab “,Jurnal Edukasi@Elektro Vol. 1, No. 1, Oktober 2004.

[6] K. Ogata, “Modern Control Engineering Fifth Edition,” Prentice Hall, 2010.

[7] K. Ogata, “Discrete-Time Control Systems Second Edition,” Prentice Hall, 1995.

[8] D. Handaya, Y.P. Nugraha, I.A. Akhinov, P. Bakti, A. Adiwilaga. “Simulasi Mesin Cuci Industri Tekstil Berbasis Kendali Fuzzy dan Interface LabVIEW 2014”. Simposium Inovasi dan Pembelajaran Sains, FMIPA ITB,2016.