

PENGGUNAAN TEKNIK MODEL *REFERENCE ADAPTIVE CONTROL* PADA TUNING PENGONTROL PI PADA SISTEM KONTROL KECEPATAN SEPEDA

Mochammad Hesa Ibrahim.¹, Erni Yudanings², Rahmadwati.²

¹Mahasiswa Teknik Elektro Universitas Brawijaya, ²Dosen Teknik Elektro Universitas Brawijaya

Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya

Jalan MT. Haryono 167, Malang 65145, Indonesia

E-mail: mochammadhesaibrahim@gmail.com

ABSTRAK

Saat ini, penelitian dan pengembangan sepeda listrik (*electric vehicle*) sudah menjadi pusat perhatian bagi kalangan industri. Sudah banyak kendaraan listrik yang diproduksi secara masal di dunia. Beberapa dari kendaraan listrik tersebut menggunakan motor *Brushless Direct Current* (BLDC) sebagai penggerak utamanya. Motor BLDC memiliki efisiensi yang tinggi dan respon yang cepat, namun masih memiliki *error steady state* (offset). Oleh karena itu dibutuhkan suatu kontroler yang tepat dan sesuai agar tidak memiliki *error steady state*. Kontrol kecepatan dengan menggunakan Kontrol PI dapat menghilangkan *error steady state* pada respon motor, namun berdampak pula pada kecepatan respon yang lambat dalam mencapai *steady state*. Diantara teknik perancangan sistem kontrol adalah dengan menggunakan teknik perancangan kontrol adaptif, yaitu *Model Reference Adaptive Control* (MRAC) yang memiliki ide dasar untuk membuat respon sistem yang dikontrol agar dapat menyerupai perilaku yang sama dengan model referensi. MRAC yang disertai dengan parameter kontrol digunakan untuk pengontrolan kecepatan respon motor BLDC berdasarkan model *plant*. Respon *plant* setelah diberi kontroler dapat meminimalkan *error steady state*.

Kata Kunci—motor BLDC, kontrol kecepatan, kontroler PI, MRAC.

ABSTRACT

Nowadays, research and development about electric vehicle has become the center of attention for industrial circle. There has been many electric vehicle which has produced massively around the world. Some of these electric vehicle use *Brushless Direct Current* (BLDC) motor for its prominent activator. BLDC motor has high efficiency and fast responses, but still has *error steady state*. Therefore a suitable and appropriate controller is needed to omit the *error steady state*. A speed control with PI controller can omit the *error steady state* in motor responses, but has an effect on response speed which slow on reaching *steady state*. Among the control system plan is with the use of *Model Reference Adaptive Control* (MRAC) which has a basic idea for making system responses that controlled in order to has a similar behavior with the reference model. MRAC which come along with control parameter used to controlling the response speed of BLDC motor based on *plant* model. *Plant* responses after added with controller can minimize the *error steady state*.

Keyword: BLDC motor, speed control, PI controller, MRAC.

I. PENDAHULUAN

Saat ini, seperti kita telah ketahui bersama bahwa cadangan dan persediaan bahan bakar fosil dari tahun ke tahun cenderung menipis. Kabar ini tentu saja menjadi hal yang buruk bagi dunia otomotif yang mengandalkan konsumsi bahan bakar fosil yang saat ini menjadi sumber utama penggerak kendaraan bermotor. Tentu saja, hal ini mendorong umat manusia untuk mencari bahan bakar alternatif dan lebih efisien dibandingkan bahan bakar fosil. Saat ini, telah banyak ditemukan berbagai macam kendaraan bermotor berbasis energi alternatif yang lebih murah dan efisien dibandingkan bahan bakar fosil. contohnya adalah sepeda listrik yang saat ini diproyeksikan menjadi kendaraan masa depan.

Kendaraan listrik merupakan kendaraan yang menggunakan murni energi listrik sebagai penggerak. Beberapa dari kendaraan listrik tersebut menggunakan motor *Brushless Direct Current* (BLDC) sebagai penggerak utamanya. Motor BLDC memiliki respon yang cepat, torsi awal yang tinggi dan kecepatan yang tinggi, namun masih memiliki *error steady state* (offset).

Diperlukan upaya agar respon motor BLDC memiliki *error steady state* dan *overshoot* yang minimum.

Kontrol kecepatan motor BLDC dengan menggunakan kontroler *Proportional Integral* (PI) dapat menghilangkan *error steady state* pada respon motor, namun berdampak pada kecepatan respon yang lambat dalam mencapai nilai *steady state*^[1].

Banyak teknik perancangan sistem yang dapat diterapkan untuk mendesain suatu sistem, antara lain teknik perancangan dengan kontrol adaptif. Hal ini terutama untuk mengeliminasi gangguan (*disturbances*) dari luar, mengeliminasi gangguan (*disturbances*) dari dalam (perubahan parameter sistem atau pada sistem yang bekerja di luar daerah linier) dan mengatasi keterbatasan perancangan klasik yang umumnya sukar direalisasikan. Di dalam kehidupan sehari-hari, adaptasi diartikan mengubah perilaku untuk menyesuaikan dengan lingkungan baru. Secara intuisi, sebuah kontroler adaptif adalah sebuah kontroler yang mampu memodifikasi perilaku responnya untuk mengubah dinamika proses dan karakter gangguan (*disturbances*)^[2].

Banyak sekali teknik perancangan sistem kontrol dengan kontrol adaptif, beberapa diantaranya adalah

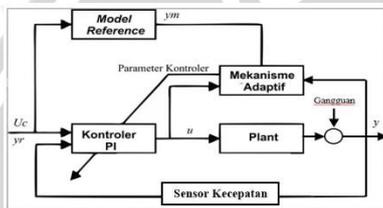
Model Reference Adaptive Control (MRAC). MRAC memiliki ide dasar untuk membuat respon sistem yang dikontrol agar dapat menyerupai perilaku yang sama dengan model referensi yang diberikan^[3]. Sehingga diharapkan dengan penambahan model referensi, respon yang lambat tersebut dapat dihilangkan.

Pada skripsi ini pengontrolan kecepatan motor BLDC dengan menggunakan kontrol MRAC dalam tuning PI bertujuan untuk meminimalkan overshoot agar tidak terjadi lonjakan pada starting motor dan dapat meminimalkan error steady state agar tidak terjadi lonjakan arus yang dapat memperpendek usia kontroler dan baterai.

II. METODE PENELITIAN

1. Diagram Blok Sistem

Perancangan diagram blok (lihat Gambar 1) sistem bertujuan agar dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana.



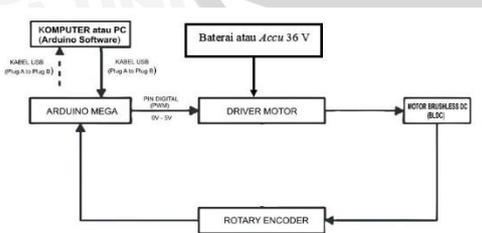
Gambar 1 Diagram blok sistem dengan *Model Reference Adaptive Control* (MRAC)

2. Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya, hal ini dimaksudkan agar sistem kontrol kecepatan motor BLDC dengan kontrol adaptif dapat berjalan sesuai deskripsi awal yang telah direncanakan.

Pembuatan perangkat keras yang dilakukan meliputi:

1. Skema pembuatan perangkat keras (Gambar 2).
2. Penentuan modul elektronik yang digunakan meliputi:
 - Komputer atau PC.
 - Baterai atau Accu 36 V.
 - Mikrokontroler Arduino Mega 2560.
 - Driver motor Inverter 3 Fasa.
 - Motor *Brushless Direct Current* (BLDC).
 - Rotary encoder (DI- Rotary encoder).



Gambar 2 Skema pembuatan perangkat keras

3. Desain Kontroler PI Menggunakan Teknik MRAC

Fungsi alih dari motor BLDC merupakan orde dua, maka fungsi alih motor BLDC adalah

$$\frac{Y(s)}{U(s)} = \frac{b}{s^2 + a_1s + a_2} \quad (1)$$

Sebuah sistem lup tertutup yang terdiri atas blok fungsi alih kontroler PI dan fungsi alih motor BLDC (Persamaan 1) dengan *feedback* adalah 1 disederhanakan menjadi sebuah sistem lup terbuka akan didapatkan Persamaan 2.

$$\frac{Y(s)}{U_c(s)} = \frac{bK_p s + bK_i}{s^3 + a_1s^2 + (a_2 + bK_p)s + bK_i} \quad (2)$$

Model referensi ditentukan melalui derajat relatif sistem. Derajat relatif model referensi sistem harus sama dengan derajat relatif *plant*. Derajat relatif adalah selisih antara derajat *polinomial pole* dengan derajat *polinomial zero*. Orde model referensi disesuaikan dengan orde *plant*^[4], sehingga model referesi dapat diasumsikan dalam bentuk fungsi alih orde 3 pada Persamaan 3.

$$\frac{Y_m(s)}{U_c(s)} = \frac{\beta s + \alpha_3}{s^3 + \alpha_1s^2 + \alpha_2s + \alpha_3} \quad (3)$$

Dimana:

$$\beta = bK_p, \quad \alpha_1 = a_1, \quad \alpha_2 = a_2 + bK_p, \quad \alpha_3 = bK_i$$

Telah diketahui bahwa perubahan parameter kontroler terhadap waktu dinyatakan dalam model pendekatan gradien, dimana *error* didefinisikan sebagai selisih antara respon *plant* y dan respon model referensi y_m ($e = y - y_m$). Sehingga memungkinkan untuk didapatkan parameter kontroler K_p dan K_i menerapkan aturan MIT $\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}$ dengan θ didapat dari penurunan Persamaan 4 dan 5.

$$\frac{\partial e}{\partial \theta} = \frac{\partial (y - y_m)}{\partial \theta} \quad (4)$$

$$\frac{d\theta}{dt} = s\theta = -\gamma e \frac{\partial e}{\partial \theta} \quad (5)$$

Dari Persamaan 4 dan 5 didapatkan bahwa

$$\theta = \frac{-\gamma}{s} e \frac{\partial e}{\partial \theta} \quad (6)$$

Dengan mengganti θ dengan K_p dan K_i

$$\frac{dK_p}{dt} = -\gamma_p \left(\frac{\partial J}{\partial K_p} \right) = -\gamma_p \left(\frac{\partial J}{\partial e} \right) \left(\frac{\partial e}{\partial y} \right) \left(\frac{\partial y}{\partial K_p} \right) \quad (7)$$

$$\frac{dK_i}{dt} = -\gamma_i \left(\frac{\partial J}{\partial K_i} \right) = -\gamma_i \left(\frac{\partial J}{\partial e} \right) \left(\frac{\partial e}{\partial y} \right) \left(\frac{\partial y}{\partial K_i} \right) \quad (8)$$

Dimana

$$e = y - y_m \Rightarrow \frac{\partial e}{\partial y} = 1$$

$$\frac{\partial J}{\partial y} = e$$



Dengan menggunakan hubungan Persamaan 7 dan 8 didapatkan

$$\frac{dK_p}{dt} = -\gamma_p \left(\frac{\partial J}{\partial K_p} \right) = -\gamma_p e \left(\frac{\partial y}{\partial K_p} \right) \quad (9)$$

$$\frac{dK_i}{dt} = -\gamma_i \left(\frac{\partial J}{\partial K_i} \right) = -\gamma_i e \left(\frac{\partial y}{\partial K_i} \right) \quad (10)$$

Cara agar mendapatkan $\frac{\partial y}{\partial K_p}$ dan $\frac{\partial y}{\partial K_i}$ dengan mendiferensialkan Persamaan 2, sehingga

$$\frac{dK_p}{dt} = -\gamma_p e \frac{bs}{s^3 + a_1s^2 + (a_2 + bK_p)s + bK_i} (u_c - y) \quad (11)$$

$$\frac{dK_i}{dt} = -\gamma_i e \frac{b}{s^3 + a_1s^2 + (a_2 + bK_p)s + bK_i} (u_c - y) \quad (12)$$

Dari Persamaan 11 dan 12, *gain parameters* tidak dapat ditemukan ketika nilai a_1 , a_2 dan b tidak diketahui. Jadi rumus yang diturunkan dari aturan MIT ini belum dapat digunakan. Sebagai gantinya, beberapa pendekatan diperlukan. Jika berfikir sistem merupakan model yang sempurna, bandingkan hubungan antara input-output dari sistem dengan model referensi^[5], Persamaan 13 merupakan pendekatan yang akan digunakan.

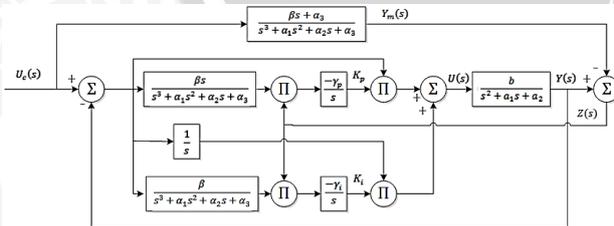
$$\begin{cases} \beta = b \\ \alpha_1 = a_1 \\ \alpha_2 = bK_p + a_2 \\ \alpha_3 = bK_i \end{cases} \quad (13)$$

Sehingga didapatkan Persamaan 14 dan 15.

$$K_p = \frac{-\gamma_p}{s} e \frac{\beta s}{s^3 + \alpha_1s^2 + \alpha_2s + \alpha_3} (u_c - y) \quad (14)$$

$$K_i = \frac{-\gamma_i}{s} e \frac{\beta}{s^3 + \alpha_1s^2 + \alpha_2s + \alpha_3} (u_c - y) \quad (15)$$

Sebuah representasi diagram dari skema adaptif berdasarkan penurunan rumus diatas dapat dilihat pada Gambar 3.



Gambar 3 Diagram blok penggunaan teknik MRAC pada tuning kontroler PI

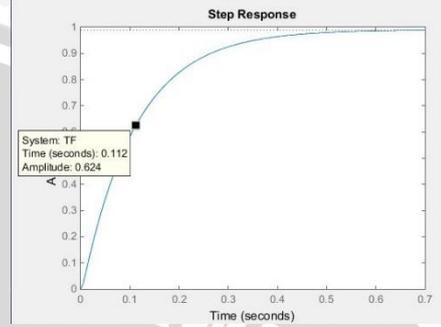
4. Penentuan Fungsi Alih Motor BLDC

Fungsi alih motor BLDC didapatkan dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS) pada mikrokontroler yang kemudian menjadi sinyal masukan motor BLDC. Dari data PRBS dan kecepatan motor BLDC, selanjutnya dilakukan identifikasi dengan menggunakan sintaks *ident* pada *software* Matlab.

Sehingga didapatkan fungsi alih motor BLDC yang memiliki *best fits* sebesar 82.49 pada Persamaan 16.

$$\frac{Y(s)}{U(s)} = \frac{2811}{s^2 + 318,6s + 2838} \quad (16)$$

Dengan memberikan masukan *unit step* pada program Matlab didapatkan respon pada Gambar 4, dimana *time constant* fungsi alih adalah 0.122 seconds.



Gambar 4 Respon *unit step* pada fungsi alih

5. Validasi Fungsi Alih Motor BLDC

Validasi fungsi alih motor BLDC dilakukan dengan cara membandingkan respon fungsi alih dan respon kecepatan motor BLDC yang dapat dari pembacaan *rotary encoder* dengan memberikan masukan *unit step*. Berikut perbandingan kedua respon yang didapat dengan menggunakan Matlab pada Gambar 5.



Gambar 5 Respon *unit step*, fungsi alih dan kecepatan Motor BLDC

Dari grafik diatas dapat dilihat bahwa respon fungsi alih yang telah didapat dari proses identifikasi hampir menyerupai respon kecepatan motor BLDC. Jadi fungsi alih yang telah didapatkan dianggap dapat mewakili pemodelan *plant* motor BLDC.

6. Penentuan Model Referensi

Penentuan model referensi dilakukan dengan cara *try and error* sehingga didapatkan respon yang stabil yang mengikuti karakteristik dari *plant*. Model referensi yang akan ditentukan tidak memiliki *error steady state*, tidak terdapat osilasi dan memiliki *settling time* yang cepat. Fungsi alih dari model referensi yang akan digunakan adalah:

$$\frac{Y_m(s)}{U_c(s)} = \frac{307,3s + 1291}{s^3 + 71,87s^2 + 583,7s + 1291} \quad (17)$$

7. Penetapan Parameter Kontroler

Berdasarkan hasil desain kontroler PI dengan teknik MRAC yang telah dijelaskan sebelumnya (lihat Persamaan 14 dan 15), maka akan didapatkan bentuk Persamaan 18 dan 19.

$$K_p = \frac{-\gamma_p}{s} e \frac{307,3s}{s^3 + 71,87s^2 + 9583,7s + 1291} (u_c - y) \quad (18)$$

$$K_i = \frac{-\gamma_i}{s} e \frac{307,3}{s^3 + 71,87s^2 + 958,3s + 1291} (u_c - y) \quad (19)$$

Pada Persamaan diatas terdapat dua parameter kontrol yang harus ditentukan nilainya terlebih dahulu dalam mendesain sistem yang akan dikontrol, yaitu parameter γ_p dan γ_i . Dimana cara menentukan nilai parameter tersebut dilakukan dengan cara *try and error* pada simulasi Matlab.

8. Desain Persamaan Beda

Model referensi dari sistem telah dinyatakan dalam bentuk frekuensi (lihat Persamaan 17). Dengan menggunakan metode *backward difference*, ganti operator s pada Persamaan 17 dengan Persamaan 20.

$$s = \frac{1 - z^{-1}}{T_s} \quad (20)$$

Dimana T_s = periode sampling

Maka akan didapatkan Persamaan 21 yang merupakan Persamaan model referensi dalam fungsi z .

$$Y_m(z) = \frac{A}{P} Y_m(z) z^{-1} - \frac{B}{P} Y_m(z) z^{-2} + \frac{C}{P} Y_m(z) z^{-3} + \frac{D}{P} U_c(z) - \frac{E}{P} z^{-1} U_c(z) \quad (21)$$

Dalam bentuk time domain, dengan mengganti operator z^0 menjadi waktu saat ini (k), maka akan didapatkan Persamaan 22 yang merupakan penurunan persamaan beda dari model referensi.

$$y_m(k) = \frac{A}{P} y_m(k-1) - \frac{B}{P} y_m(k-2) + \frac{C}{P} y_m(k-3) + \frac{D}{P} u_c(k) - \frac{E}{P} u_c(k-1) \quad (22)$$

Dimana:

$$A = 583,75T_s^2 + 143,74T_s + 3$$

$$B = 71,87T_s + 3$$

$$C = 1$$

$$D = 1291T_s^3 + 307,3T_s^2$$

$$E = 307,3,3T_s^2$$

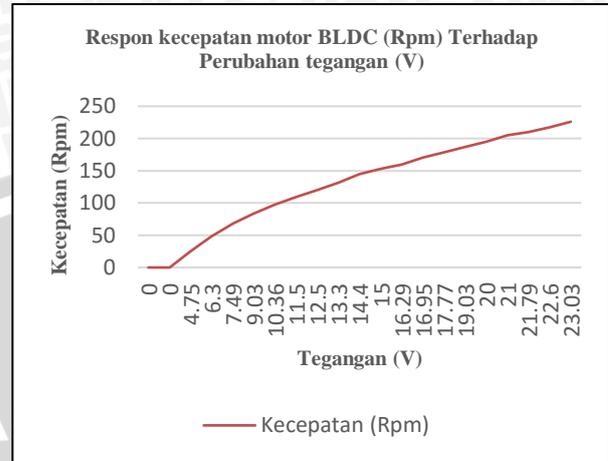
$$F = 307,3,3T_s^3$$

$$G = 1291T_s^3 + 583,75T_s^2 + 71,87T_s$$

Fungsi (k) adalah kondisi sekarang, dan ($k-1$) adalah kondisi sebelumnya. Persamaan diatas kemudian dimasukkan kedalam program pada mikrokontroler. Kemudian dengan cara yang sama akan didapatkan persamaan beda dari parameter kontroler.

9. Pengujian Motor BLDC

Pengujian motor BLDC dilakukan untuk mengetahui kelinieran kecepatan motor BLDC terhadap perubahan tegangan dapat dilihat dalam gambar 6.

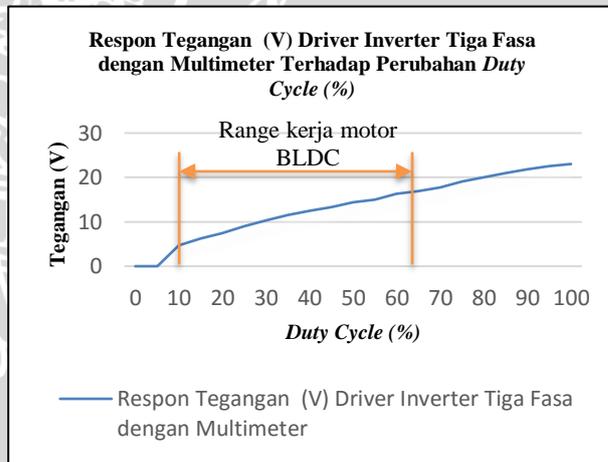


Gambar 6 Respon perubahan kecepatan motor BLDC terhadap tegangan

Dapat dilihat pada Gambar 6, bahwa hubungan perubahan kecepatan terhadap tegangan adalah linear.

10. Pengujian Driver Motor

Pengujian ini bertujuan untuk mengetahui karakteristik *driver* motor terhadap perubahan *duty cycle*, karena nilai tegangan keluaran dari *driver* akan mempengaruhi kecepatan motor BLDC. Respon *driver* motor dapat dilihat dalam gambar 7.

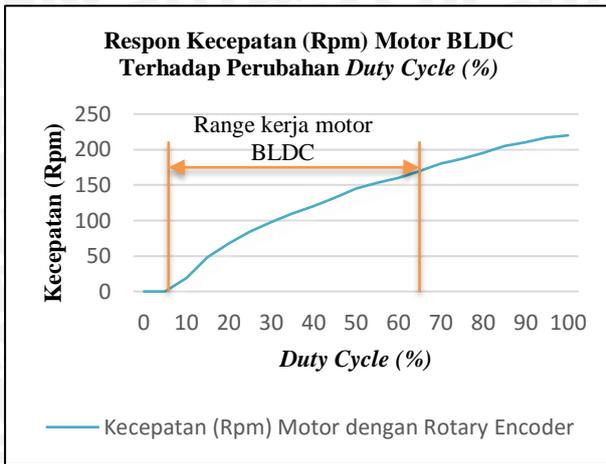


Gambar 7 Respon perubahan tegangan output *driver* terhadap *duty cycle*

Berdasarkan Gambar 7, dapat dilihat bahwa kurva tidak linier, sehingga pemilihan range kerja sistem yang akan digunakan dalam proses pengontrolan adalah dari *duty cycle* PWM 10%-65%.

11. Pengujian Rotary Encoder

Pengujian *rotary encoder* dilakukan untuk mengetahui hasil pembacaan kecepatan motor BLDC dengan *rotary encoder* terhadap *duty cycle*. Respon perubahan kecepatan motor BLDC terhadap *duty cycle* dapat dilihat dalam gambar 8.



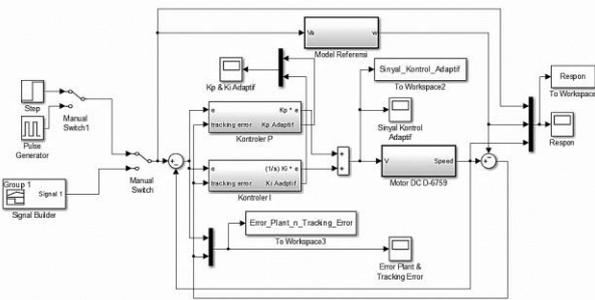
Gambar 8 Respon perubahan kecepatan motor BLDC terhadap *duty cycle*

Pada Gambar 8 dapat diketahui bahwa pemilihan range kerja sistem berada pada range kecepatan 0-150 rpm, yaitu 100, 120 dan 140 rpm.

III. HASIL DAN PEMBAHASAN

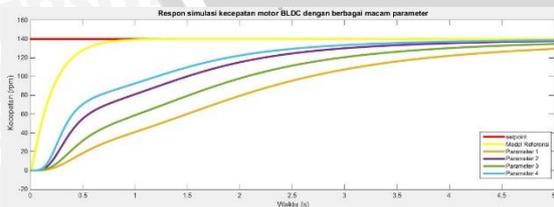
1. Simulasi Matlab

Simulasi dengan blok simulink pada Matlab (lihat Gambar 9) bertujuan mendesain sistem terlebih dahulu dan menentukan parameter kontrol yang akan digunakan.



Gambar 9 Blok simulink pada Matlab

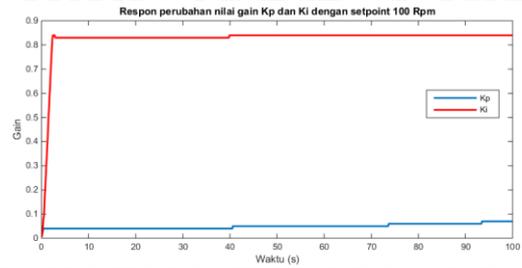
Berdasarkan hasil simulasi yang telah dilakukan (lihat Gambar 10), didapatkan nilai parameter kontrol yang memenuhi kestabilan sistem adalah $\gamma_p = 0,0001$ dan $\gamma_i = 0,0009$.



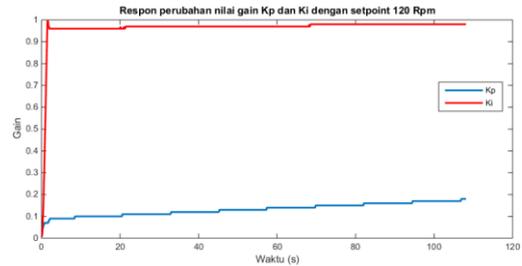
Gambar 10 Respon sistem dengan variasi nilai parameter γ_p dan γ_i

2. Implementasi

Implementasi dari sistem menggunakan 3 nilai *setpoint*, yaitu 100, 120 dan 140 rpm. Data yang diperoleh adalah perubahan nilai Kp dan Ki pada setiap nilai *setpoint* (Gambar 11, 12 dan 13) dan data nilai respon dari motor BLDC pada setiap *setpoint* (Gambar 14, 15 dan 16).



Gambar 13 Respon perubahan nilai Kp dan Ki *setpoint* 100 rpm

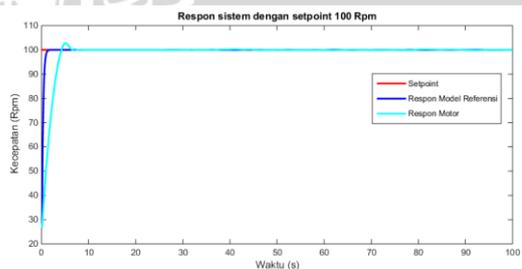


Gambar 14 Respon perubahan nilai Kp dan Ki *setpoint* 120 rpm

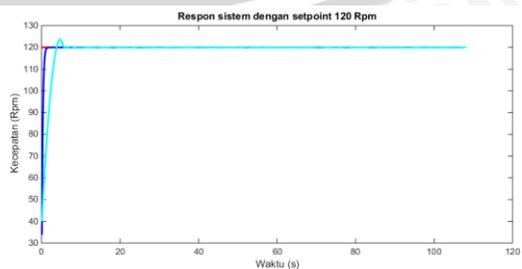


Gambar 15 Respon perubahan nilai Kp dan Ki *setpoint* 140 rpm

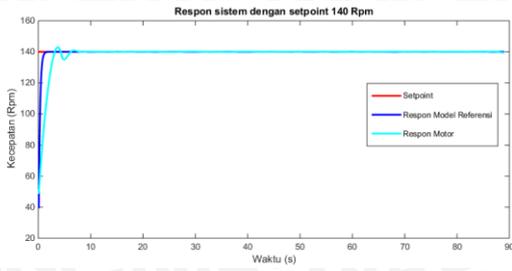
Dari (Gambar 11, 12 dan 13), nilai perubahan Kp dan Ki saat motor mulai dinyalakan mengalami kenaikan nilai yang tinggi. Sedangkan ketika motor sudah mencapai keadaan *steady state*, kenaikan nilai Kp dan Ki secara perlahan.



Gambar 14 Respon motor BLDC dengan *setpoint* 100 rpm



Gambar 15 Respon motor BLDC dengan *setpoint* 120 rpm

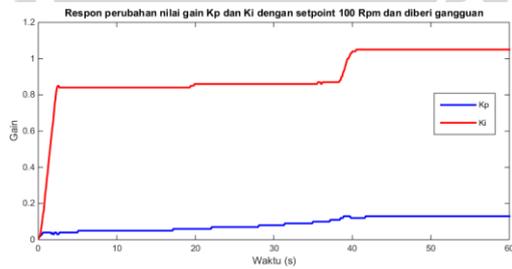


Gambar 16 Respon motor BLDC dengan *setpoint* 140 rpm

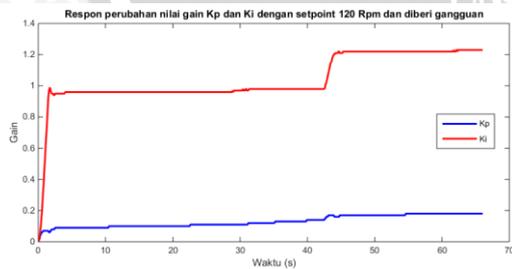
Pada respon motor BLDC (Gambar 14, 15 dan 16) dengan nilai *setpoint* 100 rpm memiliki nilai *maximum overshoot* sebesar 3%, nilai *settling time* adalah 6 detik dengan nilai *error steady state* rata-rata berada dibawah toleransi 2% yaitu 1%. Respon sistem dengan nilai *setpoint* 120 rpm memiliki nilai *maximum overshoot* sebesar 2,5%, nilai *settling time* adalah 5,4 detik dengan nilai *error steady state* rata-rata berada dibawah toleransi 2% yaitu 0,83%. Respon sistem dengan nilai *setpoint* 140 rpm memiliki nilai *maximum overshoot* sebesar 2,14%, nilai *settling time* adalah 6 detik dengan nilai *error steady state* rata-rata berada dibawah toleransi 2% yaitu 0,71%.

6. Implementasi dengan Gangguan

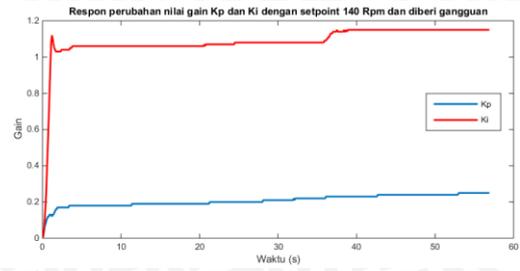
Pada implementasi sistem dengan memberikan gangguan pada saat sistem sedang berada pada keadaan *steady state*, maka akan didapatkan grafik respon Kp dan Ki dan respon motor BLDC seperti pada Gambar 17, 18 dan 19.



Gambar 17 Respon gain Kp dan Ki dengan *setpoint* 100 rpm dan diberi gangguan

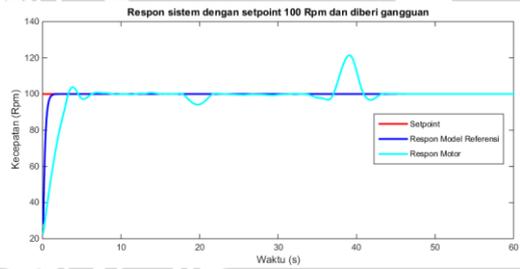


Gambar 18 Respon gain Kp dan Ki dengan *setpoint* 120 rpm dan diberi gangguan

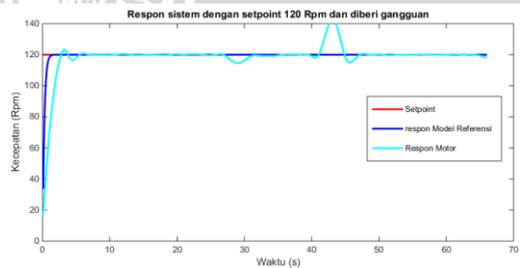


Gambar 19 Respon gain Kp dan Ki dengan *setpoint* 140 rpm dan diberi gangguan

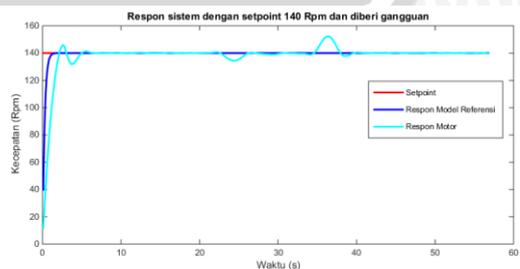
Pada respon sistem (Gambar 20, 21 dan 22) dengan diberikan gangguan pada *setpoint* 100 rpm, respon akan mengalami perlambatan dan *recovery time* respon adalah 3,1 detik. Ketika gangguan yang diberikan dilepaskan, respon akan mengalami percepatan dan *recovery time* respon adalah 8,5 detik. Pada *setpoint* 120 rpm, respon akan mengalami perlambatan dan *recovery time* respon adalah 3,8 detik. Ketika gangguan yang diberikan dilepaskan, respon akan mengalami percepatan dan *recovery time* respon adalah 6,8 detik. Pada *setpoint* 140 rpm, respon akan mengalami perlambatan dan *recovery time* respon adalah 3,1 detik. Ketika gangguan yang diberikan dilepaskan, respon akan mengalami percepatan dan *recovery time* respon adalah 5,3 detik.



Gambar 20 Respon motor BLDC dengan *setpoint* 100 rpm dan diberi gangguan



Gambar 21 Respon motor BLDC dengan *setpoint* 120 rpm dan diberi gangguan



Gambar 22 Respon motor BLDC dengan *setpoint* 140 rpm dan diberi gangguan



IV. KESIMPULAN DAN SARAN

A. KESIMPULAN

1. Berdasarkan hasil identifikasi *plant* menggunakan sinyal uji PRBS dan sintaks ident pada *software* Matlab, fungsi alih motor BLDC memiliki *best fit* sebesar 82.49 dan telah divalidasi.
2. Berdasarkan hasil simulasi didapatkan nilai parameter kontrol yang memenuhi kestabilan respon sistem adalah $\gamma_p = 0,0001$ dan $\gamma_i = 0,0009$.
3. Berdasarkan hasil implementasi tanpa gangguan, respon sistem dengan nilai *setpoint* 100 rpm, 120 rpm dan 140 rpm memiliki nilai *error steady state* rata-rata berada dibawah toleransi 2% dengan nilai masing-masing adalah 1%, 0,83%, dan 0,71%. Sedangkan rata-rata *settling time* pada setiap *setpoint* adalah 6 detik.
4. Berdasarkan hasil implementasi saat sistem diberi gangguan, pada *setpoint* 100 rpm, 120 rpm dan 140 rpm, respon akan mengalami perlambatan dan *recovery time* respon kembali pada keadaan *steady state* dengan waktu masing-masing adalah 3,1 detik, 3,8 detik dan 3,1 detik. Ketika gangguan yang diberikan dilepaskan, respon akan mengalami percepatan dan *recovery time* respon masing-masing adalah 8,5 detik, 6,8 detik dan 5,3 detik.

B. SARAN

Saran yang dapat diberikan untuk penelitian selanjutnya adalah dengan menggunakan inverter yang lebih baik lagi dan dapat mengimplementasikan motor BLDC sebagai aktuator seperti pada lift barang, konveyor, dan sebagainya.

DAFTAR PUSTAKA

- [1] Arundhati Shyam and Febin Daya J L, 2013, "A Comparative Study on the Speed Response of BLDC Motor Using Conventional PI Controller, Antiwindup PI Controller and Fuzzy Controller", India IEEE, 2013 International Conference on Control Communication and Computing (ICCC).
- [2] Astrom, K.J. and B. Wittenmark. 1995. *Adaptive Control*. Addison-Wesley Publishing Company, Inc. USA.
- [3] Ali, A.T., Eisa B. and Omar B. 2012. *Adaptive PID Controller for DC Motor Speed Control*. International Journal of Engineering Inventions 1(5), 26-30.
- [4] Butler, H. 1992. *Model Reference Adaptive Systems, From Theory to Practice*. UK: Prentice-Hall, Inc.
- [5] Sar, S.K. and lillie D. 2014. *MRAC Based PI Controller for Speed Control of DC Motor Using Lab View*. WSEAS Transactions on Systems and Control.