

DESAIN KONTROL KECEPATAN MOTOR INDUKSI TIGA FASA MENGGUNAKAN *FUZZY PID* BERBASIS *IDIRECT FIELD ORIENTED CONTROL*

Ridwan*, Era Purwanto#, Hary Oktavianto#, Muhammad Rizani Rusli#, Handri Toar*

* Jurusan Teknik Elektro, Politeknik Negeri Batam
Jl. Ahmad Yani, Tlk. Tering, Batam kota, Batam, 29461
E-mail: ridwan@polibatam.ac.id, toar@polibatam.ac.id

Departemen Teknik Elektro, Politeknik Elektronika Negeri Surabaya
Jl. Raya ITS, Keputih, Sukolilo, Surabaya, 60111
E-mail: era@pens.ac.id, hary@pens.ac.id, ruslirizani@ieee.org

Abstrak

Motor induksi tiga fasa (MITF) umumnya digunakan di berbagai aplikasi di industri karena keandalannya, biaya rendah, konstruksi kokoh, perawatan rendah, dan efisiensi yang tinggi. Namun untuk mengontrol MITF tidak semudah seperti mengontrol motor DC, karena MITF merupakan motor yang tidak linear. Penggunaan metode *indirect field oriented control* (IFOC) dengan kontroler *fuzzy proportional integrator and derivative* (FPID) dipilih untuk dapat mengatur kecepatan MITF. Metode IFOC akan membuat MITF dapat dikontrol seperti motor DC penguat terpisah. Kontroler FPID yang di desain dengan mengganti kontroler PID konvensional. Performa kontroler FPID yang di desain dibandingkan dengan kontroler PID konvensional. Performa respon yang dibandingkan seperti *rise time*, *settling time*, *overshoot*, *steady state error*, dan *undershoot*. Hasil simulasi yang dibuat menunjukkan bahwa dengan menggunakan kontroler FPID lebih baik dibandingkan dengan kontroler PID. Dimana respon *overshoot* untuk kontroler FPID 0% sedangkan kontroler PID adalah 0.23%. Begitu pula dengan respon *undershoot* untuk kontroler FPID adalah 2.88% sedangkan kontroler PID adalah 6.78%. Untuk respon *rise time*, *settling time*, dan *steady state error* tidak jauh berbeda dari kedua kontroler. Sistem yang sudah di buat disimulasikan di platform LabVIEW.

Kata kunci: Motor Induksi Tiga Fasa, IFOC, FPID, LabVIEW

Abstract

Three-phase induction motor (TPIM) is generally used in various applications in the industry because of its reliability, low cost, robust construction, low maintenance, and high efficiency. However, to control an TPIM is not as easy as controlling a DC motor, because an TPIM is a non-linear motor. The use of the indirect field oriented control (IFOC) method with a fuzzy proportional integrator and derivative (FPID) controller was chosen to be able to adjust the speed of the TPIM. The IFOC method will make a TPIM can be controlled like a separately excited DC motor. FPID controller designed by replacing a conventional PID controller. FPID controller performance designed compared to conventional PID controllers. The response performance is compared like rise time, settling time, overshoot, steady-state error, and undershoot. The simulation results show that using the FPID controller is better than the PID controller. Where the overshoot response for the FPID controller is 0% while the PID controller is 0.23%, similarly, the undershoot response for FPID control is 2.88% while the PID controller is 6.78%. For response rise time, settling time, and steady-state error are not much different from the two controllers. The system that has been created is simulated on the LabVIEW platform.

Keywords: Three-phase induction motors, IFOC, FPID, LabVIEW

1. Pendahuluan

MITF umumnya digunakan di berbagai aplikasi di industri karena keandalannya, biaya yang rendah, konstruksi yang kokoh, perawatan yang rendah, dan efisiensi yang tinggi [1][2]. Pada saat yang sama, industri menuntut aplikasi berkinerja tinggi dalam hal respon dinamis yang baik untuk perubahan beban atau kecepatan. Pengaturan motor induksi tidak semudah seperti pada motor DC, karena motor induksi merupakan motor *nonlinear*, tetapi setelah ditemukan metode *field oriented control* (FOC) yang dapat mengubah sistem *couple* menjadi *decouple*, sehingga permasalahan pada TPIM dapat diselesaikan, dengan menggunakan metode FOC, motor induksi tiga fasa bekerja seperti motor DC penguat terpisah. Metode FOC, terdapat dua teknik kontrol yaitu *direct* FOC (DFOC) dan *indirect* FOC (IFOC). Kedua teknik tersebut bertujuan mendapatkan *decoupling* torsi dan fluks bahkan dengan persamaan matematika yang kompleks untuk motor induksi [3]. IFOC memiliki keunggulan dalam hal teknik estimasi karena tanpa perlu menempatkan sensor di *air gap* untuk mendapatkan fluks rotor. Selain itu IFOC memiliki desain lebih sederhana dalam penerapannya menjadikannya lebih populer di aplikasi industri [4].

Salahsatu cara untuk meningkatkan performa dari motor induksi adalah dengan mengendalikan kecepatan motor induksi. Banyak teknik kontrol yang digunakan untuk mengatur kecepatan motor induksi berdasarkan metode IFOC termasuk kontrol konvensional seperti kontrol *proportional integral* (PI) dan teknik kontrol *proportional integral derivative* (PID) [5]. Teknik kontrol konvensional dianggap sebagai teknik yang baik karena desainnya mudah digunakan, biaya rendah dengan struktur sederhana. Namun, kontroler konvensional sulit karena perubahan Medan magnet, frekuensi, suhu, dan faktor lainnya. Untuk dapat mengatasi masalah ini, salahsatu teknik dengan menggunakan logika fuzzy untuk tujuan mengendalikan kecepatan motor induksi.

Dalam penelitian sebelumnya yang berkaitan dengan kontrol fuzzy diantaranya metode kontrol hybrid seperti fuzzy SMC [6], algoritma QLSA [2], *Fuzzy Gain Scheduling of PID* [7] dan lainnya telah diusulkan untuk meningkatkan kinerja kontroler. Prof. Zadeh pertama kali memperkenalkan logika Fuzzy. Beberapa keuntungan dari logika fuzzy, tidak memerlukan model matematika yang tepat, dapat menangani sistem linear dan *nonlinear* dan didasarkan pada logika manusia, dapat didefinisikan pada aturan linguistik menggunakan struktur umum IF-THEN [8].

Dalam paper ini, penulis mendesain kontrol kecepatan dengan metode FPID berbasis IFOC. Kontroler FPID menunjukkan kesamaan dengan kontroler PID konvensional dengan asumsi tertentu. Paper ini telah disimulasikan ke kontroler kecepatan untuk MITF. Kinerja FPID telah berhasil dibandingkan dengan kontroler PID konvensional.

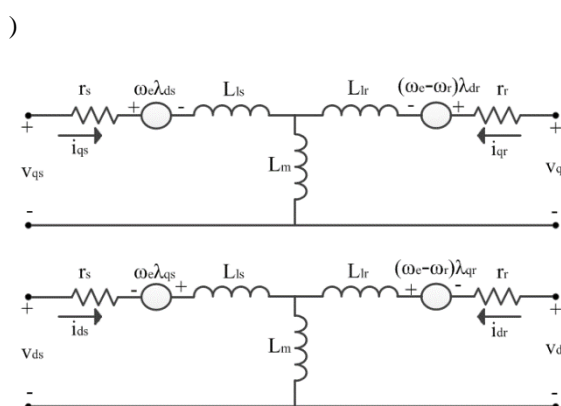
2. Metode

2.1 Model motor induksi

Model motor induksi dapat dihasilkan dari rangkaian ekuivalen motor induksi yang ditunjukkan pada gambar 1 [9].

Kerangka referensi sinkron pada persamaan motor induksi meliputi tegangan stator, rotor dan *flux linkage*. Analisa tegangan stator dan rotor dapat dinyatakan dalam persamaan (1).

$$\begin{aligned} V_{qs} &= r_s i_{qs} + \omega_e \lambda_{ds} + \lambda_{qs} \frac{d}{dt} \\ V_{ds} &= r_s i_{ds} - \omega_e \lambda_{qs} + \lambda_{ds} \frac{d}{dt} \\ V_{qr} &= r_r i_{qr} + (\omega_e - \omega_r) \lambda_{dr} + \lambda_{qr} \frac{d}{dt} \\ V_{dr} &= r_r i_{dr} - (\omega_e - \omega_r) \lambda_{qr} + \lambda_{dr} \frac{d}{dt} \end{aligned} \quad (1)$$



Gambar 1. Rangkaian ekuivalen sumbu d-q dalam kerangka referensi sinkron

Untuk persamaan *flux linkage* stator ditunjukkan pada persamaan berikut:

$$\begin{aligned} \lambda_{qs} &= L_{ls} i_{qs} + L_m (i_{qs} + i_{qr}) = L_s i_{qs} + L_m i_{qr} \\ \lambda_{ds} &= L_{ls} i_{ds} + L_m (i_{ds} + i_{dr}) = L_s i_{ds} + L_m i_{dr} \end{aligned} \quad (2)$$

Sedangkan untuk persamaan *flux linkage* rotor ditunjukkan pada berikut:

$$\begin{aligned} \lambda_{qr} &= L_{lr} i_{qr} + L_m (i_{qs} + i_{qr}) = L_r i_{qr} + L_m i_{qs} \\ \lambda_{dr} &= L_{lr} i_{dr} + L_m (i_{ds} + i_{dr}) = L_r i_{dr} + L_m i_{ds} \end{aligned} \quad (3)$$

Berikutnya persamaan (2) dan (3) disubsitusikan kedalam persamaan (1) sehingga didapatkan persamaan dalam bentuk matrik berikut:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{qr} \\ v_{dr} \end{bmatrix} = \begin{bmatrix} r_s + pL_s & \omega_e L_s \\ -\omega_e L_s & r_s + pL_s \\ pL_m & (\omega_e - \omega_r) L_m \\ -(\omega_e - \omega_r) L_r & pL_m \end{bmatrix}$$

$$\begin{bmatrix} pL_m & \omega_e L_m \\ -\omega_e L_m & pL_m \\ r_r + pL_r & (\omega_e - \omega_r)L_r \\ -(\omega_e - \omega_r)L_r & r_r + pL_r \end{bmatrix} x \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (4)$$

Dimana:

V_{qs} Dan V_{qr} adalah tegangan stator dan rotor quadrature.

V_{ds} Dan V_{dr} adalah tegangan stator dan rotor direct.

r_s Dan r_r adalah tahanan stator dan rotor.

i_{qs} Dan i_{qr} adalah arus stator dan rotor quadrature.

i_{ds} Dan i_{dr} adalah arus stator dan rotor direct.

λ_{qs} Dan λ_{qr} adalah flux stator dan rotor quadrature.

λ_{ds} Dan λ_{dr} adalah flux stator dan rotor direct.

ω_r Dan ω_e adalah kecepatan rotor dan sinkron.

L_{ls} Dan L_{lr} leakage induktansi stator dan rotor.

L_m Adalah magnetizing induktansi.

p Adalah $\frac{d}{dt}$

Persamaan torsi yang diberikan adalah [10]:

$$T_e = \left(\frac{3}{2}\right) \left(\frac{p}{2}\right) (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) \quad (5)$$

Dimana p adalah jumlah pasang pole.

Dari persamaan (4), persamaan tegangan rotor dinyatakan sebagai:

$$v_{qr} = 0 = L_m \frac{di_{qs}}{dt} + (\omega_e - \omega_r)L_m i_{ds} + \left(r_r i_r + L_r \frac{di_{qr}}{dt}\right) + (\omega_e - \omega_r)L_r i_{dr} \quad (6)$$

$$v_{dr} = 0 = L_m \frac{di_{ds}}{dt} + (\omega_e - \omega_r)L_m i_{qs} + \left(r_r i_r + L_r \frac{di_{qr}}{dt}\right) + (\omega_e - \omega_r)L_r i_{dr} \quad (7)$$

Dengan menggunakan persamaan (1) maka,

$$\frac{d\lambda_{qr}}{dt} + r_r i_{qr} + (\omega_e - \omega_r)\lambda_{dr} = 0 \quad (8)$$

$$\frac{d\lambda_{dr}}{dt} + r_r i_{dr} + (\omega_e - \omega_r)\lambda_{qr} = 0 \quad (9)$$

Dari persamaan (3) diperoleh,

$$i_{qr} = \frac{1}{L_r} \lambda_{qr} - \frac{L_m}{L_r} i_{qs} \quad (10)$$

$$i_{dr} = \frac{1}{L_r} \lambda_{dr} - \frac{L_m}{L_r} i_{ds} \quad (11)$$

Arus rotor i_{qr} dan i_{dr} dapat dihilangkan dari persamaan (8) dan (9) dengan menggunakan persamaan (10) dan (11). Maka diperoleh,

$$\frac{d\lambda_{qr}}{dt} + \frac{L_r}{r_r} \lambda_{qr} - \frac{L_m}{L_r} r_r i_{qs} + (\omega_e - \omega_r)\lambda_{dr} = 0 \quad (12)$$

$$\frac{d\lambda_{dr}}{dt} + \frac{L_r}{r_r} \lambda_{dr} - \frac{L_m}{L_r} r_r i_{ds} + (\omega_e - \omega_r)\lambda_{qr} = 0 \quad (13)$$

2.2 Indirect Field Oriented Control

IFOC merupakan suatu metoda yang digunakan untuk mengatur medan pada motor induksi, dimana sistem *coupled* diubah menjadi sistem *decoupled*. Untuk dapat mengubah sistem ini maka komponen *flux* (λ_r) pada arus I_d harus di tempatkan selaras dengan sumbu d, dan komponen torsi (T_e) pada arus (I_q) harus ditempatkan selaras dengan sumbu q. Dengan sistem ini arus *flux* dan arus torsi dapat diatur secara terpisah. Blok diagram IFOC ditunjukkan pada gambar 2. Untuk memenuhi *decoupled*, maka:

$$\lambda_{qr} = 0 \text{ Dan } \lambda_r = \lambda_{dr} \quad (14)$$

Fluk rotor harus tetap dijaga konstan, maka:

$$\frac{d\lambda_{dr}}{dt} = 0 = \frac{d\lambda_{qr}}{dt} \quad (15)$$

Substitusikan persamaan (14) kedalam persamaan (9) dan (10), maka diperoleh:

$$I_{qr} = -\frac{L_m}{L_r} I_{qs} \quad (16)$$

$$I_{dr} = \frac{\lambda_r}{L_r} - \frac{L_m}{L_r} I_{ds} \quad (17)$$

Selanjutnya dari persamaan (15) di substitusikan ke persamaan (12) dan (13):

$$(\omega_e - \omega_r) = \omega_{sl} = \frac{L_m r_r}{\lambda_r L_r} i_{qs} \quad (18)$$

$$\text{Dan } \frac{L_r}{r_r} \frac{d\lambda_r}{dt} + \lambda_r = L_m i_{ds} \quad (19)$$

persamaan (19) dapat juga ditulis kedalam bentuk lain yaitu:

$$\lambda_r = \frac{L_m i_{ds}}{s\tau_r + 1} \quad (20)$$

Dimana:

$$\tau_r = \frac{L_r}{r_r} \text{ adalah konstanta waktu rotor dan } \frac{d}{dt} = s.$$

Substitusikan persamaan untuk arus i_{qr} dan i_{dr} persamaan (16) dan (17) kedalam persamaan (2), maka persamaan *flux linkag* menjadi:

$$\lambda_{qs} = \left(L_s - \frac{L_m^2}{L_r}\right) i_{qs} \quad (21)$$

$$\lambda_{ds} = \left(L_s - \frac{L_m^2}{L_r}\right) i_{ds} + \frac{L_m}{L_r} \lambda_{dr} \quad (22)$$

Selanjutnya substitusikan λ_{qs} dan λ_{ds} dari persamaan (21) dan (22) kedalam persamaan torsi pada persamaan (5), maka,

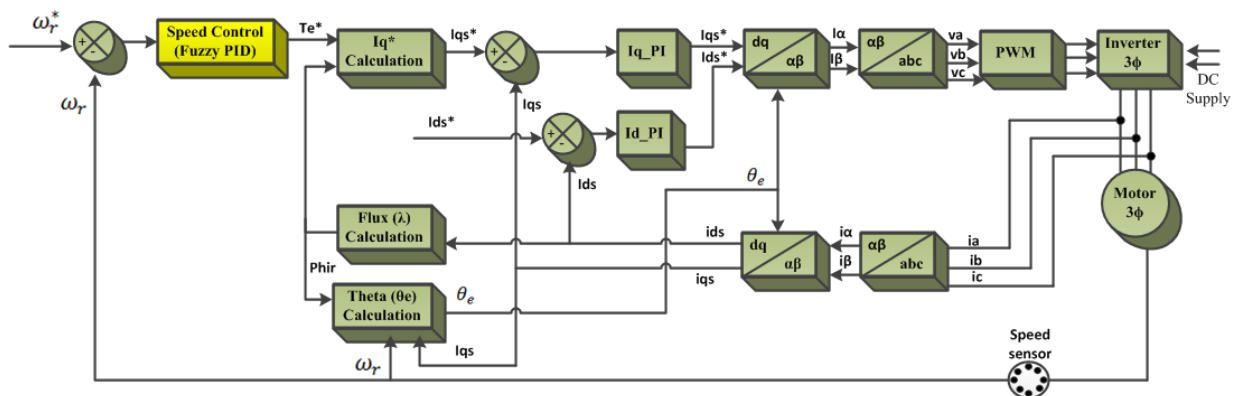
$$T_e = \left(\frac{3}{2}\right) \left(\frac{P}{2}\right) \frac{L_m}{L_r} \lambda_r i_{qs} \quad (23)$$

Pada diagram blok IFOC terdapat empat sinyal *feedback* yang dibutuhkan sistem untuk diproses. Sinyal pertama adalah kecepatan rotor yang dihasilkan dari sensor kecepatan yang digunakan sebagai *feedback* kontroler kecepatan (ω_r) untuk menghasilkan torsi referensi (T_e^*). Selanjutnya T_e^* dijadikan sebagai input referensi arus stator *quadrator* (i_{qs}^*) melalui persamaan (24). Sedangkan referensi arus stator *direct* dibutuhkan i_{ds}^* .

Untuk persamaan (24) arus stator *quadrator* (i_{qs}^*) diperoleh dari persamaan (23).

$$i_{qs}^* = \frac{2}{3} \cdot \frac{2}{P} \cdot \frac{L_r}{L_m} \cdot \frac{T_e^*}{\lambda_r} \quad (24)$$

Untuk ketiga sinyal *feedback* yang lain diperoleh dari pengukuran arus fasa stator. Selanjutnya arus fasa stator akan diubah kedalam transformasi Clark (blok abc ke $\alpha\beta$) persamaan (25). Luaran dari konversi ini akan menghasilkan arus i_α dan i_β . Selanjutnya kedua arus ini masuk kedalam blok transformasi Park (blok $\alpha\beta$ ke dq) persamaan (26) menghasilkan arus dalam kerangka arus i_{ds} dan i_{qs} . Kemudian arus i_{ds} dan i_{qs} akan dibandingkan dengan arus referensi i_{ds}^* dan i_{qs}^* dari perbandingan arus tersebut akan menghasilkan error. Selanjutnya error dari kedua arus dikontrol menggunakan kontroler PI (blok i_q_PI dan i_d_PI).



Gambar 2. Blok diagram IFOC

Luaran dari kedua arus ini akan diubah kedalam invers transformasi Park (blok dq ke $\alpha\beta$) persamaan (27). Selanjutnya luaran dari transformasi ini akan dikonversikan menjadi tegangan v_a , v_b , dan v_c dengan menggunakan invers transformasi Clark persamaan (28). Dari luaran transformasi ini selanjutnya diberikan ke *pulse wide modulation* (PWM). Luaran dari blok ini memberikan sinyal yang mengendalikan inverter untuk menjalankan motor induksi. Disini transformasi Park dan Invers Park memerlukan posisi flux rotor θ_e .

Arus i_{abc} yang didapat dari pengukuran merupakan arus tiga fasa sinusoidal. Untuk mengubah koordinat i_{abc} kedalam sistem koordinat dua fasa (sumbu-dq) dibutuhkan transformasi. Transformasi ini dapat dibagi kedalam dua step yaitu transformasi Clark dan Park.

Transformasi Clark digunakan untuk untuk mengubah koordinat tiga fasa kedalam koordinat dua fasa ($\alpha\beta$), sedangkan untuk transformasi Park digunakan untuk mengubah koordinat dua fasa stasioner ($\alpha\beta$) kedalam koordinat kerangka putar *synchronously* (dq) yang tetap pada rotor. Persamaan (25) menunjukkan transformasi Clark.

$$i_\alpha = i_a$$

$$i_\beta = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (25)$$

Sedangkan transformasi Park ditunjukkan pada

persamaan berikut:

$$i_d = i_\alpha \cos\theta + i_\beta \sin\theta$$

$$i_q = -i_\alpha \sin\theta + i_\beta \cos\theta \quad (26)$$

Untuk dapat memberikan sinyal ke PWM maka koordinat arus Id dan arus Iq harus di ubah menjadi koordinat tegangan v_a , v_b , dan v_c . Perubahan ini menggunakan invers transformasi Park dan Clark. Persamaan (27) menunjukkan persamaan invers transformasi Park.

$$i_\alpha = i_d \cos\theta - i_q \sin\theta$$

$$i_\beta = i_d \sin\theta + i_q \cos\theta \quad (27)$$

Sedangkan invers transformasi Clark ditunjukkan pada persamaan berikut:

$$i_a = i_\alpha$$

$$i_b = -\frac{1}{2}i_\alpha + \frac{\sqrt{3}}{2}i_\beta$$

$$i_c = -\frac{1}{2}i_\alpha - \frac{\sqrt{3}}{2}i_\beta \quad (28)$$

Transformasi Park dan invers Park membutuhkan sudut flux rotor ($\theta = \theta_e$) yang ditentukan pada persamaan (29).

$$\theta_e = \int (\omega_r + \omega_{sl}) dt \quad (29)$$

2.3 Desain kontroler FuzzyPID untuk kontrol kecepatan

Desain kontroler digunakan untuk mengembalikan respon *setpoint* kecepatan yang diinginkan, bahkan jika variabel *setpoint* diubah. Tahapan desain dimulai dengan membentuk kontroler PID untuk mendapatkan parameter tuning PID. Selanjutnya, mengganti persamaan PID dengan kontroler linier fuzzy yaitu FPID. Berdasarkan gambar 2, pada blok diagram IFOC terdapat blok *Speed controller* (blok warna kuning). Blok ini menjadi fokus dari penelitian yang dibuat.

2.3.1 Kontroler Proportional Integral Derivative (PID)

Kontroler PID merupakan kombinasi dari tiga metoda kontrol yaitu kontrol proporsional, integral dan *derivative* yang diperlukan untuk memenuhi persyaratan kinerja spesifik. Persamaan untuk kontroler PID dapat dinyatakan sebagai:

$$u_n(t) = K_p * e(t) + K_i * \int e(\tau) * d\tau + K_d \frac{de}{dt} \\ = K_p \left(e(t) + \frac{1}{T_i} \int e(\tau) * d\tau + T_d \frac{de}{dt} \right) \quad (30)$$

$$e(t) = Ref(t) - y(t) \quad (31)$$

Dimana:

$e(t)$ Adalah sinyal error.

$Ref(t)$ adalah *setpoint*.

$y(t)$ Adalah proses output.

$T_i = K_p/K_i$ Adalah integral time.

$T_d = K_p/K_d$ Adalah derivative time.

K_p, K_i dan K_d Adalah gain proporsional, integral dan derivative.

Jika persamaan (30) diubah kedalam bentuk *discrete*, maka akan menjadi:

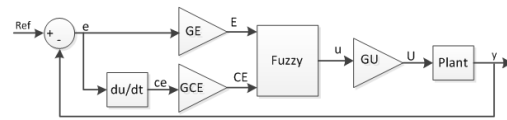
$$u_n = K_p \left(e_n + \frac{1}{T_i} \sum_{j=1}^n e_j T_s + T_d \frac{e_n - e_{n-1}}{T_s} \right) \quad (32)$$

Kontrol proporsional bertanggung jawab ketika output kontroler harus proporsional dengan perbedaan antara variabel *setpoint* dengan nilai variabel saat ini. Kontrol integral bertanggung jawab untuk mengurangi *error steady state*. Sedangkan kontrol derivative memprediksi tindakan kontrol dengan menggunakan laju perubahan sinyal kesalahan sebagai input [11].

2.3.2 Kontroler Logika Fuzzy

Kontrol fuzzy banyak digunakan dalam kontrol industri, terutama dalam situasi di mana teknik desain kontrol konvensional sulit diterapkan. Dalam mendesain FPID, dimulai dari desi FPD dan FPI. Selanjutnya dari kedua desain tersebut akan digabungkan menjadi kontroler FPID.

2.3.2.1 Fuzzy proportional derivative



gambar 3. kontroler Fuzzy PD (FPD)

Gambar 3 menunjukkan diagram blok kontroler FPD yang dapat ditulis ke dalam persamaan (33).

$$u_{FPD} = f(GE * e_n, GCE * ce_n) * GU \quad (33)$$

Fungsi f mewakili pemetaan input dan output fuzzy dari kontroler [12]. Jika persamaan non-linear menggunakan pendekatan linier, maka diperoleh:

$$u_{FPD} = (GE * e_n + GCE * ce_n) * GU \\ = GE * GU * \left(e_n + \frac{GCE}{GE} * ce_n \right) \quad (34)$$

$$ce_n = \frac{e_n - e_{n-1}}{T_s} \quad (35)$$

Jika variabel $T_i = 0$ pada persamaan (32), maka diperoleh:

$$u_n = K_p \left(e_n + T_d \frac{e_n - e_{n-1}}{T_s} \right) \quad (36)$$

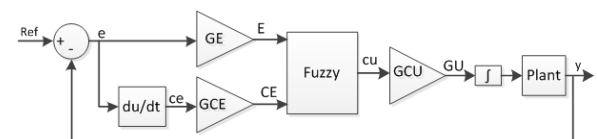
Sebagai perbandingan persamaan (34) dan (36), rasio gain adalah:

$$GE * GU = K_p \quad (37)$$

$$\frac{GCE}{GE} = T_d \quad (38)$$

Dari perbandingan gain pada persamaan (37) dan (38), perhatikan bahwa kerja dari kontroler FPD seperti kontroler PD.

2.3.2.2 Fuzzy proportional integral



Gambar 4. Kontroler Fuzzy PI (FPI)

Kontroler FPI pada gambar 4 hampir sama dengan konfigurasi kontroler FPD kecuali integrator pada output. Aksi integral akan menaikkan sinyal kontrol jika terjadi kesalahan positif kecil, sedangkan jika terjadi kesalahan negatif, sinyal kontrol akan menurunkannya. Kontroler dengan aksi integral akan selalu kembali ke nol dalam *steady state* [12].

Gambar 4 menunjukkan diagram blok kontroler FPI yang dapat ditulis dalam persamaan (39).

$$u_{FPI} = f(GE * e_n, GCE * ce_n) * GCU * \int * T_s \quad (39)$$

Jika persamaan non-linear menggunakan pendekatan linear, pada persamaan (39) menjadi:

$$\begin{aligned} u_{FPI} &= \sum_{i=1}^n (E_i + CE_i) * GCU * T_s \\ &= GCE * GCU * \left[\frac{GE}{GCE} \sum_{i=1}^n e_i * T_s + e_n \right] \end{aligned} \quad (40)$$

Jika variabel $T_d = 0$ pada persamaan (32), maka:

$$u_n = Kp \left(e_n + \frac{1}{T_i} \sum_{j=1}^n e_j T_s \right) \quad (41)$$

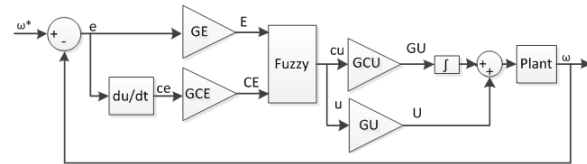
Sebagai perbandingan persamaan (40) dan (41), maka gain rasionya adalah:

$$GCE * GCU = Kp \quad (42)$$

$$\frac{GE}{GCE} = \frac{1}{T_i} \quad (43)$$

Dari perbandingan gain pada persamaan (42) dan (43), perhatikan bahwa kerja dari kontroler FPI seperti kontroler PI.

2.3.2.3 Fuzzy proportional derivative integral



Gambar 5. Kontroler Fuzzy PID (FPID)

Desain kontroler FPID merupakan kombinasi dari kontroler FPI dan FPD dapat dinyatakan dalam persamaan (41) [13][14]. Gambar 5 menunjukkan diagram blok kontroler FPID.

$$u_{FPID} = u_{FPI} + u_{FPD} \quad (44)$$

Tahap selanjutnya, mensubsitusikan persamaan (34) dan (40), hasil substitusi ditunjukkan pada persamaan (45).

$$\begin{aligned} u_{FPID} &= GU * GE * Ref - (GCU * GCE + GU * GE) * y \\ &\quad + GCU * GE * \sum_{i=0}^k e_i * T_s + GU * GCE * \frac{ce}{T_s} \end{aligned} \quad (45)$$

Jika dibandingkan Persamaan. (30) dan (45) bahwa FPID bekerja seperti kontrol PID konvensional. Bobot gain *Setpoint*, proporsional, integral, dan derivative persamaannya adalah

$$K_p * = GU * GE \quad (46)$$

$$K_p = GCU * GCE + GU * GE \quad (47)$$

$$K_I = \frac{K_p}{T_i} = GU * GE \quad (48)$$

$$K_D = K_p * T_d = GU * GCE \quad (49)$$

Dengan asumsi bahwa kontrol FPID memiliki referensi

Input maksimum adalah *max.error*, karena rentang

input *Error* adalah [-1 1], sehingga untuk menyelesaikan GE, GCU, GU, dan GCE sebagai berikut:

$$GE = \frac{1}{\text{max.error}} \quad (50)$$

$$GCU = \frac{K_i}{GE} \quad (51)$$

$$GU = \frac{K_d}{GCE} \quad (52)$$

$$GCE = \frac{GE * (K_p - \sqrt{K_p^2 - 4K_i K_d})}{2K_i} \quad (53)$$

Di mana GE adalah faktor normalisasi, GU adalah respon faktor de-normalisasi, GCU adalah respon perubahan faktor de-normalisasi, GCE adalah perubahan faktor normalisasi *error*. Struktur logika fuzzy terdiri dari 4 bagian utama, yaitu fuzzifikasi, *rule base*, *inference engine*, dan defuzzifikasi [8].

2.3.2.4 Fuzzifikasi

Tahapan fuzzifikasi adalah tahap awal dimana ada proses pemetaan nilai crisp (tegas) pada himpunan. Input dari fungsi keanggotaan (MF) yang digunakan memiliki dua input, yaitu *error* dan *change error* sedangkan untuk *outputnya* hanya satu. Untuk interval ditentukan [-1 1] baik untuk input maupun *output*. Fungsi keanggotaan input *error* dan *change error* ditunjukkan pada gambar 6 dan 7. Sedangkan untuk fungsi keanggotaan *output* ditunjukkan pada gambar 8. Semua input dan *output* fuzzy dibagi menjadi Lima bagian. Label untuk fungsi keanggotaan yaitu NB (*Negative Big*), NM (*Negative Middle*), NS (*Negative Small*), ZE (*Zero*), PS (*Positive Small*), PM (*Positive Middle*), dan PB (*Positive Big*).

2.3.2.5 Knowledge base and inference engine

Knowledge base melibatkan data dari linguistik dan database fuzzy. Database mendefinisikan kumpulan informasi input dan *output* fuzzy yang digunakan untuk mendefinisikan aturan kontrol linguistik. *Rule base* menentukan kontrol yang akan diterapkan dengan menyesuaikan aturan kontrol linguistik. *Inference engine* digunakan untuk menggambarkan proses menghasilkan *output* fuzzy dari input untuk mengevaluasi IF-THEN dan menjalankan aturan 5 x 5 seperti yang ditunjukkan pada Tabel 1.

2.3.2.6 Defuzzifikasi

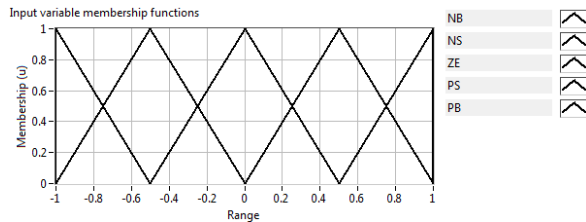
Pada langkah ini, proses memetakan nilai fuzzy menjadi nilai *crisp*. Nilai *crisp* ini akan digunakan dalam penerapan dan analisis akhir, metode defuzzifikasi yang digunakan adalah *method center of area* (CoA). Persamaan (54) [15] digunakan untuk menentukan pusat geometri area ini.

$$CoA = \frac{\int_{x_{min}}^{x_{max}} f(x) \cdot x \, dx}{\int_{x_{min}}^{x_{max}} f(x) \, dx} \quad (54)$$

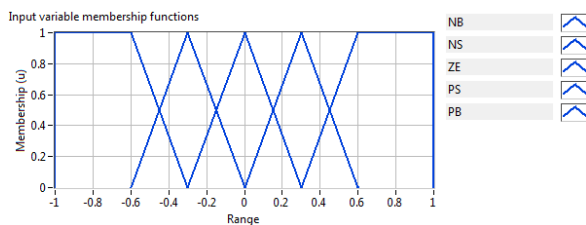
Di mana CoA adalah *center of area*, x adalah nilai variabel linguistik, dan x_{max} dan x_{min} mewakili range variabel linguistik.

TABEL 1. RULE BASE FUZZY

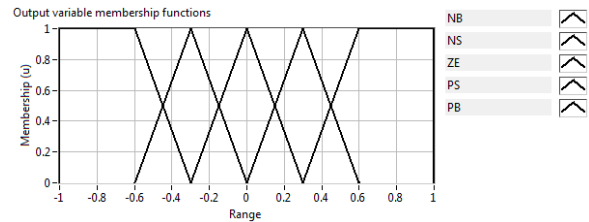
| | | | | | |
|------|----|----|----|----|----|
| CE/E | NB | NS | ZE | PS | PB |
| NB | NB | NB | NS | NS | ZE |
| NS | NB | NS | NS | ZE | PS |
| ZE | NS | NS | ZE | PS | PS |
| PS | NS | ZE | PS | PS | PB |
| PB | ZE | PS | PS | PB | PB |



Gambar 6. Fungsi keanggotaan input Error (E)



Gambar 7. Fungsi keanggotaan input change error (CE)



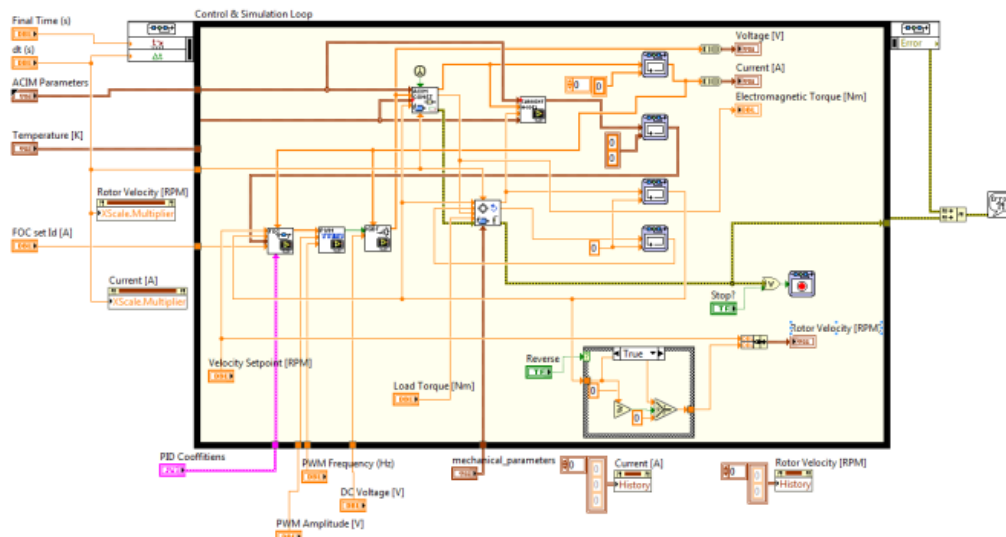
Gambar 8. Fungsi keanggotaan output

3. Hasil dan pembahasan

Dalam pengujian yang dilakukan digunakan motor induksi tiga fasa dengan daya 7500 Watt. Parameter motor induksi yang digunakan terdapat pada Tabel 2.

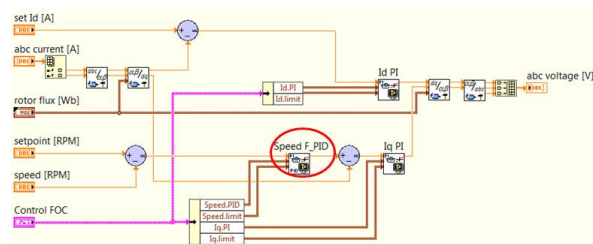
TABEL 2 PARAMETER MOTOR INDUKSI

| | |
|-----------------------------|--------------------|
| Daya Nominal | 7.5 kW |
| Kecepatan Nominal | 1440 RPM |
| Tegangan | 380 V |
| Frekuensi | 50 Hz |
| Jumlah pasang kutub | 2 |
| Tahanan Stator (R_s) | 0.7384 Ω |
| Induktansi stator (L_s) | 0.003045 H |
| Tahanan Rotor (R_r) | 0.003045 Ω |
| Induktansi rotor (L_r) | 0.7402 H |
| Mutual Induktansi (L_m) | 0.1241 H |
| Inersia (J) | 0.0343 $kg.m^2$ |
| Faktor friksi | 0.000503 N.m.s/rad |



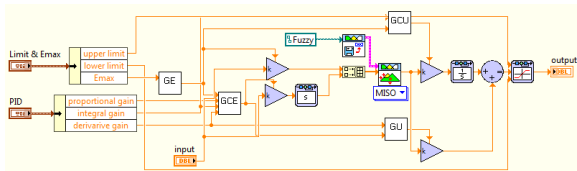
Gambar 9. Model kontroler kecepatan untuk MITF berdasarkan metode IFOC di LabVIEW

Pengujian simulasi yang dilakukan untuk mengatur kecepatan motor induksi tiga fasa dengan kontroler FPID berdasarkan metode IFOC. Simulasi telah dijalankan di bawah platform LabVIEW. Model kontrol kecepatan untuk *drive* motor induksi ditunjukkan pada gambar 9. Gambar 10 menunjukkan model blok FOC. Di sisi lane, diagram blok FPID ditunjukkan pada gambar 11.



Gambar 10. Model blok FOC pada LabVIEW

Objek yang di lingkari garis merah pada model blok FOC pada gambar 10 adalah blok kontrol kecepatan. Untuk lebih jelasnya ditunjukkan pada gambar 11.



Gambar 11. Blok Program LabVIEW Fuzzy PID

Sebelum menjalankan simulasi, parameter sistem kontrol pada motor induksi dan kontroler logika Fuzzy PID harus ditala (*tuning*) terlebih dahulu, agar diperoleh respon kinerja motor yang baik. Penalaan parameter sistem dilakukan secara coba-coba atau perkiraan, hingga diperoleh hasil sesuai dengan yang diinginkan. Adapun hasil penalaan ditunjukkan pada Tabel 3.

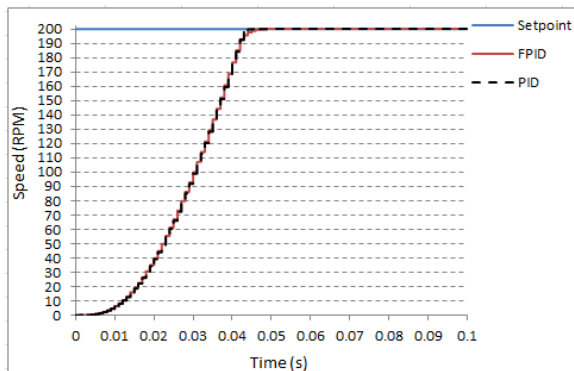
TABEL 3 PARAMETER TUNING

| | |
|------------------------|--------|
| Time Sampling (TS) | 6E-5 s |
| Gain Error (GE) | 0.001 |
| Gain Proportional (KP) | 200 |
| Gain Integrator (KI) | 4 |
| Gain Derivative (KD) | 0.2 |

Simulasi dijalankan dengan dua keadaan yaitu keadaan tanpa beban dan keadaan berbeban. Untuk menguji kemampuan dan keandalan model yang dikembangkan, maka respon sistem fuzzy PID dibandingkan dengan kontroler konvensional (kontroler PID).

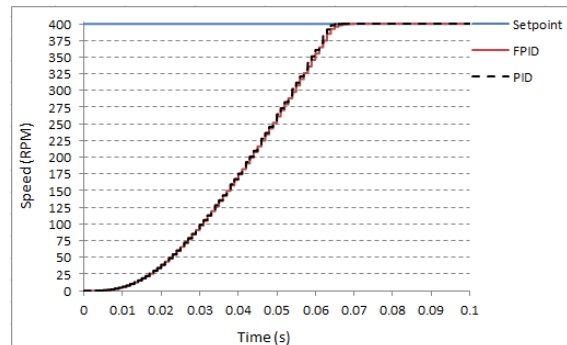
3.1 Pengujian tanpa beban

Pada gambar 12 adalah hasil pengujian kecepatan motor tanpa beban torsi disajikan. Dalam kondisi ini, model yang diuji diberi kecepatan *setpoint* 200 RPM. Respon yang diamati adalah perubahan kecepatan yang dimulai dari 0 RPM hingga 200 RPM. Berdasarkan hasil pengamatan, *rise time* dan *overshoot* untuk kontroler FPID dan PID memiliki respon yang sama, yaitu 25 ms dan 0%. *Settling time* yang dibutuhkan oleh kontroler FPID adalah 44 ms sedangkan PID adalah 43 ms. Respon steady state error yang diamati adalah 0,301 RPM untuk kontroler FPID sedangkan untuk kontroler PID adalah 0,117 RPM.



Gambar 12. Respon kecepatan pada 200 RPM

Selanjutnya pengujian dilakukan dengan *setpoint* 400 RPM yang disajikan pada gambar 13. Demikian pula, seperti dalam pengujian sebelumnya, telah dilakukan simulasi tanpa beban dengan respon diamati untuk perubahan kecepatan dari 0 RPM hingga 400 RPM. Dari respon kecepatan, *overshoot*, *rise time*, dan *settling time* diperoleh masing-masing pada 0%, 39 ms, dan 64 ms sebagai hasil respon terhadap FPID dan Kontroler PID. Respon *steady state error* pada FPID adalah 0,421 RPM sedangkan kontroler PID adalah 0,333 RPM

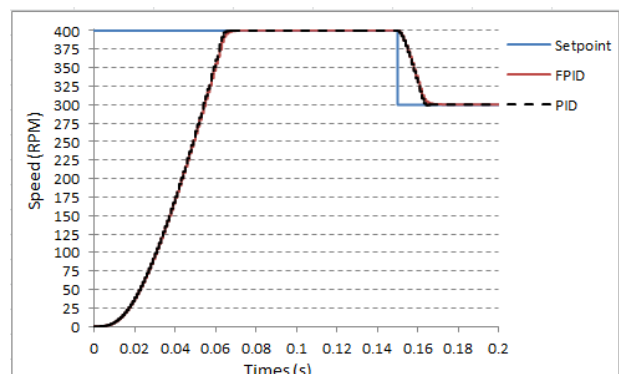


Gambar 13. Respon kecepatan pada 400 RPM

Pada gambar 14, model yang diuji adalah perubahan *setpoint*. *Setpoint* awal ditetapkan pada 400 RPM, dan setelah simulasi berjalan 0,15 detik (dalam kondisi *steady*), *setpoint* dikurangi menjadi 300 RPM. Simulasi dimulai dari kecepatan awal motor 0 RPM untuk mencapai posisi *setpoint*, dengan waktu pengambilan sampel dari 0 hingga 0,2 detik. Respon yang diamati dan dianalisis hanya pada 300 RPM dari kecepatan awal 400 RPM. Kontroler PID memiliki respon dengan *undershoot* 0,44%, sedangkan untuk kontroler FPID memiliki respon *undershoot* 0%. Dari hasil pengujian yang dilakukan sesuai dengan gambar 11 dan 12 ditunjukkan pada Tabel 4.

TABEL 4. HASIL PENGUJIAN TANPA BEBAN

| Parameter | PID | | Fuzzy PID | |
|--------------------------------|-----------|-----------|-----------|-----------|
| | 200 RPM | 400 RPM | 200 RPM | 400 RPM |
| <i>Overshoot</i> | 0 % | 0 % | 0 % | 0 % |
| <i>Rise time</i> (T_r) | 25 ms | 39 ms | 25 ms | 39 ms |
| <i>Settling time</i> (T_s) | 43 ms | 64 ms | 44 ms | 64 ms |
| <i>Error steady state</i> | 0.117 RPM | 0.333 RPM | 0.301 RPM | 0.421 RPM |



Gambar 14. Respon kecepatan dengan *setpoint* yang berubah

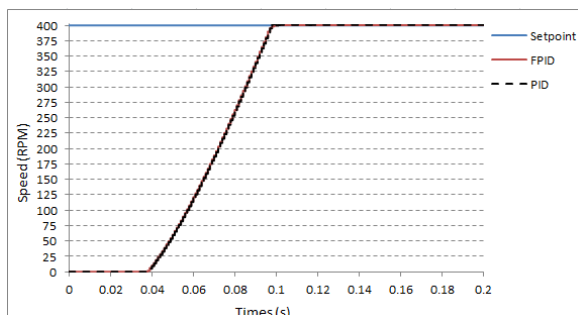
Berdasarkan hasil uji simulasi kecepatan motor induksi seperti yang ditunjukkan pada gambar 11 hingga 13, dapat dianalisis, bahwa kontroler FPID dikembangkan memiliki kemampuan untuk mengendalikan kecepatan dengan baik bila dibandingkan dengan kontroler PID, dengan rise time dan settling time yang relatif cepat, *overshoot* 0%, *error steady state* kurang dari 0,5 RPM, dan respon *undershoot* 0%.

3.2 Pengujian dengan beban

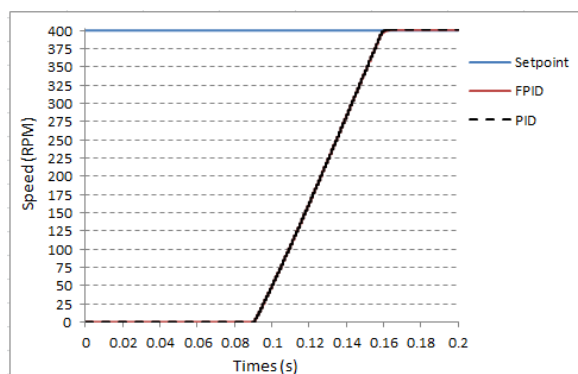
Pada kondisi berbeban yang ditunjukkan pada gambar 15 dan 16, model yang diuji dengan memasukkan torsi beban konstan yaitu 15 Nm dan 30 Nm. Pada pengujian ini dilakukan dengan mengubah beban pada motor induksi sedangkan kecepatan diatur tetap yaitu pada 400 RPM. Hasil uji yang diamati adalah perubahan kecepatan dari 0 RPM hingga 400 RPM. Hasil pengujian ditunjukkan pada Tabel 5. Pada pengujian dengan beban 15 Nm, didapat *overshoot* untuk respon kecepatan dengan kontroler FPID adalah 0% sedangkan kontroler PID 0.66%. untuk pengujian dengan beban 30 Nm, *overshoot* untuk respon kecepatan kontroler FPID tetap 0% sedangkan kontroler PID adalah 0.23%.

TABEL 5 HASIL PENGUJIAN DENGAN BEBAN

| Parameter | PID | | Fuzzy PID | |
|--------------------------------|-----------|-----------|-----------|-----------|
| | 15 Nm | 30 Nm | 15 Nm | 30 Nm |
| <i>Overshoot</i> | 0.66 % | 0.23 % | 0 % | 0 % |
| <i>Rise time</i> (T_r) | 46 ms | 55 ms | 46 ms | 55 ms |
| <i>Settling time</i> (T_s) | 118 ms | 154 ms | 118 ms | 153 ms |
| <i>Error steady state</i> | 0.323 RPM | 0.694 RPM | 0.579 RPM | 0.891 RPM |

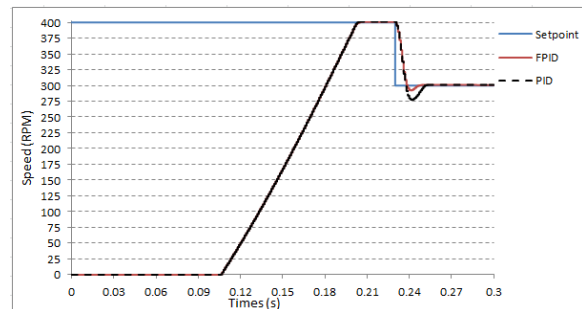


Gambar 15. Respon kecepatan pada 15 Nm



Gambar 16. Respon kecepatan pada 30 Nm

Hasil grafik pada gambar 17, model yang diuji adalah perubahan *setpoint* dengan beban konstan 30 Nm. *Setpoint* awal diatur 400 RPM, dan setelah simulasi berjalan 0,23 detik (dalam kondisi *steady*), *setpoint* dikurangi menjadi 300 RPM. Simulasi dimulai dari kecepatan awal motor 0 RPM untuk mencapai posisi *setpoint*, dengan waktu pengambilan sampel dari 0 hingga 0,3 detik. Respon yang diamati dan dianalisis hanya pada 300 RPM dari kecepatan awal 400 RPM. Kontroler PID memiliki respon *undershoot* sebesar 6,78%, sedangkan untuk kontroler FPID respon *undershoot*nya sebesar 2,88%.



Gambar 17. Respon kecepatan pada beban 30 Nm dengan *setpoint* yang berubah

Berdasarkan hasil pengujian simulasi kecepatan motor induksi seperti yang ditunjukkan pada gambar 15 hingga gambar 17, dapat dianalisa, bahwa kontroler FPID yang dikembangkan memiliki kemampuan untuk mengendalikan kecepatan dengan baik bila dibandingkan dengan kontroler PID, dengan *rise time*, *settling time* relatif cepat, dengan *overshoot* 0%, *error steady state* kurang dari 1 RPM, dan respon *undershoot* 2,88%.

4. Kesimpulan

Berdasarkan hasil simulasi dan pembahasan dapat disimpulkan bahwa desain kontroler FPID untuk kontrol kecepatan motor induksi tiga fasa berdasarkan metode IFOC dapat memberikan hasil yang baik. Hasil uji simulasi dibandingkan antara kontroler FPID dan PID untuk mengevaluasi kinerjanya. Untuk pengujian tanpa beban, hasilnya hampir sama antara kedua kontroler. Sedangkan dalam pengujian dengan beban dinamis, kontroler FPID lebih baik bila dibandingkan dengan kontroler PID. Dimana *overshoot* untuk kontroler FPID adalah 0% sedangkan kontroler PID adalah 0,23% dengan beban 15 Nm dan setpoint 400 RPM. Dengan kecepatan yang sama, respon *undershoot* dengan beban 30 Nm untuk kontroler FPID adalah 2,88% sedangkan kontroler PID adalah 6,78%. Untuk *steady state error* kontroler PID lebih unggul dibandingkan dengan kontroler FPID. Secara umum kontroler FPID menghasilkan respon *error steady state* masih dibawah 1 RPM dan *settling time* didapat hasil yang sama dengan kontroler PID.

Terima kasih

Penulis mengucapkan terimakasih kepada Politeknik

Elektronika Negeri Surabaya dan group riset laboratorium mesin listrik dan kontrol yang telah menyediakan sarana dan prasana selama kegiatan pelaksanaan penelitian

Daftar pustaka

- [1] B. Praharsena, E. Purwanto, A. Jaya, and M. R. Rusli, "Stator Flux Estimator Using Feed-Forward Neural Network for Evaluating Hysteresis Loss Curve in Three Phase Induction Motor," *Emit. Int. J. Eng. Technol.*, vol. 6, no. 1, pp. 168–184, 2018.
- [2] M. A. Hannan *et al.*, "Quantum-Behaved Lightning Search Algorithm to Improve Indirect Field-Oriented Fuzzy-PI Control for IM Drive," *IEEE Trans. Ind. Appl.*, vol. 54, no. 4, pp. 3793–3805, 2018.
- [3] M. A. Hannan, J. A. Ali, A. Mohamed, and A. Hussain, "Optimization techniques to enhance the performance of induction motor drives: A review," *Renew. Sustain. Energy Rev.*, vol. 81, no. April, pp. 1611–1626, 2018.
- [4] I. Ferdiansyah, M. R. Rusli, B. Praharsena, H. Toar, and E. Purwanto, "Speed Control of Three Phase Induction Motor Using Indirect Field Oriented Control Based on Real-Time Control System," *2018 10th Int. Conf. Inf. Technol. Electr. Eng.*, pp. 438–442, 2018.
- [5] H. Aziri, F. A. Patakor, M. Sulaiman, and Z. Salleh, "Comparison Performances of Indirect Field Oriented Control for Three-Phase Induction Motor Drives," *Int. J. Power Electron. Drive Syst.*, vol. 8, no. 4, pp. 1682–1692, 2017.
- [6] A. Saghafinia, H. W. Ping, M. N. Uddin, S. Member, and K. S. Gaied, "Adaptive Fuzzy Sliding-Mode Control into Chattering-Free IM Drive," vol. 9994, no. c, 2014.
- [7] I. Ferdiansyah, E. Purwanto, and N. A. Windarko, "Fuzzy Gain Scheduling of PID (FGS-PID) for Speed Control Three Phase Induction Motor Based on Indirect Field Oriented Control (IFOC)," *Emit. Int. J. Eng. Technol.*, vol. 4, no. 2, pp. 237--258, 2016.
- [8] A. Bennassar, A. Abbou, M. Akherraz, and M. Barara, "Fuzzy Logic Based Adaptation Mechanism for Adaptive Luenberger Observer Sensorless Direct Torque Control of Induction Motor," *J. Eng. Sci. Technol.*, vol. 11, no. 1, pp. 46–59, 2016.
- [9] M. H. Rashid, *Power Electronics Handbook*. 2018.
- [10] S. P. Paul Krause, Oleg Wasynczuk, Scott Sudhoff, *Analysis of Electric Machinery and Drive Systems*. 2013.
- [11] Michael A. Johnson Mohammad H. Moradi, *PID Control New Identification and Design Methods*. London: Springer, 2005.
- [12] J. Jantzen, "Tuning of Fuzzy PID Controllers," *Tech. Univ. Denmark, Dep. Autom.*, vol. 871, no. 98-H, pp. 1–22, 1998.
- [13] P. J. Escamilla-Ambrosio and N. Mort, "A novel design and tuning procedure for PID type fuzzy logic controllers," *2002 1st Int. IEEE Symp.*, vol. 1, no. September, pp. 36–41, 2002.
- [14] N. M. P. J. Escamilla, Ambrosio, "Auto-Tuning of Fuzzy PID Controllers," *Trienn. World Congr.*, vol. 1, no. 1, pp. 1294–1294, 2002.
- [15] Labview, "PID and Fuzzy Logic Toolkit User Manual," no. June, 2009.