

PENGATURAN MOTOR INDUKSI MENGGUNAKAN OBSERVER SELF CONSTRUCTING FUZZY NEURAL NETWORK DENGAN METODE ALGORITMA PELATIHAN LEVENBERG-MARQUARDT

Oleh : Suhariningsih
Pembimbing : Prof. Dr. Ir. H Subagio, MSEE
Prof. Dr. Ir H. Maudhi Hery P, M Eng

ABSTRAK

Dalam penelitian ini dikembangkan pengaturan kecepatan motor induksi 3 phase tanpa sensor yang dioperasikan dengan metoda Field Oriented Vector(FOC). Kecepatan motor diestimasi oleh suatu observer dengan suatu metoda Self Constructing fuzzy Neural Network (SCFNN) dimana pelatihannya menggunakan metode algoritma pelatihan Levenberg-Marquardt(LM), yang menggantikan metode Backpropagasi karena metode ini kurang cepat mencapai konvergen . Metode SCFNN mempunyai kemampuan untuk menggabungkan Fuzzy dan Neural Networks

Hasil simulasi menunjukkan sistem dapat mengestimasi fluksi dan kecepatan dengan kekonvergenan yang lebih cepat dari metode backpropagasi. Hasil estimasi dapat digunakan untuk mengidentifikasi kecepatan rotor motor induksi

Kata kunci: Pengaturan Kecepatan, Motor Induksi tanpa sensor, FOC, SCFNN Observer, metode Levenberg Marquardt.ta

1. PENDAHULUAN

Motor DC merupakan jenis motor yang paling ideal untuk pengemudian elektrik karena kecepatannya dapat diatur dengan mudah dan tidak memerlukan konverter. Kelemahan motor dc adalah harganya relatif mahal, ukurannya relatif besar, adanya komutator dan sikat-sikat dalam motor, sehingga memerlukan perawatan yang rumit dan harus dilakukan rutin. Selama perawatan operasi sistem terhenti, tentu ini sangat tidak dikehendaki dalam industri, karena akan sangat mengganggu proses dan mengurangi hasil (produksi) industri, yang berdampak pada kerugian perusahaan . Kelebihan motor induksi selain kokoh, konstruksinya sederhana juga perawatannya mudah. Kelemahan motor induksi antara lain : merupakan motor yang tidak linier , metode untuk mengatur kecepatan rumit, disamping itu diperlukan suatu konverter yang menimbulkan harmonisa. Namun setelah diketemukan suatu metode Field Oriented Control(FOC) yang mengubah sistem couple menjadi decouple sehingga kesulitan dapat teratasi. Penggunaan metode ini menyebabkan motor induksi berfungsi seperti motor dc penguat terpisah. Dengan demikian motor induksi menggeser peranan motor dc dalam industri..

Dalam sistem pengaturan kecepatan motor induksi yang dioperasikan dengan metode FOC diperlukan suatu sensor kecepatan untuk mengamati nilai kecepatannya. Hasil pengamatan dari sensor kecepatan motor ini dibandingkan dengan kecepatan setpoint, yang kemudian diumpankan ke kontroller untuk bisa mengendalikan kecepatan sehingga sesuai dengan harga setpoint yang diinputkan. Biasanya letak sensor terlalu

jauh dari sistem kontrol maka proses pengamatan sensor pada motor induksi ini hasil pengukuran kecepatannya menjadi kurang akurat . Untuk mengatasi masalah tersebut diperlukan suatu observer yang berfungsi untuk mengamati besar torka dan arus, sehingga kecepatan motor dapat diprediksi. Maka kemudian dikembangkan suatu observer menggunakan metode Self Constructing Fuzzy Neural Network(SCFNN).

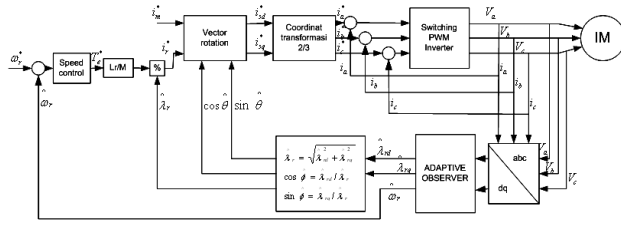
Sasaran dari penelitian ini mengembangkan suatu observer metode Self Constructing Fuzzy Neural Network (SCFNN) dengan metode algorithm learning Levenberg-Marquardt yang menggantikan SCFNN dengan metode Backpropagasi yang dilakukan peneliti sebelumnya [17] untuk mengontrol kecepatan motor induksi 3 phase , sehingga motor mempunyai kinerja yang lebih bagus dengan pencapaian kekonvergenan yang lebih cepat.

Penelitian ini diharapkan memberi kontribusi terhadap upaya untuk terus mengembangkan metode pengaturan kecepatan pada motor induksi sehingga dapat meningkatkan kinerja yang lebih baik dari sistem yang menggunakan motor induksi. Dengan perbaikan kinerja sistem diharapkan meningkatkan efisiensi kerja pada dunia industri.

2. MODELING

2.1 Model Sistem

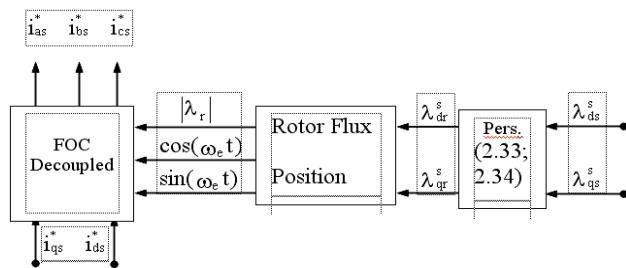
Mengembangkan dari beberapa penelitian yang telah dilakukan oleh Seong-Hwan Kim, dkk. [2], Faa-Jeng Lin, dkk [5] dan Iradiratu DPK. [16], maka blok diagram system yang dikembangkan pada penelitian ini adalah seperti Gambar 1.



Gambar 1. Konfigurasi Sistem Speed-Sensorless Vector Control Untuk Motor Induksi dengan SCFNN [16]

2.2. Field Oriented Control

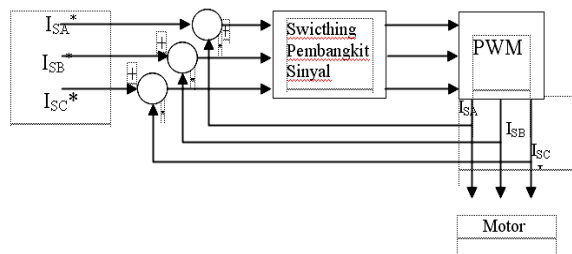
Field Oriented Control (FOC) adalah suatu metode pengaturan medan pada motor ac, dimana dari sistem coupled diubah menjadi sistem decoupled. Dengan sistem ini arus penguatan dan arus beban motor dapat dikontrol secara terpisah, dengan demikian torka dan fluksi juga dapat diatur secara terpisah. Diagram blok yang menggambarkan prinsip dasar sistem decoupled field oriented control (FOC Decoupled) motor induksi ditunjukkan pada gambar 2 [13].



Gambar 2. Diagram Blok FOC Decoupled Motor Induksi

2.3. Model Inverter

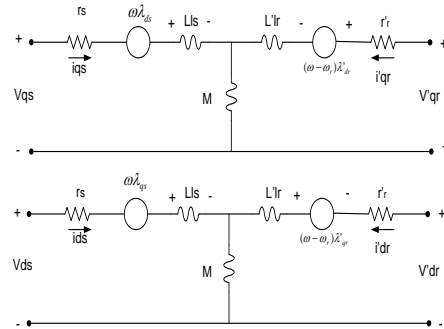
Vektor rotasi terhadap arus magnetisasi dan arus torsi menghasilkan arus fase referensi yang digunakan untuk sinyal control PWM inverter. Tegangan yang dihasilkan inverter akan digunakan oleh stator motor induksi. Model PWM inverter ditunjukkan pada Gambar 3.



Gambar 3. PWM Inverter

2.4. Model Motor Induksi

Rangkaian ekivalen motor induksi dalam koordinat d-q dapat dilihat pada Gambar 4.



Gambar 4. Rangkaian Ekivalen Motor Induksi Dalam Koordinat d-q

Rangkaian ekivalen motor induksi dalam koordinat d-q, dengan memasukkan tegangan rotor ($V_r = 0$), maka didapat besaran tegangan stator yang merupakan fungsi dari arus stator dan arus rotor dalam bentuk matrik, sebagai berikut :

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s + pL_s & -\omega_s L_s & pM & -\omega_s M \\ \omega_s L_s & R_s + pL_s & \omega_s M & pM \\ pM & -(\omega_s - \omega_r)M & R_r + pL_r & -(\omega_s - \omega_r)L_r \\ (\omega_s - \omega_r)M & pM & (\omega_s - \omega_r)L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} \quad (1)$$

dengan : $p = \frac{d}{dt}$

jika ditinjau pada koordinat stationer ($\omega_s = 0$), maka persamaan (1) menjadi:

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s + pL_s & 0 & pM & 0 \\ 0 & R_s + pL_s & 0 & pM \\ pM & \omega_r M & R_r + pL_r & \omega_r L_r \\ -\omega_r M & pM & -\omega_r L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} \quad (2)$$

3. SELF CONSTRUCTING FUZZY NEURAL NETWORK DENGAN METODE PELATIHAN LEVENBERG MARQUARDT

Kontroler ini masukannya adalah sebuah kontroler fuzzy, sehingga masukannya adalah data numeric berupa nilai error.. Struktur dasar dari sebuah fuzzy neural network adalah seperti pada Gambar 5.[8]

Pada lapisan pertama hanya terjadi proses masukan berupa data crisp yaitu error (X_1) dan Delta error (X_2) untuk meneruskan sinyal ke lapisan berikutnya.

Pada lapisan ke dua terjadi proses fuzzyfikasi dan pembentukan membership fungsi. Fungsi yang dipergunakan adalah fungsi Gaussian.

$$u_{A_i} = \exp\left(-\frac{(x_i - m_{ji})^2}{\sigma_{ji}^2}\right) \quad (3)$$

dengan m_{ji} dan σ_{ji} adalah rata-rata (*mean*) dan standar deviasi.

Pada Lapis ketiga, merupakan penentuannya kondisi awal dari aturan fuzzy. Langkah ini adalah untuk memperoleh hasil perkalian antara semua komponen input dari error dan delta error dengan persamaan :

Untuk j th rule node persamaan:

$$u_j = u_{A_j}(x_1)u_{A_j}(x_2) \cdots u_{A_j}(x_n) = \prod_i u_{A_j}(x_i) \quad (4)$$

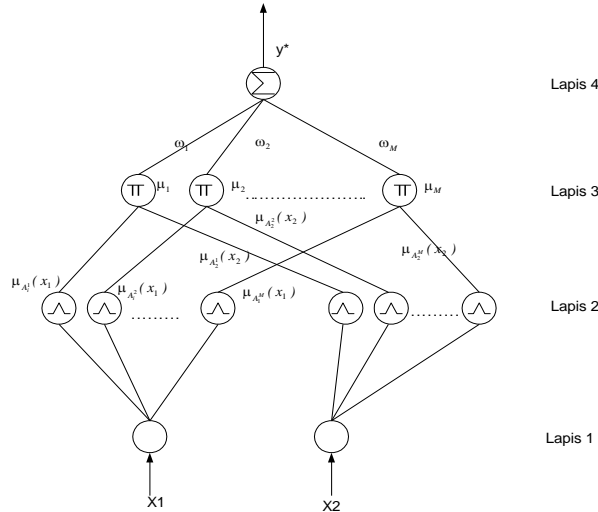
Dengan u_j adalah output node rule ke- j .

Lapis ke empat: Berfungsi untuk menjumlahkan seluruh sinyal masukan yang disimbulkan dengan Σ kemudian dirumuskan dalam persamaan y^* yang selanjutnya dilakukan proses defuzzikasi [8];

$$y^* = \sum_{j=1}^M w_j u_j \quad (5)$$

3.1 Algoritma On-Line pada SCFNN dengan metode pelatihan Levenberg Marquart

Pada SCFNN terdapat dua jenis tipe algoritma learning yaitu learning struktur dan learning parameter. Learning struktur digunakan untuk mencari space input fuzzy logic partition dan fuzzy logic subject yang bertujuan : meminimalkan jumlah rule dan meminimalkan fuzzy set dalam semesta pembicaraan dari setiap variabel input. Learning parameter menggunakan algoritma supervised learning, sedangkan untuk menentukan pembobot dan parameter dari membership function diatur dengan learning algoritma backpropagasi.



Gambar 5. Struktur dasar SCFNN

3.2 Phase Learning Struktur

Langkah pertama pada learning struktur adalah menentukan perlu tidaknya melakukan learning struktur. Jika $e_{\min} \leq |e|$ atau $\Delta e_{\min} \leq |\Delta e|$. Dengan e_{\min} dan Δe_{\min} adalah konstanta positif, maka learning struktur diperlukan. Selanjutnya menentukan node baru (membership function) pada layer 2 dan menghubungkan fuzzy logic rule pada layer 3. Jika adanya satu cluster diberikan pada input akan menyebabkan adanya sebuah rule fuzzy logic pada layer 3, maka persamaan kekuatan penyulutan (*firing strength*) dari sebuah rule untuk setiap data masukan x_i dapat ditunjukkan sebagai sudut dimana

data masukan memiliki hubungan terhadap cluster data. *Firing strength* diperoleh dari (4) yang digunakan sebagai pengukuran sudut :

$$D_j = u_j \quad j = 1, \dots, Q(t) \quad (6)$$

Dengan $Q(t)$ adalah jumlah rule yang ada pada waktu t . Kriteria pembentukan fuzzy rule baru untuk data masukan baru dinyatakan sbb. Dengan menentukan pengukuran sudut maksimum D_{\max}

$$D_{\max} = \max_{1 \leq j \leq Q(t)} D_j \quad (7)$$

Jika $D_{\max} \leq \bar{D}$, maka dibentuk membership function dengan $D \in (0,1)$. Kemudian mean dan standar deviasi dari membership function yang baru dinyatakan lebih dulu dengan nilai tertentu secara heuristik atau cara lain. Jadi mean dan standar deviasi dari membership function baru sbb:

$$m_i^{(new)} = x_i \quad (8)$$

$$\sigma_i^{(new)} = \sigma_i \quad (9)$$

dengan x_i adalah data masukan yang baru dan σ_i adalah standar deviasi.

Untuk menghindari membership function yang baru sama dengan yang telah ada, kesamaan antara membership function yang lama dan baru harus diperiksa, yaitu dengan asumsi bahwa jika terdapat dua fuzzy set A dan B dengan membership function adalah $u_A(x) = \exp [-(x - m_1)^2/\sigma_1^2]$ dan $u_B(x) = \exp [-(x - m_2)^2/\sigma_2^2]$. Dan asumsikan $m_1 \geq m_2$.

Kemudian $|A \cap B|$ dapat dihitung :

$$|A \cap B| = \frac{1}{2} \frac{h^2(x) [m_2 - m_1 + \sqrt{\pi}(\sigma_1 + \sigma_2)]}{\sqrt{\pi}(\sigma_1 + \sigma_2)} + \frac{1}{2} \frac{h^2(x) [m_2 - m_1 + \sqrt{\pi}(\sigma_1 + \sigma_2)]}{\sqrt{\pi}(\sigma_1 + \sigma_2)} + \frac{1}{2} \frac{h^2(x) [m_2 - m_1 + \sqrt{\pi}(\sigma_1 + \sigma_2)]}{\sqrt{\pi}(\sigma_1 + \sigma_2)} \quad (10)$$

dengan $h(x) = \max\{0, x\}$.

$$E(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{\sigma_1 \sqrt{\pi} + \sigma_2 \sqrt{\pi} |A \cup B|^*} \quad (11)$$

Pemeriksaan dilakukan pada semua variabel input x_i . Sedangkan nilai maksimumnya E_{\max} didapat dengan:

$$E_{\max} = \max_{1 \leq j \leq Q(t)} \{u(m_j^{(new)}, \sigma_j^{(new)}), u(m_{j1}, \sigma_{j1})\} \quad (12)$$

dengan $u(m_{j1}, \sigma_{j1})$ adalah membership function Gaussian dengan mean m_{j1} dan standar deviasi σ_{j1} ; $M(t)$ adalah jumlah membership function ke- i dari variabel input. Jika $E_{\max} \leq \bar{E}$ dengan $\bar{E} \in (0,1)$ adalah nilai yang sudah ditentukan, maka gunakan membership function baru dan jumlah $M(t)$ adalah:

$$M(t+1) = M(t) + 1 \quad (13)$$

Jadi pembentukan membership function berhubungan dengan pembentukan rule fuzzy baru dan pembobot $\omega^{(new)}$.

3.3 Phase Learning Parameter

Algoritma learning parameter dari SCFNN adalah menentukan rule yang adaptif untuk mengatur parameter-parameter jaringan, berdasarkan pasangan input-output. Jika parameter jaringan terdiri dari vektor parameter, maka proses learning memperhitungkan vektor determinasi dari fungsi energi. Metode ini umumnya berdasar rule learning backpropagasi karena vektor gradien dihitung dalam arah berlawanan terhadap keluaran setiap node, untuk menjelaskan algoritma learning parameter SCFNN menggunakan metode supervised gradient decent. Asumsikan fungsi energi E didefinisikan sebagai:

$$E = \frac{1}{2}(\omega_m - \omega_r)^2 = \frac{1}{2}e_m^2 \quad (14)$$

Kemudian algoritma learning parameter berdasarkan backpropagasi dijelaskan sebagai berikut [8]:

Layer 4: Bentuk error dipropagasi dihitung sebagai:

$$\delta^4 = -\frac{\partial E}{\partial y^*} = \left[-\frac{\partial E}{\partial e_m} \frac{\partial e_m}{\partial y^*} \right] = \left[-\frac{\partial E}{\partial e_m} \frac{\partial e_m}{\partial \omega_r} \frac{\partial \omega_r}{\partial y^*} \right] \quad (15)$$

dan pembobot diupdate besarnya

$$\Delta \omega_j = -\eta_\omega \frac{\partial E}{\partial \omega_j} = \left[-\eta_\omega \frac{\partial E}{\partial y^*} \right] \left(\frac{\partial y^*}{\partial \omega_j} \right) = -\eta_\omega \delta^4 u_j \quad (16)$$

dengan faktor η_ω adalah parameter learning-rate dari pembobot. Pembobot pada layer 4 diupdate sesuai persamaan:

$$\omega_j(N+1) = \omega_j(N) + \Delta \omega_j \quad (17)$$

dengan N jumlah iterasi dari j th

Layer 3: Pada layer ini hanya error yang perlu dihitung dan dipropagasi:

$$\delta_j^3 = -\frac{\partial E}{\partial u_j} = \left[-\frac{\partial E}{\partial y^*} \right] * \left[\frac{\partial y^*}{\partial u_j} \right] = \delta^4 \omega_j \quad (18)$$

Layer 2: Error dihitung sbb:

$$\delta_{ji}^2 = -\frac{\partial E}{\partial u_{A_j}} = \left[-\frac{\partial E}{\partial y^*} \frac{\partial y^*}{\partial u_j} \right] \left[\frac{\partial u_j}{\partial u_{A_j}} \right] = \delta_j^3 \quad (19)$$

Hukum update dari m_{ji} adalah:

$$\begin{aligned} \Delta m_{ji} &= -\eta_m \frac{\partial E}{\partial m_{ji}} = \left[-\eta_m \frac{\partial E}{\partial u_{A_j}} \frac{\partial u_{A_j}}{\partial m_{ji}} \right] \\ &= \eta_m \delta_{ji}^2 \frac{2(x_i^2 - m_{ji})}{(\sigma_{ji})^2} \end{aligned} \quad (20)$$

Hukum update dari σ_{ji} adalah:

$$\begin{aligned} \Delta \sigma_{ji} &= -\eta_\sigma \frac{\partial E}{\partial \sigma_{ji}} = \left[-\eta_\sigma \frac{\partial E}{\partial u_{A_j}} \frac{\partial u_{A_j}}{\partial \sigma_{ji}} \right] \\ &= \eta_\sigma \delta_{ji}^2 \frac{2(x_i^2 - m_{ji})}{(\sigma_{ji})^2} \end{aligned} \quad (21)$$

dengan η_m dan η_σ adalah parameter learning-rate dari mean dan standar deviasi fungsi Gaussian. Mean dan standar deviasi dari membership function pada layer ini diupdate dengan:

$$m_{ji}(N+1) = m_{ji}(N) + \Delta m_{ji} \quad (22)$$

$$\sigma_{ji}(N+1) = \sigma_{ji}(N) + \Delta \sigma_{ji} \quad (23)$$

Setelah didapatkan persamaan persamaan ini maka disimulasikan rangkaian kontrol dengan Self Constructing Fuzzy Neural Networks dengan plant motor induksi.

Pada penelitian ini metode pelatihan Backpropagasi diganti dengan metode pelatihan Levenberg-Merquard. Metode (LM). Metode pelatihan LM ini merupakan kombinasi algoritma Newton dengan Steepest Decent. Bila metode Gradient Descent dinyatakan sebagai persamaan :

$$W_{kj}(t+1) = W_{kj}(t) + \alpha \cdot \delta_k \cdot Z_j \quad (24)$$

Persamaan di atas dapat disederhanakan menjadi

$$W_{k+1} = W_k + \alpha \cdot g \quad (25)$$

Dimana g adalah vektor gradien. Bentuk persamaan Newthonnya adalah :

$$W_{k+1} = W_k - A_k^{-1} \cdot g \quad (26)$$

A_k adalah matrik Hessian(elemennya adalah turunan kedua galat terhadap penimbang) sebagai berikut

$$A = \begin{bmatrix} \frac{\partial^2 E}{\partial W_1^2} & \frac{\partial^2 E}{\partial W_1 \partial W_1} & \cdots & \cdots & \frac{\partial^2 E}{\partial W_n \partial W_1} \\ \frac{\partial^2 E}{\partial W_1 \partial W_2} & \frac{\partial^2 E}{\partial W \partial W_2^2} & \cdots & \cdots & \frac{\partial^2 E}{\partial W_n \partial W_2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 E}{\partial W_1 \partial W_n} & \frac{\partial^2 E}{\partial W_2 \partial W_n} & \cdots & \cdots & \frac{\partial^2 E}{\partial W \partial W_n^2} \end{bmatrix} \quad (27)$$

Matrik A dapat ditulis menjadi :

$$A = 2J^T J \quad (28)$$

dimana :

J adalah matrik Jacobian

Persamaan perbaikan penimbang metode pelatihan LM adalah :

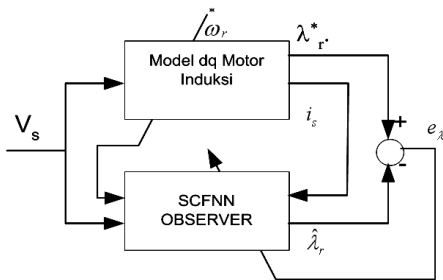
$$W_{k+1} = W_k - (J_k^T J_k + \mu I)^{-1} J_k^T e \quad (29)$$

Bila nilai $\mu = 0$, maka metode pelatihan LM akan sama dengan metode Gauss Newthon, sedangkan bila μ maka metode pelatihan LM akan sama dengan Backpropagasi(steepest descent) [2].

Setelah didapatkan persamaan persamaan ini maka dapat disimulasikan ke rangkaian kontrol dengan Self Constructing Fuzzy Neural Networks metode pelatihan Levenberg Marquardt dengan plant motor induksi.

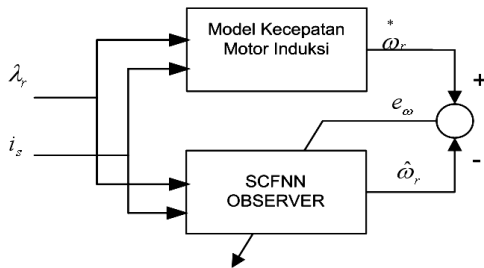
4. BLOK DIAGRAM SELF CONSTRUCTING FUZZY NEURAL NETWORK OBSERVER (SCFNNO)

Blok diagram system kendali kecepatan motor induksi tanpa sensor kecepatan dari gambar 1, pada bagian adaptive observer menggunakan self constructing fuzzy neural network, seperti terlihat pada Gambar 6.



Gambar 6. Struktur Estimasi Fluksi

Gambar 6, struktur estimasi untuk mendapatkan estimasi fluksi yang masing-masing terdiri dari fluksi direct λ_{dr} dan fluksi quadratur λ_{qr} , hasil dua parameter ini digunakan untuk input estimasi kecepatan, terlihat pada Gambar 7.



Gambar 7. Struktur Estimasi Kecepatan

Parameter motor induksi untuk data pembelajaran SCFNN yang digunakan dalam mendapatkan target:

- Fluksi direct λ_{dr} terdiri I_{ds} , V_{ds} , V_{qs} dan ω_r
- Fluksi quadratur λ_{qr} terdiri I_{qs} , V_{qs} , V_{qs} dan ω_r
- Kecepatan ω_r terdiri λ_{dr} , λ_{qr} , I_{ds} , dan I_{qs} .

4.1. Pembelajaran Off-Line

Metode pembelajaran untuk estimasi fluksi identifikasi kecepatan motor induksi tiga fasa menggunakan self constructing fuzzy neural network dimana jaringannya terdiri dari empat layer, yaitu input sebanyak 4, linguistic, precondition dan outputnya 1.

Linguistic, precondition dan output untuk mendapatkan nilai λ_{dr} , λ_{qr} dan ω_r .

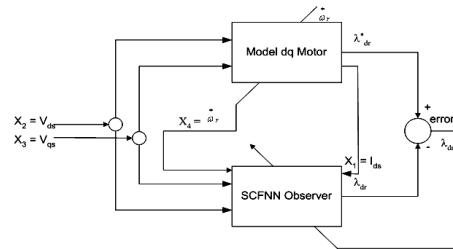
Proses pembelajaran menggunakan 4 neuron input yaitu

V_s , I_s , ω_r dan λ_r , pembelajaran dilakukan sebanyak 5 epoch. Apabila hasil pembelajaran belum konvergen atau tidak sesuai target maka akan terjadi penambahan membership fungsi baru, penambahan akan berhenti bila hasil pembelajaran konvergen.. Harga awal pembobot ditentukan antara 0 dan 1 untuk mencari parameter optimal yang menghasilkan kinerja yang terbaik.

Dalam proses estimasi ada tiga SCFNN untuk menyelesaikan estimasi fluksi direct λ_{dr} , fluksi quadratur λ_{qr} , setelah selesai proses ini ada satu SCFNN menyelesaikan estimasi kecepatan ω_r .

4.2. Pembelajaran Off-line Fluksi Direct λ_{dr}

Pembelajaran Off-Line self constructing fuzzy neural network observer untuk identifikasi fluksi direct λ_{dr} , ditunjukkan pada Gambar 8.



Gambar 8. Struktur Estimasi Fluksi Direct λ_{dr} Menggunakan SCFNNO

Gambar 8, menggambarkan tentang struktur dari estimator fluksi direct λ_{dr} menggunakan SCFNN, dan input terdiri dari I_{ds} , V_{ds} , V_{qs} dan ω_r , masuk ke blok SCFNNO.

Pada blok model motor, menentukan nilai kecepatan

(kecepatan referensi ω_r) sehingga didapatkan fluksi direct referensi λ_{dr}^* . Untuk bagian SCFNNO menghasilkan fluksi direct pembelajaran $\hat{\lambda}_{dr}$, selisih nilai referensi dan pembelajaran didapatkan error atau fluksi direct estimasi.

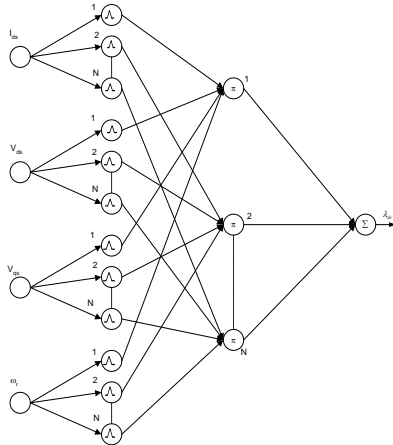
Output dari SCFNNO didefinisikan sebagai fluksi direct pembelajaran $\hat{\lambda}_{dr}$, yang kemudian digunakan sebagai input yang dapat diubah-ubah.

Jika fluksi direct yang diestimasi merupakan deviasi dari fluksi direct sesungguhnya dan error hubungan antara flux dari model fluksi direct pembelajaran $\hat{\lambda}_{dr}$ dan fluksi direct referensi λ_{dr}^* . Maka error merupakan backpropagasi dari SCFNN dan pembebanan dari SCFNN adalah adjusted on line untuk mengurangi error. Akhirnya,

output dari SCFNN merupakan model fluksi direct yang sesungguhnya.

4.3. Internal Struktur Dari SCFNN

Gambar 9, menggambarkan tentang struktur dari SCFNN dan dapat dilatih dengan algoritma backpropagasi, untuk mendapatkan hasil SCFNN yang mengikuti kecepatan yang sesungguhnya.



Gambar 9. Internal Struktur SCFNN

Jumlah SCFNN blok ada 3, yaitu blok untuk SCFNN λ_{dr} , λ_{qr} dan ω_r , blok SCFNN untuk λ_{dr} dan λ_{qr} dihitung dulu, baru hasilnya dimasukkan ke blok SCFNN ω_r .

Pada saat pembelajaran, jumlah rule yang tercipta untuk masing-masing input bisa berbeda dari rule yang tercipta untuk satu blok SCFNN, karena karakteristik I_{ds} , I_{qs} , V_{ds} , V_{qs} dan ω_r tidak sama.

5. DATA MOTOR INDUKSI TIGA FASA

Data motor induksi tiga fasa yang dipakai untuk simulasi adalah:

- $R_s = 176$ resistansi stator (ohm) Tegangan 115 Volt
- $R_r = 190$ resistansi rotor (ohm) Jumlah pasang kutub 2
- $L_s = 3,79$ induktansi stator (H) Frekuensi 60 Hz
- $L_r = 3.31$ induktansi rotor (H)
- $M = 3.21$ induktansi gandeng (H)
- $J = 0.0000105$ momen enersia (Kg.m^2)
- $K_d = 1.9e-5$ konstante gesek ($\text{Kg.m}^2/\text{s}$)

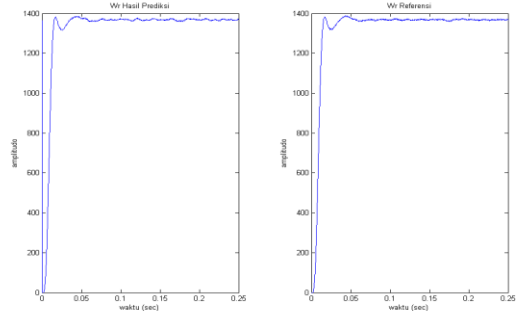
6. SIMULASI

Gambar 5.1 sampai Gambar 5.18 merupakan hasil simulasi yang diperoleh untuk estimasi fluksi dan kecepatan rotor dengan menggunakan bahasa pemrograman C^{++} . Performansi yang ditunjukkan menampilkan pengambilan data pada kecepatan referensi 1350 rpm. Gambar yang ditampilkan adalah hasil observer menggunakan SCFNN pelatihan Levenberg

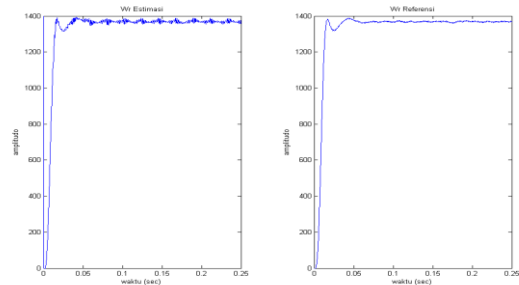
Marquardt dan SCFNN pelatihan Backpropagasi yang telah dilakukan peneliti sebelumnya [17].

(a) $\hat{\omega}_r$ Prediksi (b) ω_r^* Output Model Motor

Gambar 14. Bentuk Amplitudo Fungsi Waktu Untuk Kecepatan,



dengan SCFNN Levenberg Marquardt Nilai MSE 0,017175 %



(a) $\hat{\omega}_r$ Prediksi (b) ω_r^* Output Model Motor

Gambar 15. Bentuk Amplitudo Fungsi Waktu Untuk Kecepatan, dengan SCFNN Backpropagasi Nilai MSE 0,030675%

Nilai MSE hasil simulasi SCFNN pelatihan LM untuk untuk kecepatan rotor $\hat{\omega}_r$ dengan kecepatan referensi 1350 rpm (0,017175 %) lebih kecil dari dengan pelatihan BP (0,030675%) .

Tabel 5.1 Pengujian SCFNN dengan Pelatihan Levenberg Marquardt dan SCFNN dengan Pelatihan Backpropagasi untuk beberapa Data yang Berbeda

| NO | Kecepatan referensi (rpm) | Hasil prediksi pada | SCFNN Observer | | | |
|----|---------------------------|---------------------|---------------------|----------|---------------|----------|
| | | | Levenberg Marquardt | | Backpropagasi | |
| | | | Jumlah epoch | MSE (%) | Jumlah epoch | MSE (%) |
| 1 | 1350 | λ_{dr} | 2 | 0,010360 | 50 | 0,020035 |
| | | λ_{qr} | 2 | 0,019923 | 50 | 0,081165 |
| | | ω_r | 2 | 0,017175 | 50 | 0,030675 |
| 2 | 750 | λ_{dr} | 2 | 0,003539 | 50 | 0,056819 |
| | | λ_{qr} | 2 | 0,024703 | 50 | 0,080676 |
| | | ω_r | 2 | 0,021564 | 50 | 0,036961 |
| 3 | 500 | λ_{dr} | 2 | 0,027750 | 50 | 0,060614 |
| | | λ_{qr} | 2 | 0,018828 | 50 | 0,125130 |
| | | ω_r | 2 | 0,020052 | 50 | 0,049025 |

6. Kesimpulan

Dari pengamatan dan analisis hasil simulasi SCFNN

pelatihan Levenberg Marquardt pada penelitian ini dan kemudian dibandingkan dengan SCFNNO pelatihan Backpropagasi yang telah dilakukan peneliti sebelumnya, maka dapat disimpulkan:

1. Pada simulasi pelatihan pada simulasi SCFNNO Levenberg Marquardt diperlukan jumlah epoch yang lebih sedikit dengan Error yang lebih kecil dibanding dengan SCFNNO Backpropagasi, ini artinya metode SCFNNO pelatihan Levenberg Marquardt lebih cepat konvergen dibanding dengan SCFNNO dengan pelatihan Backpropagasi.
2. Simulasi estimasi kecepatan ($\hat{\omega}_r$) dengan SCFNNO pelatihan Levenberg Marquardt dengan pengambilan referensi kecepatan yang berbeda (yaitu 1350 rpm, 750 rpm, 500 rpm) nilai Mean Square Error (MSE) kisarannya antara 0,017175% sampai 0,021564%, dengan SCFNNO pelatihan.

Dari data tersebut dapat disimpulkan bahwa nilai MSE SCFNNO pelatihan Levenberg Marquardt mempunyai kisaran nilai yang lebih rendah dari SCFNNO pelatihan Backpropagasi.

Hal ini menunjukkan bahwa Self Constructing Fuzzy Neural Network Observer yang dirancang mampu mengestimasi fluksi dan kecepatan rotor dengan baik karena nilai yang diijinkan Standart Error Estimasi (SEE) kurang dari 5%,

Dengan demikian metode SCFNNO pelatihan Levenberg Marquardt yang dirancang mempunyai performen yang lebih baik dari pada SCFNNO pelatihan Backpropagasi sehingga dapat digunakan sebagai piranti alternatif untuk identifikasi kecepatan rotor motor induksi.

DAFTAR PUSTAKA

- [1]. Boldea Ion, Nazar Syed A, Vector Control AC Drives, CRC Press, Inc.1992
- [2]. Seong-Hwan Kim, Tae-Sik Park, Ji-Yoon Too, and Gwi-Tae Park, Speed-Sensorless Vector Control Of An Induction Motor Using Neural Network Speed Estimation, IEEE Trans. On Industry Application, Vol. 48, No. 3, June 2001.
- [3]. K. S.Narendra and K. Parthasarathy, Identification And Control Of Dynamical Systems Using Neural Networks, IEEE Trans Neural Networks, vol. 1, pp. 4-27, Mar.1990.
- [4]. F.J.Lin, W.J. Huang, and R.J.Wai, A Supervisory Fuzzy Neural Network Control System For Tracking Periodic Inputs, IEEE Trans. Fuzzy Syst, vol.7, pp.41-52, Feb.1999
- [5]. T.Fukuda and T. Shibata, Theory And Applications Of neural Network For Industrial Control System, IEEE Trans.Ind.Electron, Vol. 39, pp. 472-489, Dec 92
- [6]. C. F. Juang and C. T.Lin, An On - Line Self - Constructing Neural Fuzzy Inference Network And Its Application, IEEE Trans. Fuzzy Syst, vol.6, pp. 12- 32, Feb. 1998
- [7]. C. T. Lin, A Neural Fuzzy Control System With Structure And Parameter Learning, Fuzzy Sets Syst., vol. 70, no. 2-3, pp. 183-212, Mar. 1995
- [8]. Faa-Jeng Lin, and Chih-Hong Lin, A Permanent-magnet Synchronous Motor Servo Drive Using Self -Constructing fuzzy Neural Network Controller, IEEE Trans. On Energy Conversion, Vol. 19, No. 1, March 2004.
- [9]. Faa Jeng Lin, Rong Jong Wai, Chih-Hong Lin dan Da Chung Liu" Decouple Stator-Flux Oriented Induction Motor Drive with Fuzzy Neural Networks Uncertainty Observer, IEEE Trans. On Industrial Electronics, Vol.47, No. 2, pp.356 - 367, April, 2000.
- [10]. R.Krishnan, Virginia Tech. Blacksburg, Va Electric Motr Drives Modeling, Analysis and Control Prentice Hall International. Inc. New Jersey, 2001
- [11]. Soebagio, Era Purwanto, Algoritma genetika Untuk Optimasi Penentuan parameter Motor Induksi Dengan model d-q, SMED, 13 Juli 2000, hal. II-1
- [12]. Soebagio, Mohammad Zuhri, Pengaruh Perubahan Parameter Terhadap Kinerja Motor Induksi Rotor Sangkar Dan Rotor Belit, Proc. SITIA, jurusan Tek.Elektro ITS, 2 Mei 2005, hal. 63-67.
- [13]. Purwanto Era, Studi Pengaturan Motor Induksi Dengan Metoda Vektor, Master Tesis, Universitas Shizuoka Jepang, 1995.
- [14]. Mauridhi Hery P, Agus Kurniawan, Supervised Neural Networks dan aplikasinya, Graha Ilmu, 06.
- [15]. Soebagio, Peran Pengemudian Elektris dalam menghadapi Kompetisi Global Penguatan Guru Besar, Surabaya, Januari, 2003.
- [16]. Iradiratu, Mauridhi Hery Purnomo, Era purwanto, Perancangan Model Observer Untuk Identifikasi Kecepatan Motor Induksi, Proceedings SMED, Sep02
- [17]. Sutedjo, Soebagio dan Mauridhi Hery Purnomo, Kendali Kecepatan Motor Induksi Tanpa Sensor Kecepatan Menggunakan Self Constructing Fuzzy Neural Network, Seminar Nasional XII FTI-Its 29 Maret 2005, hal. 414.
- [18]. Wiryajati, K, dkk, Kontroller Konvensional sebagai model referensi pada pengaturan motor Induksi berbasis Field Oriented Control, CECI&SITIA, Proc. Control and Intelligent Technology Application, D40-D43, Surabaya, 2003.