

Pengaturan Kecepatan Motor Induksi Tanpa Sensor Kecepatan dengan Metoda *Direct Torque Control* Menggunakan *Observer Recurrent Neural Network*

Epyk Sunarno^{1,2)}, Soebagio²⁾, Mauridhi Heri Purnomo²⁾

¹⁾Politeknik Elektronika Negeri Surabaya- ITS

²⁾Jurusan Teknik Elektro FTI, Institut Teknologi Sepuluh Nopember Surabaya

Kampus ITS Keputih Sukolilo Surabaya 60111

Email : epyk@eepis-its.edu

ABSTRAK

Penelitian ini membahas pengembangan kontrol pada kecepatan motor induksi tiga fasa tanpa sensor kecepatan (*speed sensorless*) yang dioperasikan dengan metoda *Direct Torque Control* (DTC). Kecepatan motor induksi diidentifikasi oleh suatu *observer*. Estimasi kecepatan motor oleh *observer* memerlukan masukan arus dan tegangan stator. *Observer* untuk identifikasi kecepatan motor menggunakan metode *Artificial Neural Network* (ANN) dengan algoritma pembelajaran menggunakan *Recurrent Neural Network* (RNN). Hasil simulasi menggunakan matlab-simulink menunjukkan saat motor diberikan kecepatan referensi 77,9743 rad/detik terjadi *overshoot* 7,0224% , *rise time* 0,0125 detik dan *settling time* 0,364 detik.

Kata Kunci: *direct torque control, speed sensorless, recurrent neural network*

ABSTRACT

This paper describes about development of sensorless control for three phase induction motor speed which is operated by Direct Torque Control (DTC). Induction motor speed is identified by an Observer. Current supply and Stator Voltage are required by Observer to gain Motor Speed Estimation. Observer for motor speed identification is developed using Artificial Neural Network (ANN) Method and Recurrent Neural Network (RNN) learning algorithm. The simulation results using MathLab/Simulink show that on PI controller with Recurrent Neural Network (RNN) observer, there are the overshoot 7,0224%, rise time 0,0125 second and settling time 0,364 second with reference speed 77,9743 rad./sec.

Keywords: *direct torque control, speed sensorless, recurrent neural network*

PENDAHULUAN

Motor induksi banyak digunakan di industri, pusat bisnis transportasi dan aplikasi lain. Konstruksi yang kokoh (*robust*) merupakan keuntungan yang luar biasa dari motor induksi disamping harga yang murah dan mudah untuk merawatnya.

Untuk mengatur kecepatan motor induksi pada kecepatan tetap dan kecepatan variabel diperlukan konverter daya. Apabila motor induksi mendapat tegangan melalui konverter daya, maka bentuk gelombang tegangan tidak lagi sinusoida. Dalam kondisi ini, pemodelan motor tidak lagi menggunakan model trafo.

Pemodelan dilakukan dengan menggunakan pemodelan dalam koordinat d-q-n untuk melakukan analisa. Model ini lebih fleksibel dibandingkan dengan model motor induksi menggunakan model trafo. Bentuk tegangan sumber tidak harus sinusoida dan parameter mesin bisa diubah[1]. Selain digunakan untuk analisa kondisi *steady state* pemodelan motor induksi dalam koordinat d-q-n juga bisa untuk kondisi *transient*. Selain itu konverter daya ini juga difungsikan untuk mengatur parameter-parameter motor induksi, sehingga dengan mengatur parameter motor induksi akan dapat mempengaruhi kecepatan putar motor. Dengan pengaturan kecepatan motor akan dapat memperbaiki kualitas, kuantitas produk, dan membuat umur motor lebih panjang. Ada beberapa strategi yang dipakai dalam mengatur kecepatan motor induksi diantaranya pengaturan

Catatan: Diskusi untuk makalah ini diterima sebelum tanggal 1 Desember 2008. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 9, nomor 1, Maret 2009.

tegangan, pengaturan frekuensi, pengaturan tahanan rotor untuk rotor lilit dan pengaturan tidak langsung. Pengaturan tidak langsung diantaranya adalah pengaturan fluksi melalui V/f konstan, pengaturan arus slip dengan menggunakan pengaturan frekuensi variabel, pengaturan fluksi melalui arus stator dan slip frekuensi, *Field Oriented Control* (FOC) dan *Direct Torque Control* (DTC).

Dari beberapa strategi pengaturan kecepatan motor induksi dan beberapa kelemahan di atas, maka dalam penelitian ini penulis mengajukan strategi pengaturan motor induksi melalui kontrol vektor torka (*Direct Torque Control*) estimasi putaran dari persamaan *state* yang diharapkan dari estimasi ini menghasilkan nilai putaran, torka dan fluks stator.

Sasaran dari penelitian ini adalah pengaturan kecepatan motor induksi tanpa sensor dengan metoda *Direct Torque Control* menggunakan observer *Recurrent Neural Network*. Metoda ini digunakan untuk memperbaiki putaran motor induksi pada kondisi start, beban berubah, dan referensi putaran berubah, sehingga dengan kondisi perubahan di atas kecepatan menjadi linier. Identifikasi arus dan tegangan stator untuk memisahkan fluks dan torka adalah melalui observer, menggunakan metode Neural Network (NN) dengan pembelajaran menggunakan *Recurrent Neural Network* (RNN). Hasil output putaran estimator dibandingkan dengan putaran referensi digunakan oleh kontrol PI untuk mengontrol torka. Torka, fluks stator, dan sinyal sudut fluks stator dari estimator digunakan untuk membangkitkan sinyal pensaklaran ke rangkaian inverter, untuk menggerakkan motor induksi.

DIRECT TORQUE CONTROL (DTC)

Ide dasar dari DTC adalah perubahan torka sebanding dengan slip antara fluks stator dan fluks rotor pada kondisi fluks bocor stator tetap[2]. Hal ini banyak dikenali untuk pengaturan torsi dan fluks cepat dan *robust*. Perubahan cepat dari torsi elektromagnetik dapat dihasilkan dari putaran fluks stator, sebagai arah torsi. Dengan kata lain fluks stator dapat seketika mempercepat atau memperlambat dengan menggunakan vektor tegangan stator yang sesuai. Torsi dan fluks kontrol bersama-sama dan *decouple* dicapai dengan pengaturan langsung dari tegangan stator,

dari error respon torsi dan fluks. DTC biasanya digunakan sesuai vektor tegangan dalam hal ini untuk memelihara torsi dan fluks stator dengan dua daerah histerisis, yang menghasilkan perilaku dan variasi prosedur frekuensi pensaklaran dan *ripple* fluks, torsi dan arus yang penting.

Untuk menentukan putaran motor dapat digunakan persamaan rangkaian motor induksi. Tegangan dan arus motor diukur pada *reference frame* yang dapat dipilih dalam *stasioner frame*, maka persamaan tegangan stator dalam *referensi frame* diberikan[1] dengan persamaan:

$$\bar{\lambda}_{dr} = \frac{L_r}{L_m} v_{ds} - \frac{L_r}{L_m} \left(R_s + \sigma L_s \frac{d}{dt} \right) i_{ds} \quad (1)$$

$$\bar{\lambda}_{qr} = \frac{L_r}{L_m} v_{qs} - \frac{L_r}{L_m} \left(R_s + \sigma L_s \frac{d}{dt} \right) i_{qs} \quad (2)$$

λ adalah fluks bocor; L adalah induktansi; v adalah tegangan; R adalah tahanan; i adalah arus; dan $\sigma = 1 - L_m^2 / (L_r L_s)$ koefisien bocor motor. Subkrip r dan s berturut-turut menyatakan rotor dan stator, dan subkrip d dan q adalah berturut-turut menyatakan komponen dalam sumbu d dan sumbu q pada referensi stasioner. Persamaan fluks rotor dalam referensi stasioner adalah[1] adalah:

$$\bar{\lambda}_{dr} = \frac{L_m}{T_r} i_{ds} - \omega_r \lambda_{qr} - \frac{1}{T_r} \lambda_{dr} \quad (3)$$

$$\bar{\lambda}_{qr} = \frac{L_m}{T_r} i_{qs} + \omega_r \lambda_{dr} - \frac{1}{T_r} \lambda_{qr} \quad (4)$$

ω_r adalah putaran rotor dalam radian per detik dan $T_r = L_r / R_r$ adalah time konstan rotor. Sudut θ_r dari vektor fluks rotor ($\bar{\lambda}_r$) dalam frame stasioner dengan sumbu d frame stasioner yang ditentukan melalui persamaan:

$$\theta_r = \arctan \left(\frac{\lambda_{qr}}{\lambda_{dr}} \right) \quad (5)$$

Turunan dari persamaan 5 adalah:

$$\dot{\theta}_r = \omega_e = \frac{\lambda_{dr} \dot{\lambda}_{qr} - \lambda_{qr} \dot{\lambda}_{dr}}{\lambda_{dr}^2 + \lambda_{qr}^2} \quad (6)$$

Dari persamaan 3 dan 4 substitusi ke persamaan 6 didapatkan:

$$\omega_e = \omega_r - \frac{L_m}{T_r} \left(\frac{\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}}{\lambda_{dr}^2 + \lambda_{qr}^2} \right) \quad (7)$$

Kemudian substitusi persamaan 6 ke persamaan 7, didapat:

$$\omega_r = \frac{1}{\lambda_r^2} \left[\lambda_{dr} \dot{\lambda}_{qr} - \lambda_{qr} \dot{\lambda}_{dr} - \frac{L_m}{T_r} (\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}) \right] \quad (8)$$

dengan $\lambda_r^2 = \lambda_{dr}^2 + \lambda_{qr}^2$

SPACE VECTOR PULSE WIDTH MODULATION (SVPWM)

Space vector modulation (SVM) adalah teknik modulasi yang dikembangkan melalui *space vector* (vektor ruang) yang akan menjadi tegangan acuan dengan periode *sampling* tertentu ke *inverter*, dengan menggunakan dua tegangan acuan yaitu v_d, v_q dan sudut. Teknik SVPWM dapat digunakan untuk membangkitkan pola *switching inverter* 3 fasa jenis sumber tegangan (VSI). Bila tegangan acuan 3 fasa diberikan dalam bentuk v_a, v_b dan v_c maka diperlukan transformasi dari 3-fasa ke 2-fasa. *Space vector* dengan persamaan 3-fasa dapat dinyatakan dengan $x_a(t), x_b(t)$ dan $x_c(t)$ yang masing-masing mempunyai pergeseran fasa sebesar 120° .

Persamaan dalam *vector* dapat dinyatakan sebagai berikut[3]:

$$\bar{x} = \frac{2}{3} [x_a(t) + ax_b(t) + a^2 x_c(t)] \quad (9)$$

$$a = e^{j\frac{2\pi}{3}} = \cos\left(\frac{2\pi}{3}\right) + j \sin\left(\frac{2\pi}{3}\right) = 0,5 + j0,8667 \quad (9)$$

$$a^2 = e^{j\frac{4\pi}{3}} = \cos\left(\frac{4\pi}{3}\right) + j \sin\left(\frac{4\pi}{3}\right) = -0,5 - j0,8667 \quad (10)$$

Fungsi x dapat berupa fungsi tegangan, arus atau fluks dan tidak harus dalam bentuk fungsi sinusoida. Bila x sebagai fungsi tegangan, maka persamaan tegangan dapat diberikan dengan persamaan berikut:

$$\left. \begin{aligned} v_a(t) &= V_m \sin(\omega t) = x_a(t) \\ v_b(t) &= V_m \sin(\omega t - 120^\circ) = x_b(t) \\ v_c(t) &= V_m \sin(\omega t + 120^\circ) = x_c(t) \end{aligned} \right\} \quad (11)$$

dalam bentuk persamaan *space vector* untuk persamaan jumlah tegangan menjadi:

$$\bar{v} = \frac{2}{3} [v_a(t) + av_b(t) + a^2 v_c(t)] \quad (12)$$

Tabel 1. Kombinasi posisi saklar Sa, Sb dan Sc

Vektor Tegangan	Sa	Sb	Sc
v ₀	0	0	0
v ₁	1	0	0
v ₂	1	1	0
v ₃	0	1	0
v ₄	0	1	1
v ₅	0	0	1
v ₆	1	0	1
v ₇	1	1	1

Karena kombinasi $v_0(000)$ dan $v_7(111)$ pada tabel 1 mempunyai tegangan *output* sama dengan nol, maka tinggal 6 (enam) vektor tegangan yang akan membentuk persegi enam dan mempunyai panjang rusuk masing-masing sebesar $2/3 V_{dc}$. Jika dilihat dari salah satu segitiga yang terbentuk, akan terlihat menjadi segitiga sama kaki dengan garis tinggi sebesar $1/\sqrt{3}V_{dc}$.

KONTROL PI

Kontroler proporsional merupakan kontroler yang aksi kontrolnya sebanding dengan sinyal kesalahan. Sinyal kesalahan $e(t)$ sebagai masukan kontroler dan keluaran sinyal kontrol adalah $u(t)$. Hubungan antara masukan kontroler $e(t)$ dan keluaran kontroler $u(t)$ adalah:

$$u(t) = K_p \cdot e(t) \quad (13)$$

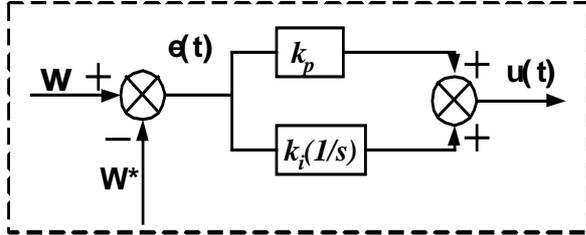
dengan K_p adalah penguatan proporsional. Kontroler integral merupakan kontroler yang aksi kontrolnya merupakan integral terhadap sinyal kesalahan. Sinyal kesalahan $e(t)$ sebagai masukan kontroler dan keluarannya adalah sinyal $u(t)$. Pada kontrol integral ini nilai keluaran $u(t)$ diubah pada laju integral dari sinyal kesalahan $e(t)$ sehingga hubungan antara masukan kontroler dengan keluaran kontroler $u(t)$ adalah:

$$u(t) = K_i \int_0^t e(t) dt \quad (14)$$

Penggabungan dari kedua kontroler tersebut diharapkan memberikan hasil yang lebih bagus. Persamaan kontrol PI dapat diberikan sebagai berikut:

$$u(t) = K_p \left\{ e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt \right\} \quad (15)$$

Dengan τ_i adalah waktu integral, parameter K_p dan τ_i dapat ditentukan. Nilai perkalian antara K_p dengan $1/\tau_i$ menghasilkan nilai K_i . Model simulasi dari kontrol PI terlihat pada gambar 1.



Gambar 1. Model simulasi kontrol PI

ARTIFICIAL NEURAL NETWORK

Pemodelan sistem *Artificial Neural Network* (ANN) biologi ke dalam model matematika dalam bentuk ANN didasarkan atas beberapa asumsi, yaitu [4]:

- Pemrosesan informasi terjadi pada elemen-elemen prosesor sederhana yang dinamakan *neuron*.
- Sinyal yang dilewatkan di antara *neuron-neuron* melalui suatu penghubung.
- Setiap penghubung memiliki bobot tertentu, dimana sinyal yang dikirim akan dikalikan.
- Setiap neuron memiliki fungsi aktivasi (biasanya *non-linear*) terhadap masukan-masukannya untuk menghasilkan sinyal keluaran.

Karakteristik ANN ditentukan berdasarkan:

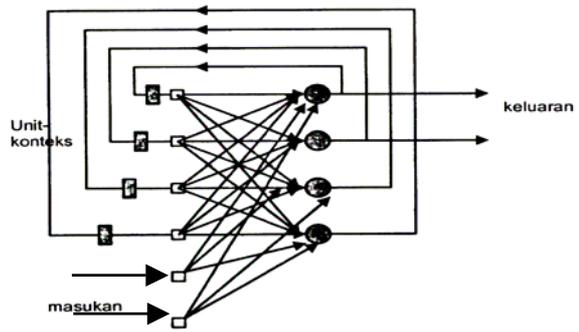
- Pola hubungan antar *neuron* (arsitektur),
- Metoda penentuan bobot (algoritma pelatihan atau pembelajaran)
- Fungsi aktivasi.

Aturan Pembelajaran *Reccurent*

Metode *Artificial Neural Network* (ANN) *Reccurent* dikembangkan oleh Elman[5]. ANN *Reccurent* adalah ANN dengan fasilitas umpan balik menuju *neuron* itu sendiri maupun *neuron* yang lain, sehingga aliran informasi dari masukan mempunyai arah jamak (*multidirectional*) seperti pada gambar 2.

Keluaran ANN tidak hanya tergantung pada masukan saat itu saja, tetapi juga tergantung pada kondisi masukan ANN untuk waktu sebelumnya. Kondisi ini dimaksudkan untuk menampung kejadian sebelumnya diikutkan pada proses komputasi berikutnya. Hal ini penting untuk

problematika yang cukup rumit, dan tanggapan keluaran ANN berkaitan dengan variasi waktu (*time-varying*), sehingga ANN memiliki *sense* terhadap waktu dan memori kondisi sebelumnya. Hal ini erat dengan suatu cabang ilmu yang disebut *automaton* (*finite automata/finite state machine*). Pada prinsipnya ANN-*reccurent* sama dengan ANN-*backpropagation* dengan tambahan unit konteks yang hanya menerima masukan internal (masukan balik dari lapis tersembunyi atau keluaran).



Gambar 2. *Reccurent neural network*

Jika d_k adalah keluaran yang diinginkan pada node k dan x_k adalah keluaran sebenarnya dari objek pada node k maka *error* pengukuran kuadrat (*squared error measure*) untuk pasangan masukan-keluaran ke- p didefinisikan sebagai [6]:

$$E_p = \sum_k (d_k - x_k)^2 \quad (16)$$

Maka kesalahan vektor gradien dari parameter untuk titik i adalah:

$$\varepsilon_i = \frac{\partial E_p}{\partial x_i} \quad (17)$$

Jika w_{ki} adalah pembobot pada penghubung titik k ke i , maka untuk meng-*update* w_{ki} pada pembelajaran secara langsung dapat dihitung dengan menggunakan aturan berantai berikut:

$$\Delta w_{ki} = \eta \frac{\partial E_p}{\partial x_i} \frac{\partial x_i}{\partial w_{ki}} \quad (18)$$

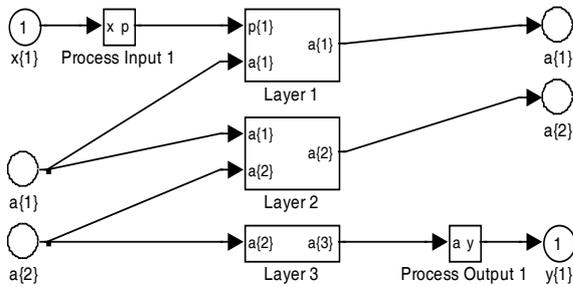
dimana η adalah konstanta pembelajaran (*learning rate*).

Observer Reccurent Neural Network

Perancangan RNN *observer* diawali dengan menyusun terlebih dahulu arsitektur RNN yang dianggap kompleks yaitu jumlah *neuron* tiap

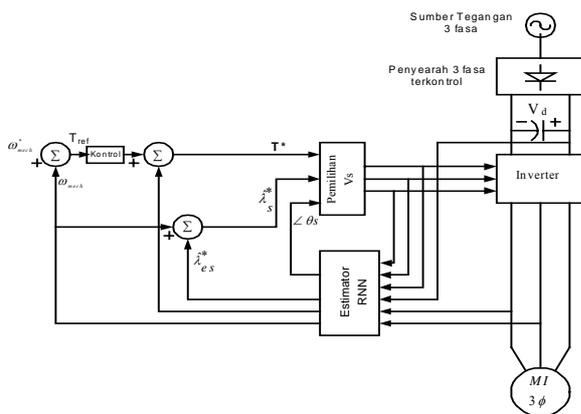
lapis tersembunyi dan jumlah lapis tersembunyi. Jumlah *neuron* yang digunakan adalah 10-10-1, dan struktur ANN dapat dilihat pada gambar 3.

Observer Recurrent Neural Network (RNN) terdiri dari 3 lapisan. Secara berurutan adalah lapis tempat masuk, lapis kompetitif, dan lapis tempat keluar. Jumlah data masukan dan *output* target yang digunakan masing-masing sebanyak 5.000 data. Jumlah *node* dalam satu lapis tersembunyi. Konstanta pembelajaran $\alpha = 0,5$ untuk error maksimum $= 1 \times 10^{-6}$ dan iterasi maksimum 1.000. Setiap *neuron* pada lapis tersembunyi mempunyai *loop* umpan balik yang berisi operator *unit delay* (z^{-1}). Fungsi aktivasi dari lapis pertama sampai ketiga sesuai dengan pemrograman menggunakan MATLAB M-File berturut turut adalah: *tansig*, *purelin* dan *traingdx*.



Gambar 3. *Observer recurrent neural network* (RNN) yang diusulkan

MODEL SISTEM PENGATURAN KECEPATAN MOTOR INDUKSI TIGA FASA



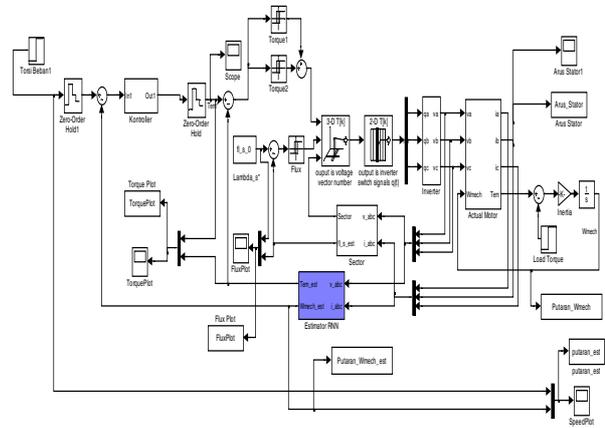
Gambar 4. Blok diagram pemodelan sistem

Dari gambar 4 di atas secara keseluruhan sistem dapat dibagi dalam beberapa blok:

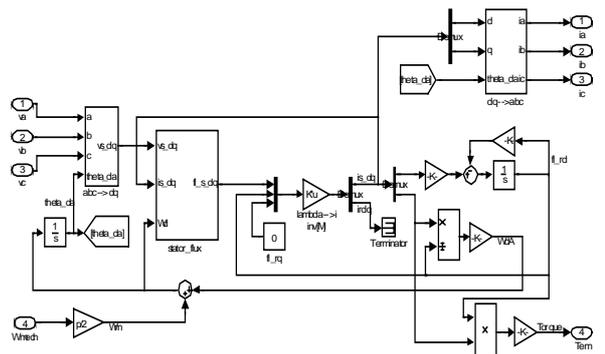
1. Model motor induksi tiga fasa
2. Model rangkaian *inverter* 3 fasa

3. Model *observer Recurrent Neural Network*
4. Model kontroler PI

Pada gambar 5 ditunjukkan diagram secara menyeluruh model pengaturan kecepatan motor induksi tiga fasa. Sedangkan gambar 6 menunjukkan model motor induksi tiga fasa.



Gambar 5. Blok diagram pemodelan sistem



Gambar 6. Model motor induksi 3 fasa

HASIL SIMULASI DAN ANALISA

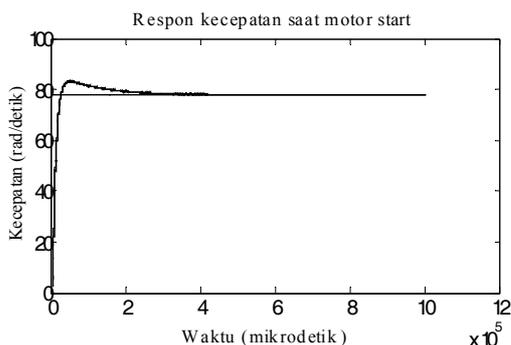
Data teknis dari motor induksi tiga fasa yang dipakai untuk simulasi dalam penelitian ini dapat dilihat pada tabel 2. Pada simulasi pengaturan putaran motor induksi tiga fasa ini akan dilakukan dengan uji pada kondisi *start*, putaran referensi turun 50%. Hal ini difungsikan untuk melihat perfromansi dari motor induksi apabila terjadi *setting* putaran turun. Disamping menggunakan *observer Recurrent Neural Network* (RNN) akan dilakukan pula dengan *Observer Feedforward Neural Network* (FNN). Untuk pengaturannya pada penelitian ini menggunakan kontroler PI di-*tuning* dengan metoda Zigler-Nichols dengan nilai $K_p = 0,54$ dan $K_i = 7,81125$.

Tabel 2. Data teknik motor induksi tiga fasa

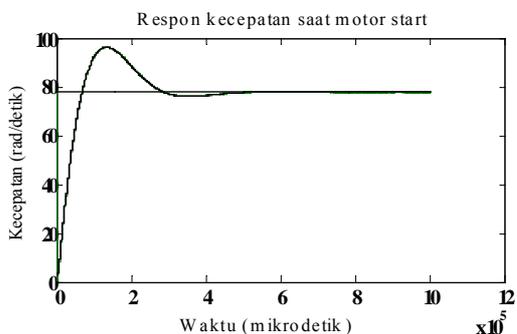
Daya (P)	7,5	HP/kW
	5,6	kW
Tegangan (V)	380	V(L-L,rms)
Frekuensi (f)	50	Hz
Fasa (F)	3	
Jumlah kutub (p)	4	
Arus beban penuh (I)	4	A
Tahanan stator (Rs)	1,77	Ω
Tahanan rotor (Rr)	1,34	Ω
Reaktansi stator (Xs)	5,25	Ω
Reaktansi rotor (Xr)	4,75	Ω
Reaktansi gandeng (Xm)	1,39	Ω
Slip beban penuh (s)	1,72	Ω
Momen enersia motor (J)	0,025	kg.m ²

Simulasi motor dengan beban saat start

Respon putaran awal motor induksi terhadap waktu pada saat diberi beban sebesar 12,64 Nm dengan kecepatan referensi 77,9743 rad/detik dapat dilihat pada gambar 7 dan gambar 8.



Gambar 7. Putaran pada saat *start* dengan *observer* RNN



Gambar 8. Putaran pada saat *start* dengan *observer* FNN

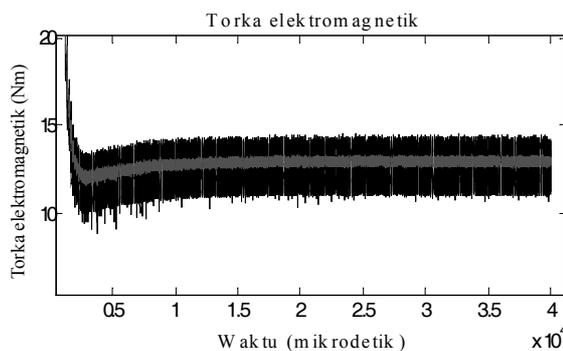
Dari gambar 7 dan gambar 8, kinerja sistem kontroler PI dengan *observer Recurrent Neural Network* (RNN) dan kinerja sistem menggunakan

kontrol PI dengan *observer Feedforward Neural Network* (FNN) ditunjukkan pada tabel 3.

Tabel 3. Hasil antara kontrol PI dengan *observer* FNN dan kontrol PI dengan *observer* RNN

No	Performansi	PI	PI dengan
		<i>observer</i> FNN	<i>observer</i> RNN
		K _p =0,54, K _i =7,8125	K _p =0,54, K _i =7,8125
1	Overshoot (%)	23,6048	7,0224
2	Rise Time (dtk)	0,044	0,0125
3	Settling Time (dtk)	0,495	0,364
4	Peak Time (dtk)	0,128	0,045
5	Peak (rad/detik)	96,38	83,45

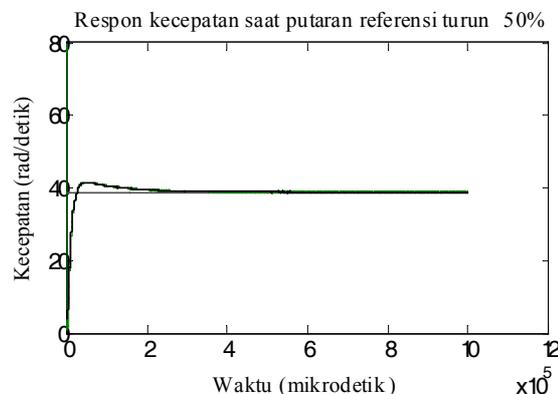
Torka elektromagnetik dapat dilihat pada gambar 9. Pada kondisi start, sistem akan stabil setelah 0,1 detik.



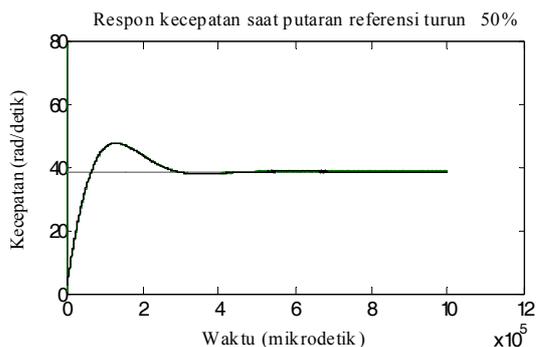
Gambar 9. Torka elektromagnetik motor

Simulasi dengan putaran referensi turun 50%

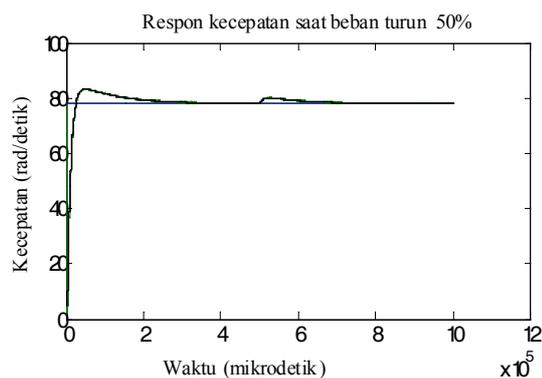
Respon putaran motor induksi terhadap waktu pada saat putaran referensi turun 50% ditunjukkan pada gambar 10 dan gambar 11.



Gambar 10. Putaran pada saat referensi turun 50% dengan kontroler PI menggunakan *observer* RNN



Gambar 11. Putaran pada saat referensi turun 50% dengan kontroler PI menggunakan *observer* FNN

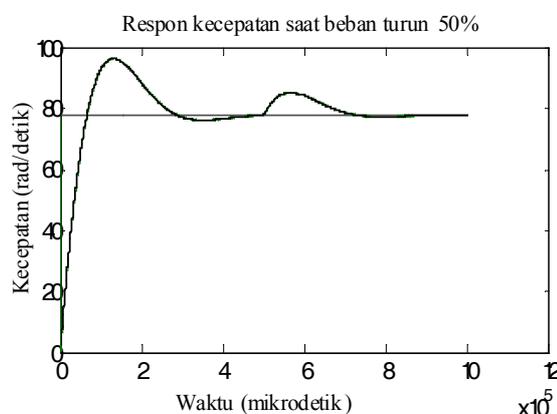


Gambar 13. Putaran pada saat ada perubahan beban turun 50% dengan kontroler PI menggunakan *observer* RNN

Dari gambar 10. dan gambar 11. kinerja sistem kontroler PI dengan *observer* RNN dan kinerja sistem menggunakan kontrol PI dengan *observer* FNN ditunjukkan pada tabel 4.

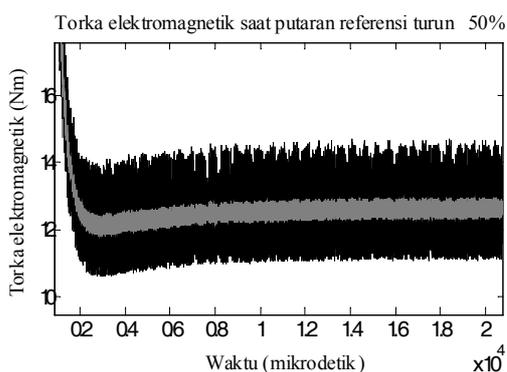
Tabel 4. Hasil kinerja sistem kedua *observer* untuk putaran turun 50%

No	Performansi	PI dengan <i>observer</i> FNN	PI dengan <i>observer</i> RNN
		Kp=0,54, Ki=7,8125	Kp=0,54, Ki=7,8125
1	Overshoot (%)	22,0529	6,4965
2	Rise Time (dtk)	0,0426	0,0132
3	Settling Time (dtk)	0,475	0,246
4	Peak Time (dtk)	0,125	0,052
5	Peak (rad/detik)	47,585	41,52



Gambar 14. Putaran pada saat ada perubahan beban turun 50% dengan kontroler PI menggunakan *observer* RNN

Torka elektromagnetik dapat dilihat pada gambar 12 pada kondisi putaran referensi turun 50%.



Gambar 12. Torka elektromagnetik untuk putaran referensi turun 50%

Dari gambar 13 dan gambar 14 kinerja sistem kontroler PI dengan *observer* RNN dan kinerja sistem menggunakan kontrol PI dengan *observer* FNN ditunjukkan pada tabel 5.

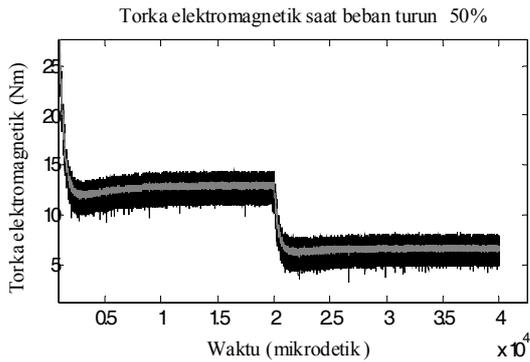
Tabel 5. Hasil kinerja sistem kedua *observer* untuk perubahan beban turun 50%

No	Performansi	PI dengan <i>observer</i> FNN	PI dengan <i>observer</i> RNN
		Kp=0,54, Ki=7,8125	Kp=0,54, Ki=7,8125
1	Overshoot (%)	9,2887	2,5030
2	Rise Time (dtk)	0,0258	0,0103
3	Settling Time (dtk)	0,4226	0,322
4	Peak Time (dtk)	0,258	0,0145
5	Peak (rad/detik)	85,217	79,926

Simulasi dengan perubahan beban turun 50%

Respon putaran motor induksi terhadap waktu pada saat ada perubahan beban turun 50% ditunjukkan pada gambar 13 dan gambar 14.

Torka elektromagnetik dapat dilihat pada gambar 15 pada kondisi perubahan beban turun 50%.



Gambar 15. Torka elektromagnetik untuk perubahan beban turun 50%

KESIMPULAN

Dari hasil simulasi yang telah dilakukan dapat ditarik kesimpulan yaitu :

1. Pada saat *start* motor dengan beban 12,64 Nm dan putaran nominal 77,9743 rad/detik terjadi *overshoot* 7,0224%, *rise time* 0,0125 detik dan *settling time* 0,364 detik.
2. Pada saat motor terjadi perubahan putaran referensi turun 50% menjadi 38,98 rad/detik dengan beban 12,64 N-m terjadi *overshoot* 6,4965%, *rise time* 0.0132 detik dan *settling time* 0.246 detik.
3. Pada saat motor terjadi perubahan beban turun 50% dari 12,64 Nm menjadi 6,32 N-m pada $t = 0,5$ detik, maka terjadi *overshoot*

2,5030 %, *rise time* 0.0103 detik dan *settling time* 0,322 detik.

4. Dengan membandingkan hasil-hasil percobaan dapat dilihat bahwa parameter *overshoot*, *rise time*, *settling time*, *peak time* dan *peak* untuk kontrol PI dengan observer RNN lebih baik dibandingkan dengan kontrol PI dengan observer FNN.

DAFTAR PUSTAKA

- [1] Soebagio, *Model mesin AC pada koordinat d-q-n*, Materi Kuliah Mesin Listrik Lanjut, ITS, 2006
- [2] D. Casadei, G. Serra, A. Tani, and L.Zarri, "Assessment of direct torque control for induction motor drives", *Buletin of the Polish academy of science tech. sciences*, vol. 54, No.3, 2006.
- [3] Bimal Kr. Bose, *Modern Power Electronics and AC drives*, Prentice Hall PTR, 2002.
- [4] Sri Kusumadewi, *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excellink*, Edisi 1, Graha Ilmu, Yogyakarta 2004.
- [5] Mauridhi Hery Purnomo dan Agus Kurniawan, *Supervised Neural Networks dan Aplikasinya*, Yogyakarta : Graha Ilmu, 2006.
- [6] A. Damiano, P.Vas etal, "Comparison of speed sensorless DTC induction motor drives", *Proc. PCIM*, Nuremberg, Germany, 1997, pp. 1-11.
- [7] D. Casadei, Giovanni Serra, "FOC and DTC: two variable scheme for induction motors torque control", *Trans. On Power Electronics*, Vol. 17, No. 5, September 2002.