

DESAIN KONTROLER ADAPTIF JARINGAN SYARAF TIRUAN UNTUK PENGATURAN KECEPATAN STASIONER PADA MESIN BENSIN

Achmad Abdul Jabbar^{*)}, Aris Triwiyatno, and Budi Setiyono

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: achmadabdulj@gmail.com

Abstrak

Kecepatan stasioner dapat dijadikan sebagai acuan untuk mengetahui performansi dari mesin bensin. Kecepatan stasioner yang terlalu cepat akan menyebabkan pemakaian bahan bakar semakin boros. Sedangkan kecepatan stasioner yang terlalu lambat akan menyebabkan tenaga yang dihasilkan oleh putaran mesin kecil, sehingga kendaraan tidak mampu untuk menghidupkan aksesoris-aksesoris kendaraan bahkan kendaraan dapat mati. Oleh karena itu diperlukan sebuah pengontrolan agar kecepatan stasioner tetap dalam keadaan yang ideal, sehingga diperoleh performansi mesin yang baik. Pada penelitian ini menggunakan kontroler adaptif jaringan syaraf tiruan untuk mengatur kecepatan mesin saat keadaan stasioner terhadap gangguan berupa torsi beban yang diwakilkan oleh sinyal step, square dan random. Struktur jaringan yang digunakan adalah NARX (Nonlinear AutoRegressive Exogeneous). Algoritma pelatihan yang digunakan adalah metode Levenberg Marquardt. Model mesin bensin dan kontroler adaptif jaringan syaraf tiruan disimulasikan dengan software komputasi. Berdasarkan hasil simulasi, kontroler adaptif jaringan syaraf tiruan mempunyai performansi lebih baik dibandingkan dengan kontroler PID dalam mempertahankan kecepatan stasioner mesin dengan nilai IAE untuk gangguan step sebesar 71,77, nilai IAE untuk gangguan square sebesar 191 dan nilai IAE untuk gangguan random sebesar 243,8.

Kata kunci: Mesin bensin, kecepatan stasioner, adaptif jaringan syaraf tiruan

Abstract

Idle speed can be used as reference to determine the performance of gasoline engine. When idle speed is too fast, engine use fuel more wasteful. When the idle speed is too slow, it will cause the engine power become small, so the vehicle accessories can't turn on. Therefore we need to control idle speed under ideal condition in order to obtain good engine performance. In this study, adaptive neural network controller was used to control the idle speed of engine against various load torque disturbance that was represented by step, square and random signals. Nonlinear Auto Regressive Exogeneous (NARX) was used as network structure and Levenberg Marquardt method was used as training algorithm. Gasoline engine model and adaptive neural network controller was simulated using computation software. Based on the simulation results, adaptive neural network controller has better performance than PID controller in maintaining idle speed with IAE value 71.77 for step signal disturbance, 191 for square signal disturbance and 243.8 for random signal disturbance.

Keywords: Gasoline engine, idle speed, adaptive neural network

1. Pendahuluan

Permasalahan utama yang dibahas dalam kontrol kecepatan stasioner adalah menjaga kecepatan mesin agar tetap pada referensi meskipun terdapat gangguan berupa torsi beban. Torsi beban dapat dihasilkan dari penggunaan AC (*air conditioner*), penyalaan unit elektronik, atau torsi yang dihasilkan oleh penggunaan *power steering*^[1]. Kecepatan stasioner yang terlalu cepat akan menyebabkan pemakaian bahan bakar semakin boros. Kecepatan

stasioner yang terlalu lambat akan menyebabkan tenaga yang dihasilkan oleh putaran mesin kecil, sehingga kendaraan tidak mampu untuk menghidupkan aksesoris-aksesoris kendaraan bahkan kendaraan dapat mati. Untuk itu kecepatan stasioner perlu dijaga tetap dalam keadaan ideal kendaraan agar mendapatkan performansi mesin yang baik. Mengatur kecepatan stasioner dapat dilakukan dengan mengatur waktu pengapian, injeksi bahan bakar, dan bukaan *throttle*.

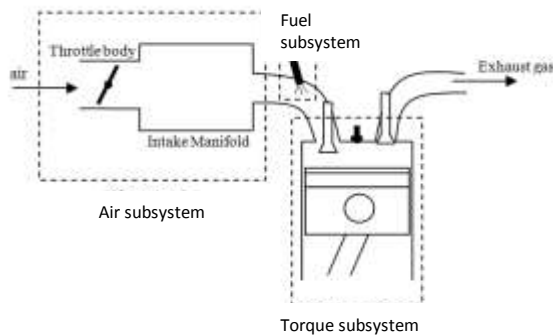
Tujuan dalam penelitian ini adalah mendesain kontroler adaptif jaringan syaraf tiruan untuk mempertahankan

kecepatan mesin saat kondisi stasioner. Kontroler adaptif jaringan syaraf tiruan mempunyai kelebihan dalam menangani sifat nonlinieritas sistem, karena kontroler adaptif jaringan syaraf tiruan mampu mengadaptasi parameter-parameter terhadap perubahan yang terjadi

2. Metode

2.1 Model Mesin Bensin

Mesin bensin memiliki 3 subsistem utama^[2], yaitu subsistem udara, subsistem bahan bakar, dan subsistem torsi. Seperti yang ditunjukkan pada Gambar 1.



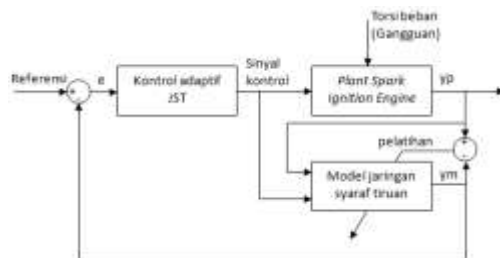
Gambar 1. Sistem mesin bensin.

Subsistem udara merupakan bagian yang menjelaskan mekanisme aliran udara mulai dari *throttle body* hingga keluar dari *intake manifold*. Subsistem bahan bakar menjelaskan mekanisme penginjeksian bahan bakar yang akan masuk ke ruang pembakaran. Subsistem torsi merupakan bagian mesin yang menjelaskan proses penghasilan torsi hingga kecepatan mesin.

2.2 Kontrol Adaptif Jaringan Syaraf Tiruan

Kontrol adaptif dapat diklasifikasikan menjadi dua metode^[3], yaitu kontrol adaptif *direct* dan kontrol adaptif *indirect*. Gambar 2 menunjukkan struktur kontrol adaptif *indirect* dengan jaringan syaraf tiruan.

Berdasarkan Gambar 2, model jaringan syaraf tiruan merupakan hasil identifikasi dari *plant*. Model ini digunakan untuk mengadaptasi parameter kontroler dan sinyal kontrol.



Gambar 2. Blok struktur kontrol adaptif jaringan syaraf tiruan

Jaringan syaraf tiruan mempunyai beberapa macam arsitektur, salah satunya adalah NARX (*Nonlinear Auto Regressive Exogeneous*)^[4]. Struktur NARX dapat dimodelkan dalam persamaan matematika diskrit sebagai berikut:

$$y_m(k+1) = f(y_p(k), \dots, y_p(k-N_y), u(k), \dots, u(k-N_u)) \quad (1)$$

Dimana y_m merupakan keluaran model jaringan syaraf tiruan. y_p adalah keluaran sistem yang akan dimodelkan. u adalah masukan sistem. N_y dan N_u masing-masing adalah orde keluaran dan masukan sistem.

Jaringan syaraf tiruan NARX dapat digunakan untuk memodelkandan mengendalikan sistem yang nonlinier. Keluaran hasil pemodelan jaringan syaraf tiruan direpresentasikan dengan parameter y_m dan dapat diketahui dengan menggunakan Persamaan 2 dan Persamaan 3.

$$z(k) = f(LW1 \cdot x(k) + B1.bias) \quad (2)$$

$$y_m(k+1) = LW2 \cdot z(k) + B2.bias \quad (3)$$

dimana z adalah keluaran lapis tersembunyi, x adalah masukan, $LW1$ adalah matriks parameter yang menghubungkan antara lapis masukan ke lapis tersembunyi, $LW2$ adalah matriks parameter antara lapis tersembunyi dengan lapis keluaran, $B1$ adalah matriks parameter bias, $B2$ adalah parameter bias pada lapis keluaran dan $f()$ merupakan fungsi aktivasi *hyperbolic tangent sigmoid* yang mempunyai persamaan sebagai berikut^[5]:

$$f(znet) = \frac{2}{1+e^{-2znet}} - 1 \quad (4)$$

Pelatihan jaringan digunakan untuk meng-*update* parameter jaringan dengan tujuan menghasilkan model yang lebih akurat. Pelatihan jaringan dapat dilakukan secara *offline* dan *online*. Pelatihan jaringan secara *offline* dilakukan dengan menggunakan sekumpulan data hubungan masukan dan keluaran sistem yang akan dimodelkan. Sedangkan pelatihan secara *online* dilakukan dengan meng-*update* parameter jaringan secara periodik berdasarkan galat yang ada.

Terdapat beberapa metode yang dapat digunakan dalam pelatihan jaringan, salah satunya adalah metode *Levenberg-Marquardt*. Metode LM dirumuskan dalam Persamaan 5^[4].

$$\Delta\theta = -(J^T J + \lambda I)^{-1} J^T e \quad (5)$$

dimana $\Delta\theta$ adalah perubahan parameter jaringan, λ adalah konstanta LM, I adalah matriks identitas, J merepresentasikan penurunan galat terhadap parameter atau dalam rumus $J = \frac{\partial e}{\partial \theta}$, dan e adalah galat.

Algoritma pelatihan jaringan secara *online* dengan metode LM adalah sebagai berikut:

1. Menentukan inisial konstanta faktor LM (λ)

2. Menentukan kriteria pemberhentian pelatihan.. Jika kriteria tidak terpenuhi, maka melakukan tahap 3 – 6.
3. Menghitung perubahan parameter dengan menggunakan Persamaan 5.
4. Menghitung parameter-parameter baru dengan menggunakan persamaan sebagai berikut:

$$\theta(k) = \theta(k - 1) + \Delta\theta(k - 1) \quad (6)$$

dimana θ adalah $LW1, LW2, B1$ dan $B2$.

5. Meng-update λ dengan menggunakan persamaan berikut:

Jika fungsi objektif yang baru lebih besar daripada nilai sebelumnya, maka

$$\lambda_{baru} = \lambda_{lama} \beta \quad (7)$$

Sebaliknya, jika fungsi objektif yang baru lebih kecil daripada nilai sebelumnya, maka

$$\lambda_{baru} = \frac{\lambda_{lama}}{\beta} \quad (8)$$

6. Kembali ke langkah 2.

3. Hasil dan Analisa

3.1 Perancangan Model Mesin Bensin

Pada penelitian ini menggunakan model mesin bensin empat langkah hasil penelitian yang dilakukan oleh Polonskii^[6]. Model mesin bensin mempunyai satu variable masukan berupa sudut bukaan *throttle (degree)*, satu keluaran berupa kecepatan mesin (RPM) serta terdapat satu gangguan berupa torsi beban (Nm). Model mesin bensin dapat direpresentasikan ke dalam blok simulasi pada Gambar 3.

Berikut keterangan untuk Gambar 3:

Ω : kecepatan sudut mesin

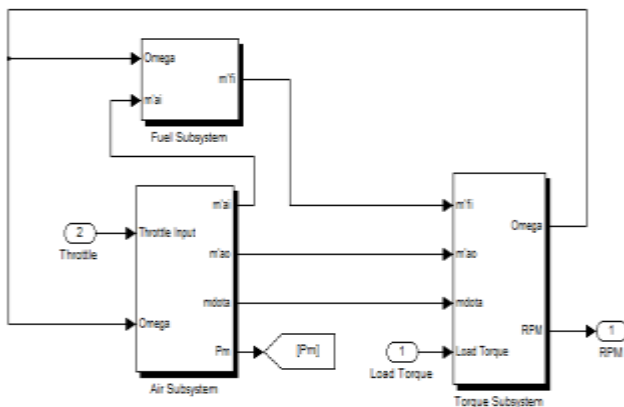
\dot{m}_{ai} : massa aliran udara yang masuk ke *intake manifold*

\dot{m}_{ao} : massa aliran udara yang masuk ke ruang pembakaran

\dot{m}_a : massa aliran udara yang berada di *intake manifold*

\dot{m}_{fi} : massa aliran bahan bakar yang masuk ke ruang pembakaran

P_m : tekanan pada *intake manifold*

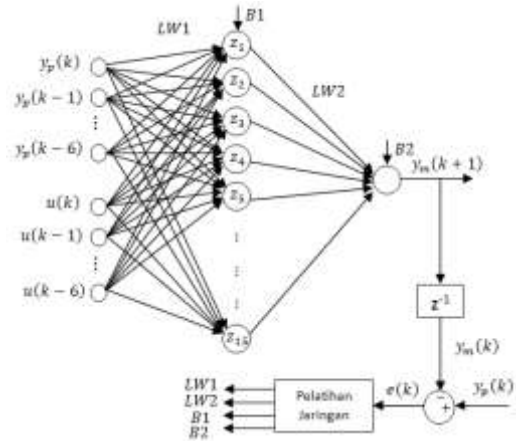


Gambar3. Model simulasi mesin bensin.

3.2 Perancangan Kontroler Adaptif Jaringan Syaraf Tiruan

3.2.1 Pemodelan

Pada penelitian ini pemodelan jaringan syaraf tiruan menggunakan struktur model seri paralel NARX. Arsitektur jaringan disajikan pada Gambar 4.



Gambar 4. Struktur jaringan syaraf tiruan untuk model

Model jaringan syaraf tiruan terdiri dari 3 lapis, yaitu 1 lapis masukan, 1 lapis tersembunyi yang tersusun oleh 15 *neuron*, dan 1 lapis keluaran. Pada lapis masukan, semua nilai masukan dinormalisasi atau ditransformasi dengan rentang nilai [-1, 1]. Hal ini dikarenakan pada lapis tersembunyi menggunakan fungsi aktivasi *hyperbolic tangent sigmoid* yang rentang keluarannya [-1,1]. Berikut ini adalah persamaan matematika untuk menormalisasi nilai masukan:

$$x_i' = \frac{2(x - \min)}{\max - \min} - 1 \quad (9)$$

dimana x_i' adalah nilai hasil dinormalisasi, x merupakan nilai masukan asli, \max adalah data maksimum, dan \min adalah data minimum.

Pada lapis keluaran juga terdapat blok yang digunakan untuk denormalisasi atau mentransformasikan kembali ke nilai asli. Persamaan 10 digunakan untuk denormalisasi.

$$x_o' = \frac{(\max - \min)(x + 1)}{2} + \min \quad (10)$$

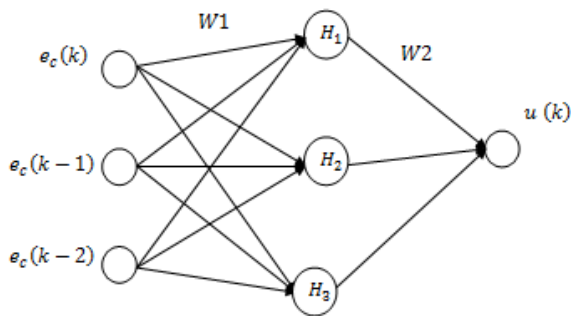
dimana x_o' adalah nilai hasil denormalisasi, x merupakan nilai keluaran, \max adalah data maksimum, dan \min adalah data minimum.

Pelatihan jaringan untuk pemodelan menggunakan metode *offline* dan *online*. Metode *offline (Batch learning)* digunakan untuk menentukan inisial awal parameter jaringan syaraf tiruan. Sedangkan metode *online* digunakan untuk optimisasi parameter agar didapatkan performa model dan kontroler yang baik. Metode pelatihan menggunakan *Levenberg-Marquardt*.

Algoritma pelatihan seperti pada Subbab 2.2, dengan inisial konstanta faktor LM (λ) sebesar 0,001 dan pelatihan akan dihentikan jika $e^2(k) < 0$.

3.2.2 Kontroler

Pada kontroler, struktur jaringan terdiri dari 3 lapis, yaitu 1 lapis masukan, 1 lapis tersembunyi yang tersusun oleh 3 neuron, dan 1 lapis keluaran yang terdiri dari 1 neuron. Untuk sinyal masukan terdiri dari $e_c(k)$, $e_c(k-1)$, $ec(k-2)$.



Gambar 5. Struktur jaringan untuk kontroler

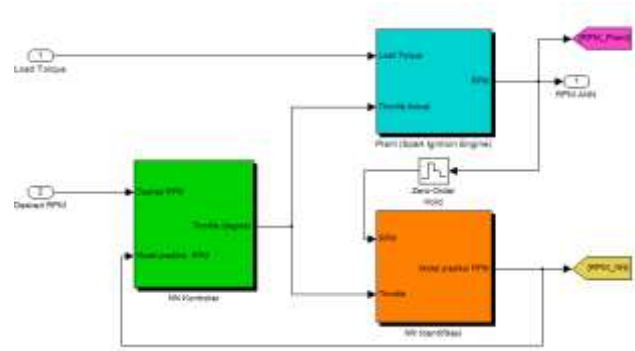
Dalam optimisasi parameter, kontroler menggunakan metode *Levenberg-Marquardt*. Algoritma yang digunakan untuk kontroler adalah sebagai berikut:

1. Menentukan inisial konstanta faktor LM (λ)
Dalam penelitian ini inisial $\lambda = 0.001$.
2. Menghitung galat dengan menggunakan persamaan sebagai berikut:
$$e_c(k+1) = y_{ref}(k+1) - y_m(k+1) \quad (11)$$
3. Menentukan kriteria pemberhentian pelatihan.
Pelatihan akan dihentikan jika $e_c^2(k+1) < 0$. Jika kriteria tidak terpenuhi, maka melakukan langkah 4-8.
4. Meng-*update* parameter $LW1$, $LW2$, $B1$ dan $B2$ pada model jaringan syaraf tiruan.
5. Meng-*update* parameter $W1$ dan $W2$ pada kontroler adaptif jaringan syaraf tiruan dengan menggunakan Persamaan 5 dan Persamaan 6.
6. Menghitung sinyal kontrol baru dengan menggunakan persamaan sebagai berikut :
$$u(k) = u(k-1) + \Delta u(k-1) \quad (12)$$

$$\Delta u(k) = W2(k) \cdot z_c(k) \quad (13)$$
7. Sinyal kontrol yang baru $u(k)$ dijadikan masukan untuk *plant* dan model jaringan syaraf tiruan.
8. Kembali ke langkah 2

3.2.3 Perancangan Sistem Keseluruhan

Gambar 6 menunjukkan perancangan sistem pengaturan kecepatan stasioner dengan menggunakan kontroler adaptif jaringan syaraf tiruan.



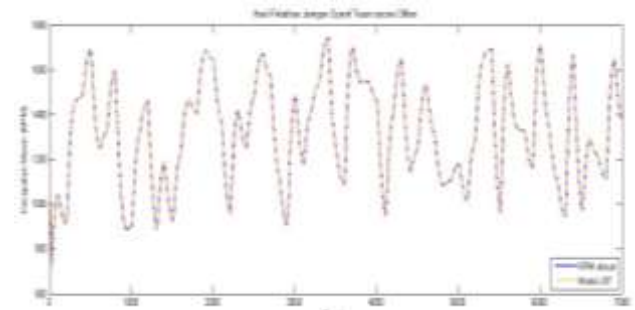
Gambar 6. Perancangan sistem keseluruhan kontroler adaptif jaringan syaraf tiruan

Referensi ditentukan konstan sebesar 954,9 RPM. Hal ini sesuai dengan kecepatan mesin bensin dalam keadaan stasioner.

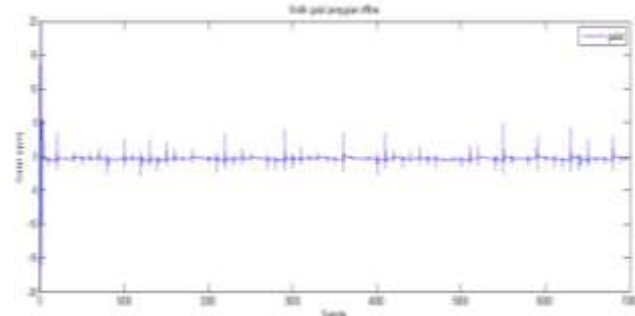
3.3 Analisa Unjuk Kerja

3.3.1 Pengujian Model Jaringan

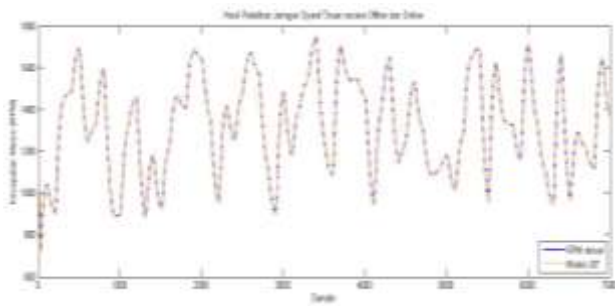
Pengujian dilakukan untuk model yang dilatih secara *offline* dan model yang dilatih secara *offline* dan *online*. Pengujian pertama dilakukan dengan memberikan masukan *throttle* berupa sinyal *random* dengan *sampling time* 1 detik.



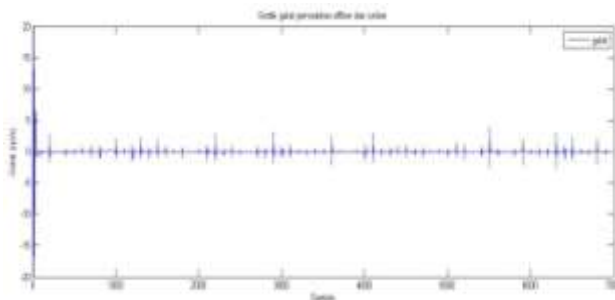
Gambar 7. Hasil pemodelan offline dengan masukan random



Gambar 8. Grafik galat pemodelan offline dengan masukan random



Gambar 9. Hasil pemodelan offline dan online dengan masukan random



Gambar 10. Grafik galat pemodelan offline dan online dengan masukan random

Gambar 7 sampai 10, merupakan grafik hasil pemodelan dan galat untuk masukan throttle berupa sinyal random dengan sampling time 1 detik. Hasil pemodelan offline dan hasil pemodelan offline dan online sudah dapat mengikuti keluaran plant sebenarnya.

Keakuratan dari model jaringan syaraf tiruan dapat diamati dari grafik galat pada Gambar 8 dan Gambar 10. Berdasarkan gambar tersebut, pemodelan offline dan online mempunyai hasil pemodelan yang lebih baik daripada pemodelan offline saja. Hal ini dapat dilihat dari nilai IAE untuk pemodelan offline dan online sebesar 9,728, sedangkan untuk pelatihan offline saja sebesar 26,62.

Pengujian kedua dilakukan dengan memberikan masukan throttle berupa sinyal sinusoidal. Pengujian ketiga dengan memberikan masukan throttle berupa sinyal square. Pengujian keempat dengan memberikan masukan throttle berupa sinyal gergaji. Nilai IAE dari keempat pengujian disajikan pada Tabel 1.

Tabel 1. Nilai IAE pengujian model jaringan syaraf tiruan

Masukan	IAE	
	Offline	Offline dan Online
Random	26,62	9,728
Sinusoidal	26,32	6,804
Square	23,99	8,769
Gergaji	29,39	7,102

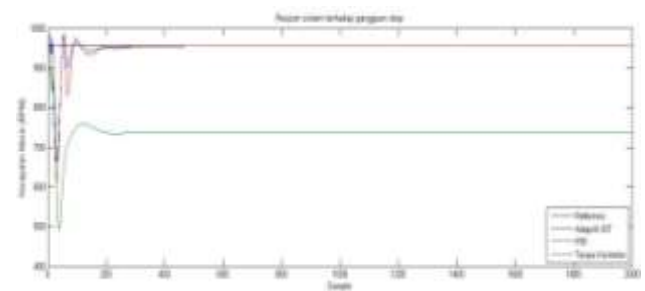
Berdasarkan Tabel 1, pemodelan jaringan syaraf tiruan secara offline dan online menghasilkan hasil pemodelan yang jauh lebih akurat daripada pemodelan secara offline saja. Dengan demikian model jaringan syaraf tiruan yang telah dirancang pada penelitian ini sudah layak digunakan sebagai model plant mesin bensin.

3.3.2 Pengujian Kontroler

Pengujian kontroler dilakukan dengan memberikan gangguan torsi beban berupa sinyal step, square dan random. Gangguan torsi beban yang diberikan sebesar 0-10 Nm. Referensi kecepatan mesin saat kondisi stasioner sebesar 954,9 RPM. Respon kontroler adaptif jaringan syaraf tiruan akan dibandingkan dengan respon kontroler PID dan respon sistem tanpa kontroler.

1. Respon sistem terhadap gangguan sinyal step

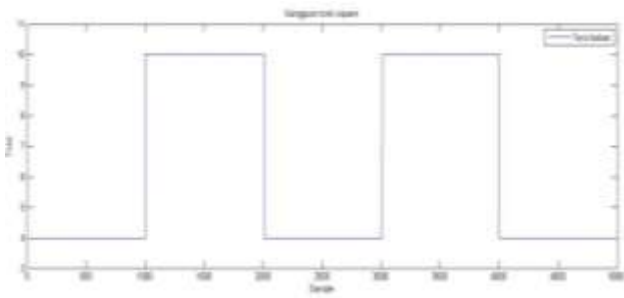
Pengujian dilakukan dengan memberikan gangguan torsi beban sebesar 10Nm. Berdasarkan Gambar 11, respon kontroler adaptif jaringan syaraf tiruan memiliki rise time (t_r) sebesar 0,4791 detik, waktu mantap (t_s) sebesar 0,9792 detik dan error steady state (e_{ss}) sebesar 0. Kontroler PID memiliki rise time sebesar 0,4757 detik, waktu mantap sebesar 1,0575 detik dan error steady state (e_{ss}) sebesar $2,32e-09$. Sedangkan respon sistem tanpa kontroler tidak mampu untuk mencapai referensi. Secara keseluruhan respon sistem yang paling baik untuk mengatasi gangguan berupa step 10 Nm adalah respon sistem adaptif jaringan syaraf tiruan. Hal ini karena sistem adaptif jaringan syaraf tiruan mampu mengatasi gangguan berupa torsi beban lebih cepat jika dilihat dari rise time, waktu mantap dan e_{ss} , serta nilai IAE sistem adaptif jaringan syaraf tiruan yang lebih kecil, dimana nilai IAE kontroler adaptif jaringan syaraf tiruan sebesar 71,77, untuk kontrol PID sebesar 106,9 dan tanpa kontroler sebesar 4389.



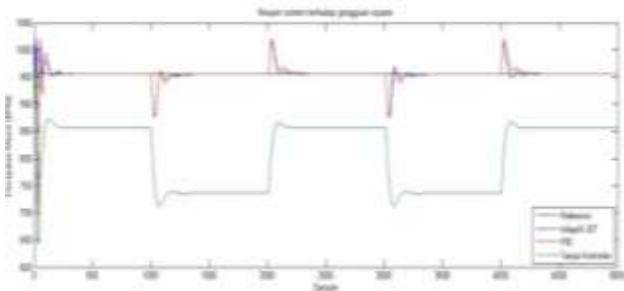
Gambar 11. Respon sistem terhadap gangguan step

2. Respon sistem terhadap gangguan sinyal square

Pengujian kontroler dengan memberikan gangguan berupa sinyal square. Sinyal square diberikan dengan sampling time 10 detik. Nilai minimal yang diberikan sebesar 4 (empat) dan nilai maksimalnya sebesar 10 (sepuluh). Gangguan berupa sinyal square disajikan pada Gambar 12.



Gambar12. Gangguan torsi beban berupa sinyal *square*



Gambar13. Respon sistem terhadap gangguan *square*

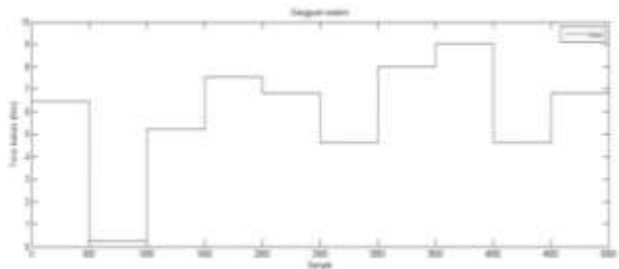
Respon sistem terhadap gangguan *square* ditampilkan pada Gambar 13. Berdasarkan gambar tersebut, respon sistem adaptif jaringan syaraf tiruan memiliki *rise time* sebesar 0,4847 detik, waktu mantap sebesar 1,104 detik dan e_{ss} sebesar $6,55e-07$. Kontroler PID memiliki *rise time* sebesar 0,4618 detik, waktu mantap sebesar 1,204 detik dan e_{ss} sebesar -0,00024. Sedangkan sistem tanpa kontroler tidak mampu untuk mencapai referensi. Respon kontroler adaptif jaringan syaraf tiruan semakin cepat dan tetap stabil. Namun respon sistem adaptif jaringan syaraf tiruan memiliki nilai puncak yang semakin besar tetapi nilainya masih termasuk kecil. Secara keseluruhan performansi kontroler adaptif jaringan syaraf tiruan dalam mempertahankan kecepatan stasioner lebih baik daripada kontroler PID dan tanpa kontroler. Hal ini bisa dilihat juga dari nilai IAE, dimana nilai IAE kontroler adaptif jaringan syaraf tiruan sebesar 191, kontroler PID sebesar 255,6 dan tanpa kontroler sebesar 7364.

3. Respon sistem terhadap gangguan sinyal *random*

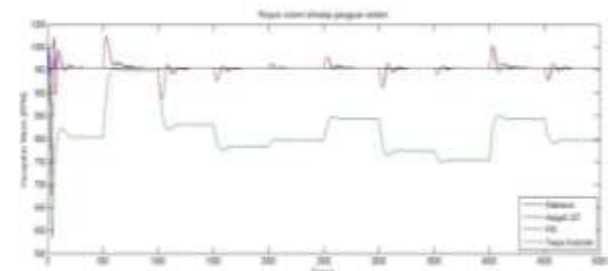
Pada penelitian ini sistem diberikan gangguan berupa sinyal *random* dengan *sampling time* 5 detik. Gangguan torsi yang diberikan memiliki nilai minimal sebesar 0 (nol) dan nilai maksimal sebesar 10 (sepuluh). Sinyal gangguan disajikan pada Gambar 14. Sedangkan respon sistem disajikan pada Gambar 15.

Berdasarkan Gambar 15, kontroler adaptif jaringan syaraf tiruan memiliki *rise time* sebesar 0,4552 detik, waktu mantap sebesar 1,1832 detik dan e_{ss} sebesar -0,02786. Kontroler PID memiliki *rise time* sebesar 0,4641 detik, waktu mantap sebesar 1,1568 detik dan e_{ss} sebesar

0,03761. Sedangkan sistem tanpa kontroler tidak mampu untuk mempertahankan kecepatan stasioner. Kecepatan mesin untuk sistem tanpa kontroler menurun akibat adanya gangguan berupa torsi beban. Secara keseluruhan performansi kontroler adaptif jaringan syaraf tiruan dalam mempertahankan kecepatan stasioner terhadap gangguan berupa sinyal *random* lebih baik daripada kontroler PID dan tanpa kontroler. Hal ini bisa dilihat juga dari nilai IAE, dimana nilai IAE kontroler adaptif jaringan syaraf tiruan sebesar 243,8, kontroler PID sebesar 291,9 dan tanpa kontroler sebesar 6917.



Gambar14. Gangguan torsi beban berupa sinyal *random*



Gambar15. Respon sistem terhadap gangguan *random*

4. Kesimpulan

Berdasarkan hasil simulasi dan analisis sistem dapat disimpulkan beberapa hal:

1. Pada pengujian pemodelan, *integrated absolute error* yang dimiliki oleh model jaringan syaraf tiruan memiliki nilai yang kecil. Ini membuktikan bahwa perancangan model jaringan syaraf tiruan sudah baik.
2. Dari seluruh hasil pengujian kontroler dengan gangguan, kontroler adaptif jaringan syaraf tiruan mampu lebih baik dibandingkan dengan kontroler PID dalam mempertahankan kecepatan stasioner mesin yang ideal sebesar 954,9 RPM dengan nilai IAE untuk gangguan step sebesar 71,77, nilai IAE untuk gangguan *square* sebesar 191 dan nilai IAE untuk gangguan *random* sebesar 243,8.

Referensi

- [1]. Yildiz, Y., dan Annaswamy, A. M., *Spark Ignition Engine Idle Speed Control: An Adaptive Control Approach*, pp. 1-25.

- [2]. Moskwa, John J., *Automotive Engine Modeling for Real Time Control*, Department of Mechanical Engineering Massachusetts Institute of Technology, 1988.
- [3]. Narendra, Kumpati S., dan Parthasarathy, Kannan, *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks, Vol. 1, No. 1, March 1990
- [4]. Van Gorp, Jurgen, *Nonlinear Identification with Neural Networks and Fuzzy Logic*, Dienst Algemene Elektriciteit and Instrumentatie (ELEC) Pleinlaan 2, B-1050 Brussel, Belgium, 2000.
- [5]. M. Dorofki, A. H. Elshafie, O. Jaafar, O. A. Karim, S. S. A. Mastura, *Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data*, International Conference on Environment, Energy and Biotechnology, 2012.
- [6]. Polonskii, M. M., *Complex Systems Simulation Using Matlab/Simulink*, The Electronic Journal for Engineering Technology Volume 4 No.1, Winter 2000.