

Perancangan Pengendali Modus Luncur untuk Motor DC dengan Optimasi Algoritma Genetika

Ahmad Riyad Firdaus

Politeknik Batam
Program Studi Teknik Elektronika
Parkway St. Batam Center Batam 29461, Indonesia
email: rifi@polibatam.ac.id

Abstrak – Pengendali Modus Luncur (PML) memiliki kekokohan yang baik dalam mengendalikan sistem linear maupun nonlinear. Kinerja pengendali ini sangat ditentukan oleh pemilihan parameter pengendali dari penguat pensaklaran dan permukaan luncur. Pada makalah ini, algoritma genetika dibuat untuk mengoptimasi pemilihan parameter pengendali dalam melacak nilai-nilai parameter pengendali yang optimal agar menghasilkan kinerja PML yang diinginkan. Pelacakan nilai-nilai parameter ini dilakukan dengan mengevaluasi fungsi kepastian (“fitness”) dari kromosom yang didefinisikan dalam algoritma. Hasil simulasi memperlihatkan, nilai-nilai parameter PML yang dihasilkan oleh algoritma genetika memberikan kinerja pengendali yang lebih baik dibanding dengan pemilihan nilai-nilai parameter yang dilakukan melalui uji coba (“trial and error”). Kinerja yang lebih baik ini diperlihatkan dengan kecilnya tanggapan waktu dan kesalahan penjejakan dari status keluaran. Metode ini memberikan solusi dalam mengoptimasi nilai-nilai parameter PML dalam perancangan sistem kendali.

Kata Kunci: Pengendali Modus Luncur (PML), tanggapan waktu, kesalahan penjejakan, algoritma genetika.

1. PENDAHULUAN

Pengendali Modus Luncur (PML) atau istilah populernya sering disebut dengan “Sliding Mode Controller” (SMC) merupakan sebuah kendali umpan balik pensaklaran berkecepatan tinggi (“high speed switching feedback control”) yang efektif dan kokoh dalam mengendalikan sistem linear maupun non-linear [1]. Sistem kendali ini kokoh karena menyediakan sebuah metoda perancangan sistem yang tidak peka terhadap ketidakpastian parameter lingkungan dan gangguan dari luar [3].

Pada prinsipnya, PML menggunakan sebuah hukum kendali pensaklaran berkecepatan tinggi (“high-speed switching”) untuk membawa trajektori status dari sistem linear/non-linear ke dalam sebuah permukaan (“hyperplane”) tertentu dalam ruang status (disebut permukaan luncur/“sliding surface”), kemudian trajektori status tersebut dipelihara agar tetap meluncur pada permukaan tersebut. Proses

pemeliharaan trajektori status pada permukaan luncur mengakibatkan terjadinya osilasi pada permukaan luncur. Osilasi ini sering disebut dengan “chattering”. Fenomena “chattering” pada permukaan luncur akan berdampak pada stabilitas dari sistem kendali.

Pada PML, waktu yang dibutuhkan oleh status dalam mencapai permukaan luncur akan berdampak terhadap kecepatan tanggapan sistem terhadap waktu. Waktu yang dibutuhkan oleh status untuk mencapai permukaan luncur sering disebut dengan “hitting time”. Salah satu keuntungan dari PML adalah ketika sistem mengenai permukaan luncur, maka sistem tersebut tidak akan peka terhadap ketidakpastian parameter lingkungan dan gangguan dari luar. Untuk menjaga perilaku sistem sehingga tidak peka terhadap perubahan lingkungan, maka dibutuhkan “hitting time” yang kecil agar sistem cepat mengenai permukaan luncur. “Hitting time” dan “chattering” yang kecil merupakan dua hal yang sangat penting dalam merancang sistem PML. Oleh karena itu, fokus penelitian ini adalah bagaimana meminimalisasi “hitting time” dan fenomena “chattering” pada PML.

Untuk meminimasi “hitting time” ini, beberapa penelitian sebelumnya sudah dilakukan, diantaranya: Young et al. menggunakan sebuah umpan balik penguatan yang tinggi (“high gain feedback”) untuk mempercepat tanggapan transien menuju permukaan luncur, tetapi metode ini berakibat pada peningkatan “chattering” sepanjang permukaan luncur dimana fenomena ini sangat tidak diinginkan oleh sistem fisik [3]. Untuk mengurangi fenomena “hitting time” dan “chattering” ini, Ching-Chang Wong dan Shin-Yu Chang [3] melakukan optimasi pemilihan penguat pensaklaran (“switching gain”) dengan algoritma genetika. Metode optimasi yang hanya memperhatikan penguat pensaklaran (“switching gain”) hanya bisa memperkecil fenomena “chattering” atau “hitting time” saja. Ng dan Li [4] melakukan optimasi 9 buah parameter “hard-switching” dengan algoritma genetika. Selain dibutuhkan waktu komputasi yang cukup lama, juga hasilnya masih memperkecil fenomena chattering saja. Riko [6] dalam tesisnya, “perancangan pengendali modus luncur untuk motor listrik” melakukan penalaan parameter penguat pensaklaran dan permukaan luncur dengan uji coba (“trial and error”). Metode ini belum efektif dalam

memecahkan permasalahan fenomena “chattering” dan “hitting time”.

Setelah melihat beberapa hasil penelitian yang sudah dilakukan sebelumnya dengan berbagai macam metode, untuk memperbaiki permasalahan “hitting time” dan “chattering” ini diajukan sebuah metode yang lain dan diharapkan bisa menjadi alternatif pilihan dalam memperbaiki permasalahan yang ada dalam merancang sistem PML. Sistem kendali yang dirancang diharapkan bisa memiliki “hitting time” dan “chattering” yang kecil secara simultan. Metoda yang diajukan adalah menggunakan algoritma genetika untuk mengoptimasi pemilihan penguat pensaklaran (“switching gain”) dan permukaan luncur (“sliding surface”) secara simultan.

2. PENGENDALI MODUS LUNCUR

2.1 Dasar Teori

Sebuah sistem linear SISO direpresentasikan dengan “persamaan (1) dan (2)” yang merupakan persamaan status dari sistem.

$$\dot{x} = Ax(t) + Bu(t) \dots\dots\dots(1)$$

dengan

$$A = \begin{bmatrix} 0 & 1 & 0 & K & 0 & 0 \\ 0 & 0 & 1 & \Lambda & 0 & 0 \\ M & M & M & M & M & M \\ 0 & 0 & 0 & \Lambda & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \Lambda & -a_2 & -a_1 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ M \\ 0 \\ 0 \\ 1 \end{bmatrix} \dots\dots\dots(2)$$

dimana $x(t) = [x_1 \ \Lambda \ x_n]^T$ merupakan sebuah vektor status berdimensi- n , A merupakan sebuah matrik sistem $n \times n$, dan B adalah sebuah vektor input berdimensi- n .

Secara umum ada dua tahap dalam perancangan PML, yaitu:

1. perancangan permukaan luncur

Permukaan luncur dirancang dengan membuat tanggapan sistem dibatasi dengan “persamaan (3)”

$$\sigma(x) = Sx(t) = 0 \dots\dots\dots(3)$$

dimana,

$$S = [s_1 \ \Lambda \ s_n] \dots\dots\dots(4)$$

adalah vektor koefisien permukaan luncur. Maka, dengan substitusi “persamaan (3) dan (4)” akan diperoleh persamaan permukaan luncur seperti pada “persamaan (5)” sebagai berikut:

$$\sigma(x) = s_1x_1(t) + s_2x_2(t) + \Lambda + s_nx_n(t) = 0 \dots\dots\dots(5)$$

2. perancangan pengendali modus luncur

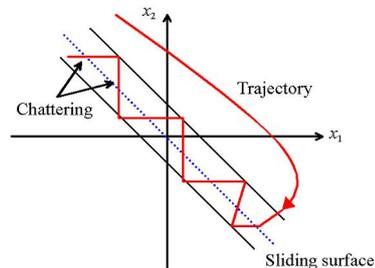
Pada perancangan ini hukum kendali $u(t)$ dibuat dengan menggunakan syarat kestabilan Lyapunov

$\sigma^T \dot{\sigma} < 0$. Secara umum hukum kendali dapat dipisah menjadi dua bagian sinyal kendali yaitu u_{eq} dan u_n , sehingga hukum kendali sistem diperoleh dengan

menjumlahkan kedua bagian sinyal kendali tersebut, seperti yang terlihat pada “persamaan (6)”

$$u(t) = u_{eq} + u_n = -kx \dots\dots\dots(6)$$

dimana u_{eq} merupakan sinyal kendali ekivalen yang akan membawa trajektori status ke permukaan luncur, sedangkan u_n merupakan sinyal kendali natural untuk menjaga agar trajektori status tetap berada pada permukaan luncur, seperti yang terlihat pada gambar 1 di bawah ini



Gambar 1. Diagram fasa trajektori status

Dinamika lingkaran tertutup sistem (“persamaan (7)”) diperoleh dari “persamaan (1) dan (6)”:

$$\begin{aligned} \dot{x} &= Ax + B(u_{eq} + u_n) \\ &= Ax + Bu_{eq} + Bu_n \dots\dots\dots(7) \end{aligned}$$

Sementara sinyal kendali ekivalen (“persamaan (8)”) diperoleh dengan mensubstitusi “persamaan (3) dan (7)”, dengan syarat $Bu_n=0$.

$$\begin{aligned} &= sx = 0 \\ \dot{x} &= s\dot{x} = s(Ax + Bu_{eq}) = 0 \\ u_{eq} &= -(sB)^{-1}sAx \dots\dots\dots(8) \end{aligned}$$

$$K_{eq} = -(sB)^{-1}sA \dots\dots\dots(9)$$

Dengan substitusi “persamaan (1) dan (8)” diperoleh dinamika lingkaran tertutup sistem pada modus luncur, seperti yang terlihat pada “persamaan (10)”

$$\dot{x} = [A - B(sB)^{-1}sA]x \dots\dots\dots(10)$$

Untuk mempertahankan trajektori status pada permukaan luncur, maka harus dipenuhi syarat keberadaan di permukaan luncur

$$\sigma^T \dot{\sigma} = \sigma(sBu_n) = \sigma u_n^* < 0. \text{ “Persamaan (11)”}$$

merupakan sinyal kendali untuk mempertahankan trajektori status pada permukaan luncur.

$$u_n = -k(sB)^{-1}sign(\sigma) \text{ jika } (sB) \text{ invertible } \dots\dots\dots(11)$$

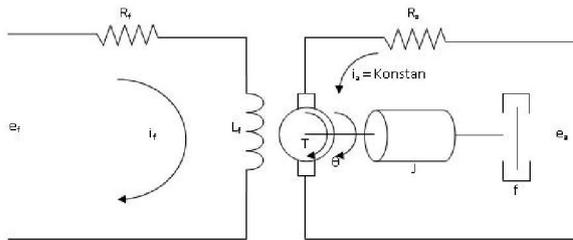
2.2 Pengendalian Motor DC dengan PML

Sistem servo merupakan sistem kendali umpan balik yang keluarannya berupa posisi mekanik, kecepatan, ataupun percepatan. Sistem kendali ini banyak digunakan di industri modern, seperti dalam sistem kendali untuk tangan robot di industri manufaktur, atau sistem pendaratan otomatis pesawat. Salah satu “plant” yang sering digunakan dalam sistem servo

adalah motor DC, disebabkan kemudahan (karena hanya dialiri arus listrik konstan) dan keragaman dalam pengendaliannya [6].

2.2.1 Pemodelan Motor DC

Model motor DC yang digunakan adalah model motor DC seperti yang digunakan oleh Riko [6]. Pada bagian ini akan diberikan pemodelan motor DC dengan tegangan medan yang terkendali (“field-controlled DC motor”). Keluaran motor DC, yaitu kecepatan dikendalikan oleh tegangan medan (“field voltage”). Sementara arus jangkar (“armature current”) dijaga konstan. Diagram skematik dari motor DC ini diperlihatkan pada gambar 2.



Gambar 2. Diagram skematik motor DC

Dari [4] diperoleh fungsi alih motor DC sebagai berikut:

$$\frac{\Omega(s)}{E_f(s)} = \frac{K_2}{(L_f s + R_f)(J s + f)} \dots\dots\dots(12)$$

masukan dari “plant” ini adalah tegangan medan (ef) dan keluarannya adalah kecepatan sudut rotor (ω). Berikut ini adalah nilai parameter-parameter yang digunakan dalam simulasi:

- ef = Masukan “plant” = 110 V
- if = Arus jangkar = 15 A (konstan)
- Lf = Induktansi medan = 20 H
- Rf = Resistansi medan = 120
- J = Inersia “plant” = 1 lb-ft-sec²
- F = Koefisien gesekan = 0.5 lb-ft/rad/sec
- ia = Arus jangkar = 15 A (konstan)
- K2 = Konstanta motor = 68,5 lb-ft/A

Dengan memasukkan parameter-parameter tersebut dan mendefinisikan peubah status x1= kecepatan sudut motor, x2=percepatan sudut rotor, dan keluaran yang diinginkan yaitu x1, maka diperoleh matrik status sebagai berikut:

$$A = \begin{bmatrix} 0 & 1 \\ -3 & -6.5 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 3.425 \end{bmatrix}, C = [1 \ 0], D = 0 \dots\dots\dots(13)$$

2.2.2 Perancangan Pengendali Modus Luncur

Tujuan pengendalian pada motor DC ini adalah membuat keluaran (x1) mengikuti masukan acuan (x1r) konstan, dan status lainnya (x2) menuju nol. Didefinisikan status error dari sistem adalah sebagai berikut:

$$x_e = \begin{bmatrix} x_1 - x_{1r} \\ x_2 - 0 \end{bmatrix} \dots\dots\dots(14)$$

dengan xe merupakan kesalahan penjejakan status.

Perancangan pengendali modus luncur dimulai dengan merancang permukaan luncur untuk sistem yang akan dikendalikan. Dengan menggunakan “persamaan (3)”, diperoleh:

$$\sigma(x) = Sx_e(t) = Sx - Sx_e \dots\dots\dots(15)$$

dengan S = [s1 s2]; dimana s1 s2 > 0

Dari “persamaan (10)” pemilihan S berkaitan dengan dinamika sistem yang akan mempengaruhi tanggapan sistem terhadap waktu (“settling time”). Dengan memilih S yang tepat, maka kutub-kutub pada matrik karakteristik sistem lingkaran tertutup akan dapat disesuaikan dengan tujuan pengendalian.

Selanjutnya dirancang masukan kendali tak kontinyu un yang akan menjaga status sistem tetap berada dalam permukaan luncur/dalam kondisi luncur. Pada perancangan masukan kendali tak kontinyu seperti yang dapat dilihat pada “persamaan (11)”, dipilih parameter k (penguat pensaklaran) yang sesuai untuk meminimalisasi fenomena “chattering” dari tanggapan sistem.

Dengan menjumlahkan masukan kendali tak kontinyu un dengan masukan kendali ekivalen ueq maka diperoleh masukan kendali total yang akan diumpanbalikan ke “plant”

3. ALGORITMA GENETIKA

Algoritma genetika merupakan metode penyelesaian optimasi yang terinspirasi oleh prinsip genetika dan seleksi alam yang dikemukakan oleh Darwin (Teori Evolusi Darwin). Algoritma ini banyak digunakan untuk memperoleh penyelesaian yang tepat dalam berbagai permasalahan optimasi

Perbedaan dengan algoritma pelacakan konvensional [9], algoritma genetika berangkat dari himpunan solusi yang dihasilkan secara acak. Himpunan ini disebut populasi. Sedangkan setiap individu dalam populasi disebut kromosom yang merupakan representasi dari solusi. Secara umum, sebuah algoritma genetika sederhana terdiri dari tiga operator genetik dasar [3]: (a). Reproduksi; (b). Pindah silang (“crossover”); dan (c). Mutasi (“mutation”).

Struktur umum dari algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut: 1. Membangkitkan populasi awal
Populasi awal ini dibangkitkan secara random sehingga diperoleh solusi awal. Populasi itu sendiri

terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Dalam membentuk generasi baru digunakan tiga operator yang telah disebut di atas yaitu operator reproduksi/seleksi, crossover dan mutasi. Proses ini dilakukan secara berulang sehingga diperoleh jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru.

3. Evaluasi solusi

Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai kepastan (“fitness”) setiap kromosom. Setelah beberapa generasi maka nilai kepastan (“fitness”) akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal.

4.OPTIMASI PARAMETER PML DENGAN ALGORITMA GENETIKA

Algoritma genetika bekerja untuk mencari parameter-parameter PML agar menghasilkan kinerja pengendali sesuai dengan yang diinginkan. Parameter-parameter yang akan dioptimasi pemilihannya adalah penguat pensaklaran k (“switching gain”) dan konstanta permukaan luncur S (“sliding surface”). Agar dapat melakukan pemilihan parameter-parameter PML $R=\{k,S\}$ dengan algoritma genetika, dipilih R sebagai sebuah kumpulan parameter dan mengkodekannya sebagai sebuah kromosom, kemudian pilih sebuah fungsi obyektif dan fungsi kepastan (“fitness”) yang digunakan untuk melacak sebuah solusi terbaik dalam ruang parameter tertentu. Fungsi obyektif yang dibuat harus merepresentasikan kebutuhan perancangan sistem kendali yang diinginkan, yaitu mengurangi “hitting time” trayektori status ke permukaan luncur dan fenomena “chattering” dari input kendali (u) dan tanggapan keluaran (x_1) sistem. Untuk mempermudah dalam merumuskan fungsi obyektif, didefinisikan “hitting time” adalah “settling time” (t_s) tanggapan waktu dari sistem; “chattering” adalah rata-rata kuadrat (MSE) kesalahan penjejakan (“tracking error”) dari status x_1 (X_{err}) dan input kendali u (U_{err}). Fungsi obyektif didefinisikan pada “persamaan (16)”.

$$F = c_1(T_r)^2 + c_2(X_{err})^2 + c_3(U_{err})^2 \dots\dots\dots(16)$$

dimana c_1, c_2, c_3 adalah konstanta-konstanta pengali untuk memperlihatkan prioritas optimasi dari fungsi obyektif tersebut. Fungsi kepastan (“fitness”) dapat didefinisikan pada “persamaan (17)”.

$$f = \frac{1}{F + 1} \dots\dots\dots(17)$$

Fungsi obyektif perlu ditambah 1 untuk menghindari kesalahan program yang diakibatkan pembagian oleh 0. Seiring dengan mengecilnya fungsi obyektif, fungsi kepastan (“fitness”) akan bertambah besar sampai

konvergen pada satu nilai tertentu. Hal ini merepresentasikan algoritma genetika bekerja dengan baik. “Persamaan (18)” dioperasikan untuk mencari nilai kepastan tertinggi yang berelasi dengan solusi terbaik dalam algoritma genetika.

$$MAX\{f(R)\} \dots\dots\dots(18)$$

dimana R adalah sebuah kromosom yang merepresentasikan sebuah nilai tertentu yang berada pada ruang pelacakan P . Tiga operasi dasar algoritma genetika dapat diterapkan untuk memilih parameter-parameter $\{k, s_1, s_2\}$ untuk meningkatkan indeks kinerja PML dalam ruang pelacakan P . Jika kromosom dengan nilai kepastan (“fitness”) terbaik telah diperoleh, maka nilai kromosom tersebut dapat dipilih sebagai sebuah parameter penguat pensaklaran k dan konstanta permukaan luncur S dari PML.

Secara umum prosedur pemilihan parameter PML dengan algoritma genetika dapat dirumuskan sebagai berikut:

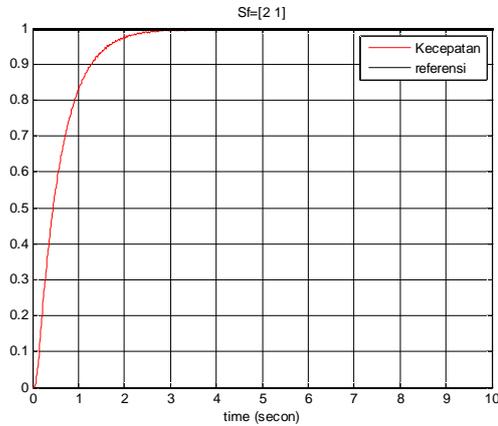
1. Rumuskan parameter-parameter PML
2. Buat polulasi awal dari kromosom secara random
3. Kodekan masing-masing kromosom dalam populasi dan evaluasi kinerja dari tanggapan sistem
4. Evaluasi nilai kepastan (“fitness”) untuk masing-masing kromosom
5. Reproduksi kromosom-kromosom tersebut berdasarkan nilai kepastan (“fitness”) yang telah dihitung pada tahap 4
6. Buat kromosom-kromosom baru dengan melakukan proses persilangan (“crossover”) dan mutasi, dan mengganti kromosom yang lama dengan kromosom yang baru.
7. Ulangi tahap ke-3 sampai maksimum iterasi diperoleh atau nilai kepastan konvergen pada satu nilai tertentu.

5. SIMULASI KINERJA OPTIMASI

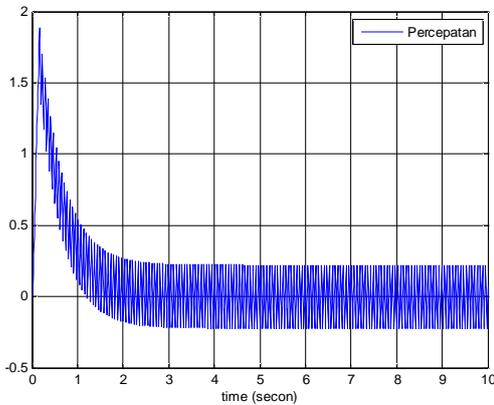
Pada simulasi ini, akan diperlihatkan kinerja optimasi algoritma genetika dengan metode konvensional [4] yang diterapkan pada “plant” orde dua motor DC. Dalam penelitian ini, diambil nilai awal nol untuk kedua status, dan masukan yang diberikan merupakan fungsi tangga satuan (“step function”). Untuk dapat membandingkan secara proporsional semua jenis pengendali, maka dipilih “settling time” (t_s) sebagai indikator pembanding. Indikator kinerja dianggap baik jika tanggapan sistem memiliki spesifikasi “settling time” (t_s) $2s$; kesalahan penjejakan (“tracking error”) dari: $x_1 < 0.03, x_2 < 0.2, u < 15$; dan kesalahan pada kondisi setimbang pada x_1 (“error steady state”) $e_{ss} < 0.002$. Toleransi yang diberikan untuk kondisi settling time adalah sekitar 1%.

A. Metode konvensional

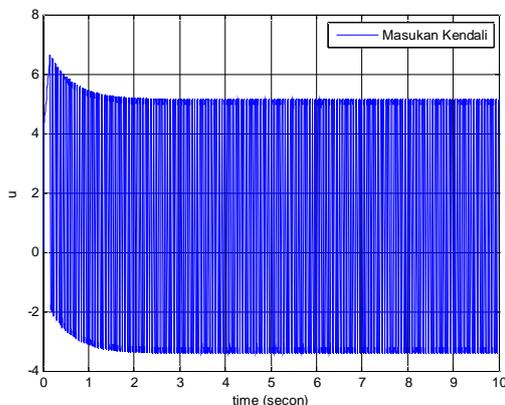
Pada metode ini, pemilihan penguat pensakaran dan konstanta permukaan luncur dilakukan dengan uji coba (“trial and error”). Penguat pensaklaran dan konstanta permukaan luncur yang dipilih adalah: $k=15$ dan $S=[2 \ 1]$. Tanggapan sistem yang diperoleh seperti yang terlihat pada gambar 3.



Gambar 3a. Tanggapan kecepatan vs waktu



Gambar 3b. Tanggapan percepatan vs waktu



Gambar 3c. Masukan kendali

Dari hasil simulasi data-data sebagai berikut:

- K : 15
- S : $[s_1=2 \ s_2=1]$
- t_s (secon) : 2.52 secon
- $MSE \ x_1$: 0.0341

- $MSE \ x_2$: 0.1097
- $MSE \ u$: 17.8675
- e_{ss} : 0.0038

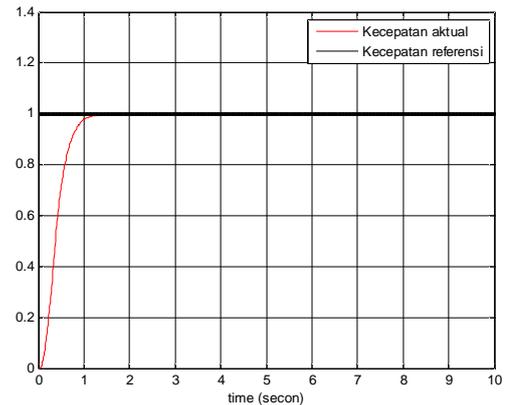
B. Metode optimasi algoritma genetika

Metode ini digunakan untuk memperoleh kombinasi parameter penguat pensaklaran (“gain switching”) k dan konstanta permukaan luncur (“sliding surface”) S terbaik. Dalam metode konvensional, penalaan parameter-parameter ini dilakukan dengan uji coba (“trial and error”). Berikut ini adalah parameter-parameter yang digunakan dalam algoritma genetika untuk kegunaan optimasi parameter PML: probabilitas “crossover”=0.8; probabilitas mutasi=0.05; panjang bit kromosom=12 x 3 bit; maksimum generasi=100; populasi=30; batas ruang pelacakan P: $k=0 - 15$; s_1 dan $s_2 =1 - 10$. Konstanta-konstanta untuk fungsi obyektif $c_1=1$; $c_2=10^7$; dan $c_3=10$.

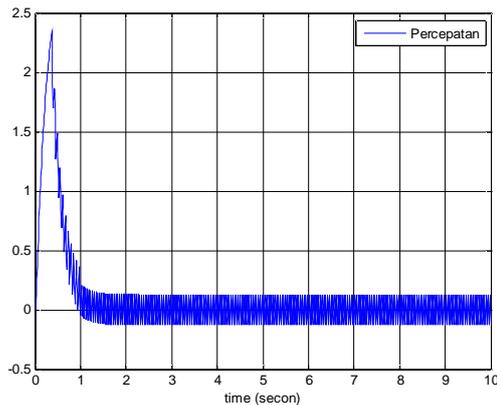
Dari hasil optimasi, diperoleh data-data parameter PML sebagai berikut:

- K : 13.095238
- S : $[s_1=4.292308 \ s_2=1.017582]$
- t_s : 1.19 secon
- $MSE \ x_1$: 0.028612
- $MSE \ x_2$: 0.152247
- $MSE \ u$: 13.803405
- e_{ss} : 0.0014

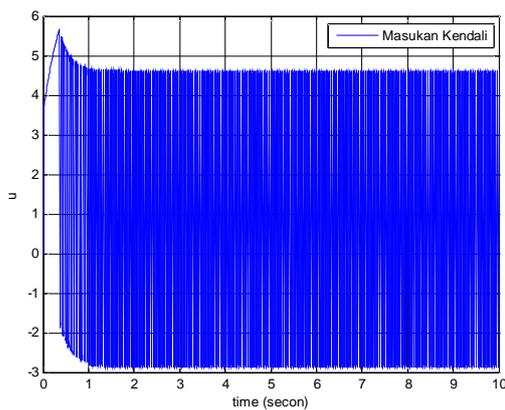
Tanggapan sistem yang diperoleh seperti yang terlihat pada gambar 4.



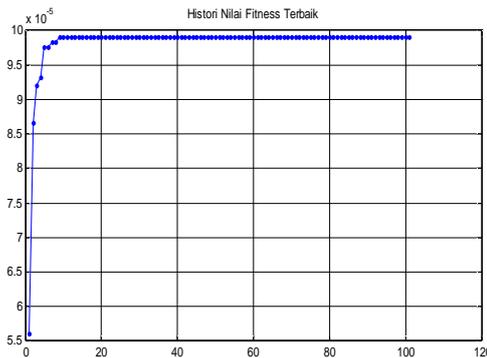
Gambar 4a. Tanggapan kecepatan vs waktu



Gambar 4b. Tanggapan percepatan vs waktu



Gambar 4c. Masukan sinyal kendali



Gambar 4e. Grafik konvergensi fungsi kepatasan

Dari gambar 3 dapat dilihat bahwa data “settling time” dari status x_1 , e_{ss} , dan kesalahan penjejakan masih lebih besar dari nilai kebutuhan perancangan yang sudah ditetapkan sebelumnya. Sedangkan pada gambar 4, data-data menunjukan seluruh kebutuhan dari perancangan terpenuhi yaitu dengan $t_s=1.19$ detik, $e_{ss}=0.0014$, $MSE_{x_1}=0.028612$, $MSE_{x_2}=0.152247$, dan $MSE_u=13.803405$. Secara keseluruhan pemilihan parameter PML dengan metode algoritma genetika menghasilkan kinerja PML yang lebih baik dibanding dengan metode konvensional.

Waktu yang dibutuhkan untuk melacak parameter-

parameter PML dengan metode algoritma genetika lebih singkat dibanding dengan menggunakan uji coba (“trial and error”). Dari gambar 4e, memperlihatkan bahwa nilai kepatasan (“fitness”) konvergen pada iterasi atau generasi ke delapan. Artinya algoritma tidak sulit dalam melacak parameter yang cocok untuk PML.

6. KESIMPULAN

Pada makalah ini telah dibahas bagaimana meningkatkan kinerja pengendali modus luncur, yaitu dengan semakin cepatnya tanggapan status keluaran mencapai nilai referensi dan kecilnya kesalahan penjejakan yang dimiliki oleh status tersebut. Metode algoritma genetika terbukti efektif dalam meningkatkan kinerja dari pengendali modus luncur yang dirancang.

Keuntungan dalam menggunakan metode algoritma genetika adalah tidak diperlukannya pengetahuan dan analisis matematika yang lebih terhadap sistem fisik. Selama proses pelacakan oleh algoritma genetika, yang dievaluasi hanya fungsi kepatasan (“fitness”) dari kromosom yang didefinisikan [3]. Algoritma dapat mempetimbangkan beberapa obyek permasalahan dan memilih parameter-parameter kendali dengan hanya mengevaluasi fungsi kepatasan (“fitness”) yang berhubungan langsung dengan tanggapan waktu dan kesalahan penjejakan dari status keluaran yang dipilih.

Metode ini memberikan pendekatan berbasis komputer dalam merancang sistem kendali yang lebih kompleks. Selain itu, dalam perancangan sistem kendali dapat dengan mudah memasukan kriteria kekokohan yang berbasiskan penolakan gangguan dan ketidakpastian dinamika sistem.

DAFTAR REFERENSI

- [1] Raymon A. DeCarlo, Stanislaw H.Zak, Gregory P.Matthews, "Variable Structure Control of Nonlinear Multivariable System: A Tutorial", *Proceeding of The IEEE*, vol. 76, no. 3, March 1988, pp.212-224.
- [2] Slotine, J-J E., Weiping Li, *Applied Nonlinear Control*, Prentice-Hall International, Inc. 1991, 276-284.
- [3] Ching-Chang Wong, Shih-Yu Chang, "Parameter Selection in the Sliding Mode Control Design Using Genetic Algorithms", *Tamakang Journal of Science and Engineering*, vol. 1, no. 2. 1998, pp. 115-122.
- [4] Y. Li, K.C. Ng, D.J. Murray-Smith, G.J. Gray, and K.C. Sharman, "Genetic Algorithm Automated Approach to Design of Sliding Mode Control Systems", *International Journal of Control*.
- [5] Li, K.C. Ng, D.J. Murray-Smith, and K.C. Sharman,"Genetic Algorithms Applied to Fuzzy Sliding Mode Control Design", *International Journal of Control*.
- [6] Riko Nofendra, "Pengendali Modus Luncur Untuk Motor Listrik", Tesis Program Studi Teknik Elektro ITB, 2006, hal. 22-30
- [7] Robert E. King, *Computational Intelligence in Control Engineering*, Marcel Dekker, Inc.,1999.
- [8] Denny Hermawanto, "Algoritma Genetika dan Contoh Aplikasinya", *Ilmu Komputer.com*, 2003, hal 1.
- [9] Arief Syamsudin,"Pengenalan Algoritma Genetika", *Kuliah Umum Ilmu Komputer.com*, 2004, hal 1
- "Micromouse ELIND BOT'1" di jurnal ilmiah LIPI pada tahun 2004, "CNC Flame Cutting" Riset Unggulan Kemitraan RISTEK RI, "Fire Fighting Robot" KRI-KRCI 2005, dan mengikuti seminar nasional dan asia pasific pada tahun 2007 sebagai pembicara dengan topic "Sliding Mode Controller".

BIOGRAFI

Ahmad Riyad Firdaus (Dosen Teknik Elektronika merangkap Kepala Bagian Perencanaan dan Pengendalian Mutu Politeknik Batam) lahir di Purwakarta, 1 Mei 1976. Beliau menyelesaikan pendidikan sarjana (S1) di Jurusan Fisika ITB pada tahun 2000, dan magister (S2) di Jurusan Teknik Elektro ITB pada tahun 2008.

Setelah lulus dari pendidikan sarjana (S1) tahun 2000, beliau bergabung dengan Politeknik Batam sebagai dosen di Program Studi Elektronika Industri, pada tahun 2002 beliau ditugaskan selama satu tahun di BSDC (Batam Skill Development Center) proyek AUS AID di SMKN 1 Batam. Tahun 2003 beliau diangkat menjadi Kepala Program Studi Elektronika Industri Politeknik Batam. Beberapa penelitian dan tulisan ilmiah yang pernah buat antara lain: