

Penentuan Matriks Pembobot Pada Kontrol Optimal Menggunakan Adaptive Particle Swarm Optimization

Khozin Mu'tamar¹, Janson Naiborhu²

¹Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Riau
Email: khozin.mutamar@unri.ac.id

²Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, ITB

ABSTRAK

Artikel ini menyajikan sebuah pendekatan dalam pemilihan matriks pembobot pada desain kontrol optimal menggunakan adaptive particle swarm optimization. APSO merupakan modifikasi PSO standar dengan mempertimbangkan pembobotan dan penerapan multi globalbest. APSO akan digunakan untuk menentukan matriks pembobot yang tepat guna meningkatkan kinerja dari kontrol. Simulasi diberikan untuk mengilustrasikan prosedur dalam desain kontrol dan perbandingan performa dari metode APSO dan PSO standar ditinjau dari nilai integral absolut error (IAE).

Kata Kunci: Adaptive particle swarm optimization, integral absolut error, kontrol optimal, matriks pembobot

ABSTRACT

This article present a new approach to determine weighting matrices in optimal control using adaptive particle swarm optimization. APSO is modified from standard PSO which have weighting and multi globalbest. APSO used to improve control performance by selecting appropriate weighting matrices. Simulation is given to illustrate the design procedure and to comparison of control performance between PSO and APSO using integral absolute error (IAE).

Keywords: Adaptive particle swarm optimization, integral absolute error, optimal control, weighting matrices

Pendahuluan

Masalah dalam kontrol optimal adalah bagaimana menentukan besarnya kontrol pada setiap waktu sehingga diperoleh fungsi biaya yang minimum. Fungsi biaya dalam kontrol optimal ditentukan oleh nilai dari matriks pembobot. Nilai matriks ini ditentukan secara trial-error atau berdasarkan analisis data pengembang. Bryson dan Ho [5] mengembangkan suatu metode yang kemudian dikenal dengan Aturan Bryson untuk menentukan nilai dari matriks pembobot. Salah satu bentuk kelemahan aturan ini adalah jika nilai kontrol tidak diketahui batasan maksimumnya, nilai pembobot tidak dapat ditentukan dengan tepat. Kondisi tersebut akan membuat aturan Bryson menjadi metode trial-error.

Pada tahun 1995, Eberhart dan Kennedy menemukan metode pencarian yang didasarkan dari perilaku sekawanan serangga yang kemudian dikenal sebagai metode swarm optimization. Serangga ini berangkat dari satu titik pencarian yang acak untuk kemudian bergerak berdasarkan suatu informasi dari sekawanan lainnya menuju lokasi-lokasi pencarian [4]. Metode ini tidaklah jauh berbeda dibandingkan dengan metode metaheuristik lain seperti algoritma genetika. Perbedaan mendasar adalah tidak adanya proses crossover dan mutation yang melibatkan seluruh populasi sehingga memperkecil kebutuhan komputasi [11].

Perkembangan teknologi komputer membuat kemajuan yang sangat pesat di bidang metode metaheuristik. Beberapa penelitian terkait metaheuristik telah dilakukan dalam bidang kontrol.

menggunakan algoritma genetika dalam masalah penerbangan. Dracopoulos [8] memanfaatkan pemrograman genetika dan Mobayen [10, 11] menggunakan PSO dan algoritma evolusi pada kontrol autoland. Byoung-Mun Min [7] menggunakan PSO dalam kontrol manuver helikopter. Abdelbar [1,2] mengembangkan pembobotan pada PSO menggunakan fungsi distribusi dan membandingkan antara fungsi distribusi Cauchy dan Gaussian.

Kontrol Optimal LQR

Diberikan persamaan *state-space*

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

$$y(t) = Cx(t) + Du(t) \quad (2)$$

dengan performa indeks

$$J = \frac{1}{2}x^T Px + \frac{1}{2} \int_{t_0}^T x^T Qx + u^T Ru dt \quad (3)$$

Persamaan 3 akan dihitung menggunakan persamaan $J = \frac{1}{2}x_0^T S(t_0)x_0$. S(t) merupakan solusi dari persamaan diferensial Riccati [6,9]

$\dot{S}(t) = S(t)A - S(t)BR^{-1}B^T S(t) + A^T S(t) + Q$ dengan syarat batas $S(T) = P$. Q dan R adalah matriks semi-definit positif dan definit positif. Nilai kontrol $u(t)$ yang akan meminimalkan Persamaan (3) adalah

$$u(t) = -R^{-1}B^T S(t)x(t) \quad (5)$$

Adaptive Particle Swarm Optimization

Misalkan posisi dan kecepatan awal partikel ke- $i = 1, 2, \dots, n$ adalah X_1^i dan V_1^i berupa nilai random diantara domain $I = [a, b]$. Posisi dan kecepatan partikel pada iterasi $k = 1, 2, \dots, M$ diberikan oleh

$$X_{k+1}^i = X_k^i + V_{k+1}^i \quad (6)$$

dan

$$V_{k+1}^i = V_k^i + c_1\gamma(Pb_k^i - X_i^k) + c_2\gamma(Gb_k - X_i^k) \quad (7)$$

c_1, c_2 adalah nilai kognitif dari setiap swarm, optimal di angka 1.4 [4], Pb adalah nilai terbaik sebuah swarm sepanjang pencarian, Gb adalah nilai terbaik seluruh swarm pada seluruh iterasi dan γ merupakan bilangan random $[0,1]$.

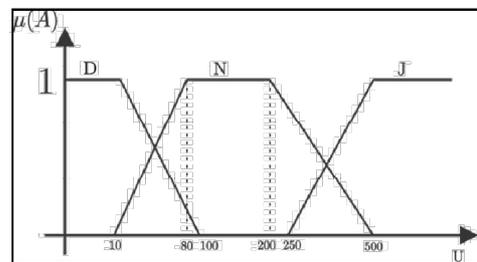
Adaptive PSO merupakan modifikasi PSO standar dengan menambahkan pembobotan dan

mempertimbangkan bobot multi globalbest, sehingga diperoleh

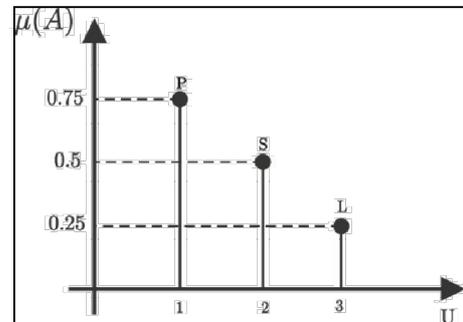
$$V_{k+1}^i = \omega_k V_k^i + c_1\gamma(Pb_k^i - X_i^k) + c_2\gamma \sum_{l=1}^D \psi(l) Gb_l - X_i^k \quad (8)$$

dengan ω adalah nilai pembobotan dan D adalah jumlah swarm yang akan menjadi globalbest. Nilai $\psi(l)$ merupakan bobot masing-masing globalbest yang ditentukan dengan logika fuzzy.

Fuzzifikasi menggunakan input berupa variabel linguistik jarak dan relasi fitness yang kurva fungsi keanggotaannya digambarkan dalam Gambar 1 dan Gambar 2.



Gambar 1. Fungsi keanggotaan untuk input jarak fitness



Gambar 2. Fungsi keanggotaan untuk input relasi fitness

Setiap dari input akan diterjemahkan dalam proses inferensi, yang aturannya diberikan pada Tabel 1

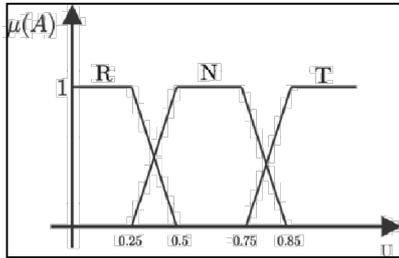
Tabel 1. Tabel inferensi

Relasi	Jarak		
	Dekat (D)	Normal (N)	Jauh (J)
Primer (P)	Tinggi	Normal	Tinggi
Sekunder (S)	Rendah	Tinggi	Tinggi
Low (L)	Rendah	Tinggi	Tinggi

Inferensi akan dihitung dengan menggunakan aturan Mamdani Addictif, yang dinyatakan dalam

$$\mu_{a \cap b} = \min(1, \mu(A) + \mu(B)) \quad (9)$$

Selanjutnya, proses defuzzifikasi dengan fungsi keanggotaan adalah bobot globalbest, ditunjukkan pada Gambar 3.



Gambar 3. Fungsi keanggotaan untuk output bobot globalbest

Fungsi keanggotaan bobot globalbest akan diterjemahkan menggunakan Metode Centroid yaitu

$$y^* = \frac{\sum_{i=1}^M \bar{y}^i \cdot \mu_{y_i}(\bar{y}^i)}{\sum_{i=1}^M \mu_{y_i}(\bar{y}^i)} \quad (10)$$

Algoritma menentukan matriks pembobot menggunakan APSO ditunjukkan pada proses berikut

1. Tentukan nilai awal posisi dan kecepatan secara random.
2. Cek nilai IAE sebagai fitness. IAE didefinisikan sebagai

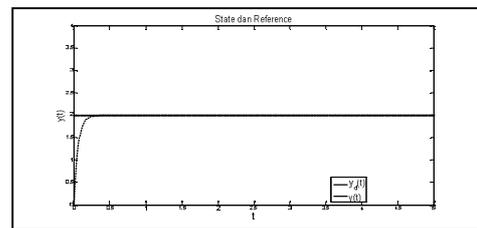
$$\int_{t_0}^T |y_d(t) - y(t)| dt \quad (11)$$
3. Tentukan personalbest dan globalbest pada iterasi awal.
4. Proses perulangan
 - a. Update nilai kecepatan masing-masing swarm menggunakan persamaan 8.
 - b. Update nilai posisi dari masing-masing swarm dengan persamaan 6.
 - c. Cek nilai IAE sebagai fitness dengan menggunakan persamaan 11.
 - d. Tentukan personalbest dengan mengambil nilai terkecil fitness dari suatu swarm dari seluruh iterasi.
 - e. Tentukan jenis bobot globalbes melalui fuzzifikasi dan inferensi menggunakan fungsi keanggotaan pada Gambar 1, Gambar 2 dan persamaan 9.
 - f. Tentukan nilai bobot multi globalbest dengan Metode Centroid pada Tabel 1 dan persamaan 10.
 - g. Cek stopping kriteria berupa iterasi maksimum dan error maksimum dari fitness. Jika kriteria tidak terpenuhi kembali ke langkah ke-3 sampai iterasi maksimum
5. Iterasi selesai dan hitung IAE final.

Simulasi Numerik

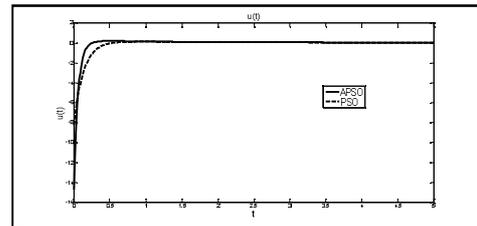
Contoh 1: Misalkan diberikan sistem plant

$$\dot{x}(t) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 2 \end{pmatrix} u(t)$$

Dengan output $y(t) = (1 \ 1)x(t)$ yang diinginkan untuk mengikuti lintasan $y_d = 2$. Hasil simulasi berupa output dan tracking line diberikan pada Gambar 4. Kontrol optimal sepanjang t ditunjukkan pada Gambar 5.

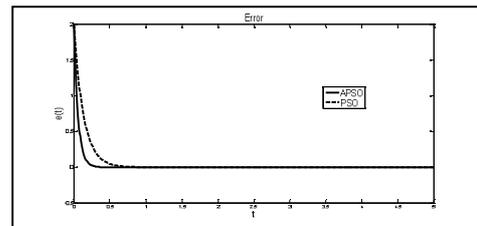


Gambar 4. Output sistem dan lintasan yang ditracking

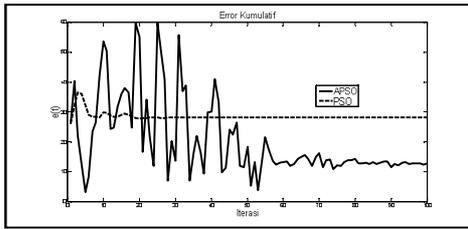


Gambar 5. Kontrol u(t) sepanjang t

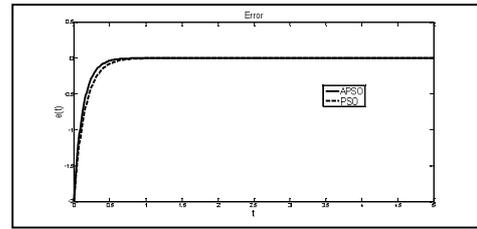
Hasil simulasi pada Contoh 1 menunjukkan bahwa output sistem yang dikontrol dapat mengikuti lintasan yang diberikan. Selanjutnya perbandingan error state antara PSO dan APSO ditunjukkan pada Gambar 6 sedangkan IAE ditunjukkan pada Gambar 7.



Gambar 6. Error state antara PSO dan APSO



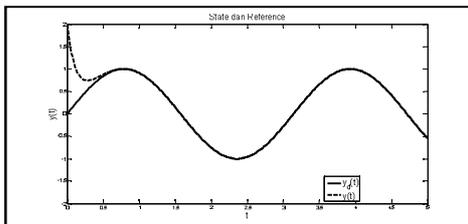
Gambar 7. IAE antara PSO dan APSO



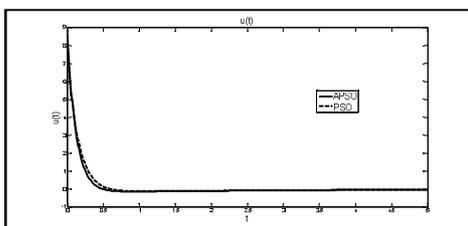
Gambar 10. Error state antara PSO dan APSO

Nilai error menunjukkan bahwa output sistem dengan menggunakan APSO lebih cepat mendekati tracking line dibandingkan dengan PSO standar. Berdasarkan perbandingan nilai IAE, APSO memiliki nilai IAE yang lebih kecil pada iterasi yang dilakukan. Hal ini berarti APSO mampu menemukan nilai pembobot yang lebih baik di setiap iterasinya yang berakibat kontrol dapat menarik output lebih dekat ke tracking line dibandingkan dengan PSO standar.

Contoh 2 : Misalkan diberikan sistem plant pada Contoh 1 dengan mengganti tracking line menjadi $y_d = \sin 2t$. Hasil simulasi berupa output dan tracking line diberikan pada Gambar 8. Kontrol optimal sepanjang waktu t ditunjukkan pada Gambar 9.

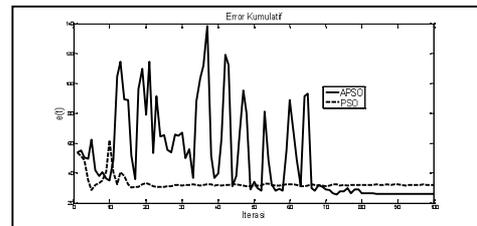


Gambar 8. Output sistem dan lintasan yang ditracking



Gambar 9. Nilai kontrol $u(t)$

Selanjutnya adalah perbandingan error dan IAE antara PSO dan APSO ditunjukkan pada Gambar 10 dan Gambar 11.



Gambar 11. IAE antara PSO dan APSO

Contoh 2 merupakan sistem yang sama dengan Contoh 1, dengan merubah tracking line yang lebih kompleks. Simulasi ini ingin menunjukkan bahwasanya pada tracking lines yang lebih kompleks, APSO juga mampu menemukan matriks pembobot yang lebih baik dibandingkan dengan PSO standar. Gambar 8 dan Gambar 9 menunjukkan bahwa kontrol mampu membuat output dari sistem untuk mengikuti tracking line dengan nilai kontrol yang masih berhingga. Kontrol juga mampu memaksa output untuk men-track lintasan pada waktu yang sangat cepat, yaitu pada kisaran $t=0.5$. Selanjutnya, berdasarkan nilai dari IAE, APSO juga masih mendapatkan nilai error dan IAE yang lebih kecil dibandingkan dengan PSO standar.

Kesimpulan

Hasil pada Contoh 1 dan Contoh 2 menunjukkan bahwasanya APSO dapat menghasilkan nilai pembobot yang lebih tepat sehingga menghasilkan kontrol yang lebih baik dalam membawa output sistem plant mengikuti lintasan tracking dan memiliki IAE yang lebih kecil dibandingkan dengan PSO standar.

Daftar Pustaka

- Abdelbar, A.M., Abdelshahid, S., dan Wunsch, D.C.**, Gaussian versus cauchy membership functions in fuzzy PSO. *Proceeding of International Joint Conference on Neural Network*, 978-1-4244-1379-9, Halaman 2902-2907, Agustus 2007.
- Abdelbar, A.M., Abdelshahid, S., dan Wunsch,** Fuzzy pso : A generalization of particle swarm optimization. *Proceeding of International Joint Conference on Neural Network*, (0-7803-9048-2/05):1086-1091, Juli 2005.
- Aranda, J., de la Cruz, J.M., Parrilla, M., and Rui Perez, P.**, Design of a linear quadratic optimal control of aircraft flight control by genetic algorithm. *Portuguese Conference of Automatic Control*, Halaman 394-399, 2000.
- Bratton, D. dan Kennedy, J.**, Defining a standard for particle swarm optimization. *Proceeding of the 2007 IEEE Swarm Intelligence Symposium*, (1-4244-0708-7/07), 2007
- Bryson, A.E. dan Y.C. Ho.**, *Applied Optimal Control*, Hemisphere, New York, 1975.
- Burl, J.F.**, *Linear Optimal Control H₂ and H_∞ Methods*, Addison-Wesley, California, 1998.
- Byoung-Mun Min, Hyo-Sang Shin, dan Min-Jea Tahk.** Control system design for autonomous helicopter using particle swarm optimization. *25th International Congress of the Aeronautical Science*, Halaman 1-7, 2006.
- Dracopoulos, D.C.**, Autolanding of commercial aircrafts by genetic programming. *Proceeding of The World Congress on Engineering I*, 978-988-98671-5-7, Juli, 2007.
- Lewis, F.J. dan Syrmos, V.L.**, *Optimal Control 2nd edition*. John Wiley & Sons, New York, 1995.
- Mobaieen, S., Mohamady, B., Ghorbani, H., dan Rabii, A.**, Optimal control design using evolutionary algorithms with application to an aircraft landing system, *Journal of Basic and Applied Scientific Research*, (2(2):1876-1882):1876-1882, 2012.
- Mobayen, S., Rabiei, A., Moradi, M., dan Mohammady, B.**, Linear quadratic optimal control system design using particle swarm optimization algorithm. *International Journal of the Physical Sciences*, 6(30):6958-6966, November, 2011.