

Desain Optimal Kontrol PID Menggunakan Craziness Particle Swarm pada Permanent Magnet Synchronous Motor

Faisal^{1*}, Muhammad Ruswandi Djalal²

^{1,2}Jurusan Teknik Mesin, Politeknik Negeri Ujung Pandang, Makassar

ABSTRACT

Pengontrol Proporsional Integral Derivatif (PID) pada motor sinkron banyak digunakan karena struktur yang sederhana, ketangguhan, kekuatan, dan kemudahan penggunaannya. Penggunaan pengontrol PID memerlukan pengaturan parameter yang tepat untuk kinerja optimal pada motor. Solusi yang sering digunakan adalah metode uji-coba untuk menentukan parameter yang benar untuk PID, tetapi hasil yang diperoleh tidak membuat pengontrol PID optimal. Baru-baru ini, banyak penelitian telah dilakukan untuk mengoptimalkan pengontrol PID dengan metode cerdas. Untuk itu, penelitian ini akan menggunakan metode Craziness Particle Swarm Optimization (CRPSO) untuk mengoptimalkan dan menentukan parameter yang tepat dari PID. Metode CRPSO adalah metode yang memberikan inovasi pada fungsi kecepatan partikel yang didistribusikan dalam metode PSO. Dari hasil simulasi, kinerja CRPSO lebih optimal dibandingkan PSO. Dari hasil penyetelan parameter PID yang benar, diperoleh respons overshoot minimum dengan beberapa variasi kecepatan. Selain itu, juga diperoleh peningkatan torsi awal PMSM menggunakan CRPSO

Kata Kunci: *Synchronous Motor, PID, CRPSO, PSO, Overshoot*

ABSTRACT

A Proportional Integral Derivative (PID) controller is widely used in synchronous motors due to its simple structure, robustness, effectiveness, and ease of use. Proper parameter settings are crucial for achieving optimal motor performance with a PID controller. Typically, the trial-and-error method is employed to determine the correct PID parameters; however, this approach often fails to achieve the best results. Recently, many studies have focused on optimizing PID controllers using intelligent methods. Consequently, this research employs the Craziness Particle Swarm Optimization (CRPSO) method to optimize and determine the appropriate PID parameters. The CRPSO method introduces an innovative velocity function for the particles in the Particle Swarm Optimization (PSO) technique. Simulation results show that CRPSO performs more effectively than traditional PSO. With the correctly tuned PID parameters, a minimal overshoot response was achieved across various speed conditions. Additionally, the starting torque of the Permanent Magnet Synchronous Motor (PMSM) was also improved using CRPSO.

Keywords: *Synchronous Motor, PID, CRPSO, PSO, Overshoot*

1. PENDAHULUAN

Motor listrik AC telah banyak digunakan sebagai pengganti motor DC karena arus AC dapat dihasilkan dan didistribusikan dengan biaya yang lebih rendah dibandingkan arus DC. Selain itu, motor listrik AC memiliki keunggulan dalam hal harga, ukuran, dan berat serta memerlukan pemeliharaan yang lebih sedikit dibandingkan motor DC. Salah satu jenis motor listrik AC adalah motor sinkron. Motor sinkron telah banyak digunakan di industri, dan pengontrol Proporsional Integral Derivatif (PID) digunakan sebagai pengendali [1].

Penggunaan peralatan PID juga memiliki masalah, yaitu nilai gain yang masih diatur secara manual. Kontrol PID memiliki kinerja yang baik, sehingga banyak digunakan di dunia industri. Namun, kekurangannya adalah memerlukan perhitungan matematis yang rumit untuk menentukan setiap nilai gain dari PID [2].

Algoritma cerdas saat ini banyak dibahas sebagai metode optimasi untuk sistem kontrol [3]. Selain itu, algoritma ini juga digunakan di berbagai bidang lainnya. Penyetelan parameter kontrol, seperti PID saat ini, juga telah dilakukan. Algoritma cerdas adalah metode yang mengadopsi habitat atau kebiasaan tertentu dan kemudian mengubahnya menjadi algoritma yang dapat menyelesaikan suatu masalah [4]. Berbagai metode optimasi berbasis metode konvensional dan metode cerdas telah banyak digunakan untuk mengoptimalkan parameter PID pada motor listrik, termasuk Artificial Bee Colony [5, 6], Evolutionary [7, 8], Bio-Inspired Algorithm [9, 10], Bacterial Foraging [11, 12], dan beberapa metode lainnya yang dapat menemukan parameter PID yang optimal. Dari hasil penelitian ini, ditemukan bahwa kinerja PID yang optimal diperoleh dengan parameter PID yang benar. Keuntungan menggunakan algoritma cerdas adalah sistem dapat secara otomatis dan adaptif menyesuaikan parameter yang tepat dengan fungsi objektif yang diberikan.

Salah satu algoritma cerdas yang telah banyak diterapkan adalah Particle Swarm Optimization (PSO) untuk penyetelan PID [13, 14]. Kelemahan PSO biasa adalah cenderung mencapai konvergensi pada optima lokal terlalu cepat [20]. terutama pada masalah yang bersifat intermittente. Akibatnya, hasilnya hanya berupa optima lokal, bukan optima global [15]. Seiring waktu, telah terjadi perkembangan signifikan dalam metode optimasi PSO, salah satunya adalah Craziiness Particle Swarm Optimization (CRPSO).

CRPSO adalah metode yang memberikan inovasi pada fungsi kecepatan partikel yang didistribusikan dalam metode PSO. Perbedaan antara PSO dan CRPSO terletak pada fungsi kecepatan yang digunakan. Disebut "crazy" partikel karena, pada kesempatan tertentu, yang ditentukan oleh probabilitas kegilaan, fungsi kecepatan partikel tidak akan menuju posisi dengan hasil optimal. Sebaliknya, partikel ini akan bergerak secara bebas dalam ruang masalah untuk mencari nilai terkecil lainnya yang mungkin [15, 16].

Penelitian ini mengusulkan metode CRPSO untuk mengoptimalkan parameter PID pada motor sinkron untuk mengatasi masalah yang disebutkan di atas. Sebagai perbandingan, metode Particle Swarm Optimization konvensional digunakan.

2. PERMODELAN SYSTEM

A. Model Synchronous Motor

Gambar 1 menunjukkan diagram blok pemodelan PMSM. Persamaan sistem PMSM dimodelkan sebagai model d-q, yang dapat dinyatakan seperti pada penelitian [5]. L_d adalah induktansi sumbu langsung, yaitu induktansi yang muncul dalam persamaan hubungan fluks. L_q adalah induktansi sumbu kuadrat. Persamaan hubungan fluks ditunjukkan dalam persamaan (1) dan (2). v_{sd} adalah tegangan stator sumbu d-q, dan v_{sq} tertulis dalam persamaan (3) dan (4).

$$\lambda_{sd} = L_d I_{sd} + \lambda_{PM} \quad (1)$$

$$\lambda_{sq} = L_q I_{sq} \quad (2)$$

$$V_{sd} = R_s i_{sd} + L_d \frac{d}{dt} i_{sd} - \omega_r L_q i_{sq} \quad (3)$$

$$V_{sq} = R_s i_{sq} + L_q \frac{d}{dt} i_{sq} + \omega_r L_d i_{sd} + \omega_r \lambda_{PM} \quad (4)$$

λ_{sd} dan λ_{sq} adalah d-q hubungan axis stator flux. I_{sd} dan i_{sq} adalah d-q arus axis stator. λ_{PM} adalah hubungan permanent magnetic flux. R_s adalah stator resistansi. ω_r adalah kecepatan motor listrik. T_{EM} electromagnetic torsi dituliskan dalam persamaan equation (5), diasumsikan bahwa L_d sama dengan L_q [13].

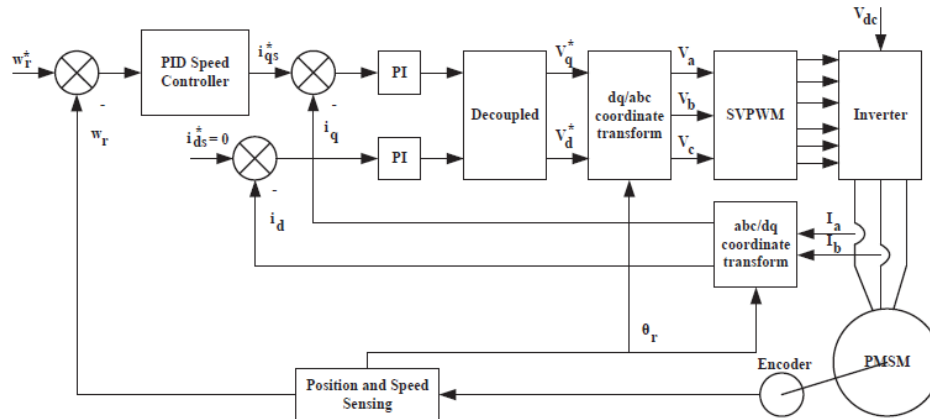
$$T_{EM} = \frac{3p}{2} \lambda_{PM} i_{sq} \quad (5)$$

Diagram vektor pada bingkai sinkron ditunjukkan pada Gambar 2. i_{sq} ditulis dalam bentuk i_s dan variasi torsi dengan δ . Hasil yang diharapkan diperoleh dalam persamaan (6).

$$T_{EM} = \frac{3p}{2} \lambda_{PM} |i_s| \sin(\delta) \quad (6)$$

Fluks yang dihasilkan oleh magnet permanen dianggap konstan, sementara torsi bervariasi dengan arus sumbu q. Oleh karena itu, torsi konstan diperoleh dengan membuat i_{sq}

konstan. Dari persamaan (6), torsi maksimum diperoleh jika sudut torsi adalah 90°. Ini sesuai dengan arus sumbu d yang nol. Jadi, dengan mengubah arus sumbu q dan menjaga arus sumbu d tetap mendekati nol, respons torsi yang cepat diperoleh dalam penggerak PMSM [13].



Gambar 1. Block Diagram PMSM [13]

B. Pemodelan PID Controller

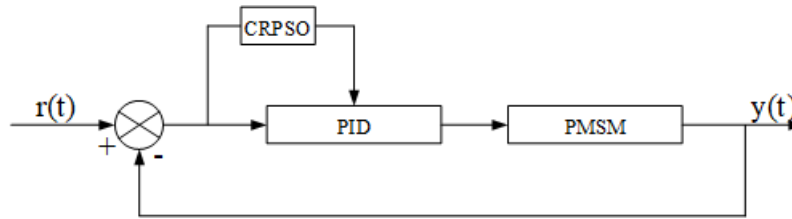
Kontrol PID adalah salah satu jenis kontrol yang telah banyak digunakan dalam aplikasi industri karena strukturnya yang sederhana. Bentuk persamaan kontrol PID ditunjukkan dalam persamaan (7).

$$u(t) = k_p [e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}] \tag{7}$$

Di mana $u(t)$ adalah nilai kontrol yang dihitung oleh pengontrol PID, K_p adalah koefisien proporsional, T_i adalah konstanta waktu integral, dan T_d adalah konstanta waktu diferensial.

Parameter proporsional menggambarkan sinyal deviasi sistem kontrol proporsional $e(t)$. Ketika sinyal $e(t)$ muncul, pengontrol PID menghasilkan efek kontrol segera untuk mengurangi drift. Parameter integral digunakan untuk menghilangkan kesalahan statis dan meningkatkan stabilitas sistem. Parameter diferensial mencerminkan perubahan dalam deviasi sinyal, memperkenalkan sinyal koreksi sebelum deviasi nilai sinyal menjadi lebih signifikan, dan mempercepat respons sistem untuk mengurangi waktu penundaan.

Oleh karena itu, merancang pengontrol PID berarti menentukan ketiga parameter yang tepat, serta cara mengonfigurasi ketiga parameter PID tersebut. Dalam penelitian ini, CRPSO diusulkan untuk mencari parameter PID yang optimal. Diagram blok sistem kontrol ditunjukkan pada Gambar 2.



Gambar 2. PID Modeling

3. CRAZINESS PARTICLE SWARM OPTIMIZATION

Metode optimasi Craziness Particle Swarm Optimization (CRPSO) adalah pengembangan dari PSO konvensional. Kennedy dan Eberhart memperkenalkan PSO pada tahun 1995. Dalam PSO, setiap kandidat didistribusikan di ruang masalah dan bergerak berdasarkan konsep kecepatan partikel untuk memperoleh hasil optimal [17, 18]. Konsep PSO dapat diformulasikan dalam Persamaan (8) dan (9) sebagai berikut.

Pembaruan kecepatan partikel:

$$v_i^{k+1} = v_i + c_1 r_1 (pBest_i - x_i^k) + c_2 r_2 (gBest_i - x_i^k) \quad (8)$$

Pembaruan posisi partikel:

$$x_i^{k+1} = x_i + v_i^{k+1} \quad (9)$$

Kelemahan dari PSO konvensional adalah kecenderungan untuk mencapai konvergensi prematur. Kondisi ini menyebabkan solusi yang diperoleh menjadi optima lokal. Saat menggunakan CRPSO, terdapat perubahan pada fungsi pembaruan kecepatan, yang memungkinkan partikel bergerak di luar aturan kecepatan pada iterasi tertentu. Iterasi ini ditentukan oleh probabilitas yang disebut sebagai partikel gila. Nilai partikel gila dipengaruhi oleh perubahan bobot yang terjadi pada setiap iterasi, seperti yang ditunjukkan dalam Persamaan (10) dan (11) di bawah ini [19, 20].

Pembaruan weight partikel:

$$w^k = (w_{max} - w_{min}) \times \frac{iter_{max} - iter^k}{iter_{max}} + w_{min} \quad (10)$$

$$P_{craz} = w_{min} - \exp\left(-\frac{w^k}{w_{max}}\right) \quad (11)$$

Perubahan yang terjadi dalam pembaruan kecepatan berdasarkan nilai P_{craz} ditunjukkan dalam Persamaan (12) sebagai berikut.

$$v_i^k = \begin{cases} rand(0, v_{max}) & \text{if } P_{craz} \leq rand(0,1) \\ v_i^k & \text{other} \end{cases} \quad (12)$$

4. PENALARAN PID DENGAN CRPSO

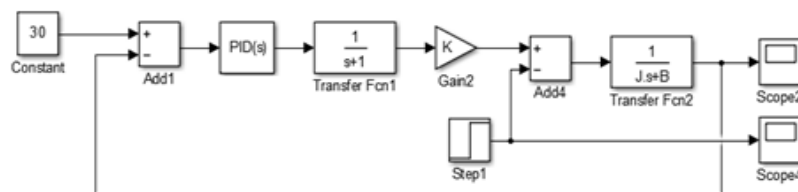
Metode CRPSO digunakan dalam penelitian ini untuk menyetel parameter PID. CRPSO bekerja dengan fungsi objektif, yaitu meminimalkan Integral Absolute Time Error (ITAE), seperti yang ditunjukkan dalam Persamaan (13).

$$ITAE = \int_0^t |\Delta\omega(t)| dt \tag{13}$$

Parameter PID yang disetel oleh CRPSO adalah K_p , K_i , dan K_d . Tabel 1 menunjukkan parameter CRPSO yang digunakan dalam simulasi. Gambar 3 menunjukkan pemodelan PMSM menggunakan Simulink-Matlab.

Tabel 1. CRPSO Parameters

Parameter	Nilai
Ukuran Populasi	20
Probabilitas Pergantian	0.8
Parameter Iterasi	50
Dimensi	3
Batas atast K_p , K_i , K_d	[1 1 0.1]
Batas bawah K_p , K_i , K_d	[0.1 0.1 0.01]



Gambar 3. Model Synchronous Motor [13]

5. HASIL DAN PEMBAHASAN

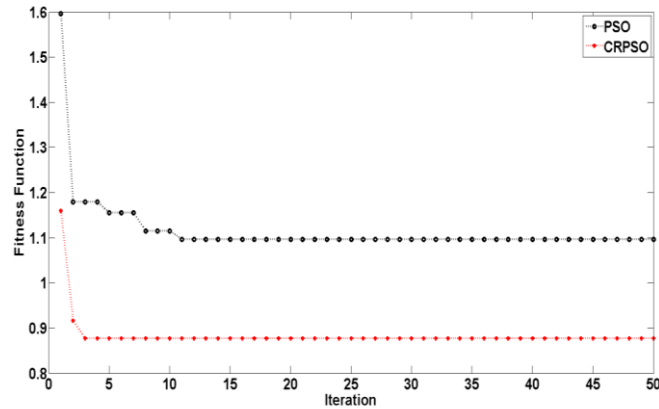
Hasil penyetelan parameter PID menggunakan metode CRPSO ditunjukkan dalam Tabel 2 di bawah ini, sebagai perbandingan dengan metode PSO konvensional.

Tabel 2. PID Parameter Results

Parameter	PSO	CRPSO
K_p	0.8481	0.2229
K_i	0.4875	0.1228
K_d	0.1832	0.1864

Proses penyetelan parameter PID ditunjukkan dalam grafik konvergensi pada Gambar 4 di bawah ini. Dari hasil simulasi, nilai fungsi fitness untuk PSO adalah 1.097, sementara dengan menggunakan CRPSO, nilai fungsi fitness adalah 0.877. Dari Gambar 4, dapat

diketahui bahwa proses konvergensi algoritma CRPSO adalah yang tercepat, yaitu pada iterasi ke-3 algoritma CRPSO telah menemukan parameter PID yang optimal. Sementara itu, proses konvergensi PSO yang optimal terjadi pada iterasi ke-11. Hal ini menunjukkan bahwa proses konvergensi CRPSO lebih cepat dalam mengoptimalkan PID PMSM.



Gambar 4. Proses Konvergensi

A. Karakteristik Kecepatan PMSM dengan Input Kecepatan 50 rpm

Analisis kinerja PMSM yang pertama adalah mengamati respons kecepatan motor dengan beberapa variasi titik set kecepatan. Eksperimen dilakukan dengan menetapkan titik set kecepatan motor pada putaran 50 rpm. Tabel 3 menunjukkan karakteristik overshoot kecepatan untuk setiap jenis eksperimen.

Tabel 3. Respons Overshoot Kecepatan PMSM dengan Input 50 rpm

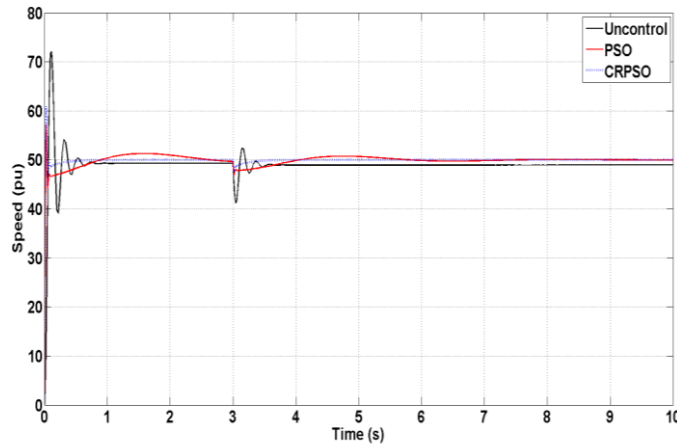
Skema	Load I		Load II
	Upper (rpm)	Lower (rpm)	Upper (rpm)
Uncontrol	72.05	41.17	52.29
PSO	60.93	46.92	50.72
CRPSO	57.25	47.68	49.87

Tabel 3 menunjukkan karakteristik respons overshoot kecepatan PMSM dengan titik set 50 rpm dalam beberapa variasi skema kontrol. Hasil uji menunjukkan bahwa pada pemuatan pertama, sistem tanpa kontrol mengalami overshoot yang berayun hingga 72.05 rpm, sementara pada pemuatan kedua, sistem mengalami perubahan beban, dengan respons yang dihasilkan berkisar antara 41.17 rpm hingga 52.29 rpm. Respons semacam ini tentu berbahaya bagi desain. Di sisi lain, hal ini dapat menyebabkan masalah bagi PMSM.

Hasil uji kedua dengan skema kontrol PID-PSO menunjukkan kinerja yang lebih baik. Dalam kondisi pemuatan pertama, PMSM mengalami overshoot sebesar 60.93 rpm.

Kemudian, dalam kondisi perubahan beban, respons sistem yang berayun diperoleh dalam kisaran 46.92 rpm hingga 50.72 rpm.

Pada kondisi beban ketiga, kinerja yang lebih optimal diperoleh dengan PID-CRPSO, di mana respons awal motor adalah 57.25 rpm, sementara pada beban kedua, respons yang dihasilkan berkisar antara 47.68 rpm hingga 49.87 rpm. Lihat respons sistem secara grafis, yang ditunjukkan pada Gambar 5.



Gambar 5. Respons Kecepatan PMSM dengan Input Kecepatan 50 rpm

Respons kecepatan PMSM dengan kontrol berbasis PID-CRPSO menghasilkan kinerja optimal dengan rata-rata kesalahan minimum sebesar 6.46%. Sementara itu, menggunakan kontrol berbasis PID-PSO, diperoleh kesalahan sebesar 9.82%. Sistem tanpa kontrol merupakan skema yang terburuk karena memiliki kesalahan yang signifikan sebesar 22.11%. Kesalahan operasi PMSM ditunjukkan dalam Tabel 4 di bawah ini.

Tabel 4. Kesalahan Respons Overshoot Kecepatan PMSM dengan Input 50 rpm

Skema	Load I		Load II	Error Average (%)
	Upper Error (%)	Lower Error (%)	Upper Error (%)	
Uncontrol	44.1	17.66	4.58	22.11333333
PSO	21.86	6.16	1.44	9.82
CRPSO	14.5	4.64	0.26	6.466666667

B. Karakteristik Kecepatan PMSM dengan Input Kecepatan 100 rpm

Untuk menguji kinerja PMSM dengan variasi input kecepatan 100 rpm. Tabel 5 menunjukkan karakteristik overshoot kecepatan dengan input 100 rpm.

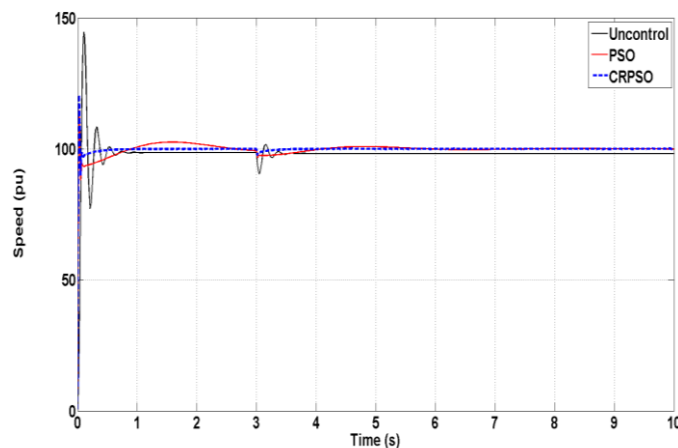
Tabel 5. Respons Overshoot Kecepatan PMSM dengan Input 100 rpm

Skema	Load I	Load II	
	Upper (rpm)	Lower (rpm)	Upper (rpm)
Uncontrol	144.6	90.45	101.7
PSO	120.7	96.73	100.8

Pada uji variasi kecepatan kedua dengan input 100 rpm, kinerja PMSM masih sangat optimal. Karakteristik PMSM masih dapat diandalkan dengan kontrol berbasis PID-CRPSO. Tabel 5 menunjukkan karakteristik respons overshoot kecepatan PMSM dengan titik set 100 rpm dalam beberapa variasi skema kontrol. Hasil uji menunjukkan bahwa pada pemuatan pertama, sistem tanpa kontrol mengalami overshoot yang berayun hingga 144.6 rpm, sementara pada pemuatan kedua, sistem mengalami perubahan beban, dengan respons yang dihasilkan berkisar antara 90.45 rpm hingga 101.7 rpm.

Hasil uji kedua dengan skema kontrol PID-PSO menunjukkan bahwa pada kondisi pemuatan pertama, PMSM mengalami overshoot sebesar 120.7 rpm. Kemudian, dalam kondisi perubahan beban, respons sistem yang berayun diperoleh dalam kisaran 96.73 rpm hingga 101.8 rpm.

Pada kondisi beban ketiga, kinerja yang lebih optimal diperoleh dengan PID-CRPSO, di mana respons awal motor adalah 114.6 rpm, sementara pada pemuatan kedua, respons yang dihasilkan berkisar antara 98.8 rpm hingga 100 rpm. Lihat respons sistem secara grafis, yang ditunjukkan pada Gambar 6.



Gambar 6. Respons Kecepatan PMSM dengan Input Kecepatan 100 rpm

Dalam pengujian dengan input 100 rpm, kinerja PMSM masih optimal dalam operasinya. Respons kecepatan PMSM dengan kontrol berbasis PID-CRPSO menghasilkan kinerja optimal dengan rata-rata kesalahan minimum sebesar 5.26%. Sementara itu, menggunakan kontrol berbasis PID-PSO, diperoleh kesalahan sebesar 8.25%. Sistem tanpa kontrol merupakan skema yang terburuk karena memiliki kesalahan yang signifikan sebesar 18.61%. Kesalahan operasi PMSM ditunjukkan dalam Tabel 6 di bawah ini.

Tabel 6. Kesalahan Respons Overshoot Kecepatan PMSM dengan Input 100 rpm

Skema	Load I		Load II	Error Average (%)
	Upper Error (%)	Lower Error (%)	Upper Error (%)	
Uncontrol	44.6	9.55	1.7	18.61667
PSO	20.7	3.27	0.8	8.256667
CRPSO	14.6	1.2	0	5.266667

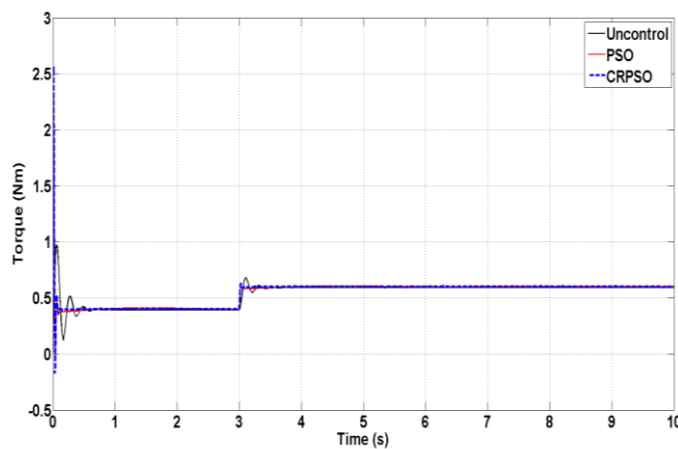
C. Karakteristik Respons Torsi PMSM dengan Input Kecepatan 50 rpm

Analisis karakteristik PMSM berikutnya adalah untuk mengamati respons torsi PMSM dengan variasi input kecepatan 50 rpm. Tabel 4 menunjukkan karakteristik torsi PMSM dengan input 50 rpm.

Tabel 7. Respons Torsi PMSM dengan Input Kecepatan 50 rpm

Uncontrol (Nm)	PSO (Nm)	CRPSO (Nm)
0.9707	2.32	2.562

Tabel 7 menunjukkan respons torsi PMSM saat memulai dengan beberapa skema kontrol. Hasil pengujian menunjukkan bahwa respons torsi maksimum menggunakan metode PID-CRPSO adalah 2.562 Nm. Sementara itu, dengan menggunakan PID-PSO, torsi yang dihasilkan adalah 2.32 Nm. Sistem tanpa kontrol menghasilkan torsi sebesar 0.9707 Nm. Gambar 7 menunjukkan respons torsi dalam bentuk grafis.



Gambar 7. Respons Torsi PMSM dengan Input Kecepatan 50 rpm

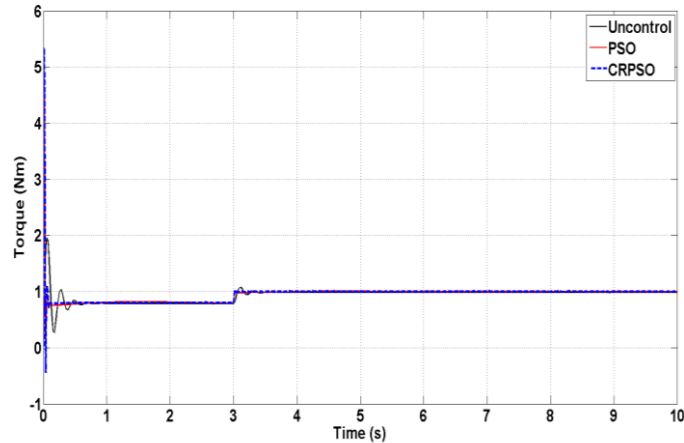
D. Karakteristik Respons Torsi PMSM dengan Input Kecepatan 100 rpm

Perbandingan karakteristik torsi dibahas dalam bagian ini. Berikut adalah uji torsi dengan input kecepatan 100 rpm. Tabel 8 menunjukkan karakteristik torsi PMSM dengan input 100 rpm.

Tabel 8. Respon Torsi PMSM dengan kecepatan Input 100 rpm

Uncontrol (Nm)	PSO (Nm)	CRPSO (Nm)
1.953	4.826	5.345

Tabel 8 menunjukkan respons torsi saat memulai PMSM dengan beberapa skema kontrol. Hasil pengujian menunjukkan bahwa respons torsi maksimum menggunakan metode PID-CRPSO adalah 5.345 Nm. Sementara itu, dengan menggunakan PID-PSO, torsi yang dihasilkan adalah 4.826 Nm. Sistem tanpa kontrol menghasilkan torsi sebesar 1.953 Nm. Gambar 8 menunjukkan respons torsi dalam bentuk grafis.



Gambar 8. PMSM Torque response with Speed Input 100 rpm

6. KESIMPULAN

Penggunaan metode optimasi cerdas berbasis CRPSO untuk mengoptimalkan parameter PID memberikan hasil yang optimal. Hal ini terlihat dari respons overshoot untuk setiap titik set input yang diberikan. Pengontrol PID dapat meredam osilasi PMSM, terutama saat terjadi gangguan dalam injeksi beban. Parameter PID dengan optimasi berbasis CRPSO adalah $K_p = 0.2229$, $K_i = 0.1228$, and $K_d = 0.1864$. Proses konvergensi CRPSO lebih cepat dibandingkan dengan PSO. Untuk CRPSO, konvergensi terjadi pada iterasi ke-3, sementara PSO baru konvergen pada iterasi ke-11. Respons torsi saat memulai motor induksi dengan metode CRPSO adalah yang terbesar dibandingkan dengan PSO

DAFTAR PUSTAKA

- [1] A. A. Kumar, "Application of GA for PID controller of a Real time industrial Process," *Sensors and transducers*, 10/10 2010.
- [2] K. Lu, W. Zhou, G. Zeng, and W. Du, "Design of PID controller based on a self-adaptive state-space predictive functional control using extremal optimization

- method," *Journal of the Franklin Institute*, vol. 355, no. 5, pp. 2197-2220, 2018/03/01/ 2018, doi: <https://doi.org/10.1016/j.jfranklin.2017.12.034>.
- [3] M. Ali *et al.*, "The comparison of dual axis photovoltaic tracking system using artificial intelligence techniques," vol. 10, no. 4, p. 901, 2021.
- [4] S. Qiu *et al.*, "Intelligent algorithm tuning PID method of function electrical stimulation using knee joint angle," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 26-30 Aug. 2014 2014, pp. 2561-2564, doi: 10.1109/EMBC.2014.6944145.
- [5] S. Pareek, M. Kishnani, and R. Gupta, "Application of Artificial Bee Colony Optimization for optimal PID tuning," in *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, 1-2 Aug. 2014 2014, pp. 1-5, doi: 10.1109/ICAETR.2014.7012817.
- [6] H. Wang, H. Du, Q. Cui, and H. Song, "Artificial bee colony algorithm based PID controller for steel stripe deviation control system," vol. 105, no. 1, p. 00368504221075188, 2022, doi: 10.1177/00368504221075188.
- [7] D. Maddi, A. Sheta, D. Davineni, and H. Al-Hiary, "Optimization of PID Controller Gain Using Evolutionary Algorithm and Swarm Intelligence," in *2019 10th International Conference on Information and Communication Systems (ICICS)*, 11-13 June 2019 2019, pp. 199-204, doi: 10.1109/IACS.2019.8809144.
- [8] U.-Q. Jorge-Humberto, H.-R. Jesús-Antonio, and M.-G. Nicolás, "Optimum PI/PID Controllers Tuning via an Evolutionary Algorithm," in *PID Control for Industrial Processes*, S. Mohammad Ed. Rijeka: IntechOpen, 2018, p. Ch. 3.
- [9] N. Katal and S. Singh, "Optimal Tuning of PID Controller for DC Motor using Bio-Inspired Algorithms," *International Journal of Computer Applications*, vol. 56, pp. 1-5, 10/01 2012, doi: 10.5120/8860-2822.
- [10] S. K. Valluru and M. Singh, "Optimization Strategy of Bio-Inspired Metaheuristic Algorithms Tuned PID Controller for PMBDC Actuated Robotic Manipulator," *Procedia Computer Science*, vol. 171, pp. 2040-2049, 2020/01/01/ 2020, doi: <https://doi.org/10.1016/j.procs.2020.04.219>.
- [11] K. M. Goher and S. O. Fadlallah, "Bacterial foraging-optimized PID control of a two-wheeled machine with a two-directional handling mechanism," *Robotics and Biomimetics*, vol. 4, no. 1, p. 1, 2017/03/23 2017, doi: 10.1186/s40638-017-0057-3.

- [12] V. Rajinikanth and K. Latha, "Bacterial foraging optimization algorithm based pid controller tuning for time delayedunstable systems," *Mediterranean Journal of Measurement and Control*, vol. 7, pp. 197-203, 01/01 2011.
- [13] M. Baskin and B. Caglar, "A modified design of PID controller for permanent magnet synchronous motor drives using particle swarm optimization," in *2014 16th International Power Electronics and Motion Control Conference and Exposition*, 21-24 Sept. 2014 2014, pp. 388-393, doi: 10.1109/EPEPEMC.2014.6980524.
- [14] E. S. Rahayu, Ma, #039, A. arif, and A. Çakan, "Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor," *2022, PID Control; PSO; Angular Speed; Arduino; Tuning* vol. 2, no. 2, p. 13 %J *International Journal of Robotics and Control Systems*, 2022-07-06 2022, doi: 10.31763/ijres.v2i2.476.
- [15] K. T. Chaturvedi, M. Pandit, and L. Srivastava, "Particle swarm optimization with crazy particles for nonconvex economic dispatch," *Applied Soft Computing*, vol. 9, no. 3, pp. 962-969, 2009/06/01/ 2009, doi: <https://doi.org/10.1016/j.asoc.2008.11.012>.
- [16] M. Saini, M. R. Djalal, and A. Yunus, "Design of a Robust PID-PSS & FACTS Using Craziiness Particle Swarm Optimization in Sulsebar System," *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 4, 2024.
- [17] R. M. Bamatraf, "DESAIN KONTROL OPTIMAL POWER SYSTEM STABILIZER (PSS) DAN FLEXIBLE AC TRANSMISSION SYSTEM (FACTS) MENGGUNAKAN CRAZINESS PARTICLE SWARM OPTIMIZATION (CRPSO) PADA SISTEM INTERKONEKSI JAWA BALI 500 KV," 2010.
- [18] M. Siswanto, M. Ali, M. A. Haikal, S. Wahyudi, S. Soedarsono, and M. R. Djalal, "Stability of Water Flow in Tanks Using Particle Swarm Optimization (PSO) Method," in *E3S Web of Conferences*, 2024, vol. 473: EDP Sciences, p. 04003.
- [19] M. R. Djalal, M. Saini, and A. S. Yunus, "Optimal design of PSS on SMIB using particle swarm optimization," *INTEK: Jurnal Penelitian*, vol. 8, no. 1, pp. 91-95, 2021.
- [20] M. SAINI, M. R. DJALAL, A. YUNUS, and A. PANGKUNG, "FACTS Devices Optimization for Optimal Power Flow Using Particle Swarm Optimization In Sulsebar System," *Przeglad Elektrotechniczny*, vol. 2024, no. 4, 2024.