

OPTIMISASI KONTROL PID UNTUK MOTOR DC MAGNET PERMANEN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION

Muhammad Ruswandi Djalal¹, Rahmat²

^{1,2}Jurusan Teknik Mesin, Program Studi Teknik Energi, Politeknik Negeri Ujung Pandang

^{1,2}Jalan Perintis Kemerdekaan km.10, Makassar 90245, Indonesia

E-mail : ¹wandi@poliupg.ac.id, ²rahmatpnup1991@gmail.com

ABSTRAK

Untuk memaksimalkan kinerja Motor DC, penggunaan controller yang tepat merupakan hal yang sangat penting. Desain controller Proportional Integral Derivative (PID) sebagai pengendali motor dc sudah sering dilakukan. Penggunaan controller PID dibutuhkan pengaturan parameter yang tepat untuk mendapatkan kinerja yang optimal pada motor. Umumnya yang sering digunakan adalah metode trial-error, untuk menentukan parameter dari controller PID, namun hasil yang didapat tidak membuat controller PID optimal dan justru akan merusak system. Untuk itu pada penelitian ini akan diusulkan salah satu metode penalaan parameter PID dengan menggunakan metode cerdas berbasis Particle Swarm Optimization (PSO), untuk mengoptimasi dan menentukan parameter yang tepat dari PID. PSO adalah salah satu metode cerdas yang terinspirasi dari perilaku partikel yang mencari sumber makanan secara berkelompok, konsep inilah yang diadaptasi dan diterapkan menjadi algoritma cerdas untuk menyelesaikan masalah optimasi. Dari hasil yang diperoleh metode PSO dapat dengan baik menala parameter PID, di mana untuk $K_p = 57.2409$, $k_i = 23.7090$, dan $k_d = 9.3575$. Respon kecepatan setelah penambaha PID, overshoot yang dihasilkan tidak ada dan settling time sangat cepat.

Kata Kunci : PID, Particle Swarm Optimization, Settling Time, Trial Error

ABSTRACT

To maximize the performance of DC motors, the proper use of the controller is crucial. Design of Proportional Integral Derivative (PID) controller on dc motor has been done frequently. The use of PID controllers requires proper parameter setting to achieve optimal performance on the motor. Commonly used is the trial-error method, to determine the parameters of the PID controller, but the results obtained do not make the PID controller optimal and will actually damage the system. Therefore, in this research we will propose one of PID parameter tuning method, that is by using Particle Swarm Optimization (PSO), to optimize and determine the proper parameters of PID. PSO is one of the smart methods inspired by the behavior of particles that search for food sources in groups, this concept is adapted and applied into intelligent algorithms to solve optimization problems. From the results obtained the PSO method can well tune the parameters of PID, where for $K_p = 57.2409$, $k_i = 23.7090$, and $k_d = 9.3575$. Response speed after PID receiver, resulting overshoot does not exist and settling time is very fast.

Keywords : PID, Particle Swarm Optimization, Settling Time, Trial Error

I. Pendahuluan

Aplikasi controller Proportional Integral Derivative (PID) sudah semakin banyak digunakan dalam pengontrolan motor dc. Beberapa keunggulan dari PID adalah struktur yang sederhana, stabilitas yang baik dan ketahanan yang kuat, memainkan peran penting dalam kontrol industri [1]. Adapun masalah utama yang sering dibahas dalam kontrol PID adalah penyesuaian parameter. Salah satu teknik yang sering digunakan adalah kontrol konvensional trial-error, namun untuk metode ini sulit untuk menyesuaikan parameter, sehingga dibutuhkan waktu lama serta akurasi kontrol tidak baik. Dalam beberapa tahun terakhir, para peneliti telah banyak menggunakan metode cerdas (*Artificial Intelligent*) untuk penentuan parameter PID motor DC. Para peneliti mulai mempelajari perilaku cerdas dari hewan untuk diaplikasikan untuk menyelesaikan masalah optimasi. Hewan

tersebut diantaranya lebah, semut, tawon dan menemukan suatu algoritma perilaku dari kebiasaan atau perilaku dari hewan tersebut.

Beberapa metode optimasi berbasis metode konvensional maupun metode cerdas telah banyak digunakan untuk mengoptimasi parameter PID motor DC magnet permanen, diantaranya Artificial Bee Colony [2], Evolutionary Algorithm [3], Bio-Inspired Algorithm [4], Tabu Search [5], Bacterial Foraging [6], Fuzzy Logic [7], dan Cuckoo Search [1].

Pada penelitian ini akan digunakan salah satu metode cerdas (*Artificial Intelligent*) untuk menala parameter PID motor DC yaitu dengan metode *Particle Swarm Optimization (PSO)* dan akan dianalisa dan dibandingkan respon kecepatan motor DC dengan metode konvensional PID trial-error serta motor DC magnet permanen tanpa controller.

II. Pemodelan Sistem

Motor DC magnet permanen adalah motor listrik yang memerlukan suplai tegangan arus searah (DC) pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/direct-unidirectional. Motor DC magnet permanen memiliki 3 bagian utama untuk dapat berputar yaitu : Magnet Permanen, Dinamo dan Komutator.

Untuk merancang dan mensimulasikan suatu sistim kontrol kecepatan motor DC diperlukan adanya model matematika dari plant yang akan dikontrol. Sehingga dalam memudahkan analisis sistem bisa digambarkan ke dalam model statis, model fisis (rangkaiian listrik) dan dinamik (diagram blok dan transfer function).

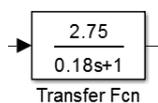
2.1. Pemodelan Motor DC

Pemodelan motor DC yang digunakan mengacu pada sebuah paper yang di mana menggunakan pemodelan transfer function. Motor DC yang digunakan di sini adalah menggunakan motor DC magnet permanen, rating kecepatan 1400 rpm dengan kecepatan saat pengukuran 1250 rpm. Dengan memasukkan kecepatan motor DC, maka transfer function motor DC adalah,

$$G(s) = \frac{K_m}{T_m s + 1} \quad (1)$$

Di mana, Km adalah gain mekanik dan Tm adalah waktu konstan mekanik.

$$G(s) = \frac{2.75}{0.18s + 1} \quad (2)$$



Gambar 1. Pemodelan Motor DC di Simulink

2.2. Pemodelan Kontroler PID [1]

Kontrol PID adalah salah satu kontrol yang sudah banyak digunakan pada aplikasi industri karena strukturnya yang sederhana [15,16].

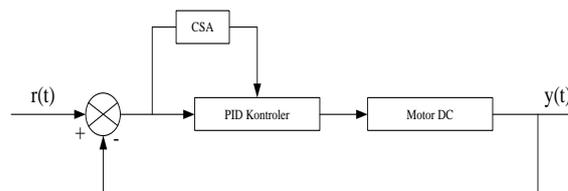
$$u(t) = k_p [e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de_t}{dt}] \quad (3)$$

Di mana, u(t) adalah nilai kontrol yang dihitung oleh kontroler PID, Kp adalah koefisien proporsional, Ti adalah integral waktu konstan dan

Td adalah diferensian time konstan. Fungsi dari ketiga elemen tersebut adalah :

- Proporsional: menggambarkan sinyal deviasi e (t) dari sistem kontrol proporsional. ketika sinyal e(t) ada, kontroler PID menghasilkan efek kontrol segera untuk mengurangi penyimpangan.
- Integral : digunakan untuk menghilangkan kesalahan statis dan meningkatkan stabilitas sistem.
- Diferensial : mencerminkan perubahan penyimpangan sinyal, memperkenalkan sinyal koreksi sebelum penyimpangan nilai sinyal menjadi lebih besar dan mempercepat respon sistem untuk mengurangi pengaturan waktu.

Oleh karena itu, merancang kontroler PID terutama berarti menentukan tiga parameter, serta bagaimana mengkonfigurasi ke tiga parameter PID (Kp, Ki, Kd). Dalam penelitian ini, *Particle Swarm Optimization* diusulkan untuk mencari parameter optimal PID. Blok diagram sistem kontrol ditunjukkan pada gambar 1.



Gambar 2. Sistem Kontroler PID-PSO

III. PARTICLE SWARM OPTIMIZATION

Prosedur PSO dapat dijabarkan dengan langkah-langkah sebagai berikut :

1. Asumsikan bahwa ukuran kelompok atau kawanan (jumlah partikel) adalah N. untuk mengurangi jumlah evaluasi fungsi yang diperlukan ntuk menemukan solusi, sebaiknya ukuran N tidak terlalu besar, tetapi juga tidak terlalu kecil, agar ada banyak kemungkinan posisi menuju solusi terbaik atau optimal. Jika terlalu kecil, sedikit kemungkinan menemukan posisi partikel yang baik. Terlalu besar juga akan membuat perhitungan jadi panjang. Biasanya digunakan ukuran kawanan adalah 20 sampai 30 partikel.
2. Bangkitkan populasi awal X dengan rentang X(B) dan X(A) secara random sehingga didapat X1, X2, ... XN. Setelah itu, untuk mudahnya, partikel j dan kecepatannya pada iterasi I dinotasikan sebagai X1(0), X2(0), ..., XN(0). Vector Xj(0), (j=1,2,...,N) disebut partikel atau vector koordinat dari partikel (seperti kromosom algoritma genetika). Evaluasi dari fungsi tujuan untuk setiap partikel
3. Hitung kecepatan dari semua partikel. Semua partikel bergerak menuju titik optimal dengan suatu kecepatan. Awalnya semua kecepatan dari

partikel diasumsikan sama dengan nol. Set iterasi $i = 1$

4. Pada iterasi ke-I, temukan 2 parameter penting untuk setiap partikel j yaitu:

- (a) Nilai terbaik sejauh ini dari $X_j(i)$ (koordinat partikel j pada iterasi i) dan nyatakan sebagai $Pbest_j$, dengan nilai fungsi obyektif paling rendah (kasus minimasi), $f[X_j(i)]$, yang ditemui sebuah partikel j pada semua iterasi sebelumnya. Nilai terbaik untuk semua partikel $X_j(i)$ yang ditemukan sampai iterasi ke-I, $Gbest$, dengan nilai fungsi tujuan paling kecil/minimum di antara semua partikel untuk semua iterasi sebelumnya, $f[X_j(i)]$.

- (b) Hitung kecepatan partikel j pada iterasi ke I dengan rumus sebagai berikut :

Di mana $c1$ dan $c2$ masing-masing adalah learning rates untuk kemampuan individu (cognitive) dan pengaruh social (group), dan $r1$ dan $r2$ bilangan random yang berdistribusi uniformal dalam interval 0 dan 1. Jadi parameter $c1$ dan $c2$ menunjukkan bobot dari memory (position) sebuah partikel terhadap memory (posisi) dari kelompok (swarm). Nilai dari $c1$ dan $c2$ biasanya adalah 2 sehingga perkalian $c1r1$ dan $c2r2$ memastikan bahwa partikel-partikel akan mendekati target sekitar setengah selisihnya.

- (c) Hitung posisi atau koordinat partikel j pada iterasi ke-I dengan cara : Evaluasi nilai fungsi tujuan untuk setiap partikel dan nyatakan sebagai :

5. Cek apakah solusi yang sekarang sudah konvergen. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka langkah 4 diulang dengan memperbarui iterasi $i = i + 1$, dengan cara menghitung nilai baru dari $Pbest_j$ dan $Gbest$. Proses iterasi ini dilanjutkan sampai semua partikel menuju ke satu titik solusi yang sama. Biasanya akan ditentukan dengan kriteria penghentian (stopping criteria), misalnya jumlah selisih solusi sekarang dengan solusi sebelumnya sudah sangat kecil.

3.1. Implementasi Particle Swarm Optimization Untuk Tuning PI

Berikut ini adalah step-step dalam Particle Swarm Optimization (PSO) untuk tuning PI:

1. Menentukan parameter awal dari PSO seperti $c = 0.2$ dan $w = 0.05$
2. Menentukan nilai K_p dan K_i dengan menggunakan metode Ziegler Nichols
3. Menentukan banyaknya swarm ($n=5$)
4. Menentukan posisi awal dari tiap swarm (sumbu $x=K_p$, sumbu $y=K_i$)

5. Mencoba nilai posisi (K_p dan K_i) tiap-tiap swarm yang telah dipilih dan lihat responnya

6. Ambil hasil nilai Overshoot, Rise Time, dan Settling Time dari masing-masing swarm yang telah dipilih. Tentukan objective function dari tiap-tiap swarm menggunakan rumus

7. Nilai yang paling kecil merupakan swarm yang paling optimal. Di mana $Pbest$ adalah nilai paling optimal di swarm itu sendiri. Dan $Gbest$ adalah nilai paling optimal dari keseluruhan swarm.

8. Cek apakah solusi yang sekarang sudah konvergen. Jika posisi Overshoot, Rise Time, dan Settling Time semakin kecil, maka hal tersebut hampir konvergen. Jika belum maka diulang dengan memperbarui iterasi $i = i + 1$, dengan cara menghitung nilai baru dari $Pbest$ dan $Gbest$.

3.2. Pemodelan Motor DC

Motor DC yang digunakan adalah Motor DC Permanen Magnet, dengan Rating Kecepatan 1400 rpm. Berikut transfer function yang digunakan [1].

$$G(s) = \frac{K_m}{T_m s + 1}$$

Dimana K_m adalah gain mekanik dan T_m adalah mekanik

$$G(s) = \frac{2.75}{0.18s + 1}$$

3.3. Desain Kontrol PID

Dengan objective function yang digunakan :

$$fit_k = \frac{1}{J(ITAE_k)}$$

Dimana $ITAE_k$ adalah nilai Integral Time Absolute Error (ITAE), k_{th} merupakan gain PID (K_p, K_i dan K_d). $j(ITAE_k)$ dirumuskan sebagai berikut :

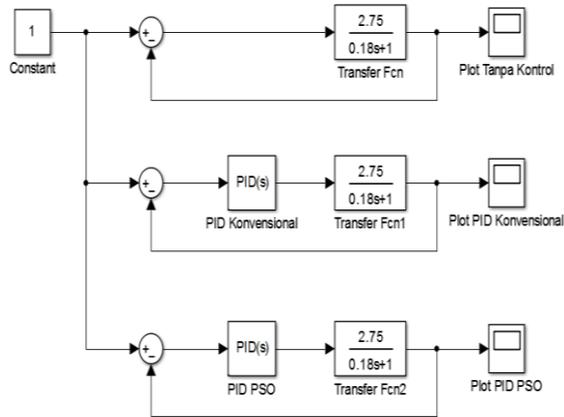
$$J(ITAE_k) = \int_0^{\infty} t |e(t)| dt$$

Dengan menggunakan fungsi tujuan di atas, maka algoritma PSO akan mencari parameter PID berdasarkan minimum error yang paling terkecil, sehingga PSO akan bekerja sampai menemukan solusi optimal parameter PID selama iterasi yang ditentukan.

IV. Hasil Simulasi

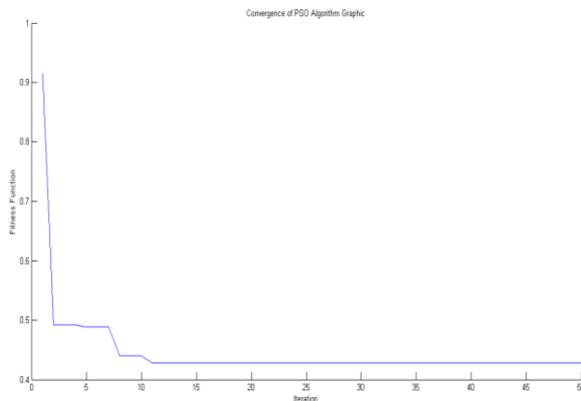
Berikut gambar pemodelan DC menggunakan software Simulink pada Matlab, di mana masing-

masing di gambarkan dengan transfer function, untuk system tanpa control, dengan PID trial error dan PID dengan PSO.



Gambar 3. Pemodelan Motor DC

Setelah menggambarkan pemodelan motor DC di atas, maka selanjutnya membuat algoritma PSO di matlab. Optimasi PSO akan bekerja berdasarkan objective function yang digunakan. Berikut hasil konvergensi PSO dalam melakukan optimasi mencari nilai PID.



Gambar 4. Grafik Konvergensi PSO

Dari grafik di atas, terlihat bahwa PSO sangat cepat dalam melakukan optimasi, di mana pada iterasi ke 12 sudah menemukan titik optimal dari nilai PID. Adapun hasil penalaan PID ditampilkan pada table berikut ini.

Tabel 1. Hasil Optimasi PID Menggunakan PSO

Iteration: 50, Fitness: 0.427260

ans =

0.4273

fitness_terbaik_global =

0.9144

kp_pso =

57.2409

ki_pso =

23.7090

kd_pso =
9.3575

Tabel di atas menunjukkan hasil optimasi parameter PID menggunakan algoritma PSO, di mana untuk $k_p = 57.2409$, $k_i = 23.7090$, dan $k_d = 9.3575$, dengan nilai fitness sebesar 0.9144.

```

Command Window
parameters dialog
> In pso at_93
Iteration: 50, Fitness: 0.427260

ans =

    0.4273

-----
Coba Simulasi Tugas UAS Pengemudian Motor Listrik
---Oleh:Muhammad Ruswandi Djalal NRP:2213201008---
-----

fitness_terbaik_global =

    0.9144

kp_pso =

    57.2409

ki_pso =

    23.7090

kd_pso =

    9.3575

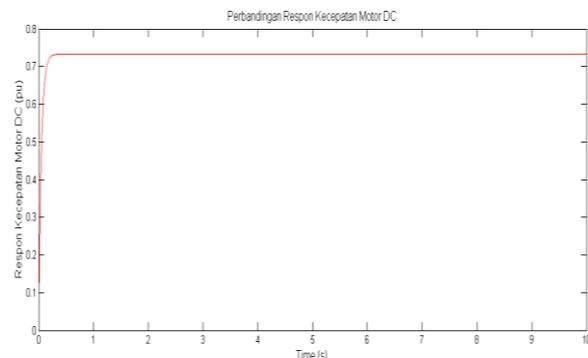
f3 >>
    
```

Gambar 5. Hasil Optimasi PID menggunakan PSO di Matlab

Adapun respon pada motor dc untuk masing-masing model pengontrolan dapat dilihat pada pembahasan berikut.

Respon kecepatan motor DC tanpa control

Pembahasan pertama akan ditampilkan bagaimana respon yang terjadi pada motor dc tanpa dilengkapi controller tambahan.



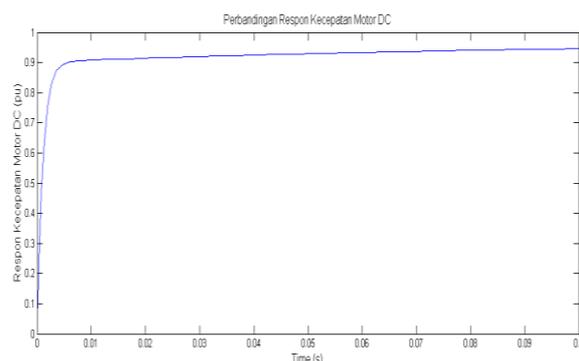
Gambar 6. Respon Kecepatan Motor DC Tanpa Kontrol

Dari grafik di atas data terlihat respon motor dc yang sangat buruk, karena tidak mencapai set point yang

sudah ditentukan dengan settling time yang sangat lama menuju ke kondisi steady state, oleh karenanya model pengoperasian motor seperti ini sangat tidak disarankan.

Respon kecepatan motor DC dengan PID konvensional

Berikut respon motor dc ketika diberikan controller tambahan berupa PID, yang di mana parameteranya di set dengan metode trial-error.

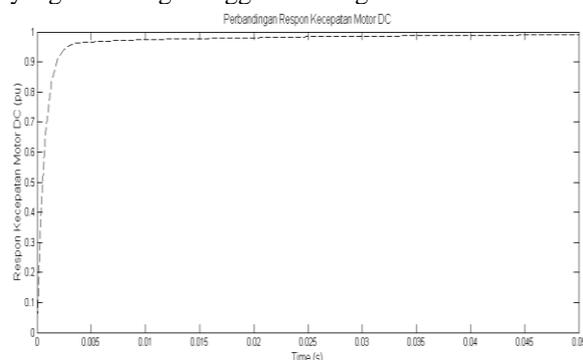


Gambar 7. Respon Kecepatan Motor DC Dengan PID Trial-Error

Dari grafik di atas terlihat respon motor yang sudah semakin cepat responnya menuju ke kondisi steady atau ke kondisi yang sama dengan set point. Hal ini karena motor sudah ada mode pengontrolannya yang dilakukan oleh PID, sehingga output dari motor ada umpan balik ke input untuk selanjutnya dikontrol sesuai dengan set point yang sudah ditentukan. Namun kinerja PID dalam mode ini masih bisa ditingkatkan untuk mendapatkan kinerja respon motor yang optimal.

Respon kecepatan motor DC dengan PID PSO

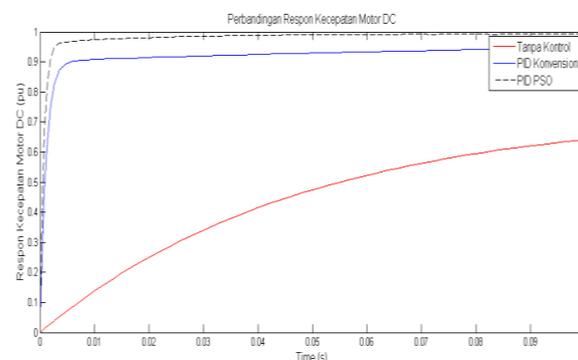
Berikut respon motor dc dengan pengendali PID yang dituning menggunakan algoritma PSO.



Gambar 8. Respon Kecepatan Motor DC PID PSO

Dapat terlihat bahwa kinerja PID sudah semakin baik dalam melakukan pengontrolan terhadap motor dc dengan parameter yang optimal. Pada mode ini,

respon motor sudah semakin baik dan sangat cepat system menuju ke kondisi steady tanpa memerlukan waktu settling time yang lama. Untuk hasil perbandingan masing-masing mode operasi motor, dapat di lihat pada gambar berikut.



Gambar 9. Perbandingan Respon Kecepatan Motor DC Tanpa Kontrol, PID Trial dan PID PSO

Desain controller pada penelitian ini diusulkan meode cerdas berbasis kecerdasan particle swarm optimization dalam mencari nilai optimal PID pada motor DC.

V. Kesimpulan

Hasil optimal optimasi parameter PID menggunakan algoritma PSO adalah $k_p = 57.2409$, $k_i = 23.7090$, dan $k_d = 9.3575$, dengan nilai fitness sebesar 0.9144. Metode kecerdasan buatan berbasis Particle Swarm Optimization dapat dengan baik mencari nilai optimal PID berdasarkan objective function yang digunakan yaitu meminimalkan ITAE (Integral Time Absolute Error).

Daftar Pustaka

- [1] M. R. Djalal, D. Ajiatmo, A. Imran, and I. Robandi, "Desain Optimal Kontroler PID Motor DC Menggunakan Cuckoo Search Algorithm," *Prosiding SENTIA*, 2015.
- [2] W. Liao, Y. Hu, and H. Wang, "Optimization of PID control for DC motor based on artificial bee colony algorithm," in *Advanced Mechatronic Systems (ICAMEchS), 2014 International Conference on*, 2014, pp. 23-27.
- [3] D. Chen, K. Fang, and Q. Chen, "Application of genetic algorithm in PID parameters optimization," *Microcomputer Information*, vol. 23, pp. 35-36, 2007.
- [4] H. He and F. Qian, "The PID parameter tuning based on immune evolutionary algorithm," *Microcomputer Information*, vol. 27, pp. 1174-1176, 2007.
- [5] N. Katal and S. K. Singh, "Optimal Tuning of PID Controller for DC Motor using Bio-Inspired Algorithms," *International Journal of Computer Applications*, vol. 56, 2012.
- [6] B. Bhushan and M. Singh, "Adaptive control of DC motor using bacterial foraging algorithm,"

Applied Soft Computing, vol. 11, pp. 4913-4920, 2011.

- [7] A. Ahuja and S. K. Aggarwal, "Design of fractional order PID controller for DC motor using evolutionary optimization techniques," *WSEAS transactions on system and control*, vol. 9, pp. 171-182, 2014.