

Optimasi kontrol PID untuk kontrol kecepatan Motor_Sniter.pdf

anonymous marking enabled

Submission date: 13-Mar-2024 11:05PM (UTC-0700)

Submission ID: 2320088266

File name: Optimasi_kontrol_PID_untuk_kontrol_kecepatan_Motor_Sniter.pdf (1.15M)

Word count: 2765

Character count: 16325

OPTIMASI KONTROL PID UNTUK KENDALI KECEPATAN MOTOR DC MENGGUNAKAN METODE METAHEURISTIK

Eddy Lybrech Talakua¹, Yoga Alif Kurnia Utama², Indra Andriyanto^{3*}
^{1,2,3}Universitas Widya Kartika

Abstrak

Hampir seluruh industri di dunia ini menerapkan perkembangan teknologi kontrol Motor *Direct Current* (DC) adalah aktuator yang sering digunakan dalam industri dan teknologi kontrol. Motor DC mempunyai karakter respon yang cepat, namun masih memiliki *error steady state*. Oleh karena itu di butuhkan suatu kendali yang tepat sesuai karakteristik Motor DC. Kontroler *Proportional Integral Difrensial* (PID) adalah kontrol aksi yang memiliki respon cepat, sehingga sesuai untuk mengontrol Kecepatan motor DC. Dalam penelitian ini kami menggabungkan Metode *metaheuristic* Algoritma Genetika dalam pencarian parameter PID yang lebih optimal dari pada metode konvensional karena *overshoot* rendah dan *steady state* yang sangat cepat dari pada metode PID konvensional yang digunakan selama ini. Dari parameter PID yang telah di dapatkan dari metode metaheuristik di implementasikan pada motor DC untuk di bandingkan antara hasil respon sistem antara metode konvensional yaitu *Trial and error*, Ziegler Nichols dan metode optimasi metaheuristik melalui simulasi Simulink pada MATLAB.

Kata kunci: Motor DC, PID, dan Metaheuristik

Abstract

Almost all industries in the world apply the development of control technology. Direct current (DC) motors are actuators that are often used in industry and control technology. The DC motor has a fast response character, but still has a steady state error. Therefore we need a precise control according to the characteristics of a DC motor. Differential Integral Proportional Controller (PID) is an action control that has a fast response, making it suitable for controlling the speed of a DC motor. In this study, we combine the Genetic Algorithm metaheuristic method in the search for PID parameters that are more optimal than the conventional method because of the low overshoot and very fast steady state than the conventional PID methods used so far. From the PID parameters that have been obtained from the metaheuristic method implemented on a DC motor to be compared between the results of the system response between conventional methods, namely Trial and error, Ziegler Nichols and metaheuristic optimization methods through Simulink simulation on MATLAB.

Keywords: Motor DC, PID, dan Metaheuristic

1. PENDAHULUAN

Dewasa ini dalam revolusi industri 4.0 Sistem kendali pada suatu Industri menjadi peranan utama dalam meningkatkan efektifitas dan efisiensi dalam suatu proses produksi. Sistem kendali untuk suatu *Plant* di dalam sistem tersebut, Karena di dalam sistem terdiri dari *Plant* yang membentuk sebuah alur produksi pada sistem produksi, Hal ini penulis coba mencontohkan sebagian *Plant* yang ada di

sistem produksi Amoniak di Petrokimia Gresik yaitu Reaktor kimia, *Heater*, *Boiler* dan sebagainya. *Plant* tersebut selalu di pantau di *Control room* dan berbagai parameter pada *Plant* di baca oleh alat *Instrument* dan dikirim ke *Control Room* untuk di *Monitoring* dan di kontrol. Dewasa ini perkembangan sistem kontrol sangat banyak mengalami pembaharuan yang sangat positif, mulai dari kontrol konvensional sampai dengan kontrol cerdas.

*)Penulis Korespondensi

Kontrol PID (*Proporsional Integral Derivatif*) merupakan pengembangan dari kontrol konvensional yang bertujuan untuk memberikan performa kontrol terbaik pada sistem kendali. Sistem Kendali (P) mempunyai keunggulan risetime yang cepat. Kendali Integral (I) mempunyai keunggulan untuk meminimalisir error. Kendali Derivatif (D) mempunyai keunggulan *overshoot*. Berdasarkan survey, 97% industri yang bergerak dalam bidang proses (seperti industri kimia, pulp, minyak, makanan, dan gas) menggunakan PID sebagai komponen utama dalam pengontrolannya (sumber: Honywell, 2000).

2. METODE PENELITIAN

2.1 Sistem Kontrol Motor DC

A. Metode Kontrol Kecepatan Motor DC

Sistem kontrol kecepatan diterapkan untuk motor DC. Kontrol kecepatan motor DC adalah dengan dua metode utama yaitu.

1. Metode kontrol *armature*

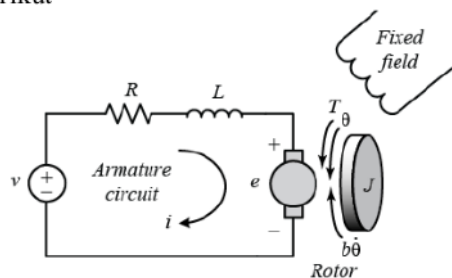
Dalam metode ini, tegangan jangkar bervariasi hingga mengontrol kecepatan motor.

2. Metode kontrol lapangan

Dalam metode ini, bidang saat ini atau saat ini stator bervariasi untuk mengendalikan kecepatan motor.

B. Pengaturan Fisik Motor DC

Sirkuit ekuivalen elektrik dari armature dan *body free* diagram rotor ditunjukkan pada gambar berikut



Gambar 2.1 Rangkaian *electric equivalent* listrik rotor

*)Penulis Korespondensi

C-49-2

Masukan dari sistem adalah sumber tegangan (V) yang diterapkan angker motor, sedangkan output adalah rotasi kecepatan poros. Rotor dan poros diasumsikan kaku. Model gesekan viskos, yaitu torsi gesek sebanding dengan kecepatan sudut poros

Tabel 2.1 Parameter fisik untuk motor dc.

Momen inersia rotor (J)	0.01 kg.m ²
Konstanta gesekan kental motor (b)	0.1 N.m.s
Konstanta gaya gerak listrik (K_e)	0.01 V/rad/sec
Konstanta torsi motor (K_t)	0.01 Nm / Amp
Hambatan listrik (r)	1 Ohm
Induktansi listrik (L)	0.5 H

C. Persamaan Fungsi Transfer dari Sistem

1 Kontrol Umpan Balik

Torsi yang dihasilkan oleh motor DC sebanding dengan arus jangkar dan kekuatan medan magnet. Di contoh ini kita akan menganggap bahwa medan magnet adalah konstan. Karena itu, torsi motor hanya proporsional arus jangkar i oleh faktor konstan K_t seperti yang ditunjukkan pada persamaan di bawah ini. Ini disebut sebagai *armature controlled motor*.

$$T = K_t i \quad (1)$$

Dimana :

T = torsi

K_t = faktor instan

I = arus jangkar

Back e.m.f, e , sebanding dengan kecepatan sudut dari poros oleh faktor konstan K_e .

$$e = K_e \theta \quad (2)$$

Dimana :

- e = back e.m.f
- K_e = Faktor konstan
- θ = Kecepatan sudut

Dalam unit SI, torsi motor dan konstanta belakang adalah sama, yaitu, $K_t = K_e$; oleh karena itu, kami akan menggunakan K untuk mewakili baik konstanta torsi motor dan konstanta belakang. Kita bisa mendapatkan yang berikut ini mengatur persamaan berdasarkan hukum ke-2 Newton dan Hukum tegangan Kirchhoff.

$$J\theta + b\theta = K_t I \quad (3)$$

$$L \frac{di}{dt} + R_i = V - K\theta \quad (4)$$

Dimana :

- J = momen inersia rotor
- B = konstanta gesekan motor *viscous*
- K_i = faktor konstan
- L = induktansi listrik
- R = resistensi listrik
- V = sumber tegangan

Menerapkan transformasi Laplace, pemodelan di atas persamaan dapat diungkapkan dalam hal variabel Laplace.

$$s(Js + b)\theta(s) = KI(s) \quad (5)$$

$$(Ls + R)I(s) = V(s) - Ks\theta(s) \quad (6)$$

Function transfer *Open-loop* berikut dengan menghilangkan $I(s)$ antara dua persamaan di atas, di mana kecepatan dianggap sebagai output dan tegangan angker dianggap sebagai masukan

$$P(s) = \frac{\theta(s)}{V(s)} = \frac{K}{(Js+b)(Ls+R)+K^2} \text{(rad/sec/V)} \quad (7)$$

*)Penulis Korespondensi

C-49-3

$$P(s) = \frac{0.01}{(0.01s+0.1)(0.5s+1)+0.01^2} \quad (8)$$

$$P(s) = \frac{0.01}{0.005s^2+0.006s+0.1001} \quad (9)$$

Sistem kontrol umpan balik sering disebut sebagai *closedloop* sistem kontrol. Dalam sistem kontrol *loop* tertutup, sinyal aktuasi, yang merupakan perbedaan antara sinyal input dan sinyal umpan balik dimasukkan ke pengontrol sehingga dapat mengurangi kesalahan dan membawa output dari sistem ke nilai yang diinginkan. Istilah kontrol *loop* tertutup selalu menyiratkan penggunaan tindakan kontrol umpan balik untuk mengurangi kesalahan sistem.

Sistem *loop* tertutup:

$$P(s) = \frac{0.01}{0.005s^2+0.006s+0.1001} \quad (10)$$

2.2 Metode Optimasi Metaheuristik

Optimasi Metaheuristik berkaitan dengan masalah optimisasi menggunakan algoritma metaheuristik. Optimalisasi pada dasarnya ada di mana-mana, dari desain teknik hingga ekonomi dan dari perencanaan liburan hingga perutean internet. Karena uang, sumber daya, dan waktu selalu terbatas, utilitas optimal dari sumber daya yang tersedia ini sangat penting. Kebanyakan optimasi dunia nyata sangat nonlinear dan multimodal, di bawah berbagai kendala yang kompleks. Tujuan yang berbeda seringkali saling bertentangan. Bahkan untuk satu tujuan, kadang-kadang, solusi optimal mungkin tidak ada sama sekali. Secara umum, menemukan solusi optimal atau bahkan solusi sub-optimal bukanlah tugas yang mudah.

Dalam pengertian yang paling sederhana, optimasi dapat dianggap sebagai masalah minimisasi atau maksimalisasi. Misalnya, fungsi $f(x) = x^2$ memiliki minimum $f_{\min} = 0$ pada $x=0$ pada di seluruh domain $-\infty < x < \infty$. Secara umum, jika suatu fungsi cukup sederhana, kita dapat menggunakan turunan pertama $f'(x) = 0$ untuk menentukan lokasi

potensial, dan menggunakan turunan kedua $f''(x)$ untuk memverifikasi apakah solusi tersebut maksimum atau minimum. Namun, untuk fungsi nonlinier, multimodal, multivarian, ini bukan tugas yang mudah. Selain itu, beberapa fungsi mungkin memiliki diskontinuitas, dan dengan demikian informasinya tidak mudah diperoleh. Ini dapat menimbulkan berbagai tantangan bagi banyak metode tradisional seperti mendaki bukit.

Optimasi Secara umum, masalah optimisasi dapat ditulis sebagai

$$f_1(x), \dots, f_i(x), \dots, f_l(x), \quad x = (x_1, \dots, x_d) \quad (11)$$

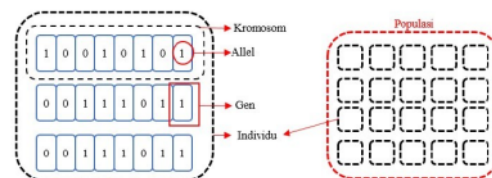
Untuk menyelesaikan masalah optimisasi, pencarian yang efisien atau algoritma optimisasi diperlukan. Ada banyak algoritma pengoptimalan yang dapat diklasifikasikan dalam banyak cara, tergantung pada fokus dan karakteristiknya.

2.3 Algoritma Genetika

Algoritma Genetika (GA) adalah metode pencarian global stokastik yang meniru proses evolusi alami. Ini adalah salah satu metode yang digunakan untuk optimasi. John Holland secara resmi memperkenalkan metode ini di Amerika Serikat pada tahun 1970 di Universitas Michigan. Peningkatan kinerja berkelanjutan sistem komputasi telah membuatnya menarik untuk beberapa jenis optimasi. Algoritma genetik dimulai tanpa pengetahuan tentang solusi yang benar dan sepenuhnya bergantung pada tanggapan dari operator lingkungan dan evolusi tersebut sebagai reproduksi, *crossover* dan mutasi untuk sampai pada solusi terbaik. Dengan memulai pada beberapa titik independen dan pencarian secara paralel, algoritma menghindari local minimal dan konvergen ke solusi sub optimal. Dengan cara ini, GA telah terbukti mampu menemukan kinerja tinggi area dalam domain kompleks tanpa mengalami kesulitan yang terkait dengannya dimensi tinggi, seperti yang mungkin terjadi dengan teknik atau metode

yang layak gradient yang mengandalkan informasi derivatif.

Ada enam komponen primer dalam algoritma genetika, yaitu teknik penyandian, Prosedur Inisialisasi, fungsi Evaluasi, Seleksi, Operator Genetika dan Penentuan Parameter

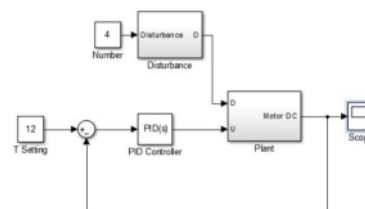


Gambar 2.2 Istilah Algoritma genetika

- A. Genotype (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.
- B. Allel, nilai dari gen.
- C. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
- D. Individu, menyatakan suatu nilai yang menyatakan salah satu solusi yang mungkin untuk suatu permasalahan optimasi.
- E. Populasi, Kumpulan individu yang akan diproses dalam satu siklus algoritma genetika.
- F. Generasi, menyatakan jumlah siklus algoritma genetika berjalan.

2.4 Pemodelan Matlab

Dari desain sistem yang digambarkan dalam diagram alir dan model matematik yg dibahas bab sebelumnya kami bisa memulai melangkah ke realisasi dalam bentuk simulasi



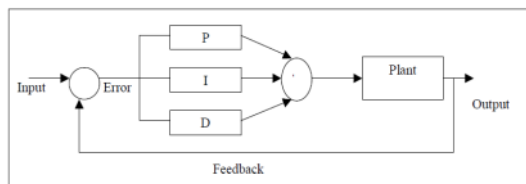
*)Penulis Korespondensi

pada fitur Simulink di program MATLAB proses ini di namakan pemodelan. Untuk lebih jelas nya lihat gambar 2.3 dimana terdapat pemodelan seluruh sistem.

Gambar 2.2 Pemodelan Simulasi Pada Simulink

2.5 PID

Kontroller PID terdiri dari Tindakan Proportional, Aksi Integral, dan Derivatif Tindakan. Biasanya mengacu pada parameter tuning PID Ziegler-Nichols. Sejauh ini algoritma kontrol yang paling umum dan banyak digunakan. Algoritma pengontrol PID sebagian besar digunakan dalam loop umpan balik. Kontroler PID bisa diimplementasikan dalam berbagai bentuk. Ini dapat diimplementasikan sebagai pengontrol yang berdiri sendiri atau sebagai bagian dari paket *Direct Digital Control* (DDC) atau bahkan *Distributed Control System* (DCS). Yang terakhir adalah sistem kontrol proses terdistribusi hierarkis yang banyak digunakan dalam pabrik pengolahan seperti pharmaceutical atau penyulingan minyak industri. Sangat menarik untuk dicatat bahwa lebih dari setengah pengontrol industri digunakan hari ini menggunakan skema kontrol PID atau PID yang dimodifikasi. Di bawah ini adalah diagram sederhana menggambarkan skema pengontrol PID. Pengaturan seperti ini dikenal sebagai bentuk paralel.



Gambar 2.3 Skema Kontroler PID - *Non-Interacting Form*

Dalam Kontrol proporsional,

$$P_{term} = K_p \times \text{Error} \quad (12)$$

Ia menggunakan proporsi kesalahan sistem untuk mengontrol sistem. Dalam aksi ini offset diperkenalkan dalam sistem.

Dalam kontrol Integral,

$$I_{term} = K_i \times \text{Error} \int \text{Error} dt \quad (13)$$

(Ini sebanding dengan jumlah kesalahan dalam sistem. Dalam aksi ini, aksi-I akan memperkenalkan kelambatan dalam sistem. Ini akan menghilangkan offset yang tadinya diperkenalkan sebelumnya oleh aksi-P.

Dalam kontrol Derivatif,

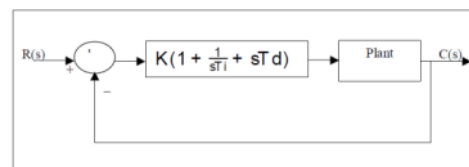
$$D_{term} = K_D \times \frac{d(\text{error})}{dt} \quad (14)$$

Ini sebanding dengan tingkat perubahan kesalahan. Dalam aksi ini, aksi-D akan memperkenalkan petunjuk dalam sistem. Ini akan menghilangkan kelambatan pada sistem yang tadinya diperkenalkan oleh I-aksi sebelumnya.

Tiga pengendali saat digabungkan bersama dapat diwakili oleh mengikuti fungsi transfer.

$$G_C(s) = K \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (15)$$

Ini dapat diilustrasikan di bawah ini dalam diagram blok berikut



Gambar 2.4 Blok diagram dari Continuous PID Kontroler

Apa yang dilakukan oleh kontroler PID pada dasarnya adalah untuk bertindak pada variabel yang akan dimanipulasi melalui kombinasi

*)Penulis Korespondensi

yang tepat dari tiga tindakan kontrol yaitu Tindakan kontrol P, tindakan kontrol I dan tindakan kontrol D. Tindakan P adalah tindakan kontrol yang proporsional dengan kesalahan penggerak sinyal, yang merupakan perbedaan antara input dan sinyal umpan balik. I aksi adalah aksi kontrol yang proporsional dengan integral dari *actuating* sinyal kesalahan. Akhirnya tindakan D adalah tindakan kontrol yang sebanding dengan turunan dari sinyal kesalahan aktuasi. Dengan integrasi ketiga tindakan, PID berkelanjutan dapat direalisasikan. Jenis pengontrol ini banyak digunakan dalam industri di seluruh dunia. Sebenarnya banyak penelitian, studi dan aplikasi telah ditemukan dalam beberapa tahun terakhir.

3. HASIL DAN PEMBAHASAN

Penelitian dimulai dengan melakukan pemodelan di Simulink pada MATLAB dan menganalisa respon sistem tanpa controller PID.

Tabel 1. Respon Motor DC Tanpa PID

No	Respon Sistem	Rise Time (S)	Over Shoot (pu)	Settling Time (s)	Steady State (pu)
1	Motor DC	1.02	0.0907	1.85	0.0908

Kemudian dilakukan *tuning* dengan metode konvensional yaitu metode trial & error dan Ziegler Nichols sebagai pembanding metode optimasi yang akan di terapkan.

Tabel 2. Respon motor DC dengan Hasil Tuning PID dengan metode Konvensional

No	Metode Tuning	Rise Time (S)	Over Shoot (pu)	Settling Time (s)	Steady State (pu)
1	Trial And Error	0.014176	16.040	0.118	0.233
2	Ziegler Nichols	0.068865	15.256	1.138	1.636

Setelah menerapkan tuning PID dengan metode konvensional kami mulai mengimplementasikan

*)Penulis Korespondensi

Algoritma Genetika terhadap kontroler PID, Dalam percobaan penelitian ini kami menguji 9 kali penerapan Algoritma genetika terhadap Kontroler PID adapun itu :

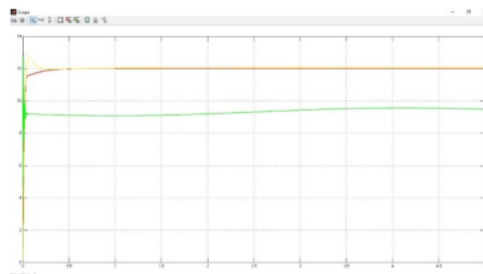
1. Populasi random 20 dibatasi 20 literasi
2. Populasi random 20 dibatasi 40 literasi
3. Populasi random 20 dibatasi 60 literasi
4. Populasi random 30 dibatasi 20 literasi
5. Populasi random 30 dibatasi 40 literasi
6. Populasi random 30 dibatasi 60 literasi
7. Populasi random 50 dibatasi 20 literasi

Dalam percobaan tersebut kami menerapkan nilai parameter K_p , K_i , dan K_d *output* dari pengoptimasian terhadap Simulink dan mendapat hasil dibawah ini :

Keterangan :

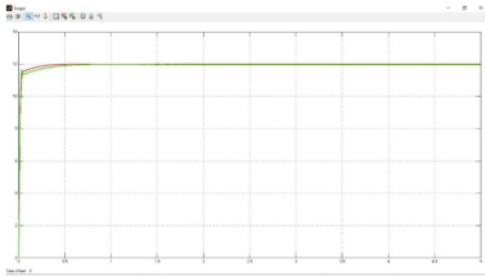
- *Output* warna merah literasi 60
- *Output* warna kuning literasi 40
- *Output* warna Hijau literasi 20

1. Populasi 20



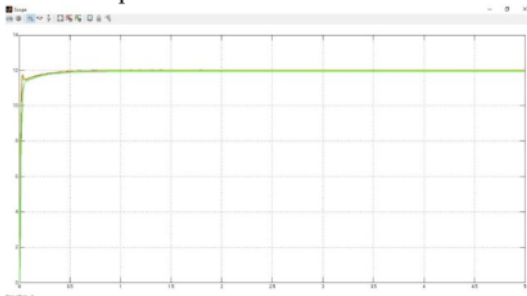
Gambar 2.5 Respon motor DC dengan Hasil Tuning PID dengan percobaan 20 populasi acak

2. Populasi 30



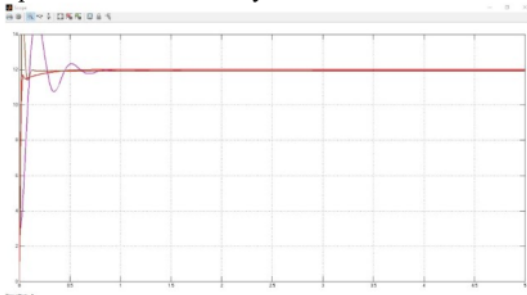
Gambar 2.6 Respon motor DC dengan Hasil Tuning PID dengan perobaan 30 populasi acak

3. Populasi 50



Gambar 2.7 Respon motor DC dengan Hasil Tuning PID dengan perobaan 50 populasi acak

Setelah percobaan optimasi menggunakan algoritma genetika langkah selanjutnya adalah membandingkan dengan hasil respon sistem dengan berbagai metode yang sudah di aplikasikan sebelumnya.



Gambar 2.8 Perbandingan Respon motor DC dengan Hasil Tuning PID Trial & Error, Ziegler Nichols dan Optimasi Algoritma Genetika

Ziegler Nichols dan Optimasi Algoritma Genetika

Keterangan :

- Output warna merah Optimasi Algoritma Genetika
- Output warna Ungu Ziegler Nichols
- Output warna Coklat Trial and Error

Dalam perbandingan tersebut didapatkan :

Tabel 3. Respon motor DC dengan Hasil Tuning PID dengan metode Konvensional

No	Metode Tuning	Rise Time (S)	Over Shoot (pu)	Settling Time (s)	Steady State (pu)
1	Trial And Error	0.014176	16.040	0.118	0.233
2	Ziegler Nichols	0.068865	15.256	1.138	1.636
3	Optimasi Algoritma Genetika	0.442057	12.013	1.30	1.30

4. KESIMPULAN

Dari hasil pengujian sistem kontrol dengan simulasi pada kontroler PID dan tuning algoritma genetika pada plant Motor DC dengan populasi random 50 di batasi lliterasi ke 60 terbukti belum cukup optimal karena nilai *settling time* yang sebesar 1.30 detik terhitung lebih lama tidak secepat metode Ziegler Nichols dengan 1.138 detik dan *Trial error* 0.118 namun metode optimasi algoritma unggul karena *overshoot* lebih kecil yaitu hanya 12.013 di bandingkan metode *trial and error* sebesar 16.040 dan Metode Ziegler Nichols Sebesar 15.256. Mungkin hasil akan lebih optimal jika optimasi tidak di batasi dengan iterasi namun membutuhkan waktu lebih panjang lagi untuk optimasi.

*)Penulis Korespondensi

Daftar Pustaka

- K.Ogata, Modern Control Engineering, Fifth ed. New Jersey: Prentice Hall, 2010.
- Grassi, E., Tsakalis, K., Dash, S., Gaikwad, S. V., Macarthur, W., & Stein, G. (2001). Integrated system identification and PID controller tuning by frequency loop-shaping. *IEEE Transactions on Control Systems Technology*.
- Khin Ei Ei Khine., Win Mote Mote Htwe., Yin Yin Mon., (2019). Simulation DC Motor Speed Control System by using PID Controller. *IJTSRD*,: 2456-6470.
- J. B. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 2B, pp. 220-222, 1993.
- Md Akram Ahmad, Electronics and Communication Engineering Department, Maryland Institute of Technology and Management, Jamshedpur, India, Kamal Kishor, Electronics and Communication Engineering Department, Ramgovind Institute of Technology, Koderma, India, Pankaj Rai, Electrical Engineering Department, BIT Sindri, Dhanbad, India,
- A. Visioli, *Practical PID Control*. London: Springer, 2006.
- Talbi, E.-G. (2009). *Metaheuristik: from design to implementation*. Hoboken, New Jersey: John Wiley & Son, Inc
- Setiawan, Iwan. 2008. *Kontrol PID untuk Proses Industri*, Jakarta: PT Elex Komputindo
- Yang, X,S, *Nature-Inspired Metaheuristic Algorithms*, Frome: Luniver Press. 2008.
- D. Chen, K. Fang, and Q. Chen, "Application of genetic algorithm in PID parameters optimization", *Microcomputer Information*, vol. 23, no.3, pp. 35-36, 2007.
- Djalal.M.R, D. Ajiatmo, A. Imran, I. Robandi, "Desain Optimal Kontroler PID Motor DC Menggunakan Cuckoo Search Algorithm", Seminar Nasional Teknologi Informasi dan Aplikasinya (SENTIA) Politeknik Negeri Malang, 2015.

Optimasi kontrol PID untuk kontrol kecepatan Motor_Sniter.pdf

ORIGINALITY REPORT

24%

SIMILARITY INDEX

22%

INTERNET SOURCES

3%

PUBLICATIONS

9%

STUDENT PAPERS

MATCHED SOURCE



spada.uns.ac.id

Internet Source

5%

5%

★ spada.uns.ac.id

Internet Source

Exclude quotes On

Exclude matches Off

Exclude bibliography On