Penalaan Optimal Kendali Motor DC dengan Algoritma Ant Colony Optimization Menggunakan Simulink

Muhammad Iqbal, Mimin Iryanti

Program Studi Fisika, FPMIPA, Universitas Pendidikan Indonesia
e-mail : iqbal2599@upi.edu

ABSTRAK

Dalam penggunaan PID sebagai sistem kontrol diperlukan adanya konfigurasi nilai dari parameter PID agar dapat mencapai performa yang maksimal. Perkembangan penalaan PID menggunaan algoritma komputer saat ini sangatlah pesat karena performansi yang dihasilkan oleh algoritma komputer sangatlah optimal. *Ant Colony Optimization* merupakan algoritma yang terinspirasi dari perjalanan kawanan semut dalam suatu ekspedisi. Dimana motor DC sebagai sistem dan PID sebagai kontroller dibuatkan pemodelan komputernya menggunakan aplikasi Simulink. Hasil penalaan PID untuk motor DC menggunakan *Ant Colony Optimization* menghasilkan nilai parameter Kp sebesar 4.9804, Ki 0.82332, dan Kd 0.68298. Dan karakteristik performansi respon *rise time* sebesar 0.293 detik, *settling time* 0.607 detik, dan *overshoot* sebesar 0.455 %. Dengan hasil performansi ini didapatkan kinerja respon motor DC yang optimal ditandai dengan responsifnya sinyal respon sistem yang mendekati nilai *set point* dibandingkan dengan sistem menggunakan PID *Tuner* maupun sistem tanpa PID.

Kata Kunci : *Ant Colony Optimization*, Motor DC, PID, Sinyal Respon

PENDAHULUAN

Industri saat ini sudah menggunakan teknologi-teknologi canggih yang menunjang kemampuan produksinya agar lebih cepat, lebih sederhana, dan lebih efisien. Perkembangan industri pun banyak menggunakan motor DC sebagai aktuator untuk melakukan berbagai macam instruksi. Motor DC merupakan sistem yang bersifat non-linear sehingga diperlukan adanya sistem kontrol. Sistem kontrol yang sering digunakan oleh industri adalah PID (*proportional integral derivative)*. PID merupakan sistem kontrol yang digunakan agar performansi suatu respon sistem dapat mencapai *set point* dengan efektif.

Dalam penggunaan PID sebagai sistem kontrol diperlukan adanya konfigurasi nilai dari parameter PID agar dapat mencapai performa yang maksimal. Metode dalam mencari parameter PID sangatlah beragam, dapat menggunakan penalaan manual, penalaan menggunakan logika *fuzzy*, maupun penalaan menggunakan bantuan algoritma.

Perkembangan penalaan PID menggunaan algoritma komputer saat ini sangatlah pesat karena performansi yang dihasilkan oleh algoritma komputer sangatlah bagus dibandingkan dengan metode yang lain. Para peneliti saat ini banyak melakukan publikasi tentang penalaan PID menggunakan algoritma computer. Salah satu algoritma yang saat ini populer yaitu *Ant Colony Optimizaiton* dimana proses algoritmanya terinspirasi dari proses perjalanan kawanan semut. Sehingga ini menjadi menarik bagi penulis untuk melakukan penalaan PID menggunakan Algoritma *Ant Colony Optimization*. Pada penelitian kali ini akan dilakukan uji performansi sistem respon menggunakan algoritma *Ant Colony Optimization* dan akan dilihat nilai performansinya dari hasil respon sistem. Sehingga masyarakat dapat mengetahui kebermanfaatan dari penggunaan algoritma ini.

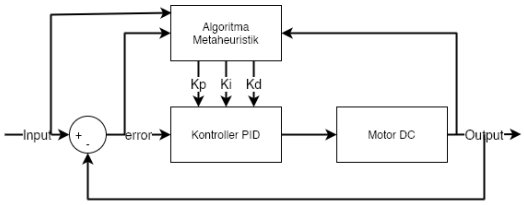
METODE

Pada penelitian kali ini saya akan melakukan penelitian menggunakan metode simulasi dan pemodelan menggunakan aplikasi Simulink matlab.

1. Diagram Blok Penelitian

Gambar 1 merupakan diagram blok dari penelitian yang akan dilakukan, dimana

kontroller PID akan ditala oleh algoritma metaheuristik untuk meningkatkan performansi dari motor DC,



Gambar 2.1 Diagram Blok Model Penelitian

Sehingga dengan model seperti ini sistem dapat menghasilkan nilai keluaran yang optimal.

2. Model Motor DC

Motor DC merupakan komponen elektrik yang dapat menghasilkan gerak rotasi karena memanfaatkan gaya gerak elektromagnetik. Pada motor DC terdapat 2 komponen penting yaitu komponen elektrik, dan komponen mekanik. Komponen elektrik pada motor DC dijelaskan pada persamaan 2.1 yang didapatkan dari penurunan rumus menggunakan hukum kirchoff.

$$V_{\mathrm{a}} = i_a R_a + L_a \frac{di_a}{dt} + E_b \quad \textbf{(2.1)}$$

$$I_A(t) = \frac{1}{La} \int [V_a(t) - R_a i_a(t) - E_b(t)] dt \ \textbf{(2.2)}$$

merupakan induktansi kumparan, ia merupakan arus yang dihasilkan, dan Eb merupakan gaya gerak listrik induksi.

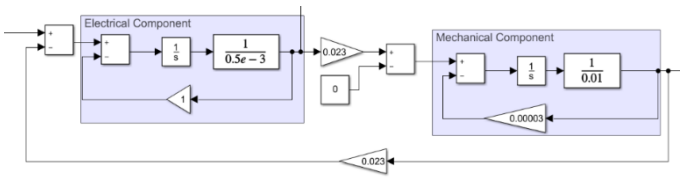
Komponen mekanik motor DC merupakan bagian yang digerakan karena adanya induksi magnetik sehingga medan magnet menggerakan bagian rotor dan menghasilkan kecepatan putaran motor DC. Persamaan 2.3 menjelaskan hubungan komponen mekanik motor DC yang diturunkan menggunakan hukum Newton.

$$T_e = J \frac{d^2\theta}{dt} + B \frac{d\theta}{dt} + T_L \quad \textbf{(2.3)}$$

$$\omega_m(t) = \frac{1}{J} \int [T_e(t) - B\omega_m(t) - T_L(t)] dt \ \textbf{(2.4)}$$

Te merupakan torsi motor (Nm), J merupakan momen inersia ($kg\ m^2$), B merupakan koefisien redaman motor, ω_m merupakan percepatan motor ($^{rad}/_{s^2}$), dan T_L merupakan torsi beban.

Dengan menggabungkan persamaan 2.3 dan 2.4 kita dapatkan model untuk motor DC seperti yang digambarkan pada gambar 2.2.



Gambar 2.2 Model Motor DC Pada Simulink

Tabel 2.1 Paramater Motor DC

Karakteristik	Nilai
L_a	1 Ohm
R_a	0.5 mH
k_T	0.023 Nm/A
T_{L}	0 (Tidak ada beban)
k_e	0.023 V/rad/s
J	0.01 kg.m^2
B	0.00003 kg m^2/s

3. Model PID *Controller*

PID Controller merupakan metode pengontrollan sistem menggunakan 3 elemen yaitu *Proportional*, *Integral*, dan *Derivative*. Masing-masing elemennya berfungsi untuk menghitung nilai error yang dihasilkan oleh sistem lalu dioperasikan dengan operasi matematik yang terdapat pada masing-masing komponen. Lalu dijumlahkan dan luaran PID menghasilkan nilai optimum untuk meningkatkan performansi motor DC. Persamaan 2.5 merupakan persamaan yang digunakan pada PID *controller* dimana e merupakan nilai error, dan K merupakan konstanta/parameter.

$$PID = K_p\, e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{de(t)}{dt} \quad \textbf{2.5)}$$

4. *Ant Colony Optimization*

Algoritma ACO merupakan algoritma untuk menyelesaikan masalah optimisasi pada suatu sistem dengan pendekatan fenomena kasus perjalanan semut dari suatu tiitk ke tiitk yang lain. Semut pada algoritma kali ini dianggap sebagai proses optimisasi parameter PID dan menghasilkan berbagai macam solusi untuk menghasilkan parameter PID yang optimal.

Pada algoritma ACO juga memiliki beberapa parameter untuk mendukung proses optimisasinya, beberapa parameternya seperti yang ditampilkan pada tabel 2.2

Tabel 2.2 Parameter Algoritma ACO

Parameter	Nilai
Jumlah Semut	30
Iterasi Maksimal	100
Intensitas Feromon	0.8
Nilai Beta	0.2
Evaporasi	0.7

Setelah kita menentukan parameter dari algoritma yang kita gunakan, maka selanjutnya kita menentukan nilai inisialisasi perubahan feromon pada tiap semutnya. Perubahan feromon ini berguna untuk mencari solusi terbaik sesuai bobot dari sistem yang kita buat, sehingga tiap semut saat dijalankan membawa bobot solusinya sendiri. Gambar 2.3 menjelaskan flowchart dari algoritma Ant Colony Optimization.

Gambar 2.3 *Flowchart Ant Colony Optimization*

Selanjutnya setelah kita memasukan parameter ACO, maka algoritma akan dijalankan sesuai dengan diagram alur pada gambar 2.3**.**

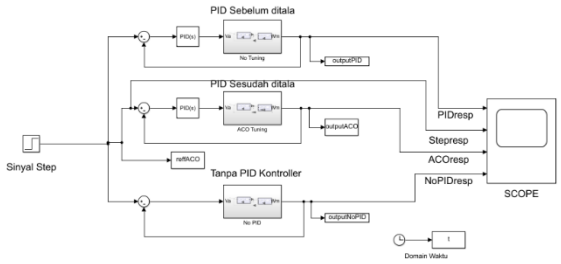
Fungsi objektif yang digunakan menggunakan *Integral Time Absolut Error (ITAE)* yang dijelaskan pada persamaan 2.11.

$$ITAE = \int_0^t t\,|\Delta e(t)|\,dt \qquad (2.11)$$

Fungsi objektif ini akan menghasilkan nilai *fitness* dari setiap proses iterasinya, semakin dekat nilai fitness dengna nilai yang kita inginkan maka algoritma akan menghasilkan nilai yang optimal untuk optimisasi penalaan PID pada motor DC.

5. Model Sistem

Setelah kita melakukan pemodelan pada motor DC, PID, dan Algoritma *Ant Colony Optimization* maka keseluruhan pemodelan tersebut dijalankan pada *software* Simulink dengan rangkaian model sistem seperti pada gambar 2.4.



Gambar 2.4 Model Sistem Penelitian

Pada kasus ini akan dilakukan perbandingan sinyal keluaran dari 3 sistem yang berbeda yaitu sistem dengan PID menggunakan proses penalaan otomatis melalui *Toolbox PID Tuner*, lalu sistem dengan PID yang ditala menggunakan algoritma *Ant Colony Optimization*, dan juga sistem tanpa adanya controller. Sehingga dengan seperti ini kita dapat melihat bagaimana performansi sistem PID yang ditala menggunakan algoritma ACO dengan kriteria performansi sebagai berikut :

1. *Error steady state* kurang dari 5% dari *set point*.
2. *Rise Time* maksimal 1s.
3. *Setting time* maksimal 2s
4. *Maximum Overshoot* sebesar 5% dari *set point.*

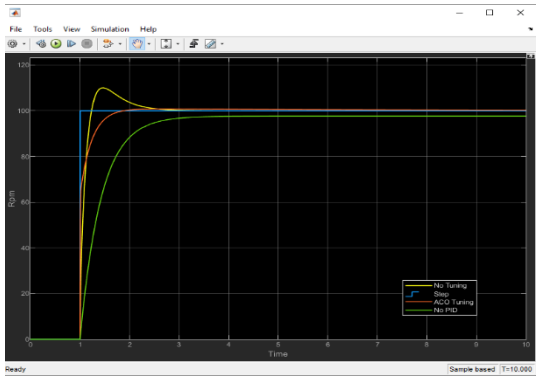
HASIL DAN PEMBAHASAN

Setelah program dijalankan selama 100 iterasi didapatkan nilai parameter PID yang menggunakan algoritma *Ant Colony Optimization* dan juga terdapat nilai PID sebagai nilai pembanding yaitu menggunakan nilai PID pada sistem yang menggunakan *Toolbox PID Tuner*. Hasil penalaannya dijelaskan pada tabel 3.1.

Tabel 3.1 Hasil Paramater PID

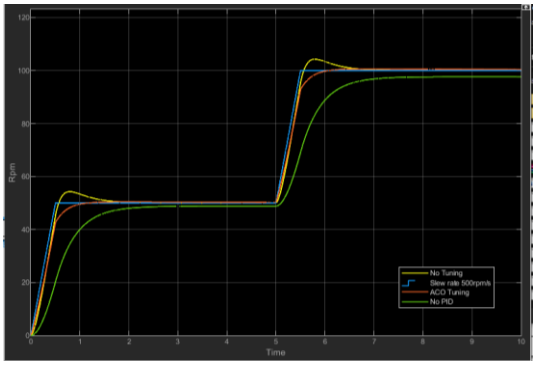
Param.	*Tuner*	Ant Colony
Kp	4.6848	4.9804
Ki	9.3483	0.82332
Kd	0.0791	0.68298

Hasil sinyal respon berbagai macam sistem tersebut dijelaskan pada gambar 3.1. Dimana warna biru menunjukkan sinyal step yang digunakan, warna kuning merupakan sinyal respon hasil penalaan PID pada matlab, sinyal merah merupakan hasil penalaan PID menggunakan ACO dan sinyal hijau merupakan sistem tanpa PID controller.

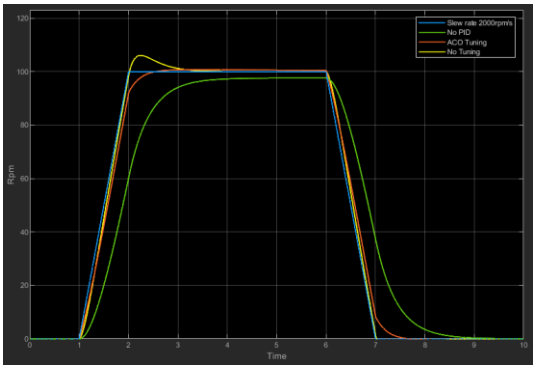


Gambar 3.1 Hasil Sinyal Respon Sistem

Dari hasil penalaan tersebut saya mencoba untuk merubah masukan sinyal step yang digunakan dengan 2 sinyal step dan juga saat motor DC dimatikan untuk melihat performansi yang dihasilkan apakah efektif atau tidak.



Gambar 3.2 Respon Sistem dengan Sinyal 2 Step



Gambar 3.3 Respon Sistem saat dimatikan

Dari hasil sinyal respon tersebut saya juga mendapatkan hasil performansi sistem dari

setiap sinyal tersebut yang dijelaskan pada tabel 3.2.

Tabel 3.2 Performansi Sinyal Respon

Perform	No PID	Tuner	ACO
Rise Time (s)	0.918	0.165	0.293
Setting Time (s)	3.08	1.22	0.607
Overshoot (%)	0.501	1 0	0.455

Dapat terlihat bahwa respon motor DC saat tidak menggunakan kontroller memiliki nilai respon yang sangat jauh bahkan tidak mencapai nilai *set point*, dikarenakan sistem tersebut tidak memakai kontroller untuk menstabilkan nilai keluaran dan nilai masukan. Akibatnya memiliki nilai *setling time* yang sangat besar namun *overshoot*nya kecil akibat tidak memiliki nilai parameter PID yang dapat meningkatkan *overshoot*. Sehingga performansi sistem ini tidak memasuki kriteria performansi optimal.

Respon motor DC menggunakan PID *Tuner* memiliki sinyal respon dengan *Overshoot* yang sangat tinggi ini dikarenakan nilai parameter Ki yang cukup besar mengakibatkan *overshoot* sistem cukuplah tinggi. Pada respon ini juga memiliki nilai *rise time* dan *settling time* yang cukup cepat dikarenakan nilai parameter Kp yang tinggi, parameter ini berguna untuk mempercepat respon sistem. Namun dengan nilai *overshoot* yang sangat tinggi mengakibatkan sistem ini tidak termasuk ke dalam sistem yang optimal.

Pada respon motor DC yang sudah menggunakan *Ant Colony Optimization* sebagai proses penalaan untuk PID, kita dapat melihat hasil respon sistem tersebut yang begitu mendekati nilai *set point*. PID pada sistem ini memiliki parameter yang dapat mempercepat respon sistem karena memiliki parameter Kp yang cukup besar sehingga membuat PID dapat meningkatkan waktu *rise time* maupun *settling time*. Lalu parameter Ki dan Kd pada sistem ini mengakibatkan hasil sinyal respon tidak memiliki *overshoot* yang tinggi. Sehingga sistem seperti ini merupakan sistem yang stabil dibandingkan dengan kedua sistem sebelumnya dikarenakan dengan menggunakan algoritma *Ant Colony Optimization* yang melakukan proses iterasi sebanyak 100 kali dapat mengevaluasi hasil penalaan PID setiap iterasinya sehingga sistem dapat menghilangkan nilai *error* sistem tersebut. Jika kita lihat juga dengan hasil yang merubah-ubah nilai step sistem dapat terlihat bahwa sistem yang menggunakan PID ACO mendapatkan performansi yang sangat baik dibandingkan dengan sistem yang lainnya. Sistem yang menggunakan PID ACO ini mempunyai kecepatan respon yang cepat dan juga stabilitas respon yang stabil. Maka sistem ini merupakan sistem penalaan PID untuk motor DC yang optimal dibandingkan dengan kedua sistem lainnya.

PENUTUP

Hasil penalaan menggunakan algoritma *Ant Colony Optimization* mendapatkan nilai performansi yang sangat baik, dan juga sinyal respon sistem menunjukan bahwa sistem memiliki respon yang sangat cepat dan stabil dibandingan dengan sistem yang lainnya. Sehingga penulis berkesimpulan bahwa penalaan menggunakan algoritma *Ant Colony Optimization* sangatlah baik sebagai proses penalaan PID untuk motor DC. Saran pada penelitian kali ini baiknya menggunakan pembanding lainnya yang memiliki kesamaan yang cukup dekat dengan *Ant Colony Optimization*. Sehingga kita dapat mengetahui secara kritis apakah metode ACO ini efektif untuk menala PID atau tidak.

DAFTAR PUSTAKA

Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. In *Computational intelligence for multimedia big data on the cloud with engineering applications* (pp. 185-231). Academic Press.

Djalal, M. R., Nurohmah, H., Imran, A., & Yunus, M. Y. (2017). Aplikasi Metode Cerdas untuk Optimasi Controller PID Motor DC Berbasis Firefly Algorithm. Jurnal Nasional Teknik Elektro, 6(2), 76-83.

Khine, M. K. E. E., Htwe, M. W. M. M., & Mon, M. Y. Y. (2019). Simulation DC Motor Speed Control System by using PID Controller.

Kim, S. H. (2017). Electric motor control: DC, AC, and BLDC motors. Elsevier.

Kommula, B. N., & Kota, V. R. (2020). Direct instantaneous torque control of Brushless DC motor using firefly Algorithm based fractional order PID controller. *Journal of King Saud*

University-Engineering Sciences, *32*(2), 133-140.

Muhammad, R. D., & Rahmat, R. (2020). Desain Kontrol Motor Dc Menggunakan Ant Colony Optimization. Jurnal Teknologi (Jurtek), 12(1), 49-56.

Ogata, K. (1991). Teknik Kontrol Otomatik, Jilid 1. Erlangga, Jakarta.

Sabir, M. M., & Khan, J. A. (2014). Optimal design of PID controller for the speed control of DC motor by using metaheuristic techniques. *Advances in artificial neural systems*, *2014*.

Yang, X. S. (2010). Nature-inspired metaheuristic algorithms. Luniver press.

Yazgan, H. R., Yener, F., Soysal, S., & Gür, A. E. (2019). Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor. *Sakarya University Journal of Science*, *23*(2), 162-174.