

STUDI PERFORMANSI PENGENDALI *SLIDING MODE* DAN PID PADA PENGENDALIAN KECEPATAN MOTOR DC

Oleh:

Dian Mursyitah

Jurusan Teknik Elektro, UIN Suska Riau, Pekanbaru,

dmursyitah@uin-suska.ac.id

Abstract

This research was proposed to study the performance of sliding mode and PID controller to control speed of DC motor. Performance study has done by simulation in Matlab/Simulink. Based on observation, the result shows that both of controllers i.e sliding mode control and PID produces good performance. However, based on analysis result, the sliding mode controller produces better performance than PID controller by comparison of transient time is 1 : 3. With the following result are $\tau = 0.1001$ seconds, $t_s = 0.5005$ s, $t_r = 0.2947$ seconds, and $t_d = 0.2487$ seconds.

Keywords: PID, sliding mode control, speed of DC motor

Abstrak

Penelitian ini diusulkan untuk mempelajari kinerja modus dan PID geser untuk mengontrol kecepatan motor DC. Studi Kinerja telah dilakukan simulasi di Matlab / Simulink. Berdasarkan pengamatan, hasilnya menunjukkan bahwa ke dua pengendali yaitu sliding mode kontrol PID dan menghasilkan kinerja yang baik. Namun, berdasarkan hasil analisis, mode pengontrol geser menghasilkan kinerja yang lebih baik dari PID controller dengan perbandingan waktu transien adalah 1 : 3 Dengan hasil berikut adalah $\tau = 0,1001$ detik, $t_s = 0,5005$ s, $t_r = 0,2947$ detik, dan $t_d = 0,2487$ detik.

Kata Kunci : PID, model pergeseran kontrol, kecepatan motor DC

1. Pendahuluan

Keberadaan pengendali dalam industri memberikan kontribusi yang besar terhadap perilaku sistem yang akan dikendalikan. Perkembangan pengendali telah mengalami kemajuan yang pesat dewasa ini, dimulai dari kendali konvensional hingga kendali modern [1].

Pesatnya perkembangan pengendali membuat desainer sistem kendali harus mampu memilih dan merancang pengendali yang tepat untuk mengatasi permasalahan yang muncul. Karena, masing-masing kendali memberikan efek yang berbeda pada sistem. Misalnya suatu sistem dengan pemodelan matematis yang kompleks dapat dikendalikan dengan pengendali modern yang memiliki pemodelan matematis yang kompleks pula. Namun, permasalahan sistem tersebut dapat pula terselesaikan hanya dengan pengendali konvensional yang memiliki pemodelan matematis sederhana. Berdasarkan hal tersebut, diperlukan suatu studi yang dapat

memberikan informasi mengenai pengendali konvensional dan modern.

Salah satu pengendali modern adalah pengendali *sliding mode*. Pengendali *sliding mode* merupakan salah satu pengendali yang memiliki pemodelan matematis yang kompleks. Pengendali *sliding mode* menggunakan teori pensaklaran berkecepatan tinggi yang berfungsi untuk membawa *trajectory state* dari sistem ke dalam sebuah permukaan luncur, kemudian *trajectory state* tersebut dipelihara dengan tujuan agar tetap pada permukaan luncur tersebut [1][2]. Dalam proses pemeliharaan permukaan luncur akan mengakibatkan terjadinya osilasi atau yang sering dikenal dengan istilah *chattering* [1][2][3][4]. Fenomena *chattering* telah diselesaikan oleh para peneliti dengan mengganti fungsi *signum* dengan fungsi saturasi atau arcus tangen[5].

Pengendali *Proportional Integral Derivative* (PID) atau yang dikenal dengan *three term control* merupakan pengendali konvensional

yang banyak digunakan dalam berbagai industri. Pengendali ini memiliki algoritma yang sederhana dan mudah untuk dipahami [7]. Namun pengendali PID ini sangat bergantung pada penalaan parameter P, I, dan D untuk menghasilkan respon yang baik.

Pada penelitian ini akan dilakukan studi performansi pengendali *sliding mode* dan PID untuk mengendalikan kecepatan motor DC. Studi performansi bertujuan untuk mengetahui pengendali mana yang lebih baik dalam mengendalikan kecepatan motor DC.

2. Landasan Teori

Pengendali *Sliding Mode*

Pengendali *sliding mode* adalah kendali umpan balik pensaklaran berkecepatan tinggi yang dapat diterapkan pada sistem linier maupun nonlinier. Prinsip kerja dari pengendali *sliding mode* adalah memaksa *trajectory state* suatu sistem menuju permukaan luncur tertentu dan mempertahankannya pada permukaan luncur tersebut [2][3][4].

a. Permukaan luncur

Permukaan luncur dipilih dengan pertimbangan status trayektori sistem dapat menuju permukaan tersebut di manapun kondisi awalnya dan dalam waktu yang terbatas serta status trayektori sistem dapat dipertahankan di sekitar permukaan luncur.

$$s(x,t) = 0 \quad (1)$$

b. Sinyal kendali

Untuk mendapatkan sinyal kendali yang mampu membawa status trayektori menuju permukaan luncur dan mempertahankan status trayektori agar tetap berada di sekitar permukaan luncur, maka diperlukan dua macam sinyal kendali. Sinyal kendali yang pertama adalah sinyal kendali ekivalen yang berfungsi untuk membawa status trayektori menuju permukaan luncur (U_{eq}) dan sinyal kendali yang kedua adalah sinyal kendali natural yang berfungsi untuk mempertahankan status trayektori agar tetap berada di sekitar permukaan luncur (U_n) [4].

$$U = U_{eq} + U_n \quad (2)$$

Pengendali PID

PID adalah salah satu pengendali otomatis yang terdiri dari perpaduan tiga aksi dasar kendali yaitu Proporsional, Integral, dan Derivatif. Pengendali PID merupakan pengendali berumpan balik yang paling populer di dunia industri [4][7].

$$u(t) = K_p e(t) + k_i \int e(t) dt + K_d \frac{de}{dt} \quad (3)$$

Dimana:

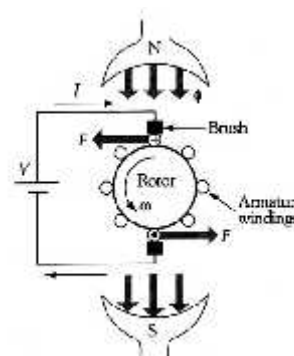
K_p = Proporsional Gain

K_i = Integral Gain

K_d = Derivatif Gain

Motor DC

Motor arus searah (motor DC) adalah motor yang digerakkan oleh energi listrik arus searah. Sebuah motor DC biasanya tersusun atas magnet permanen, kumparan jangkar, dan sikat (*brush*). Medan magnet yang besarnya konstan dihasilkan oleh magnet permanen, sedangkan komutator dan sikat berfungsi untuk menyalurkan arus listrik dari sumber di luar motor ke dalam kumparan jangkar. Letak sikat di sepanjang sumbu netral dari komutator, yaitu sumbu dimana medan listrik yang dihasilkan bernilai nol. Hal ini dimaksudkan agar pada proses perpindahan dari sikat ke komutator tidak terjadi percikan api [2][6]. Struktur motor DC ditunjukkan pada Gambar 1.



Gambar 2. Motor DC[2]

Motor DC Sangat Dikenal Pemakaiannya Yang Beraneka Ragam, Salah Satu Contohnya Adalah Digunakan Pada Sistem Manufaktur Modern Yaitu Pada Bidang Industri Produk Dan Layanan, Seperti Industri Makanan Dan Kimia,

Peralatan Mesin, Robotik, *Aeronautic* Dan Wahana Elektrik. Hal Ini Dikarenakan Motor DC Memiliki Karakteristik yang Mudah Diatur, Terutama Pada Pemakaian Yang Memerlukan Rentang Kecepatan Motor Yang Lebar Atau Pengaturan Yang Teliti Pada Keluaran Motornya. Meski Akan Diprediksi Bahwa Motor Arus Bolak Balik (Motor AC) Akan Menggantikan Motor Arus Searah (Motor DC), Namun Kenyataannya Untuk Aplikasi Kecepatan Masih Didominasi Oleh Motor DC. Hal Ini Dikarenakan Motor DC Lebih Cenderung Handal (*Reliability*), Sederhana (*Simple*) Dan Relatif Murah (*Lower Cost*) [6][8].

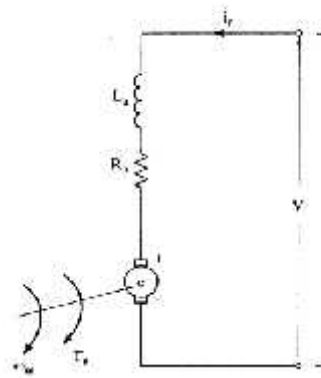
3. Metodologi Penelitian

Metodologi Penelitian Dimulai Dengan Mencari Studi Literatur Yang Berhubungan Dengan Penelitian Yaitu Pengendali *Sliding Mode*, Pengendali PID, Motor DC Dan Lain Sebagainya. Selanjutnya Dilakukan Penurunan Penurunan Rumus-Rumus Berdasarkan Hukum-Hukum Fisika Dan Matematika Melalui Rangkaian Ekuivalen Motor DC Guna Mendapatkan Pemodelan Matematis Motor DC. Pemodelan Yang Telah Diperoleh Perlu Diuji Secara Openloop Untuk Mendapatkan Perilaku Sistem Motor DC.

Tahapan Selanjutnya Adalah Merancang Pengendali *Sliding Mode* Dan PID Dengan Menurunkan Persamaan Matematis Masing-Masing Pengendali. Setelah Diperoleh Sinyal Kendali Masing-Masing Pengendali *Sliding Mode* Dan PID Dalam Bentuk Persamaan Matematis, Maka Persamaan Tersebut Akan Ditransformasikan Ke Dalam Bentuk Program Simulink Matlab. Tahapan Terakhir, Berdasarkan Hasil Simulasi Akan Dianalisa Respon Kedua Pengendali Dalam Mengendalikan Kecepatan Motor DC. Studi Performansi Dapat Dilakukan Dengan Menganalisa Hasil Perbandingan Kedua Pengendali.

Pemodelan Matematis Motor DC

Pemodelan Matematis Sistem Motor DC Dilakukan Dengan Cara Menurunkan Persamaan Berdasarkan Hukum-Hukum Fisika Yang Diilustrasikan Pada Gambar 2.



Gambar 2. Rangkaian Ekuivalen motor DC[9]

Untuk persamaan elektrik rangkaian ekuivalen motor DC dalam bentuk laplace :

$$V(s) = R_a I_a(s) + sL_a I_a(s) + K_b \omega_m(s) \quad (4)$$

Untuk persamaan mekanik rangkaian ekuivalen motor DC dalam bentuk laplace :

$$K_t I_a(s) = sJ \omega_m(s) + B_t \omega_m(s) \quad (5)$$

Dengan demikian, fungsi alih untuk motor DC menjadi:

$$\frac{\omega_m(s)}{V(s)} = \frac{K_t}{s^2 + s \left(\frac{JR_a + L_a B_t}{JL_a} \right) + \left(\frac{B_t R_a + K_t K_b}{JL_a} \right)} \quad (8)$$

Tabel 1. Parameter motor DC [9]

No	Parameter motor DC	Nilai
1	R_a	0.5 Ω
2	L_a	1 mH
3	J	0.001 kg.m ²
4	B_t	0.01 N.m rad.sec ⁻¹
5	K_b	0.001 V.sec rad ⁻¹
6	K_t	0.008 NmA ⁻¹
7	V	240 V

Sehingga dengan memasukkan parameter motor DC, maka fungsi transfer menjadi :

$$\frac{\omega_m(s)}{V(s)} = \frac{8000}{s^2 + s510 + 5001} \quad (9)$$

$$a = 510; b = 5001; c = 8000 \quad (10)$$

$$A = \begin{bmatrix} 0 & 1 \\ -5001 & -501 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -8000 \end{bmatrix} \quad (11)$$

Perancangan Pengendali *Sliding Mode*

Persamaan (8) dimisalkan menjadi fungsi transfer yang memiliki koefisien numerator dan denominator seperti persamaan (9), kemudian direpresentasikan dalam bentuk persamaan diferensial:

$$\ddot{\omega}_m + a\dot{\omega}_m + b\omega_m = cv \quad (12)$$

dengan *error* adalah:

$$e = r - \omega_m; \dot{x}_1 = r - \omega_m \quad (13)$$

Karena setpoint tetap, maka:

$$\dot{x}_1 = x_2 = -\dot{\omega}_m; \dot{x}_2 = -\ddot{\omega}_m \quad (14)$$

Substitusikan persamaan (13) dan (14) ke persamaan (12):

$$\dot{x}_2 = -ax_2 - bx_1 + br - cv \quad (15)$$

Dapat diilustrasikan dengan persamaan sistem berikut:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{x} &= Ax + Bu_{eq} + Bu_N \end{aligned} \quad (16)$$

permukaan luncur didefinisikan:

$$\begin{aligned} S\dot{x} &= 0 \\ S\dot{x}(Ax + Bu_{eq} + Bu_N) &= 0 \end{aligned} \quad (17)$$

Maka sinyal kendali ekivalen didapat dengan asumsi bahwa sinyal kendali natural 0:

$$\begin{aligned} SAx + SBu_{eq} &= 0 \\ u_{eq} &= \frac{1}{c}((1-a)x_2 - bx_1 + br) \end{aligned} \quad (18)$$

Dengan demikian sinyal kendali natural dapat diperoleh:

$$\begin{aligned} \dot{\sigma}(x) &= SAx + SBu_{eq} + SBu_N \\ \dot{\sigma}(x) &= -cu_N \end{aligned} \quad (19)$$

Berdasarkan analisa kestabilan Lyapunov:

$$S\dot{S} \leq -\eta|S| \quad (20)$$

Dengan $\eta > 0$:

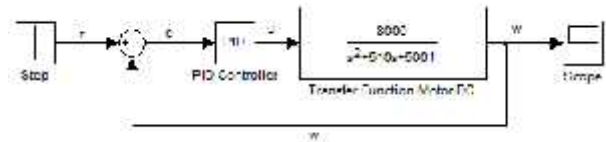
$$\begin{aligned} -cu_N &= -\eta \text{sign}(\sigma(x)) \\ u_N &= \frac{1}{c} \eta \text{sign}(\sigma(x)) \end{aligned} \quad (21)$$

Sehingga sinyal kendali total:

$$u = \frac{1}{c}((1-a)x_2 - bx_1 + br) + \frac{1}{c} \eta \text{sign}(\sigma(x))$$

Perancangan Pengendali PID

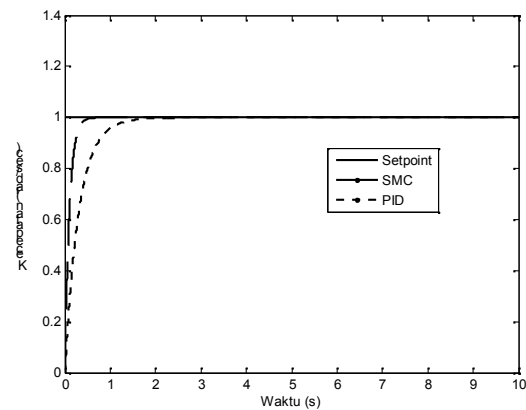
Perancangan pengendali PID dilakukan dengan menggunakan blok yang sudah tersedia di program simulink Matlab [7]. Seperti ditunjukkan pada Gambar 3.



Gambar 3. Blok kendali PID

4. Hasil Dan Pembahasan

Simulasi dilakukan untuk mengetahui performansi sistem sebelum diimplementasikan pada sistem yang sebenarnya. Simulasi menggunakan perangkat lunak *Matlab/Simulink 7.8.0 R2009a*. Pengujian dilakukan dengan memberikan masukan kecepatan sebesar 1 rad/sec. Parameter η pada pengendali *sliding mode* dipilih dengan metode *trial and error* pada nilai 10. Sedangkan parameter pengendali PID dipilih nilai $K_p = 1,5$; $K_i = 2$; $K_d = 0,4$. Hasil simulasi ditunjukkan pada Gambar 4.



Gambar 4. Respon sistem menggunakan pengendali *sliding mode* dan PID dalam mencapai *setpoint*

Dari gambar 4 ditunjukkan hasil perbandingan antara respon sistem yang menggunakan pengendali *sliding mode* dan pengendali PID. Tabel 2 berikut ini menampilkan analisa respon gabungan antara pengendali *sliding mode* dengan pengendali PID.

Tabel 2. Perbandingan analisa respon sistem antara pengendali *sliding mode* dan PID mencapai *setpoint*

Analisa Respon	Pengendali <i>sliding mode</i>	Pengendali PID
τ	0,1001 detik	0,3090 detik
t_s	0,5005 detik	1,5450 detik
t_r	0,2947 detik	0,9098 detik
t_d	0,2487 detik	0,7678 detik
e_{ss}	0	0,0001

5. Kesimpulan

Berdasarkan hasil simulasi performansi pengendali *sliding mode* dan PID, maka didapatkan hasil bahwa pengendali *sliding mode* menghasilkan performansi yang lebih baik dibandingkan pengendali PID dengan perbandingan waktu transien sebesar 1 : 3. Untuk mengoptimalkan pengendali *sliding mode* sehingga dapat menghasilkan performansi yang lebih baik lagi, maka pengendali *sliding mode* dapat dikombinasikan dengan berbagai pengendali lainnya.

Daftar Pustaka

- [1] David Young, dkk. 1999. “*A Control Engineer’s Guide to Sliding Mode Control*”. IEEE Transactions on Control System Technology, Volume 7, No. 3.
- [2] Kurniawan Subroto, Ramadhani. 2012. “*Desain dan Implementasi Kontroler Optimal Sliding Mode untuk Tracking Posisi Motor DC MS150*”. ITS. Surabaya.
- [3] Musmade, B. B dkk. 2011. “*Design of Sliding Mode Controller to Chemical Processes for Improved Performance*”. International Journal of Control and Automation Volume 4 No. 1.
- [4] Nasir, A. N. K dkk. 2010. “*Performance Comparison between Sliding Mode Control (SMC) and PID Controllers for a Nonlinear Inverted Pendulum System*”. World Academy of Science, Engineering and Technology 71.
- [5] Mursyitah, Dian dkk. 2012. “*Control of Continuous Stirred Tank Reactor (CSTR) using Decouple Sliding Mode Controller*”. SENAKI. ITS Surabaya.

- [6] Kumar Mishra, Akhilesh dkk. “*Speed Control of DC Motor Using Artificial Bee Colony Optimization Technique*”. Universal Journal of Electrical and Electronic Engineering, Volume 1, No.3, pp. 68-75. 2013
- [7] Mursyitah, Dian. 2009 “*Analisa Pengendalian Proporsional Integratif Derivatif (PID) pada Proses Mixing Tank*” UIN SUSKA Riau.
- [8] Noraraba’afyah. 2012. “*Q-Parameterization Controller Design for DC Motor System*”. UTM. Malaysia.
- [9] Rahmani, Rasoul dkk. 2012. “*Fuzzy Logic Controller Optimized by Particle Swarm Optimization for DC Motor Speed Control*”. IEEE Student Conference on Research and Development pp. 109-113.
- [10] Patre, B. M dkk. 2011 “*Sliding Mode Control of DC Drives*”. Sliding Mode Control, Prof. Andrzej Bartoszewicz (Ed). In Tech. April (<http://www.intechopen.com> diakses Februari 2014)