DAFTAR PUSTAKA

Suganthi P, Nagapavitra S, Umamaheswari S, "Modeling and Simulation of Closed Loop Speed Control for BLDC Motor," IEEE Conference on Emerging Device and Smart System (ICEDSS), 2017.

M. Mahmud, S. M. A. Motakabber, A. H. M. Z. Alam, A. N. Nordin, "Adaptive PID Controller Using for Speed Control of the BLDC Motor," International Conference on Semiconductor Electronics (ICSE), 978-1-7281-5968-3/20, 2020.

S. Agustian, E. Purwnto, O. Asrarul, "Pengaturan Kecepatan Motor Brushless Direct Current (BLDC) Dengan Vector Control," Jurnal INOVTEK SERI ELEKTRO, Vol. 3, No. 1, E-ISSN: 2716-1684, April 2021.

K. Gadeka, S. Joshi, H. Metha, "Performance Improvement in BLDC Motor Drive Using Self Tuning PID Controller," International Conference on Inventive Research in Computing Applications (ICIRCA), ISBN: 978-1-7281-5374-2, 2020.

S. Achary and V. Sharma, "Speed Control of Brushless DC Motor Using Zeta Converter," International Conference on Smart Electronics and Communication (ICOSEC), ISBN: 978-1-7281-5461-9, 2020.

K. S. Devi, R. Dhanasekaran, S. Muthulakshmi, "Improvement of Speed Control Performance in BLDC Motor Using Fuzzy PID Controller," International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), ISBN: 978-1-4673-9545-8, 2016.

Seyed Mohammad H M, Seyed Sajjad S G S, Seyed Mohsen M, and Seyed Ebrahim S G S, "Reduction of Torque Rippe and Increase of torque Capacity of BLDC Motor," International Conference on Power Electronics, Drives and Energy Systems, 2012.

Bonula S. Kalyani, Venu M. Mukkavilli, Dr. Gopichand naik, "Performance Enhancement of Permanent Magnet Brushless DC Motor Using Multilevel Inverter," International Advance Computing Conference. 2017.

Renny R, farid D Murdianto, and Muhammad W Alim, "Soft Starting & Performance Evaluation of PI Speed Controller for Brushless DC Motor Using Three Phase Six Step Inverter," International Seminar on Application for Technology of Information and Communication (Isemantic), 2018.

P. Hari Krishnan and M. Arjun, "Control of BLDC Motor Based oon Adaptive Fuzzy Logic PID Controller," International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), 2014.

K. Sarojini D, R. Dhanasekaran, and S. Muthulakshmi, "Improvement of Speed Control Performance in BLDC motor using Fuzzy PID Controller," International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2014.

M. Yildirim, M. Polat, H. Kurum, "A Survey on Comparison of Electric Motor Types and Drives Used For Electric Vehicles," 16th International Power Electronics and Motion Control Conference and Exposition, 2014.

http://repository.unika.ac.id/15044/4/11.50.0012%20B.%20Andryan%20Wicaksono%20BAB%20III.pdf (Gambar outer dan inner rotor).

R. Somanatham, P. V. N. Prassad, A. D. Rajkumar, "Modeling and Simulation of Sensorless Control of PMBLDC Motor Using Zero-Crossing Back E.M.F Detection," IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2006.

M. Fajri. Sachruddin, F. A. Samman, R. S. Sadjad, "BLDC Motor Control Using a Complex Programmable Logic Device with Hall-Sensors," IEEE International Conference on Smart-Green Technology in Electrical and Information Systems (ICSGTEIS), Bali, 2021.

Nanda R A, H. Suryoatmojo, S. Anam, "Desain Kontrol Kecepatan Motor Brushless DC Berbasis Power Factor Correction (PFC) Menggunakan Single Ended Primary Inducta Converter (SEPIC)," Jurnal Teknik ITS, Vol. 5, No. 2. ISSN: 2337-3539. 2016.

N. Hidayat, F. A. Samman, R. S. Sadjad, "FPGA Based Controller of BLDC Motor Using Trapezoid Control," IEEE 14th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, 2022.

Suhendra, T. *et al.* (2018) 'Kendali Kecepatan Motor DC dengan Metode Pulse Width Modulation menggunakan N-channel Mosfet', *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, 07(02), pp. 78–85. Available at:https://media.neliti.com/media/publications/266846-kendali kecepatan-motor-dc-dengan-metode-a7110155.pdf (Accessed: 24 June 2023).

Badarov, D.H. and Manoev, V.D. (2017) 'Teaching methodology for digital pulse width modulation and phase angle control', in *XXVI International Scientific Conference Electronics (ET)*. Available at: https://doi.org/https://doi.org/10.1109/ET.2017.8124335.

Gadgetronicx Team (2021) *PWM : Pulse Width Modulation – explained*. Available at: https://www.gadgetronicx.com/pwm-pulse-width-modulation-explained/ (Accessed: 7 August 2023).

A. Fauzan Alim, “Sistem kendali Kecepatan Motor Brushless DC dengan Pengendali PI,” Departement Teknik Elektro Universitas Hasanuddin, 2020.

Faizal A. Samman (2016). Sistem Kendali Adaptif (Desain, Pemodelan dan Simulasi Analog). Lembaga Sains, Teknologi dan Seni (Institute of Sciences, technologies and Arts – IESTA). Gowa

Rajendra, R., and Pratihar, D. K. 2011. Particle Swarm Optimization Algorithm vs Genetic Algorithm to Develop Integrated Scheme for Obtaining Optimal Mechanical Structure and Adaptive Controller of a Robot. Intelligent Control and Automation, 2011, 2, 430-449.

Arkeman, Y., dkk. 2014. Algoritma Genetika Tujuan Jamak. Bogor: PT Penerbit IPB Press.

LAMPIRAN

Lampiran 1. Pencarian Nilai Kp

No.	KP	KI	KD	RMS PID
1	0	0	0	327,15
2	0,1	0	0	911,6
3	0,2	0	0	599,2
4	0,3	0	0	451,8
5	0,4	0	0	365,2
6	0,5	0	0	308,2
7	0,6	0	0	267,8
8	0,7	0	0	237,9
9	0,8	0	0	215,3
10	0,9	0	0	197,5
11	1	0	0	183,2
12	1,1	0	0	171,7
13	1,2	0	0	161,8
14	1,3	0	0	153,9
15	1,4	0	0	147,9
16	1,5	0	0	141,2
17	1,6	0	0	136,1
18	1,7	0	0	131,8
19	1,8	0	0	128,9
20	1,9	0	0	124,6
21	2	0	0	121,7
22	2,1	0	0	119,1
23	2,2	0	0	116,7
24	2,3	0	0	114,6
25	2,4	0	0	112,8
26	2,5	0	0	111,1
27	2,6	0	0	109,5
28	2,7	0	0	108,1
29	2,8	0	0	106,9
30	2,9	0	0	105,8
31	3	0	0	104,7

Lampiran 2. Pencarian Nilai Ki

NO	KP	KI	KD	RMS PID
1	3	0	0	104,7
2	3	0,1	0	99,99
3	3	0,2	0	96,83
4	3	0,3	0	94,67
5	3	0,4	0	93,17
6	3	0,5	0	92,09
7	3	0,6	0	91,03
8	3	0,7	0	90,07
9	3	0,8	0	90,24
10	3	0,9	0	89,87
11	3	1	0	89,58
12	3	1,1	0	89,34
13	3	1,2	0	89,14
14	3	1,3	0	88,97
15	3	1,4	0	88,83
16	3	1,5	0	88,7
17	3	1,6	0	88,59
18	3	1,7	0	88,05
19	3	1,8	0	88,42
20	3	1,9	0	88,35
21	3	2	0	88,28
22	3	2,1	0	88,23
23	3	2,2	0	88,18
24	3	2,3	0	88,130
25	3	2,4	0	88,009
26	3	2,5	0	88,060
27	3	2,6	0	88,050
28	3	2,7	0	88,040
29	3	2,8	0	88,030
30	3	2,9	0	88,020
31	3	3	0	88,010

Lampiran 3. Pencarian Nilai Kd

NO	KP	KI	KD	RMS PID
1	3	2,4	0	88,009
2	3	2,4	0.1	128,4
3	3	2,4	0.2	175,5
4	3	2,4	0.3	212,9
5	3	2,4	0.4	244,8
6	3	2,4	0.5	273
7	3	2,4	0.6	298,6
8	3	2,4	0.7	322,2
9	3	2,4	0.8	344,2
10	3	2,4	0.9	364,8
11	3	2,4	1	384,4
12	3	2,4	1.1	403
13	3	2,4	1.2	420,7
14	3	2,4	1.3	437,8
15	3	2,4	1.4	454,2
16	3	2,4	1.5	470
17	3	2,4	1.6	485,5
18	3	2,4	1.7	500,7
19	3	2,4	1.8	515,5
20	3	2,4	1.9	529,9
21	3	2,4	2	544
22	3	2,4	2.1	557,9
23	3	2,4	2.2	571,5
24	3	2,4	2.3	585
25	3	2,4	2.4	598,3
26	3	2,4	2.5	611,4
27	3	2,4	2.6	624,4
28	3	2,4	2.7	637,1
29	3	2,4	2.8	649,7
30	3	2,4	2.9	662,2
31	3	2,4	3	674,6

Lampiran 4. Hasil pencarian menggunakan Metode PSO

No	Kp	Ki	Kd	RMSE (*Root Mean of Squared Error*)
1	0.406726915	0.351110459	5.04E-05	182097.0468
2	0.666931533	2.444045069	6.13E-05	56158.14341
3	0.93372566	0.974566316	8.19E-05	57038.12983
4	0.810950032	0.738684343	5.32E-05	70377.95102
5	0.484548272	1.028139663	2.02E-05	105395.2499
6	0.75674921	1.12707642	4.54E-05	67465.46943
7	0.417047454	1.639661379	4.28E-05	94599.05824
8	0.971785993	1.685760463	9.66E-05	47551.39878
9	0.987974701	1.187466686	6.20E-05	51402.71352
10	0.864147529	1.194392639	6.95E-05	58442.35391
11	0.388883776	1.546101658	7.20E-05	102777.9361
12	0.454741828	1.972591626	3.47E-05	80910.02242
13	0.246687198	2.852745595	5.17E-05	98935.41251
14	0.784423093	2.167045543	5.57E-05	52120.63668
15	0.882837606	1.200239236	1.56E-05	57060.04466
16	0.913711681	2.495614018	5.62E-05	44108.04073
17	0.558284924	0.403015025	6.95E-05	122324.6376
18	0.598868103	0.181400316	4.26E-05	125616.0913
19	0.14887672	0.252741157	8.36E-05	565389.5644
20	0.899713485	0.491694955	7.31E-05	67460.83551
21	0.450393581	0.972659761	3.60E-05	115326.2618
22	0.205672339	0.905180332	4.54E-05	221479.8982
23	0.899650991	0.035042973	3.86E-05	77963.15746
24	0.762585539	1.619715282	7.76E-05	59378.21433
25	0.882486307	0.286118078	7.34E-05	73571.62217
26	0.284950218	0.439544569	4.30E-05	248412.9372
27	0.673225986	1.893423621	6.94E-05	61893.62882
28	0.664279904	2.577961234	9.45E-05	55223.59879
29	0.122814994	2.922664894	7.84E-05	145172.58
30	0.407318423	1.712515282	7.06E-05	94140.6327
31	0.275286951	2.990550644	1.09E-05	90537.27769

Lampiran 5. Hasil Perbandingan Kecepatan *Trial and Error*

Variasi Kecepatan	Jenis Pengendali	*Settling Time* (s)	*Rise Time* (s)	*Overshoot* (%)
1000 RPM	PID	-	0,003798	42,143
	PID Adaptif	-	0,004269	10,565
1500 RPM	PID	-	0,004884	10,579
	PID Adaptif	0,019383	0,00618	4,801
2000 RPM	PID	-	0,010592	12,928
	PID Adaptif	0,014958	0,013496	1,266

Lampiran 6. Hasil simulasi paper terkait (Performace Improvement in BLDC Motor Drive Using Self-Tuning PID Controller)

Variasi Kecepatan	Jenis Pengendali	*Settling Time* (s)	*Rise Time* (s)	*Overshoot* (%)
1000 RPM	PID	0,03	0,0005	85
	Proposed Self-tuned PID	0,025	0,0025	90
1500 RPM	PID	0,0275	0,0006	86,67
	Proposed Self-tuned PID	0,0225	0,0025	33
2000 RPM	PID	0,025	0,00065	90
	Proposed Self-tuned PID	0,02	0,0035	25

Lampiran 7. Hasil simulasi paper terkait (Particle Swarm Optimization-Based BLDC Motor Speed Controller with Response Speed Consideration)

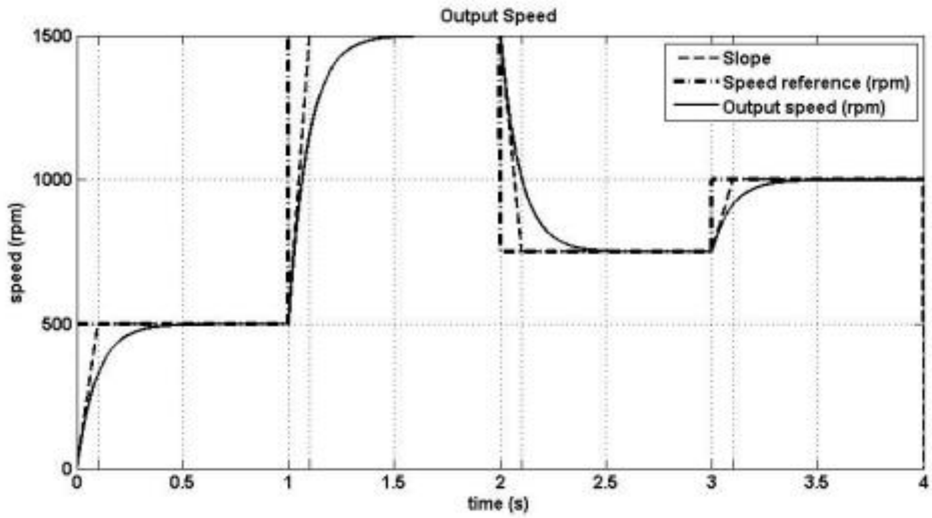
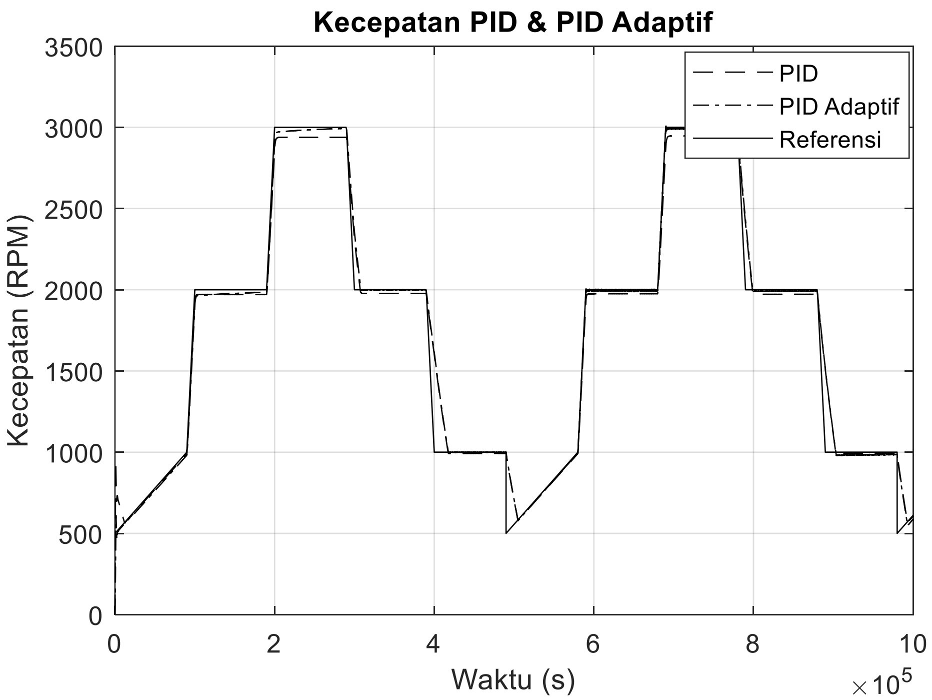


Fig 10. Output speed with various speed references ($t_{\alpha} = 0.1$ s)

Lampiran 8. Hasil simulasi kecepatan yang didapat



Lampiran 9 Paper

2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)

Performance Evaluation Brushless DC Motor System With Variable Loads

Febriyani Bahara
Department Of Electrical Engineering
Hasanuddin University
Makassar, Indonesia
baharuf20d@student.unhas.ac.id

Faizal Arya Samman
Department Of Electrical Engineering
Hasanuddin University
Makassar, Indonesia
faizalas@unhas.ac.id

Yusran
Department Of Electrical Engineering
Hasanuddin University
Makassar, Indonesia
yusran@unhas.ac.id

***Abstract*— The use of brushless DC motors will increase with the transition of conventional vehicle motors to electric vehicles (Electric Vehicles). The purpose of this study is to produce control that can maintain and increase the stability of motor rotation at changing or non-linear loads. It then performs testing of the created model. The control method used is PID. The control that is suitable for use in cases like this is the PID (Proportional Integral Derivative) control system. The best K_p, K_i, and K_d values obtained based on trial and error, and the values obtained are K_p = 1.1, K_i = 0.2, and K_d = 0.8. The response to distractions gets better and the response to speed variations gets better. In addition, the speed of the simulation results can follow the change in the speed reference with a slight overshoot in each transition of the speed value. The resulting system responds with a settling time value of 0.0035 s and a large overshoot with a value of 1.9%.**

Keywords*— *Motor Brushless DC, Commutation Logic, PID, Matlab/Simulink Software

I. Introduction

Considering that fuel oil (BBM) is increasingly scarce and prices are increasing plus the air pollution produced is increasing, motor vehicles must be immediately diverted using electric motors (Electic Vehicles), this can answer environmental problems and reduce vehicle emissions.

The electric motor commonly used in electric vehicles is a type of Permanent Magnet Brushless Direct Current (BLDC). This motor has the characteristics of reliability, efficiency, good performance, and large torque with relatively low maintenance costs when compared to DC motors that use commutators (1). So that it becomes an option in the fields of industrial automation, automotive, aviation, and mining (2).

Conventional controlling of DC motors poses many problems. Examples are the fracture of the agitator propeller rotor in the mining industry, and the damage to the pinion gear in motor vehicles due to an overshoot at the initial start of the motor. To produce a good (smooth) initial start of the motor and avoid beats, it is necessary to design a control that can maintain and increase the stability of the motor rotation at arbitrary or non-linear loads. A suitable control for use in cases like this is the PID (Proportional Integral Derivative) control system (3).

Control the rotational speed of the brushless DC motor using the PID control system makes the system more stable. The parameter in PID is K_p proportional constants used as gains, K_i are integral constants that function to improve steady-state responses, and K_d are derivative constants that can minimize the overshoot effect (4).

DC brushless motor control systems have been discussed in previous studies. Research (5) conducted is "New Control Method of DC Brushless Motor To Maximize Starting Torque". This research performs the replacement of transistors controlled with the C60Q system, with the new method being able to eliminate noise and ripples so that the Brushless DC motor can rotate and increase the starting torque. Research (6) conducted "Improved Speed Control Performance in BLDC Motors Using Fuzzy PID Controllers", the results of this study shows that using Fuzzy PIDs is suitable for high-performance motor drives.

The research discussed modeling the DC brushless motor control system using the Simulink / Matlab application and PID controllers as motor speed controllers. The difference between the two topics above is by using the trial and error method to get the best PID value.

Contribution to analyzing the potential for adaptive control purposes.

II. System Modeling

Modeling of the brushless motor control system is built on the dynamic characteristics of each component so that the right model is produced and can present the actual condition of the system. The DC brushless motor system block consists of four subsystem blocks, namely the switching control block, sensor hall block, inverter block, and DC brushless motor block. This block has two inputs and one output which is the load and the set points and pulse as the output. Modeling of the DC brushless motor system is shown in figure 1.

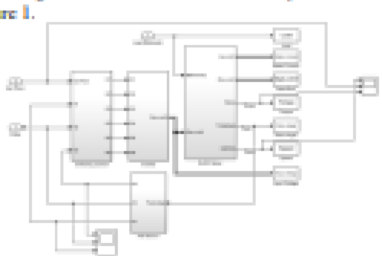


Fig. 1. Blok System Modeling

A. Brushless DC Motor

The motor modeled in this study is a three-phase DC brushless motor with a wye-connected coil and a trapezoid back EMF reverse induction voltage (7). This model of a DC brushless motor is made in electrical and mechanical equations (8). The following is the electrical equation and the parts of the BLDC motor can be shown in figure 2.

$$E_{aa}(t) = R_a . i_{aa}(t) + L_a . \frac{di_{aa}(t)}{dt} + E_{ba}(t) \quad (1)$$

$$E_{av}(t) = R_a . i_{av}(t) + L_a . \frac{di_{av}(t)}{dt} + E_{bv}(t) \quad (2)$$

$$E_{aw}(t) = R_a . i_{aw}(t) + L_a . \frac{di_{aw}(t)}{dt} + E_{bw}(t) \quad (3)$$

Where:

$E_{aa}(t), E_{av}(t), E_{aw}(t)$: Input Voltage (V)

$E_{ba}(t), E_{bv}(t), E_{bw}(t)$: Back Voltage emf (V)

$i_{aa}(t), i_{av}(t), i_{aw}(t)$: Current (A)

R_a : Resistance (Ω)

L_a : Inductance (Ω)

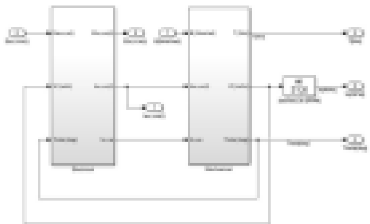


Fig. 2. DC Brushless Motor Block Model

The electric model of the DC brushless motor uses two blocks, namely the armature block and the EMF back block. The senator block contains a differential model of the electrical components of the DC brushless motor (8). While the EMF back block contains a brushless DC motor model which is a function of the rotor angle in the form of a trapezoid (9). Figure 3 below is a model of the electrical block as a whole and consists of two blocks namely the back emf block and the stator block.

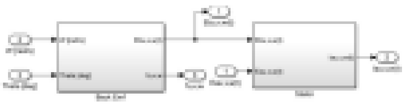


Fig. 3. Electrical Block Model

The mechanical model of the brushless DC motor contains the differential equation of the mechanical component of the motor (9), in this block there is a rotor angel block to obtain the value of the rotor angle. This rotor angle value is used as the EMF back block input which is a function of the rotor angle (10). Here's the mechanical equation.

$$T(t) = J . \frac{d\omega_m(t)}{dt} + B . \omega_m(t) \quad (4)$$

$$T(t) = T_u(t) + T_v(t) + T_w(t) \quad (5)$$

$$\omega_m(t) = \frac{d\theta_m}{dt} \quad (6)$$

$$\frac{d\omega_m(t)}{dt} = \frac{1}{J}(T(t) - B . \omega_m(t)) \quad (7)$$

$$T_{eu}(t) = K_m . i_{au} . f_u . (\theta_e) \quad (8)$$

$$T_{ev}(t) = K_m . i_{av} . f_v . (\theta_e) \quad (9)$$

$$T_{ew}(t) = K_m . i_{aw} . f_w . (\theta_e) \quad (10)$$

$$\theta_e = \frac{p}{2} . \theta_m \quad (11)$$

Where:

$T_u(t), T_v(t), T_w(t)$: Torque in Each Phase

$T(t)$: Total Torque (Nm)

J : Motor Inertia (Nms^2/rad)

B : Motor Load (Nms/rad)

$\omega_m(t)$: Motor rotation speed (rad/s)

θ_e : Electric Angle Rotor

θ_m : Rotor Mechanical Angle

The above mechanical equation is implemented in Simulink / Matlab so that it forms a mechanical block as shown in figure 4.

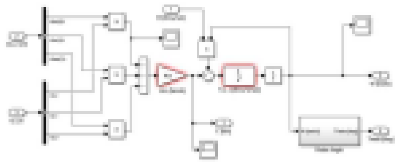


Fig. 4. Mechanical Block Model

Figure 4 contains a rotor angel block. The function of the rotor angle block is to convert the motor speed output ω (rad/s) into the position of the rotor in the form of a rotor angle. So the mechanical block output is the speed of the motor and the position of the rotor in the form of a rotor angle. Here is a model of the rotor angel block.

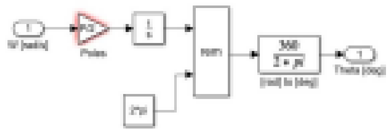


Fig. 5. Angel Rotor Block Model

The working principle of this rotor angel block is based on equation (11). The math function block serves to repeat the output value every one turn (2π rad) so that it produces the rotor angle value, then the rotor angle unit is converted from radians to degrees using the block (rad) to (deg).

B. Inverter

A three-phase inverter consists of six MOSFETs assembled in pairs to produce a three-phase voltage output. The pattern of the sixth declaration of this MOSFET is governed by the Commutation Logic block based on the Signal hall (11). Here is a three-phase inverter block model.

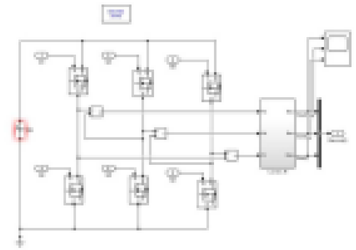


Fig. 6. Three-Phase Inverter Block Model

As seen in figure 6, the output voltage of the inverter is in the form of a line-to-line voltage, while driving the load of the motor connected to the wye requires a line to neutral voltage, therefore the LL to LN block to convert the line to line voltage into line to neutral.

C. Hall Sensors

There are three hall sensors used to detect the position of the rotor, each of which is 120 degrees apart. The hall sensor block is modeled as a function of the rotor angle (12).

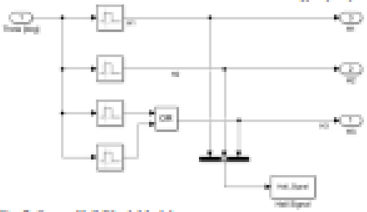


Fig. 7. Sensor Hall Block Model

TABLE I. BRUSHLESS DC MOTOR ROTOR ANGLE FUNCTION

θe	H1	H2	H3
0° - 60°	1	0	1
60° - 120°	1	0	0
120° - 180°	1	1	0
180° - 240°	0	1	0
240° -300°	0	1	1
300° - 360°	0	0	1

D. Switching Control

In the switching control block, there are two subsystem blocks, namely the PWM Generator block and the Commutation Logic block. The PWM Generator block serves as a PWM signal generator that will be used to generate a voltage in the inverter (13). The commutation logic block serves to set the commutation pattern or inverter declaring pattern. The commutation method used to drive the brushless DC motor is a six-step commutation method where commutation is carried out by activating six MOSFETs on the inverter alternately in six stages to produce one electrical cycle.

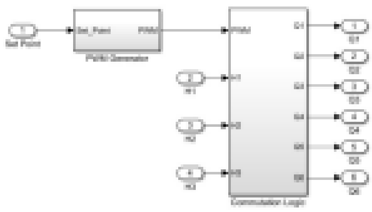


Fig. 8. Block Switching Control Model

E. PWM Generator

In the control of a brushless DC motor, the speed regulation of the motor is carried out by changing the motor input voltage large. The input voltage of this brushless DC motor can be changed by the PWM method where this PWM signal serves to modulate the voltage from the inverter to the brushless DC motor (13). The amount of voltage generated will depend on the duty cycle of the PWM signal. Here is the PWM generator block model.

Fig. 9. Block Model PWM Generator

F. Commutation Logic

The logic commutation block has four inputs namely PWM and three hall sensors, and six outputs which are MOSFET control signals. In this block, the PWM signal will be activated on each control signal periodically based on the hall sensor received on the input port. The following is a block model of logic commutation and commutation pattern of a DC brushless motor.

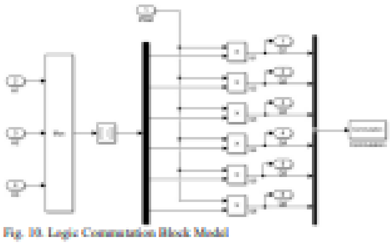


Fig. 10. Logic Commutation Block Model

TABLE II. COMMUTATION PATTERN OF DC BRUSHLESS MOTOR

Hall Sensor			Controller Output					
H1	H2	H3	Q1	Q2	Q3	Q4	Q5	Q6
1	0	1	0	0	0	1	1	1
1	0	0	0	0	1	1	1	0
1	1	0	0	1	1	1	0	0
0	1	0	1	1	1	0	0	0
0	1	1	1	1	0	0	0	1
0	0	1	1	0	0	0	1	1

The commutation pattern specified in table II can be seen as an example, of the commutation pattern I when the H1 and

H3 sensors are high and H2 low. The switched high switches are Q4, Q5, and Q6. Likewise, the other switches remain low.

G. PID Control

PID control serves as process control in the industry reaching 90%, due to its simplicity and good performance (14). This study used a parallel PID control system. The equation is the output of parallel PID control.

$$u(t) = K_p * e(t) + K_i * \int e(t)\, dt + K_d * \frac{de(t)}{dt} \qquad (12)$$

If the plant model can be obtained, then the PID parameters can be determined through mathematical calculations, but in the field, the model is very difficult to obtain because of its complexity. This study uses the heuristic method (trial and error). The steps taken are tuning the value K_p, after the K_p value is close to the set point value than combining the values K_p, K_i, K_d, until getting the right value for the parameter. In figure 10 this is the PID control created.

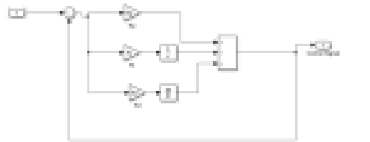


Fig. 11. PID Controler Block Model

III. Simulation Results

Control system testing is carried out in the form of model simulations on Matlab/Simulink. The test is carried out in two stages, namely, testing with load variations and testing with speed variations.

A. Load Variation Testing

The purpose of this test is to find the value of the best PID controller parameter. The results of this test in the form of a comparison of BLDC motors without PID controllers and with PID controllers are shown in figures 12 and 13.

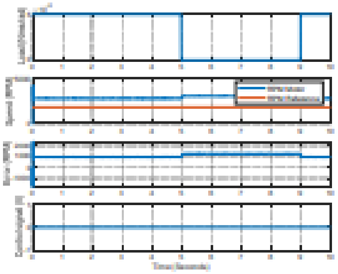


Fig. 12. Simulated Load Variation Testing Without PID

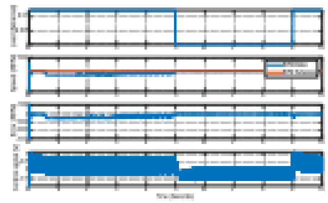


Fig. 13. Simulated Load Variation Testing Using PID

It can be seen in the figure of 11 motor speeds that the simulation results are different from the reference speed given so it produces an error. As seen in figure 12, there is effort generated by the controller so that there is a significant change in the motor speed response. The speed of the simulated motor managed to reach the given reference speed of 3000 rpm.

B. Speed Variation Testing Of DC Brushless Motors

In this test, the simulation is run for 10 seconds. This variation in speed is made by raising and lowering the speed. So the given speed variation is 3000 rpm and 5000 rpm. The purpose of this test is to see the effect of the PID controller on the response of the brushless DC motor under varying speed conditions. Figure 13, and 14,15,16 is the result of simulated testing of speed variation of a brushless DC motor with PID or without PID.

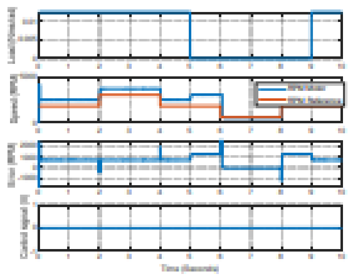


Fig. 14. Simulation Of 3000 rpm Speed Without Using PID

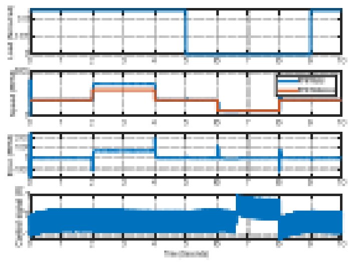


Fig. 15. Simulation Of 3000 rpm Speed Using PID

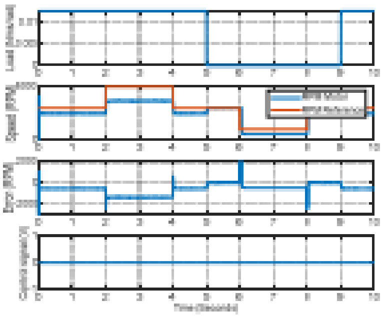


Fig.16. Simulation Of 5000 rpm Speed Without Using PID

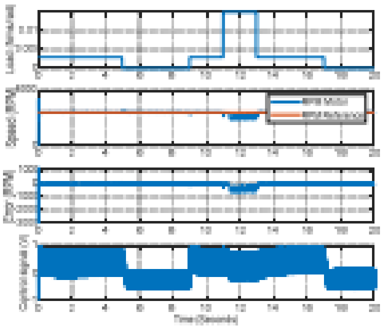


Fig. 17. Simulation of 5000 rpm Speed Using PID

From the simulation results, it can be seen that the speed of the simulation results without PID cannot keep up with the change in the given speed reference. After using the PID controller there was a significant change in the result of the speed of the BLDC motor. The speed of the simulation results can follow the change in the speed reference with only a slight overshoot occurring in each transition of the speed value.

IV. CONCLUSION

After researching the evaluation of the DC brushless motor system, a model of the DC brushless motor system has been produced using the six-step commutation method and by using the structure PID controller as the speed controller. The goal is to produce a good starting start for the bike and avoid any beats.

The best K_p, K_i, and K_d values obtained using the trial and error method are $K_p = 1.1$, $K_i = 0.2$, dan $K_d = 0.8$. The response to distractions gets better and the response to speed variations gets better. In addition, the speed of the simulation results can follow the change in the speed reference with an overshoot in the transition of the speed value.

When the controller is applied to the system, the test starts the resulting system response with a settling time value of 0.0035 s and a large overshoot with a value of 1.9%. In addition, using a PID controller can line the BLDC motor.

ACKNOWLEDGMENT

The author would like to thank the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia for supporting our research with the Kedaireka matching fund scheme with grant contract number 490/E1/KS.06.02/2022.

REFERENCES

[1] P. Sugandhi, S. Nagapavithra and S. Umamaheswari, "Modeling and Simulation of Closed Loop Speed Control for BLDC Motor," Conference on Emerging Device and Smart Systems (ICEDSS), Tiruchengode, pp. 229-233, 2017.

[2] M. Mahmud, S. M. A. Motakabber, A. H. M. Z. Alam, A. N. Nordin, "Control BLDC Motor Speed Using PID Controller," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11 (3), pp. 477-481, 2020.

[3] J. C. Gamazo, E. Vazquez dan J. Gomez, "Position and Speed Controlof Brushless DC Motor Using sensorless Techniques and Application Trends," Sensors, pp. 6901-6947, 2010.

[4] M. Jain and S. S. Williamson, "Suitability Analysis of In-Wheel Motor Direct Drives for Electric and Hybrid Electric Vehicles," 2009 IEEE Electrical Power & Energy Conference (EPEC), pp. 1-5, 2009.

[5] J. D. Irwin, Power Electronics Handbook, Florida: Academic Press, 2001.

[6] Kamil Plachta, "A new control method of brushless DC motor to maximize starting torque" Wroclaw University of Technology., IEEE,2016.

[7] K. Sarojini, R. Dhanasekaran, S. Muthulakshmi, "Improvement of Speed Control Performance in BLDC Motor Using Fuzzy PID Controller," ICACCCT. International Conference on Advance Communication Control and Computing Technologies. 2016.

[8] V. R. Walker and S. V. Markute, "Speed Control of BLDC Motor using PI & Fuzzy Approach: A Comparative Study," International Conference on Information, Communication, Engineering and Technology (ICICET), Pune, pp.1-4, 2018.

[9] A. Tashakori, M. Ektesabi dan N. Hosseinzadeh, "Modeling of BLDCmotor with Ideal Back-EMF for Automotive Applications," dalam World Congress on Engineering, London, 2011.

[10] M.-F. Tsai, T. P. Quy, B.-F. Wu dan C.-S. Tseng, "Model construction and verification of a BLDC Motor Using MATLAB/SIMULINK and FPGA Control," 6^{th} *IEEE Conference on Industrial Electronicsand Applications*, Beijing, 2011.

[11] O. Jhon, "PID Control of Brushless DC Motor and Robot Trajectory Planning and Simulation with Matlab/Simulink." Technology and Communication, University of Applied Sciences, 2009

[12] T.C. Siong, B. Ismail, S.F. Siraj, M.F. Mohammed, Fuzzy logic controller for BLDCpermanent magnet motor drives, Int. J. Electr. Comput. Sci. 11 (2) (2011) 12–17.

[13] K. Kolano, "Improved sensor control method for BLDC motors," IEEE Access, vol. 7, pp.186158-186166, 2019.

[14] M. Baszynski and S. Pirog, "A novel speed measurement method for a highspeed BLDC motor based on the signals from the rotor position sensor," IEEE Trans. Ind. Informatics, vol. 10, n0. 1 pp. 84-91, 2014.

Lampiran 10 Coding PSO

```
clc

% Inisialisasi nilai awal
populationSize = 50;
nGen = 31;
lb = [0 0 0];
ub = [1 1 1];
population = lb + (ub - lb) .* rand(populationSize, nGen);
maxGenerations = 5;
bestError = zeros(maxGenerations, 1);
error = zeros(populationSize, 1);

generation = 1;
while generation <= maxGenerations
    % Perhitungan nilai fitness
    for i = 1:populationSize
        error(i) = calculateError(population(i, :));
    end

    % Seleksi parents menggunakan seleksi turnamen
    parentIndex = selectionTournament(error, 18, populationSize);
    parents = population(parentIndex, :);

    % Operasi crossover titik potong tunggal
    offspring = crossover(parents, parentIndex, nGen);

    % Operasi mutasi
    mutationProbability = 0.1;
    mutatedOffspring = mutation(offspring, mutationProbability);

    % Evaluasi error/performa individu/kromosom hasil reproduksi
    offspringError = zeros(size(offspring, 1), 1);
    for i = 1:size(offspring, 1)
        offspringError(i) = calculateError(mutatedOffspring(i, :));
    end

    % Penggantian populasi
    [population, error] = replace(population, error, parentIndex,
offspring, offspringError, populationSize);

    % Pencatatan nilai error terbaik setiap generasi
    bestError(generation) = min(error);

    % Tampilkan hasil setiap generasi
    disp(['Generation: ', num2str(generation), ' | Best Error: ',
num2str(bestError(generation))]);

    generation = generation + 1;
end

% Hasil optimasi
[minError, minIndex] = min(error);
bestParameters = population(minIndex, :);
disp('Optimization Results:');
disp(['Best Parameters: ', num2str(bestParameters)]);
```

```
disp(['Best Error: ', num2str(minError)]);

% Fungsi untuk menghitung performa sistem kontrol PID pada SEPIC
converter
function error = calculateError(k)
    assignin('base', 'k', k);
    % Kode simulasi SEPIC converter
    sim("PlantforGAtesting.slx");
    % Fungsi Objektif
    error = ISE(length(ISE));
end

% Fungsi-fungsi bantu
function parentIndex = selectionTournament(error, m, populationSize)
    parentIndex = zeros(m, 1);
    for i = 1:m
        candidates = randperm(populationSize, 2);
        if error(candidates(1)) <= error(candidates(2))
            parentIndex(i) = candidates(1);
        else
            parentIndex(i) = candidates(2);
        end
    end
end

function offspring = crossover(parents, parentIndex, nGen)
    n = numel(parentIndex);
    offspringNum = n/2;
    crossoverPoint = randi([1, nGen-1]);
    offspring = zeros(offspringNum, nGen);
    for i = 1:offspringNum
        candidates = randperm(n, 2);
        offspring(i, :) = [parents(candidates(1), 1:crossoverPoint),
parents(candidates(2), crossoverPoint+1:end)];
    end
end

function mutatedOffspring = mutation(offspring, mutationProbability)
    mutatedOffspring = offspring;
    n = numel(mutatedOffspring);
    ranNum = 0.01*rand(n,1);
    for i = 1:n
        if ranNum < mutationProbability
            mutatedOffspring(i) = ranNum(i);
        end
    end
end

function [newPopulation, newError] = replace(population, error,
parentIndex, offspring, offspringError, populationSize)
    newPopulation = population;
    newError = error;
    m = numel(parentIndex)/2;
    sorting = sort(newError, 'descend');
    nMaxError = sorting(m);
    maxErrorIndices = zeros(m,1);
    for i = 1:populationSize
```

```
        if newError(i) >= nMaxError
            maxErrorIndices(i) = i;
        end
    end
    maxErrorIndices = maxErrorIndices(maxErrorIndices ~= 0);
    for a = 1:m
        if newError(maxErrorIndices(a)) >= offspringError(a)
            newPopulation(maxErrorIndices(a), :) = offspring(a, :);
            newError(maxErrorIndices(a)) = offspringError(a);
        end
    end
end
```

Lampiran 11 Coding *Trial and Error*

```
clear;
clc;
%MOTOR PARAMETER
Ra = 2.3; % Tahanan Jangkar [Ohm]
La = 0.00768; % Induktansi jangkar [Henry]
J = 0.00035; % Inersia [Nms2/rad]
Kb = 0.7452; % Konstanta generator [Vs/rad]
Km = 0.74; % Konstanta motor [Nm/A]
B = 0.002; % Konstanta beban (frction) [Nms/rad]
Vdc = 24; % Tegangan inverter [Volt]
Max_RPM = 3000; % RPM motor
Max_freq = Max_RPM/60; % Frekueansi maksimal [Hz]
P = 4; % Jumlah kutub

%CONTROLLER PARAMETER
Kpvar = 0:0.1:3;
Ki = 0;
Kd = 0;
M = length(Kpvar);
RMSEakhir = zeros(1,M);
RMSUakhir = zeros(1,M);
j = zeros(1,M);
Kpsave = zeros(1,M);

% Tref = 2000*Ts/(P/2);
% Dref = 50;
Vref = 5; % Tegangan referensi (maksimal) potensio (V)
Set_Point = 3; % Tegangan input potensio [V]
K = (Max_freq/Vref)*(P/2); % Gain Frekuensi pada frequency generator
[Hz/V]
RPM_ref = (Set_Point*K*60)/(P/2); % Referensi kecepatan
ErrorMax = 700;
UsahaMax = 2;
p = 1/ErrorMax;
q = 1/UsahaMax;

%MOTOR SIMULATION
for k = 1:M
    Kp = Kpvar(k);
    SamplingTime = 1e-6;
    Ts = 10*SamplingTime;
    SimulationTime = 10;
    sim('Pencari_Nilai_Kp.slx')
    N = length(Time);
    T = zeros(1,N);
    RPM = zeros(1,N);
    RPM_Total = zeros(1,N);
    RPM_rata2 = zeros(1,N);
    Error = zeros(1,N);
    Error_kuadrat = zeros(1,N);
    RMSE = zeros(1,N);
    ControlSignal_kuadrat = zeros(1,N);
    RMSU = zeros(1,N);
    T(1) = Time(1);
    RPM(1)= 2.0890e+03;
```

```
    RMSE(1)=0;
    Error(1) = 0;
    Error_kuadrat(1) = 0;
    RMSU(1) = 0;
    Usaha(1) = 0;
    ControlSignal_kuadrat(1) = 0;
    RPM_Total(1) = 0;
    for I = 2:N
        T(I) = Time(I);
            if (Pulse(I)>0.5)&&(Pulse(I-1)<=0.5)
                PERIODE = T(I)-T(I-1);
                RPM(I)= (60/PERIODE)/(P/2);
            else
                RPM(I) = RPM(I-1);
                T(I) = T(I-1);
            end
        Error(I) = RPM(I) - RPM_ref;
        Error_kuadrat(I) = Error_kuadrat(I-1) + Error(I)^2;
        RMSE(I) = sqrt(abs((Error_kuadrat(I)/I)));
        ControlSignal_kuadrat(I) = ControlSignal_kuadrat(I-1) +
ControlSignal(I)^2;
        RMSU(I) = sqrt(abs((ControlSignal_kuadrat(I)/I)));
        RPM_Total(I) = (RPM_Total(I-1)+RPM(I));
        RPM_rata2(I) = RPM_Total(I)/I;
    end
    RPMakhir = RPM_rata2(I);
    RMSEakhir(k) = RMSE(I);
    RMSUakhir(k) = RMSU(I);
    j(k) = sqrt((((p*RMSEakhir(k)).^2))+(((q*RMSUakhir(k)).^2)));
    Kpsave(k) = Kp;
end
```