PI-Based SRM Speed Control System

Ilham Akbar
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
ilham.akbar912@gmail.com

Nanang Ismail
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
nanang.is@uinsgd.ac.id

Eki Ahmad Zaki Hamidi
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
ekiahmadzaki@uinsgd.ac.id

Diky Zakaria
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
dikyzak@students.itb.ac.id

2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA) | 978-1-7281-7523-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSIMA50015.2021.9526293

***Abstract*— There are three types of motors used in electric vehicles currently, namely permanent magnet synchronous motors, induction motors, and Switched Reluctance Motors (SRM). SRM has its advantages over other types of motorcycles. The advantages of SRM are weight, reliability, fault tolerance, and acceleration time. However, the SRM motor needs to be adjusted to have the required speed and transient response. This study aims to simulate and design speed control in SRM. The Proportional Integral (PI) control simulation is firstly conducted by using MATLAB to see the results of the control before it is implemented in SRM. From the results of the SysId Matlab, it is obtained system transfer function in the first order. The transfer function of this system is used to determine the value of Kp and Ki analytically obtained the value of Kp is 0.006159 and Ki is 0.054752. SRM operates at PWM 72-84 or in a duty cycle of 28.23% - 32.94%. This PWM signal is equivalent to the driver input voltage of 1.446 Volts - 1.756 Volts. The implementation is conducted by using an Arduino microcontroller and it only uses a positive control signal that makes the motor rotate in only one direction. From the simulation and implementation that were carried out, it is found that the control system that has been designed is able to track the speed at the setpoint, namely 400 rpm, 600 rpm, and 800 rpm. The settling time, respectively, is 1.2 s, 2 s, and 1.3 s, overshoot is 52.75%, 62.16%, and 13.87%. The steady-state error for each setpoint is 10%, 0.33%, 0.5%.**

Keywords—SRM, speed control, RPM, PI

I. Introduction

Switched Reluctance Motor (SRM) is becoming the main choice for industrial needs, such as for use in electric vehicles. SRM is not recently existed, since 150 years ago it has been used. In 1842, the SRM was used for the first time as a locomotive drive. SRM is often used in various fields such as aerospace, mining and industrial needs, household appliances, etc.[1] [2].

In recent years, SRM control system is more reliable than other control systems using other machines because it has obvious fault tolerance[3]. SRM does not depend on characteristics of motor magnetic phase and electric inverter circuit phase. So, when an error occurs in one phase it will not affect the other phases. This occurs in other engine drives such as induction motor drive systems which do not have such tolerance characteristics[4][5].

In the use of SRM for industrial needs, a certain speed is often demanded, so a control system is needed. Various control methods can be used, one of which is the Proportional Integral (PI) control method. PI control is a type of control that is widely used in control engineering. PI control is a combination of two types of control, namely Proportional and Integral[6] [7]. Previously, the paper [4] described the different types of errors in the SRM driver. In addition, this paper emphasized the study of faults in open switches in inverter circuits so that a fault tolerance topology can be applied to minimize torque and speed ripples. This paper also has provided a comparison of the interference behavior of the SRM driver that has a fuzzy controller and a PID controller. Meanwhile, research [8] discussed the characteristics and parameters to analyze the performance of SRM and simulation of SRM driver models using Simulink. This paper considers hardware prototype developers because they have advantages in modeling and simulation.

In speed settings, in addition to being targeted to achieve setting points, it also requires rotational stability to the reference value and wide speed range, and ability to be implemented in a variety of applications. Therefore, data is needed to compare the response value of the closed-loop motor speed control system using the PI control and the open-loop control system[9] [10]. Motor speed control is regulated using Pulse Width Modulation (PWM), the PWM duty cycle value can change based on the calculation of the PI control signal. By using this PI control, it is expected that the use of the motor will be more efficient[6] [8] [11].

Based on the description of the background stated, in this study, a simulation and design of speed regulation on SRM using a PI controller were carried out through a linear approach with SRM modeling at the one operating point. The simulation was carried out using Matlab, and the implementation of PI control was carried out using Arduino as a microcontroller.

II. Controller Design

A. SRM Modelling

An illustration of the SRM equivalent circuit can be identified in Fig.1. The SRM equivalent circuit can be revealed by ignoring the mutual inductance between the phases. The voltage applied to a phase is proportional to the sum of the resistive voltage drops and the rate of flux linkages, which is obtained by[12]

978-1-7281-7523-2/21/$31.00 ©2021 IEEE

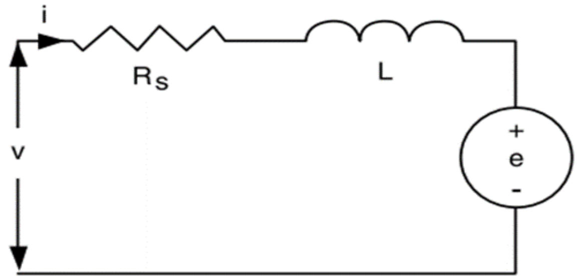


Fig. 1. Equivalent circuit of single-phase SRM

$$V = R_s i + \frac{d\lambda(\theta,i)}{dt}, \tag{1}$$

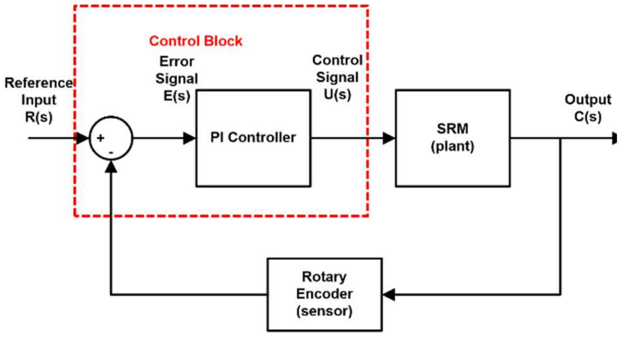


Fig. 2. Block diagram of control system.

where R_s is the resistance of each phase of the SRM and λ is the flux linkage of each phase which can be expressed by

$$\lambda = L(\theta, i)i. \tag{2}$$

L is the inductance based on the rotor position and the phase flow. The SRM phase voltage equation can be calculated by the equation

$$\begin{aligned} V &= R_s i + \frac{d\{L(\theta,i)i\}}{dt} \qquad (3) \\ &= R_s i + L(\theta,i)\frac{di}{dt} + i\frac{d\theta}{dt}.\frac{dL(\theta,i)}{d\theta} \\ &= R_s i + L(\theta,i)\frac{di}{dt} + \frac{dL(\theta,i)}{d\theta}\omega_m i. \end{aligned}$$

In Equation (3), the three terms on the right side represent resistive voltage drop, inductive voltage drop, and emf. The result is the same as the voltage equation for a series dc motor.

Emf e is obtained by equation

$$e = \frac{dL(\theta,i)}{d\theta}\omega_m i = K_b \omega_m i. \tag{4}$$

The variable K_b can be interpreted as the same constant *emf* as the *dc series excited machine*, i.e.

$$K_b = \frac{dL(\theta,i)}{d\theta}. \tag{5}$$

The electromechanical torque T can be written as

$$T = J\frac{d\omega}{dt} + D\omega + T_L, \tag{6}$$

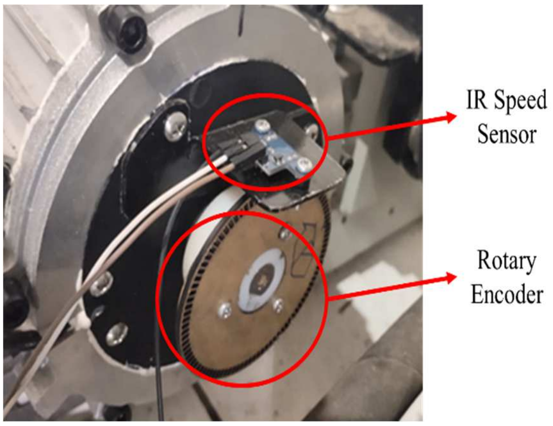


Fig. 1. Implementation IR speed sensor and Rotary Encoder.

where J is the total rotor and load inertia, ω is the rotor angular speed, D is the coefficient of viscous friction, and T_L is the load torque.

B. Block Diagram System

The general block diagram is shown in Fig.2. SRM speed detection is reversed to compare with the reference value-added. The difference between the reference value and the detected speed is identified as an error signal. This error will determine how much control signal is needed to track the given set point.

C. Speed Sensor Design

In order to digitally measure speed, it is able to choose from several types of sensors such as Hall effect, proximity IR speed sensor, and rotary encoder. This speed sensor is designed to be able to read the speed that can represent the transient conditions when the motor is moving. The implementation of the IR sensor and rotary encoder can be identified in Fig.3.

D. Non Linearity Input of SRM

The PWM signal which has a frequency of 490 Hz cannot be directly supplied to the driver. When tested by giving a PWM signal directly to the SRM driver, it causes the SRM rotation to oscillate. When the LPF (Low Pass Filter) output is given to the SRM driver, the SRM can rotate properly without oscillation. After testing the provision of PWM input steps, the data obtained that SRM only responds to PWM 72 – 84 or a duty cycle of 28.23% – 32.94%. This PWM signal is equivalent to the driver input voltage of 1.446 V – 1.756 V.

E. SRM Transfer Function

SRM is a non-linear system, while the proposed PI method is linear, so that the designed controller can be used, the operating speed area is limited. In this study, the determination of the mathematical model of the plant cannot be determined by direct parameters. This is because the SRM datasheet used is not equipped with complete parameters such as motor inductance, motor resistance, motor inertia. Therefore, mathematically modeling SRM is conducted by looking at the input and output relationships using SysId Matlab, i.e. by entering SRM response data when assigned a specific PWM value. In this case, a PWM of 75 (1.524 Volts) is assigned to the SRM driver or select an operating point at a PWM of 75 (approximately 680 rpm) as the model of SRM. The system response obtained can be seen in Fig.4 and the results of the SRM transfer function are shown in Fig.5.

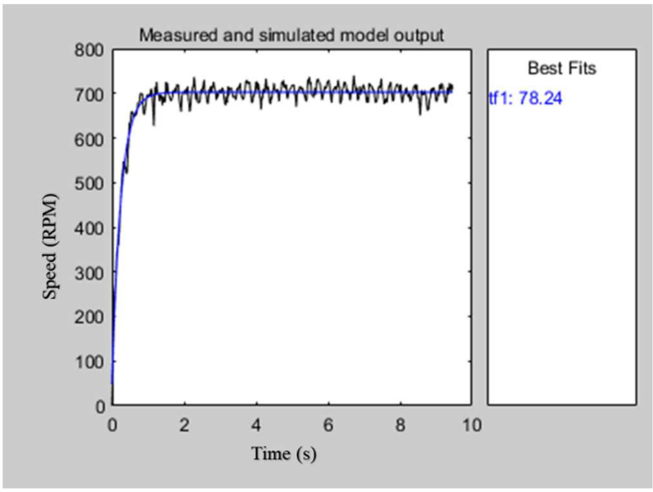
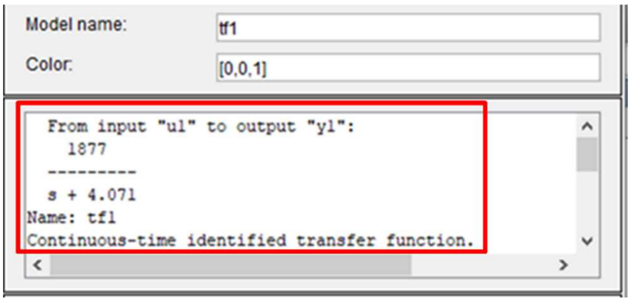


Fig. 4. SRM Response at PWM 75



```
Model name: tf1
Color: [0,0,1]

 From input "u1" to output "y1":
   1877
 ---------
 s + 4.071
Name: tf1
Continuous-time identified transfer function.
```

Fig. 5. The result of SRM transfer function

The obtained SRM transfer function is

$$SRM_{tf} = \frac{C(s)}{U(s)} = \frac{1877}{s+4{,}071} = \frac{461{,}066}{0{,}24\,s+1} \leftrightarrow \frac{K}{\tau s+1}, \tag{7}$$

where τ is the time constant.

F. PI Controller and Value of Kp and Ki

PI controllers are the most commonly used controllers in feedback control systems. Sometimes not all proportional (P), integral (I), and derivative (D) components are used simultaneously. The variations that are widely used are a combination of each component. The integral proportional controller equation in time response can be written as

$$u(t) = Kp.e(t) + Ki \int_0^t e(t)dt, \tag{8}$$

where *Kp* is the proportional constant, *Ki* is the integral constant, *e(t)* is the error signal, and *u(t)* is the control signal.

Laplace's transformation from the Equation (8) shows the switching function of a proportional plus integral controller as shown in the block diagram of Fig. 6.

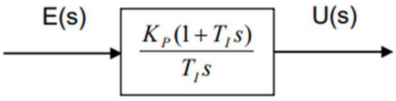


Fig. 6. Proportional-integral scheme.

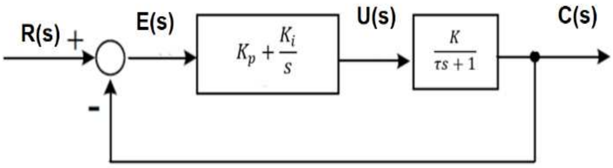


Fig. 7. Control structure diagram

The determination of *Kp* and *Ki* values is conducted analytically using a closed-loop system that can be seen in Fig.7, then approached with the general equation of the closed-loop system of second-order

$$\frac{C(s)}{R(s)} = \frac{{\omega_n}^2}{s^2 + 2\omega_n\xi s + {\omega_n}^2}, \tag{9}$$

where ω_n is the undamped natural frequency and ξ is the damping ratio. In the open-loop PI, obtained

$$G(s) = \left(K_p + \frac{K_i}{s}\right)\frac{K}{\tau s+1} = \frac{K_pKs+K_iK}{\tau s^2+s}, \tag{10}$$

and in the close-loop PI obtained:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} = \frac{\frac{K_pKs + K_iK}{\tau s^2 + s}}{1 + \frac{K_pKs + K_iK}{\tau s^2 + s}} \left(\frac{\tau s^2 + s}{\tau s^2 + s}\right)$$

$$= \frac{K_pKs + K_iK}{s^2 + \frac{(K_pK+1)s}{\tau} + \frac{K_iK}{\tau}}. \tag{11}$$

Based on Equation (9) and (11), obtained:

$$2\omega_n\xi = \frac{K_pK+1}{\tau} \tag{12}$$

$${\omega_n}^2 = \frac{K_iK}{\tau}. \tag{13}$$

By including the overshoot criteria of 2% and settling time of 0.5 seconds, the values of *Kp* and *Ki* can be obtained. Parameters *Kp* and *Ki* are listed in Table I.

TABLE I. PARAMETER RESULTS

Parameter	Value
SRM_{tf}	$\frac{461.066}{0.24s+1}$
τ (s)	0.24
Settling Time (s)	0.5
K_p	0.006159
K_i	0.054752

G. Simulation Results

The simulation was carried out to see the response of the system before implementation. The simulation was conducted on a continuous-time system. The *Kp* and *Ki* values that have been obtained in the design process are added in Simulink Matlab which is shown in Fig.8.

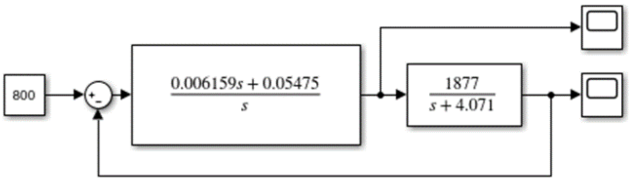
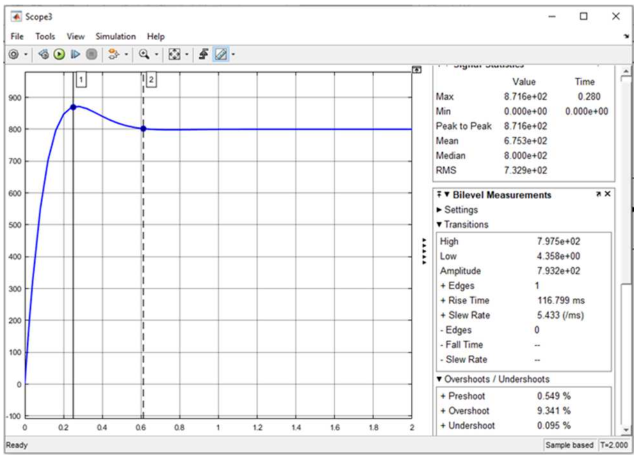


Fig. 8. Simulation of PI control



	Value	Time
Max	8.716e+02	0.280
Min	0.000e+00	0.000e+00
Peak to Peak	8.716e+02	
Mean	6.753e+02	
Median	8.000e+02	
RMS	7.329e+02	

▼ Bilevel Measurements

► Settings

▼ Transitions

High	7.975e+02
Low	4.358e+00
Amplitude	7.932e+02
+ Edges	1
+ Rise Time	116.799 ms
+ Slew Rate	5.433 (/ms)
- Edges	0
- Fall Time	--
- Slew Rate	--

▼ Overshoots / Undershoots

+ Preshoot	0.549 %
+ Overshoot	9.341 %
+ Undershoot	0.095 %

Sample based T=2.000

Fig. 9. System Respone of PI control

Based on the control simulation on the continuous-time system shown in Fig.9, it can be seen that the settling time value is in the range of 0.6 s, the overshoot is 9.35%, and the steady-state error is 0%. The steady state error value is in accordance with the desired specifications. The value of settling time has a difference of 0.1 s with the demanded value. While the overshoot has a difference of 7.35% from the demanded value, which is 2%.

H. Implementation

The implementation was carried out using an Arduino UNO microcontroller connected to an IR speed sensor to read the speed of the motor rotation. The voltage source for the LPF used is 9V which is connected to IC LM358 as a power source and connected to the reference input on the microcontroller. In addition, the Arduino pin is also connected to the LPF as a digital to analog converter (DAC). The resistor used in the LPF is 3kΩ and the capacitor is 1μF. The SRM is connected to the fluid pump as a mechanical load. This speed control implementation has not implemented load regulation. The overall implementation is identified in Fig.10.

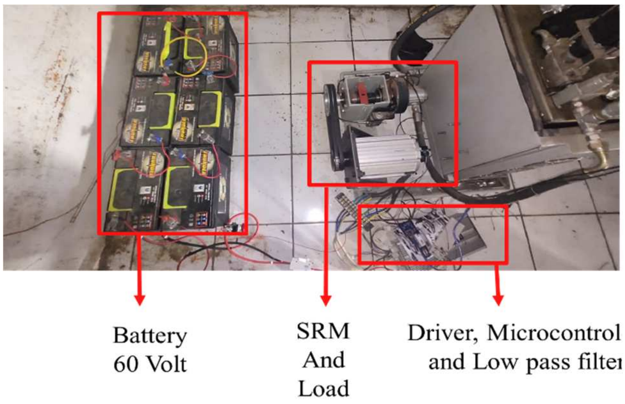


Fig. 10. Overall Implementation

III. Results and Analysis

A. Speed Sensor Testing

Speed sensor testing was carried out to see whether the sensor is feasible or not. In addition, in the design of a feedback control system that feeds the output, which output, in this case, is the speed of the motor, then the motor speed reading must be good. A good motor speed reading will give a more precise calculation of control signals as well. The speed sensor is tested by performing a speed reading with a sampling of 20ms. Testing was conducted by reading the speed of the sensor and compared to the measurement on the tachometer. Error is calculated using the formula in Equation

$$e_m = \left| \frac{RPM_{tacho} - RPM_{encoder}}{RPM_{encoder}} \right| x\ 100\% \qquad (14)$$

Table II shows the speed measurements in each PWM value.

B. PI Controller Results

SRM is a non-linear system, while the proposed PI method is linear, so that the designed controller can be used, the operating speed area is limited. In the test, setting points were given based on three areas of 400 rpm, 600 rpm, and 800 rpm.

Based on Fig.11, generally, it can be seen that the design of the PI control in SRM is able to track changing setpoints.

TABLE I. Speed Sensor Testing

No	PWM	Duty Cycle (%)	Rotary Encoder (RPM)	Tacho meter (RPM)	Measurement Error (%)
1	72	28.23	385	385	0
2	73	28.63	385	387	0.517
3	74	29.02	405	406	0.246
4	75	29.41	680	688	1.163
5	76	29.8	785	790	0.633
6	77	30.2	790	794	0.504
7	78	30.59	801	797	0.502
8	79	30.98	805	811	0.74
9	80	31.37	887	889	0.225
10	81	31.76	946	944	0.212
11	82	32.16	958	965	0.725
12	83	32.55	1008	1015	0.69
13	84	32.94	1132	1127	0.444
The average of error (%)					**0.508**

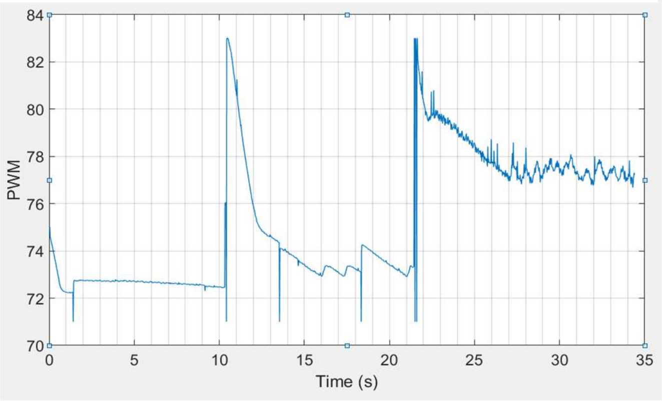


Fig. 11. The signal of PI Control.

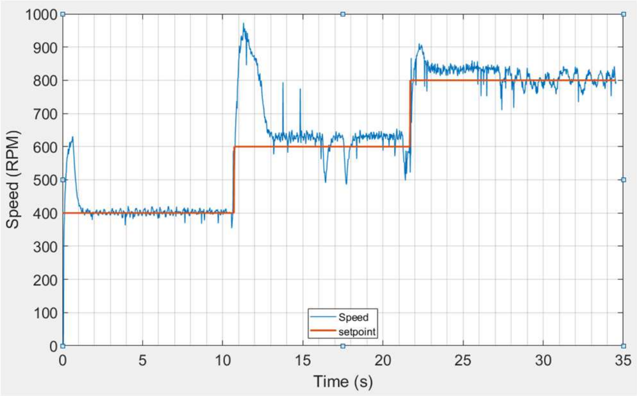


Fig. 12. System response with changing setpoints.

Changes in the control signal can be seen in Fig.12. In the test results, there are overshoots at several setpoints that are far from the specified operating point. At setpoint 400, the system gives a response similar to the simulation results, there is an overshoot and steady-state at 400. At setpoint 600 and 800, the system still has a high overshoot and has a steady-state error. In addition, at several points, jump spikes are also seen in steady-state conditions which can be seen at the setpoint of 600. To determine the performance of the PI controller against a given setpoint, an analysis was carried out per setpoint area.

At 400 rpm, it takes 1.2s to reach a steady state. When compared with the demanded specification, which is 0.5 s, there is a difference of 0.7 s. In addition, there is a steady-state error of 10%. At set point 400, there is a fairly large overshoot of 57.75%, where the peak point of the recording results is 631 rpm. In the 600 rpm area, it takes 2 s to reach a steady state. When compared with the demanded specification, which is 0.5 s, there is a difference of 1.5 s. In addition, there is a steady-state error of 3.33%. At set point 600, there is a fairly large overshoot of 62.16%, where the peak point of the recording is 973 rpm. Finally, in the 800 rpm area, to reach a steady state response, it takes 1.3 s. When compared with the demanded specification, which is 0.5 s, there is a difference of 0.8 s. In addition, there is a steady-state error of 0.5%. At set point 800, there is a fairly large overshoot of 13.87%, where the peak point of the recording results is 911 rpm.

IV. Conclusions

Speed setting uses PWM signals in the value range of 72 – 84 or in a duty cycle of 28.23% - 32.94%. The relationship between PWM and SRM speed is nonlinear. In this study, a linear approach was used, by choosing around the operating point of 680 RPM as the control model. The given control signal only uses a positive control signal or without braking effect. The system designed is implemented and it is able to track the given setpoints, which are 400 RPM, 600 RPM, and 800 RPM. In the test, the graph of the PI system response from SRM was obtained. Each speed setting is able to provide a fairly good response even though it still has a fairly high overshoot, especially at the 600 RPM set point. In the area of 400 RPM, 600 RPM, and 800 RPM respectively, it is obtained the settling time of 1.2s, 2s, and 1.3s, the overshoot was 52.75%, 62.16%, and 13.87%. Steady-state error for each set-point is 10%, 0.33%, 0.5%.

References

[1] T. Wasita Febriandi and S. Riyadi, “Pengaturan Kecepatan Motor Switched Reluctance Dengan Konverter Asymmetric Pada Mode Magnetizing Dan Demagnetizing,” *SNIKO*, pp. 269–276, 2018, doi: 10.5614/sniko.2018.32.

[2] A. W. Aditya, Ihsan, R. M. Utomo, and Hilmansyah, “Evaluasi Motor Listrik Sebagai Penggerak Mobil Listrik,” *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 3, no. 2, p. 49, 2019, doi: 10.30595/jrst.v3i2.4142.

[3] M. Chaple and S. B. Bodkhe, “The simulation and mathematical modeling of switched reluctance motor based on phase winding inductance,” *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 3048–3052, 2018, doi: 10.1109/ICECDS.2017.8390015.

[4] N. Saha, S. Panda, D. S. Choudhury, S. Rath, and S. K. Panda, “Comparison on behavior of different faults in fuzzy logic and PID controlled switched reluctance motor drives,” *2nd Int. Conf. Commun. Control Intell. Syst. CCIS 2016*, pp. 165–169, 2016, doi: 10.1109/CCIntelS.2016.7878222.

[5] A. P. Khedkar and P. S. Swami, “Comparative study of asymmetric bridge and split AC supply converter for switched reluctance motor,” *6th Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2017*, vol. 2018-Janua, pp. 522–526, 2018, doi: 10.1109/ICCPEIC.2017.8290421.

[6] J. Song, S. Song, and B. Qu, “Application of an Adaptive PI Control for a Swtiched Reluctance Motor,” *2016 IEEE 2nd Annu. South. Power Electron. Conf.*, pp. 1–5, 2016, doi: https://doi.org/10.1109/SPEC.2016.7846009.

[7] P. Charles L and H. Royce D, *Feedback control systems*. Britainia Raya: Prentice Hall, 1991.

[8] K. R. Chichate, S. R. Gore, and A. Zadey, “Modelling and Simulation of Switched Reluctance Motor for Speed Control Applications,” *ICMIA 2020*, no. Icimia, pp. 637–640, 2020, doi: 10.1109/ICIMIA48430.2020.9074845.

[9] K. Ogata, *Modern Control Engineering*, 5th ed. New Jersey: Prentice Hall, 2002.

[10] Rosalina, I. Qosim, and M. Mujirudin, “Analisis Pengaturan Kecepatan Motor DC Menggunakan Kontrol PID (Proportional Integral Derivative),” *Teknoka*, vol. 2, pp. 89–94, 2017.

[11] X. Deng, O. Ma, and P. Xu, “Sensorless Control of a Four Phase Switched Reluctance Motor Using Pulse Injection,” *Proc. 2018 IEEE 3rd Adv. Inf. Technol. Electron. Autom. Control Conf. IAEAC 2018*, no. Iaeac, pp. 1066–1070, 2018, doi: 10.1109/IAEAC.2018.8577869.

[12] R. Krishnan, *Switched Reluctance Motor Drives : Modeling, Simulation, Analysis, Design, and Applications*. New York: CRC Press, 2001.