Adaptive Fuzzy-PI for Induction Motor Speed Control

Hari Maghfiroh 1*, Joko Slamet Saputro 2, Fahmizal 3, Muhammad Ahmad Baballe 4

1, 2 Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia
3 Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Yogyakarta, Indonesia
4 Department of Computer Engineering Technology, Kano State Polytechnic, Kano State, Nigeria

Email: 1 hari.maghfiroh@gmail.com, 2 jssaputro89@staff.uns.ac.id, 3 fahmizal@ugm.ac.id, 4 mbaballe@kanopoly.edu.ng

*Corresponding Author

Abstract — An induction motor (IM) is one type of AC motor which widely used. IM is chosen due to its simplicity, reliability, efficiency, and low cost. There are many methods proposed to control the speed of IM which is known as variable speed drive (VFD). In this study, the DTC method is used since it is more robust to the parameter’s changes. The combination of the Fuzzy and PI method is used in speed control. PID performances decrease when the system condition changes. Therefore, fuzzy is used as an adaptive algorithm to vary the PID gain. This article is organized as follows. In Section II, the DTC and Fuzzy-PI are reviewed. Results and discussion are presented in Section III. The last Section IV concludes this study.

Keywords—induction motor, speed control, PID, fuzzy

I. INTRODUCTION

An induction motor (IM) is one type of AC motor which widely used. It can be found in an industrial or electric vehicle as a traction motor [1]. According to [2], IM is chosen due to its simplicity, reliability, efficiency, and low cost. There are many methods proposed to control the speed of IM which is known as variable speed drive (VFD). Fig. 1 shows the resume of the method of VFD. It is classified as a sensor less and sensored method. In this study, sensored control is chosen since it is simple and easy to implement. Scalar control and vector control are two subcategories of the sensored approach. Scalar control has a straightforward architecture and poor precision [3, 4]. Vector control, on the other hand, is intricate and provides excellent performance [5-7]. This is the rationale behind using vector control in this investigation. There are two classes of vector control: Direct Torque Control (DTC) and Field Oriented Control (FOC) (DTC). Faster reactions and high efficiency characterize FOC. The approach is highly sensitive to changes in the parameter, though. DTC, on the other hand, provides good performance and consideration for parameter variation [8].

Because the FOC and DTC only regulate torque, additional control methods are required to control IM’s speed. Proportional Integral Derivative (PID), Fuzzy Logic Control (FLC), Sliding Mode Control (SMC), Artificial Neural Network (ANN), Linear Quadratic Gaussian (LQG), and also a combination of conventional and artificial method as in [14, 15] are just a few of the speed control methods for IM that have been proposed by researchers.

In this study, the DTC method is used since it is more robust to the parameter’s changes. The combination of the Fuzzy and PI method is used in speed control. According to [16], PID performances decrease when the system condition changes. Therefore, fuzzy is used as an adaptive algorithm to vary the PID gain. This article is organized as follows. In Section II, the DTC and Fuzzy-PI are reviewed. Results and discussion are presented in Section III. The last Section IV concludes this study.

II. RESEARCH METHOD

A. Direct Torque Control (DTC)

In the DTC method, the primary controlled variable is the stator flux vector as in (1) – (4). Where \( \lambda_s \), \( v_s \), \( r_s \), are stator flux, stator voltage, stator resistance, and stator current respectively. While subscript \( qs \) and \( ds \) in the flux and current means the results from DQ-transformation. With the right choice of a voltage reference, the magnitude and angle can be adjusted because \( v_s \) can be controlled by it.\n
\[
\frac{d\lambda_s}{dt} = v_s - r_s i_s \tag{1}
\]
\[
\lambda_{qs} = \int(v_{qs} - r_s i_{qs})dt \tag{2}
\]
\[
\lambda_{ds} = \int(v_{ds} - r_s i_{ds})dt \tag{3}
\]
\[
\lambda_s = \sqrt{\lambda_{ds}^2 + \lambda_{qs}^2} \tag{4}
\]

DTC additionally dissociates flux and torque for improved FOC performance. Without current control, it does, however, directly regulate the torque and flux. A PWM generator receives the output of the flux and torque control system. A switching table is used by the basic DTC to produce PWM. The disadvantage of this method is that it performs poorly at low speeds because of variable switching frequency [17]. As a result, in this study, we employ the SVPWM technique to obtain the switching signal. The block diagram of the DTC method for speed control using basic DTC and DTC-SVPWM is shown in Fig. 2. Only the PWM generation technique differs.

Fig. 3 shows the Simulink model of the IM speed control using DTC. The IM is inside the DTC Induction Motor Drive block. The parameters of the IM used are listed in Table 1. The input is speed references and load torque. The power supply used is AC three phase 460V 60Hz. Whereas
simulation output for analysis is speed, torque, IAE, and Kp and Ki gain changes.

![Fig. 1. IM control method classification [9]](image)

**B. Proposed Fuzzy-PID Control**

The proposed method block diagram of Simulink is shown in Fig. 4. There are two different control blocks which is PI and FLC-PI which can be selected using the switch in the output. The FLC in FLC-PI is used to vary the Kp and Ki based on the speed error and error changes. The fuzzy rule used for output Kp and Ki is the same and is listed in Table 2. The fuzzy Mamdani type with triangle membership is used in this design.

![Fig. 2. DTC of IM] (image)

![Fig. 3. IM control in MATLAB Simulink] (image)

**TABLE I. PARAMETERS OF THE INDUCTION MOTOR**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power, kVA</td>
<td>149.2</td>
</tr>
<tr>
<td>Voltage, Vrms</td>
<td>460</td>
</tr>
<tr>
<td>Frequency, Hz</td>
<td>60</td>
</tr>
<tr>
<td>Inertia (J), kgm²</td>
<td>3.1</td>
</tr>
<tr>
<td>Friction factor, Nms</td>
<td>0.08</td>
</tr>
<tr>
<td>Pole pair</td>
<td>2</td>
</tr>
</tbody>
</table>

**TABLE II. FUZZY RULES FOR KP AND KI**

<table>
<thead>
<tr>
<th>E/CE</th>
<th>N</th>
<th>Z</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>P</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

**III. RESULTS AND DISCUSSION**

There are two tests used to prove the performance of the proposed method which are the no-load and loaded test. The tolerance of the speed control is ± 1%. The integral absolute of error (IAE) is used to compare the performance besides settling time and overshoot or undershoot. The IAE value is calculated as accumulative from the beginning until the end of the simulation.
A. No-Load Test

In the no-load test, only speed tracking is performed. There are two set-point speeds used in one cycle which are 500rpm and 200rpm. This represents the increasing and decreasing speed respectively. Fig. 5 shows the no-load test which consists of speed, IAE, and Kp/Ki graph. It can be seen that method can track the speed reference well. However, there are small overshoot and undershoot found in the PI responses. This makes the PI method has higher IAE compared to Fuzzy-PI as shown in Fig. 5(b). Whereas Fig. 5(c) depicts the Kp and Ki of the Fuzzy-PI method which varied based on the error (E) and error changes (CE). It shows that when there are speed changes, both E and CE will have a high value that makes both Kp and Ki increase.

In more detail, the quantitative value of performance parameters is resumed in Table 3. In the first set-point, Fuzzy-PI is superior compared to PI in terms of settling time and overshoot. The proposed method has a faster settled time of 12.31% compared with the PI method. In the second test, the result is identical with Fuzzy-PI being the best. It has a better settling time of 38.18% faster. The settling time of the second set-point is measured from the time simulation of 2s. The minus sign in OS/US means it is an undershoot. In the end, the exact IAE value of Fuzzy-PI and PI is 3.56 and 6.47, respectively. This means the proposed method has better IAE compared to PI by 44.98%. The percentage of IAE may differ if the simulation time is extended.

<table>
<thead>
<tr>
<th>Control method</th>
<th>Settling time (s)</th>
<th>OS/US (%)</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set-point-1: 500rpm</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy-PI</td>
<td>0.57</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>0.65</td>
<td>1.7</td>
<td></td>
</tr>
<tr>
<td><strong>Set-point-2: 200rpm</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy-PI</td>
<td>0.34</td>
<td>&lt; -1</td>
<td>3.56</td>
</tr>
<tr>
<td>PI</td>
<td>0.55</td>
<td>- 2.4</td>
<td>6.47</td>
</tr>
</tbody>
</table>

Fig. 5. No load test result: (a) speed, (b) IAE, (c) Kp and Ki
### B. Loaded Test

The loaded test is to prove the performance of the speed control when load changes are given. The load is in the form of torque applied to the motor shaft. Fig. 6 shows the loaded test. In terms of speed tracking under loaded conditions, the result is in Fig. 6 (a). It is seen that there is speed fluctuation when the load changes. For example, at time 1.5s, when the load decreases, the speed increases. On the other side, at time 3s when the load increases, the speed decreases. Overall, both control methods can perform speed tracking under loaded conditions well. In terms of IAE, the loaded condition has a higher IAE value. The IAE of PI is nearly close to the Fuzzy-PI one since the load can reduce the overshoot of PI responses. The Kp and Ki value of Fuzzy-PI, it has more variation since the controller wants to maintain the speed under load variation.

The quantitative result of the loaded test is listed in Table 4. In seeing that, in the first set-point, PI is superior to Fuzzy-PI since it is faster to reach the speed references by 37.5%. Whereas in terms of overshoot, the proposed method is still superior to PI. In the second set-point, the proposed method dominated both settling time and undershooting. It is faster by 39.29% than the PI response. The overshoot/undershoot of the loaded test is higher than the no-load test. The last, in terms of IAE, the proposed method is superior with lower IAE compared to the PI method by 4.47%.

#### TABLE IV. LOADED PERFORMANCE PARAMETERS

<table>
<thead>
<tr>
<th>Control method</th>
<th>Settling time (s)</th>
<th>OS/ US (%)</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set-point-1: 500rpm</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy-PI</td>
<td>0.88</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>PI</td>
<td>0.64</td>
<td>3.49</td>
<td></td>
</tr>
<tr>
<td><strong>Set-point-2: 200rpm</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy-PI</td>
<td>0.34</td>
<td>-2.37</td>
<td>12.41</td>
</tr>
<tr>
<td>PI</td>
<td>0.56</td>
<td>-6.73</td>
<td>12.99</td>
</tr>
</tbody>
</table>

### IV. Conclusion

The speed control of the induction motor based on DTC has been carried out successfully in the simulation environment. The speed tracking with two conditions, no-load and loaded, the condition was performed. The proposed Fuzzy-PI results are compared to the PI method to identify the improvement. The simulation results prove that the proposed method can perform well in speed tracking and loaded condition. It is superior in terms of settling time, overshoot/undershoot, and IAE compared to the PI method. It has lower IAE in both speed tracking and loaded conditions by 44.98% and 4.47%, respectively. The fuzzy rules in this study are determined based on experience. In the future, to get the optimal result, the fuzzy rules can be generated from the optimization method.
REFERENCES


