Adaptive Linear Quadratic Gaussian Speed Control of Induction Motor Using Fuzzy Logic

Hari Maghfiroh¹, Alfian Ma’arif², Feri Adriyanto³, Iswanto Suwarno⁴, Wahyu Caesarendra⁵

¹ Department of Electrical Engineering, Universitas Sebelas Maret, Surakarta 57126, Indonesia
² Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta 55164, Indonesia
³ Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Yogyakarta 55184, Indonesia
⁴ Faculty of Integrated Technologies, Universiti Brunei Darussalam, Gadong BE1410, Brunei Darussalam
⁵ Faculty of Mechanical Engineering, Opole University of Technology, Opole 45-758, Poland

Corresponding Author Email: hari.maghfiroh@staff.uns.ac.id

https://doi.org/10.18280/jesa.560420

Received: 10 April 2023
Revised: 31 July 2023
Accepted: 5 August 2023
Available online: 31 August 2023

Abstract

An induction motor’s speed can be managed in a variety of ways using a Variable Frequency Drive (VFD). In this study, the speed control of an induction motor will be controlled by applying Indirect Field Oriented Control (IFOC) combined with Linear Quadratic Gaussian (LQG). Conventional LQG control is a linear controller; therefore, if the system’s dynamic is high and over the linear boundary, the LQG performance will not be optimal. Therefore, Adaptive LQG (ALQG) is proposed. Fuzzy logic is used as an adaptive algorithm with low complexity and ease of implementation. The significance of this study lies in its endeavor to tackle the challenges associated with nonlinearities and high dynamics in induction motor control. The average performance of speed variation and load variation tests proves that ALQG is superior in terms of settling time and undershooting than PID and LQG. PID has the highest overshoot with the smallest Integral Absolute Error (IAE). In comparison, ALQG is superior to conventional LQG in terms of IAE with 3.59% lower.

1. INTRODUCTION

One technological development that can facilitate the production process is the electric motor. Two types of electric motors, AC and DC, depending on the supply current. The DC motor employs a mechanical commutator (brush), which increases maintenance costs even if it is easier to operate [1]. As a result, an AC motor is selected. The two types of AC motors are synchronous and asynchronous. The induction motor is one popular asynchronous motor type [2]. During installation and maintenance, the induction motor is economical, simple, and reliable [3, 4]. About 60% of the facility’s energy is mainly used by induction motors [5]. There are wide applications for induction motors, such as compressors, elevators, centrifugal pumps [6, 7], and also as traction motors in electric vehicles [8, 9].

An induction motor’s speed can be managed using a variable frequency drive (VFD) [3]. Figure 1 shows the classification of VFDs sensed (with a sensor) or unsensed (sensorless). The sensor-based control method will be employed in this study because the algorithm is simpler.

Scalar control and vector control are the two subtypes of sensor control. Scalar control is straightforward, easy, and parameter-independent [10, 11]. On the other hand, vector control has excellent controlling performance and is the most popular [12, 13]. Vector control is divided into Field Oriented Control (FOC) and Direct Torque Control (DTC). The FOC has a faster reaction and high efficiency. However, this method is sensitive to the parameter’s changes. On the other hand, DTC provides good performance and consideration for parameter variation [14]. The FOC approach is further broken down into Direct-FOC and Indirect-FOC (IFOC). According to the studies [15, 16], the IFOC approach has been widely applied. Therefore, the IFOC method is adopted in this study.

The scalar control and vector control are essentially torquing control; hence speed control requires a separate speed control mechanism. There are numerous speed control algorithms used in the speed control of induction motors, including PID [17], Linear Quadratic Regulator (LQR) [18, 19], Linear Quadratic Gaussian (LQG) [20-22], fuzzy logic controllers [23, 24], Sliding Mode Controllers (SMC) [25, 26], backstepping control [2], Artificial Neural Networks (ANNs) [27, 28] and some combination methods as in the studies [29, 30].

According to the study [31], LQG is one kind of optimal control that combines LQR and Kalman Filter. The use of the Kalman Filter to predict system states can reduce the number of sensors compared to the conventional LQR method. Conventional LQG control is a linear controller; therefore, if the system’s dynamic is high and over the linear boundary, the LQG performance will not be optimal. Therefore, in this study, an Adaptive-LQG (ALQG) is proposed. With fuzzy logic as an adaptation algorithm, it can adapt to the plant condition allowing a wide operation range.

The previous study using adaptive LQG has been presented in the studies [32, 33]. Both previous studies used the Least Square Method (LSM) to update the LQG gain. However, the use of LSM is mathematically complicated. Therefore, the ALQG method using fuzzy logic as an adaptive algorithm with low complexity and ease of implementation is proposed. The objective is to get ALQG which is an LQG that can adapt to plant conditions and has a wide range of operations.
Furthermore, the algorithm is simple and implementable. The simulation test is conducted under speed variation and load variation to prove the performance of the proposed method. The rest of the paper is organized as follows. Section 2 briefly reviews induction motor control and the proposed control design. The simulation testing results are presented in Section 3. The last, Section 4 is the conclusions.

![Figure 1. Induction motor control classification [5]](image)

2. METHOD

This study consists of several stages which are presented in Figure 2. The induction motor model was performed on the black box modelling method where the parameter value was based on a literature study. Then, the model is used in the ALQG design. The simulation is done in MATLAB Simulink. The last step is analyzing the result by comparing it with other control methods which are PID and conventional LQG.

![Figure 2. Block diagram of the research step](image)

2.1 Induction motor control

Figure 3 depicts the block diagram of VFD parts. Because the sensor base is the preferred control strategy, the speed sensor is used, in this study encoder. The controller in Figure 3 is a speed controller since the torque control is integrated into the inverter. Speed data must be collected and supplied to the inverter and controller for torque and speed control. The controller's output is a torque reference for the inverter's torque management.

![Figure 3. General component of IM VFD](image)

To validate the performance of the proposed control method, the simulation was built in the MATLAB Simulink software, as shown in Figure 4. There are two controllers in cascade form. The first one is the speed controller proposed in this study and the second one is vector control based on IFOC which employs hysteresis current control inside it. In the speed controller block, two options are using PID or LQG/ALQG since in this study we compare the performance of the proposed method with PID and conventional LQG. The induction motor parameters used in this study are listed in Table 1.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal power</td>
<td>37.3</td>
<td>kVA</td>
</tr>
<tr>
<td>Voltage (line-line)</td>
<td>460</td>
<td>V-rms</td>
</tr>
<tr>
<td>Frequency</td>
<td>60</td>
<td>Hz</td>
</tr>
<tr>
<td>Stator resistance</td>
<td>0.087</td>
<td>Ohm</td>
</tr>
<tr>
<td>Stator inductance</td>
<td>0.0008</td>
<td>H</td>
</tr>
<tr>
<td>Rotor resistance</td>
<td>0.228</td>
<td>Ohm</td>
</tr>
<tr>
<td>Rotor inductance</td>
<td>0.0008</td>
<td>H</td>
</tr>
<tr>
<td>Mutual inductance</td>
<td>0.0347</td>
<td>H</td>
</tr>
<tr>
<td>Pole pairs</td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>

The induction motor from the MATLAB model is extracted to obtain its state-space model using MATLAB System Identification Toolbox to simulate the actual situation. This strategy is considered to illustrate the step for actual implementation. This modelling method is also known as black-box modelling which only uses the input-output data of the system, as used in the study [34]. Instead of using the mathematical equation as in the studies [35, 36], this method is more suitable for hardware implementation where no system parameters are ready. In this study, the input-output data for modelling are torque and motor speed, respectively. The state-space model of the induction motor is presented in (1)
consisting of two states which are speed and stator current. The speed is measured using the encoder, whereas the stator current is predicted using the Kalman Filter in the LQG control. This can reduce the number of sensors.

\[
\dot{x} = \begin{bmatrix}
1.378 & 0.285 \\
-1.25 & 0.06
\end{bmatrix} x + \begin{bmatrix}
0.16 \\
-0.52
\end{bmatrix} u \\
y = \begin{bmatrix}
1 & 0
\end{bmatrix} x
\]  

(1)

2.2 ALQG design

The LQR is the state feedback control that minimizes a cost function, as presented in Eq. (2) [37, 38]. The matrices \(Q\) and \(R\) determine the relative importance of the error and energy expenditure [39]. LQG is the improvement of the LQR method which is a state-feedback control that requires all the state of the system. By adding the Kalman Filter (KF) [40, 41] as an observer, LQG can reduce the sensor used in the LQR method. Therefore, it can reduce equipment costs, especially the number of sensors. However, its robustness is not guaranteed [42]. Figure 5 shows the block diagram of LQG control which consists of LQR and Kalman Filter.

\[
J = \int (x^T(t)Qx(t) + u^T(t)Ru(t)) dt
\]  

(2)

\[
Q = \begin{bmatrix}
20 & 0 \\
0 & 1
\end{bmatrix}, \quad R = \begin{bmatrix}
1
\end{bmatrix}
\]  

(3)

Figure 4. Speed control of IM in MATLAB/Simulink software

Figure 5. LQG block diagram [36]

Conventional LQG control is a linear controller; therefore, if the system’s dynamic is high and over the linear boundary, the LQG performance will not be optimal. Since LQG is the combination of LQR and Kalman Filter, this method has two gains: \(K_{LQR}\) and \(K_f\) for LQR gain and Kalman Filter gain, respectively. In this study, matrix \(Q\) and \(R\) are chosen manually as shown in Eq. (3). In the matrix \(Q\), the value of \(Q\) (1,1) which corresponds to the speed state is given the weight of 20. On the other hand, the value of \(Q\) (2,2) which is the weight of the stator’s current state is set to be 1. The \(R\) method is correlated with control effort, in this test it is set to be 1.

Figure 6. Proposed ALQG

Fuzzy logic is used as an adaptive algorithm whose output is a scaling value for LQR gain. The block diagram of the proposed ALQG is presented in Figure 6. The error signal between speed reference and speed output is sent to the integrator controller to eliminate steady-state error. The error and change are also sent to the fuzzy logic controller to generate a scaling factor for LQR gain. The adaptation mechanism is FLC determines the scaling factor based on the determined rules for \(K_{LQR}\). Therefore, the \(K_{LQR}\) value can
change based on the system condition in this case the error of speed tracking.

Table 2. LQG gains

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$X_1$</th>
<th>$X_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{LQR}$</td>
<td>4.64</td>
<td>1.02</td>
</tr>
<tr>
<td>$K_f$</td>
<td>0.85</td>
<td>-0.88</td>
</tr>
</tbody>
</table>

Table 3. FLC rules

<table>
<thead>
<tr>
<th>DE/E</th>
<th>L</th>
<th>M</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>M</td>
<td>L</td>
<td>M</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>M</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

Figure 7. FLC rules mesh diagram

Figure 8. ALQG control design process

Table 2 shows the constant $K_{LQR}$ and $K_f$ gain for the conventional LQG method. Table 3 is the fuzzy rule for $K_{LQR}$ adaptation with two inputs: speed error (E) and its change of error (DE). The output of the fuzzy system is a scaling factor that scales the $K_{LQR}$ gain, where L, M, and H present low, medium, and high, respectively. The rules are tuned manually with the principle that the higher the error, the higher the scaling factor. Both the input and output of a fuzzy system consist of three members of a triangular function. The mesh diagram, which shows the surface relation between the input and output of the fuzzy system is presented in Figure 7. It informs that the scaling factor used is 0.5 to 1.5, if the E and DE are negative, the scaling is below 1, whereas if it is positive, the value will be greater than 1. The scaling factor will be 1 if the E and DE are zero. Figure 8 shows the steps for designing the ALQG which uses black-box modelling based on input and output data of the motor namely torque and speed. Once the state-space model using MATLAB System Identification is obtained, the LQG control is designed before adding fuzzy logic to form ALQG.

The performance of the proposed method is tested in the simulation in the MATLAB software as shown in Figure 4. The performance is measured in terms of settling time, overshoot/undershoot, and Integral Absolute Error (IAE).

3. RESULTS AND DISCUSSION

The model validation is conducted under conditions of speed and load fluctuation. The loaded test is done by giving torque to the motor shaft. The performance of the suggested control approach was demonstrated by comparison to the PID and LQG methods. The speed control in use has a tolerance of 2%.

3.1 Speed variation test

The objective of a speed variation is to evaluate how well the control method performs with the provided speed set points. In the simulation, the load was determined at 50Nm. The speed varied from the initial condition and increased gradually. Figure 9 displays the outcomes of the speed variation testing. A velocity graph, an Integral of Absolute Error (IAE), and a $K_{LQR}$ graph are all present.

Figure 9(a) depicts the speed response of the control methods in the first test when tracking the speed references. All the control techniques are effective. Figure 9 (b) shows the zoomed-in version in more detail. Figure 9(b) shows that the PID has the highest overshoot and undershoot values. LQG and ALQG's quick replies differ slightly from one another. While Table 4 displays the quantitative outcomes. Figure 9(c) shows the IAE graph where PID has the lowest value, followed by ALQG and LQG. Figure 9(d) shows the $K_{LQR}$ value, which adapts to the system error and change of error based on fuzzy logic. The $K_{X1}$ and $K_{X2}$ oscillate between the value of 4.64 and 1.02, respectively, which is the constant gain value of LQR. Both $K_{X1}$ and $K_{X2}$ value increases when the error increases. Table 4 shows the results of speed tracking, including three set-point speeds, which are 50, 80, and 60 rad/s. Since there are three set-point values, the settling time and overshoot or undershoot are measured in every value. In contrast, the IAE is derived as accumulation at the end of simulation time.
According to the first set-point, PID has the lowest settling time, Figure 9(a). However, it also has the highest overshoot compared to others. LQG has the highest settling time with medium overshoot compared to PID and ALQG. Whereas the settling time of ALQG is higher compared to PID by 1.31% and lower compared to LQG by 6.63%. In terms of overshoot, the proposed method has lower and higher overshoot compared to PID and LQG by 50.42% and 15.89%. In the second set-point, the same, ALQG settling time is in between PID and LQG. Whereas its overshoot is lower and upper by 27.27% and 14.29% compared to PID and LQR, respectively. In the last set-point, when speed reference decreases, ALQG dominates with the lowest settling time and no undershoot. While PID has a lower settling time with higher undershoot compared to LQG. The accumulative value of IAE shows that PID has the lowest value, LQG has the highest value, and ALQG is in the middle of them. The IAE of ALQG is higher than PID and lower than LQG by 84.08% and 3.15%, respectively.

<table>
<thead>
<tr>
<th>Control Algorithm</th>
<th>Settling Time (s)</th>
<th>Undershoot (%)</th>
<th>Overshoot (%)</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set-point 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>1.53</td>
<td>-</td>
<td>19.12</td>
<td></td>
</tr>
<tr>
<td>LQG</td>
<td>1.66</td>
<td>-</td>
<td>8.18</td>
<td>5.24</td>
</tr>
<tr>
<td>ALQG</td>
<td>1.55</td>
<td>-</td>
<td>9.48</td>
<td>5.24</td>
</tr>
<tr>
<td>Set-point 80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>1.03</td>
<td>-</td>
<td>5.09</td>
<td></td>
</tr>
<tr>
<td>LQG</td>
<td>1.42</td>
<td>-</td>
<td>3.24</td>
<td>4.23</td>
</tr>
<tr>
<td>ALQG</td>
<td>1.27</td>
<td>-</td>
<td>3.70</td>
<td>4.23</td>
</tr>
<tr>
<td>Set-point 60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID</td>
<td>0.96</td>
<td>4.23</td>
<td>23.37</td>
<td></td>
</tr>
<tr>
<td>LQG</td>
<td>1.37</td>
<td>2.92</td>
<td>-</td>
<td>44.42</td>
</tr>
<tr>
<td>ALQG</td>
<td>0.66</td>
<td>-</td>
<td>-</td>
<td>43.02</td>
</tr>
</tbody>
</table>

### 3.2 Load variation test

The second test involves load variation. The objective of this test is to determine how well the motor speed control system performs under changing load conditions. This test can
represent situations with ups and downs in the state of the roads for electric car applications.

Figure 10 shows the outcomes of the load variation test. The speed differences may be noted when the load is increased and decreased. The load starts at 50 Nm and keep constant for 3 seconds before being instantaneously increased to 100 Nm. The load is reduced to 20 Nm at the 6s and stays there until the simulation is finished.

In the load variation test, the speed graph in Figure 10(a), the settling time, overshoot, and undershoot of both ALQG and LQG look the same. For more detail, the zoomed-in version is presented in Figure 10(b). It is seen that in every speed change, each method has a different performance due to load variation. The quantitative result is resumed in Table 5. Regarding settling time, ALQG is superior, with 5.29% and 4.17% faster than PID and LQG, respectively. When the load increases, there is an undershot. In case of load decreases, all the control methods respond with overshoot. Since there are two overshoots, in the starting and when the load decreases, the maximum overshoot value is used. PID has the higher value of maximum overshoot; on the other side, LQG is the lowest one, whereas ALQG is in the middle of them. Although ALQG has a higher value of undershoot and overshoot than LQG, it can back to the references faster. Meanwhile, from the IAE perspective, Figure 10(e), the IAE of ALQG is higher than PID and smaller than LQG as in the speed variation test results. In this test, ALQG has higher and lower IAE by 25.46% and 4.06% compared to PID and LQG, respectively. Figure 10(d) shows the value changes of $K_{LQR}$, where $K_{X1}$ and $K_{X2}$ correspond to the state gain of speed and stator current. Its value adaptively changes based on the value of error and change of error of the speed. Both have the same pattern where the higher error $K_{LQR}$ will have higher value.

Table 6 summarizes the average of all the test performed. The proposed ALQG is better than others in terms of settling time and undershoot. Although it has a higher overshoot than conventional LQG, it has a lower IAE. PID methods still have the lowest IAE value with the highest overshoot. Whereas the conventional LQG has the lowest overshoot with the highest settling time and undershoot.

![Figure 10. Load variation test results: (a) speed responses; (b) zoom-in of speed responses; (c) IAE; (d) $K_{LQR}$](image-url)
Table 5. Load variation test performances

<table>
<thead>
<tr>
<th>Control Algorithm</th>
<th>Settling Time (s)</th>
<th>Undershoot (%)</th>
<th>Overshoot (%)</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>1.70</td>
<td>4.01</td>
<td>18.21</td>
<td>30.87</td>
</tr>
<tr>
<td>LQG</td>
<td>1.68</td>
<td>5.79</td>
<td>9.31</td>
<td>40.37</td>
</tr>
<tr>
<td>ALQG</td>
<td>1.61</td>
<td>6.40</td>
<td>9.75</td>
<td>38.73</td>
</tr>
</tbody>
</table>

Table 6. Average of all performances

<table>
<thead>
<tr>
<th>Control Algorithm</th>
<th>Settling Time (s)</th>
<th>Undershoot (%)</th>
<th>Overshoot (%)</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>1.31</td>
<td>2.06</td>
<td>10.61</td>
<td>13.56</td>
</tr>
<tr>
<td>LQG</td>
<td>1.53</td>
<td>2.18</td>
<td>5.18</td>
<td>21.20</td>
</tr>
<tr>
<td>ALQG</td>
<td>1.27</td>
<td>1.60</td>
<td>5.73</td>
<td>20.44</td>
</tr>
</tbody>
</table>

Table 7. Parametric sensitivity test results

<table>
<thead>
<tr>
<th>Scaling Factor of FLC Output</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5–1.5</td>
<td>43.12</td>
</tr>
<tr>
<td>0.7–1.3</td>
<td>44.17</td>
</tr>
<tr>
<td>0.2–1.8</td>
<td>48.93</td>
</tr>
</tbody>
</table>

Regarding the parametric sensitivities, the scaling factor affects the ALQG performance. As investigated on the speed tracking test, different scaling factors are tested. The results are resumed in Table 7. It informs that the scaling factor from fuzzy output needs to be carefully selected since the higher or lower range does not guarantee with best performance. Besides that, the fuzzy input and rules also affect the FLC performance as adaptive algorithms.

4. CONCLUSIONS

The ALQG method has been used to develop and simulate the speed control of the induction motor. Based on speed error and change of error, the LQG gain was scaled using fuzzy logic as an adaptive method. The objective to give the adaptable capability to the LQG method is achieved. The average results of the speed and load variations show that ALQG is superior in terms of settling time and undershoot. PID has the highest overshoot with the smallest IAE. Whereas ALQG is also superior to conventional LQG in terms of IAE. In terms of IAE, ALQG is in the middle of PID and LQG. The IAE of ALQG is higher than PID and lower than LQG by 50.72% and 3.59%, respectively. ALQG has the potential for better performance by proper tuning of fuzzy membership and or its rules. Fuzzy logic as adaptive algorithms are simple and easy to implement. Hence it can be combined with LQG for different motor loads and speed ranges. Future work to improve the performance of ALQG can be done by using artificial intelligence for fuzzy tuning to get the optimal results.

ACKNOWLEDGMENT

This work is supported by Universitas Sebelas Maret. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, who have improved the presentation.

REFERENCES


[38] Önen, Ü., Çakan, A. (2021). Multibody modeling and balance control of a reaction wheel inverted pendulum


