A NOVEL CONTROL OF AC VOLTAGE CONTROLLER UNDER INDUCTION MOTOR LOAD

Ali M. Eltamaly  A. I. Alolah  R. Hamouda  M. Y. Abdulghany
EE Department, Faculty of Engineering, King Saud University, POB 800, Riyadh, 1121 Saudi Arabia
E-mail: eltamaly@ksu.edu.sa  alolah@ksu.edu.sa  rhamouda@ksu.edu.sa  Digital_minde@yahoo.com

KEYWORDS
FPGA, Induction motor, PSIM, triac, Voltage controller.

ABSTRACT
A novel, simple and reliable digital firing scheme has been implemented for soft starting and speed control of three-phase induction motor using ac voltage controller. The system consists of three-phase supply connected to the three-phase induction motor via three triacs and its control circuit. The ac voltage controller has three modes of operation depending on the shape of supply current. The performance of the induction motor differs in each mode where the speed is directly proportional with firing angle in two modes and inversely in the third one. So, the control system has to detect the current mode of operation to choose the correct order for the firing angle of triacs. Three sensors are used to feed the line currents to control system to detect the mode of operation. The control strategy is implemented using a low cost Xilinx Spartan-3E field programmable gate array (FPGA) device. Three PI-controllers are designed on the FPGA to control the system for the three-modes. Simulation of the system is carried out using PSIM computer program. The simulation results show stable operation for different loading conditions especially in mode 2/3. The simulation results have been compared with the experimental results from laboratory prototype.

INTRODUCTION
Speed control of three-phase induction motor can be achieved by stator voltage variation using three-phase controller. This controller consists of three triacs connected as shown in Figure 1. This system has been used in speed control of three-phase induction motor in many literatures (Lipo, T.A. 1971; Hamed, S. et al.1990; Kamesh, H. 1977 –more references). Although this technique are characterized by low cost, simple and rugged design, it is suffering from many difficulties as high harmonics in the supply currents, narrow speed control range, and different performance depending on the shape of supply currents. The harmonic contents in line currents can be reduced using modern harmonic reduction techniques as active filters or third harmonic injection technique (Ali M. Eltamaly 2008; Bhuvaneswari, G. et. al.). The problem of the different performance depending on the shape of supply currents can be removed using control system that can detect the mode of operation of the system as explained and implemented in this paper. The ac voltage controller has three mode of operation depends on the shape of supply current namely 0/2, 0/2/3, and 2/3 (Lipo, T.A. 1971; I.

Aolol et. al. 2006). The best mode of operation is 2/3 because of low THD in its line currents, high efficiency and low pulsating torque. To force the motor to work in 2/3 mode of operation, it is required to follow the limits of mode 2/3. The motor speed is directly proportional to the firing angle in modes 0/2 and 0/2/3 and inversely proportional in mode 2/3. So, the control system has to detect the mode of operation to produce the correct firing angle.

COMPUTER SIMULATION
Simulation of three-phase phase ac controller under three-phase induction motor load has been carried out using PSIM 6.1 computer program and SIMULINK to validate the simulation results. The induction motor model in stationary reference frame has been used in the simulation (M. G. Solveson. 2004; M. G. Solveson, B. Miražal, and N. A. O. Demerdash, 2006). The details of the simulation of the induction motor and ac voltage controller have been shown in many researches (H. A. Ashour; R.A. Ibrahim, 2007; Gasitli.; M. M. Ahmed, 2005; Xu Dianguo; Zhao Kaiqi; Wang Yi. 2003; Rajaji, L.; Kumar, C. 2008; Ali M. Eltamaly 2008; Bhuvaneswari, G.; Charles, S.; Nair, M.G. 2008 ).

The nameplate motor data used in this simulation is; 1kW, 380 V, Y-connected, 3.0A, 4-pole, 60 Hz, its pu parameters are: \( R_s = 0.0583 \) pu, \( R_r = 0.0417 \) pu, \( X_s = 0.125 \) pu, \( X_r = 0.018 \) pu, and \( X_m = 1.05 \) pu.

Intensive simulation has been carried out for soft starting and speed control at different operating conditions to determine the operating limits of this system. Each mode for different speed and load torque for different firing angle are shown in Figure 2 and Figure 3 respectively. The motor speed is directly proportional with the firing angle in modes 0/2 and 0/2/3 and inversely proportional in mode 2/3 as shown in Figure 4. The system operates in a highly distorts supply current especially in light loads and low firing angle which occurs in modes 0/2 and 0/2/3. So it is better to avoid the operation of the motor in these modes.

The logic used in the design of digital controller is shown in the following sections. Figure 5 shows the main block diagram for the whole control system. This main block diagram has several sub-blocks each one handles a separate function that will be detailed in the following sections.
Zero crossing detector-block receives the square waveform from external zero crossing circuit for each phase voltage. Zero crossing detector blocks generate synchronization pulses to synchronize internal control logic with phase voltages. Sine-wave half cycle counter block generates three digital saw-tooth signals using free running counters with a double of the main supply frequency. Each saw-tooth counter is synchronized with one of phase voltages using zero crossing pulse. The saw-tooth counter is used to represent the value of instantaneous angle of each phase.

DIGITAL CONTROLLER

As explained before the system has to detect the mode of operation to decide to take the suitable action to control the speed. Three PI-controllers should be designed to match the performance of the motor in each mode as shown in Figure 5. The motor speed is measured using speed sensor and fed to ADC to be compared with the reference speed to determine the error signal which is fed to the three PI-controllers as shown in Figure 5. The limits decoder receives the signals from three-phase current sensors to determine the mode of operation of the motor in order to select the correct PI-controller using the control MUX block. The firing angle control block receives the calculated firing angle from the selected PI-controller, clips the firing angle to ensure that the firing angle lies between its maximum and minimum allowable values, and applies this change to current firing angle every new cycle of phase voltage. Angle detection and pulse duration counter block receives the firing angle and the saw-tooth counter values to generate three pulses for the three triacs.

Figure 1 Three-phase ac voltage controller under induction motor load.

Zero crossing Detector-Block

Zero crossing point is detected by simple logic circuit and an external analog comparator. Zero crossing comparator output is a square waveform has a +V value with the positive half cycle of phase voltage and 0 with the negative half cycle of phase voltage as shown in Figure 6. The required signal for the control system has
two pulses for each period one when the comparator output changes from 0 to +V and the other when it changes from +V to 0. The pulse duration is only one FPGA clock cycle. Comparing any two successive samples of comparator output using XOR logic function will generate the pulses whenever the comparator output changed as shown in Figure 6. Comparator output is sampled using a single bit register, and the register input and output are applied to XOR logic gate. AND gate and an inverter are used to disable zero crossing pulse generation whenever a global system reset is applied. Each phase has a separate zero-crossing detector block to synchronize the saw-tooth counter with phase’s voltages.

B Sine-wave Half Cycle Counter-Block

The output of logic circuit used in sine-wave half cycle counter block is a digital saw-tooth waveform with a frequency of two times the supply frequency. The circuit is a digital accumulator which incremented every positive edge of FPGA core clock and reset whenever it reaches its maximum value or a zero crossing is detected.

In the moment of the saw-tooth waveform reach its maximum, the saw-tooth is reset and starting count from zero for the next half cycle of sinusoidal phase voltage. Resetting the saw-tooth when the phase voltage has a zero cross; grants that the saw-tooth and phase voltage starts from zero value at the same time, this shown in Figure 7. So, the maximum value of saw-tooth waveform can be obtained from (1).

\[ C_{\text{max}} = \frac{T}{2*F_c} = \frac{f_c}{2*F} \]  

(1)

The angle of the phase voltage equals its frequency times time \( \theta = f \cdot t \). The instantaneous angle value is directly proportional to the time. Also the instantaneous value of saw-tooth is directly proportional to the instantaneous value of phase angle or \( C_{\theta} \).

Using the relation between phase angle and saw-tooth instantaneous values, equation (2) can be used to convert any given phase angle to a respective saw-tooth value.

\[ C_{\theta} = C_{\text{max}} \frac{\theta}{180} \]  

(2)

For a chosen firing angle value \( \theta_f \), the respective saw-tooth value \( C_{\theta} \) can be obtained from (2) by substituting \( \theta = \theta_f \). The firing pulse has a finite width of \( \theta_f \) degrees; this is equivalent to \( \Delta C_{\theta} \) change in saw-tooth value since the firing pulse starts. The value of \( \Delta C_{\theta} \) can be obtained from (2) by substituting \( \theta = \theta_f \). The values of \( C_{\text{max}} \) and \( C_{\theta} \) are rounded to the nearest integer value as the digital control is designed to work with fixed numbers representation only. The relation between phase voltage, saw-tooth waveform, firing pulses and FPGA core clock is shown in Figure 7.

Knowing that the FPGA works at speed of 50MHz and the line frequency is 60Hz. Using (1); saw-tooth maximum value \( C_{\text{max}} \) will equal 416666. In (2), \( C_{\theta} \) can take any value from 0 to 416666 with increment of 1; which allows firing angle to be adjusted with a step of 432*10^-6 degrees.

![Figure 6: Voltage, comparator output, and zero-crossing pulse.](image)

![Figure 7: Relation between phase voltage, saw-tooth, firing pulses, and FPGA core clock.](image)

Angle Detection and Pulse Duration Counter Block

To generate the firing pulse, the saw-tooth is compared to a constant value \( C_{\theta} \) respective to the required firing angle. This constant value can be obtained using (2). The firing pulse should starts when the value of saw-tooth equals to \( C_{\theta} \) and stays active for duration equals to \( \Delta C_{\theta} \). There is a register has an initial value of 0 and will output 1 when being enabled by the saw-tooth comparator. The output of the lower side register is the firing pulse applied to triac gate. Once enabled, the output of lower side register will enable the upper side register. The upper side register and the adder in the top-right corner of the logic diagram form together a
digital counter. The counter is incremented every positive edge of FPGA clock and keeps counting up once it’s enabled by the lower side register output. The counter will count until it reaches a maximum value of \( \Delta C_p \), in this moment the counter and the lower side register are reset in the same time which will disable both the output pulse and the counter in the same time. Also, the lower side register and the counter are reset simultaneously once the phase voltage crossing zero value.

**PI Controllers-Blocks**

Three PI controllers used to regulate the firing angle position depending on the current mode of operation of the motor and the requested motor speed. Equation (3) is the idealized model for the continuous PI algorithm.

\[
u(t) = k_p e(t) + k_i \int e(t) dt \quad (3)
\]

where: 
- \( u(t) \): is the output of PI controller at time \( t \),
- \( e(t) \): is the error signal at time \( t \),
- \( k_p \): is the proportional gain constant,
- \( k_i \): is the integral gain constant,

For small sample times the continuous time PI equation can be turned into a difference equation by discretization. The integral term is approximated using trapezoidal integration which requires storage of all past sample errors. The intermediate equation can be transformed into a recursive equation where only the previous output, current error, and the last error must be stored. The final discrete version of the PI equation is shown in the following:

\[
u(n) = u(n-1) + k_1 e(n) + k_2 e(n-1) \quad (4)
\]

The same structure used for the three PI controllers but each controller has different set of constants \( k_1 \) and \( k_2 \) based on its \( k_p \) and \( k_i \) values. Each PI controller was first designed in time domain to set the correct values of \( k_p \) and \( k_i \) constants. PSIM simulation software used to tune each PI independently. The digital structure of PI controller requires only two multiplications and three additions which is a simple and straightforward algorithm to be implemented using FPGA cells. FPGA already has embedded multipliers blocks (18x18) to be used in the multiplication processes while the addition process is efficiently mapped to FPGA logic cells by synthesis tool. Pipelining registers added after each level of PI structure to post performance and avoid any timing issues due to unbalanced delay in the digital PI structure. The sequencer or the control state machine is a simple control logic used to enable the last stage register (used for \( u(n) \)) three clock cycles after receiving new error value \( e(n) \). The three PI controllers receive the error signal results from comparing motor sensed speed output from speed sensor and ADC sub-system and the reference speed. The reference speed may be changed online using simple push-buttons.

**Limits Decoder Block**

The limits decoder block is that part of system which responsible to decide which PI controller to be used based on the current mode of motor operation. The limits decoder block uses two analog comparators for each phase current and simple decode logic inside FPGA. The motor mode of operation is decoded every one complete cycle of phase voltage as a minimum of one complete cycle is needed to determine motor mode of operation based on concept discussed in introduction. Figure 8 shows the analog/digital circuits used in limits decoder block.

Each phase current has a one of two possible states in a complete phase voltage cycle. It can have a zero value or a non-zero value. One analog comparator will generate a positive pulse only if the phase current has a positive value and the other will generate a positive pulse only if the phase current has a negative value. Both comparators will generate a zero output if the phase current has a zero value. Applying the output of two comparators to a logical OR function will result in a logic 1 only if the phase current has a non-zero value.

A stage of decode logic based on AND/NAND logic functions and single bit registers used to decode the three possible states of phase currents. The decoded current states applied to 3-inputs Look Up Table (LUT) which generates the proper control MUX selection signal (selects between 3 different PI controllers). The control MUX selection signal is updated every one complete cycle of phase voltage.

**F. Firing Angle Control-block**

PI controllers adjust the firing angle so the motor maintains its speed at different loads and mode of operations. The output of selected PI controller must be scaled and clipped to fall in the allowable firing angle range. The firing angle control block receives the selected PI controller output and scales it. The scaled output is compared to maximum and minimum allowable firing angle values and passed only if it falls within the allowable range. If the firing angle value exceeds the maximum allowable value; the value is blocked and only the maximum allowable value is passed. If the firing angle is lower than the minimum allowable value; the value is blocked and only the minimum allowable value is passed. The firing angle is updated every half cycle of phase voltage.

**DESIGN IMPLEMENTATION**

Control logic was implemented on a Xilinx Spartan-3E embedded development kit. The kit integrates a Spartan-3E FPGA device with over 33,000 of configurable logic cells (UG257, MicroBlaze). The device is intended for cost sensitive embedded
applications; however, the supported resources and speed allow efficient implementations of digital control systems with performance much better than soft implementations using microcontrollers and DSP processors.

PicoBlaze is an embedded soft processor designed by Xilinx for FPGA applications (UG129, PicoBlaze). One instance of a PicoBlaze processor is used to interface control logic to on-board analog to digital converter circuitry. The PicoBlaze runs an assembly code written independently and compiled using the assembler which produce a VHDL code to integrate into the design.

The whole design is written in VHDL using Xilinx ISE 9.2 design suite. Each part of the control system is written as a separate module to simplify simulation and debugging process. All modules integrated together, synthesized, implemented, and downloaded to the board for system evaluation.

ISE synthesis and implementation tools report a maximum clock speed of 75MHz and utilization of less than 35% of available device resources. This means that the design could be implemented on a smaller FPGA device to reduce design cost and power consumption.

**EXPERIMENTAL RESULTS**

The three-phase induction motor used in the study is a 380V, 3.81 A, 1.0 kW, 60 Hz and 4 poles, slip ring with the same ratings as used in simulation. The power circuit has been implemented on one board in the lab especially for this paper is shown in Figure 9. This board can be used with any different control devices as microcontroller or PIC controller.

The board circuit consists of the following:

1- Three power triacs (Part # BT139) has a 600V Repetitive peak off-state voltage and 16 A on-state current.
2- Three opto-coplers (Part # TLP3022).
3- 680 current limiting resistors between the triacs and optocopler.
4- One dc power supplies +5/-5V to feed the comparators required for zero crossing.
5- Three operational amplifiers for zero crossing comparators.
6- Three step down transformers to feed the phase voltages to the three comparators.
7- Snubber circuit to protect the power Triacs.
8- Three current sensors (Part# ACS754KCB) with ±150A sensible current range and 13.3mV/A sensitivity.

The experimental waveforms of three-phase voltages in modes 0/2/3, 2/3, and 0/2 are shown in Figures :10 and 11 respectively. The experimental waveforms of three-phase currents in modes 0/2/3 and 2/3 are shown in Figures 12 and 13 respectively. It is clear from these figures that the line current in the mode 2/3 has the lowest harmonic contents and has the near sinusoidal shape.
CONCLUSIONS

Three-phase ac voltage controller has been used to start and control the speed of induction motor by controlling the stator voltage. Computer simulation using PSIM and SIMULINK software has been carried out to configure the performance of the motor in each mode. From simulation results there are three modes of operations depending on the shape of supply currents. These modes namely 0/2, 0/2/3, and 2/3. The variation of speed with firing angle is different in each mode. A digital control system has been implemented using FPGA. The control system detects the mode of operation to provide the switches with the correct firing angle. All components of the control system have been designed on FPGA chip. The operating limits have been determined from the simulation results using PSIM and SIMULINK computer program. The experimental results prove the simulation results and show the superiority of the digital control system.

VII ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>THD</td>
<td>Total Harmonic Distortion</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional Integrator</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>MUX</td>
<td>Multiplexer</td>
</tr>
<tr>
<td>FF</td>
<td>Flip Flop</td>
</tr>
<tr>
<td>LUT</td>
<td>Look Up Table</td>
</tr>
<tr>
<td>VHDL</td>
<td>Very High Speed Integrated Circuits Hardware Description Language</td>
</tr>
</tbody>
</table>

LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f )</td>
<td>Frequency of phase voltage</td>
</tr>
<tr>
<td>( t_c )</td>
<td>Period of FPGA core clock signal</td>
</tr>
<tr>
<td>( f_{cl} )</td>
<td>Frequency of FPGA core clock signal</td>
</tr>
<tr>
<td>( C_{max} )</td>
<td>Maximum value of digital saw-tooth</td>
</tr>
<tr>
<td>( C_t )</td>
<td>Instantaneous value of digital saw-tooth</td>
</tr>
<tr>
<td>( \theta_i )</td>
<td>Instantaneous value of phase voltage angle</td>
</tr>
<tr>
<td>( C_{\theta} )</td>
<td>Saw-tooth value respective to a phase voltage angle of ( \theta )</td>
</tr>
<tr>
<td>( \theta_p )</td>
<td>Duration of firing pulse measured in degrees</td>
</tr>
<tr>
<td>( \Delta C_{\theta} )</td>
<td>Duration of firing pulse as the difference between saw-tooth value respective to firing angle and saw-tooth value respective to the angle where the firing pulse deactivated.</td>
</tr>
<tr>
<td>( e(t) )</td>
<td>Error signal input to analog PI controller</td>
</tr>
<tr>
<td>( u(t) )</td>
<td>Output of analog PI controller</td>
</tr>
<tr>
<td>( k_p )</td>
<td>Proportional constant of analog PI controller</td>
</tr>
<tr>
<td>( k_i )</td>
<td>Integral constant of analog PI controller</td>
</tr>
<tr>
<td>( e(n) )</td>
<td>Error signal input to digital PI controller at sample ( n )</td>
</tr>
<tr>
<td>( e(n-1) )</td>
<td>Error signal input to digital PI controller at sample ( n-1 )</td>
</tr>
<tr>
<td>( u(n) )</td>
<td>Output of digital PI controller at sample ( n )</td>
</tr>
<tr>
<td>( u(n-1) )</td>
<td>Output of digital PI controller at sample ( n-1 )</td>
</tr>
<tr>
<td>( k_1 )</td>
<td>First constant of digital PI controller</td>
</tr>
<tr>
<td>( k_2 )</td>
<td>Second constant of digital PI controller</td>
</tr>
</tbody>
</table>

REFERENCES


Gastli,; M. M. Ahmed, 2005 “ANN-Based Soft Starting of


