CHAPTER 4

INTELLIGENT SPEED CONTROLLER FOR A SWITCHED RELUCTANCE MOTOR DRIVE USING FPGA

4.1 INTRODUCTION

In this chapter, an FPGA based digital speed control scheme is presented, that is developed to overcome the drawbacks existing in the previous speed control schemes, which were proposed for switched reluctance motor (SRM) drives. It is based on discrete P, PI and PID control algorithm, and requires simple mathematical models. The scheme is implemented by using a XC2S300E FPGA. The real-time experimental results given in this chapter show that the speed control method proposed could provide accurate speed control over a wide range of speeds, and can also perform accurately at different operating conditions (steady state/transient operation under soft chopping mode). The closed loop SRM speed control system is seen to achieve 6.2 rpm speed accuracy, depending on the needed operating speed range, with a step response settling time of 0.25 to 1.05 seconds. Complete descriptions of the experimental system along with FPGA implementation are presented.

4.2 SRM DESCRIPTION

A prototype SRM has 6 stator poles and 4 rotor poles as shown in Figure 4.1.
Figure 4.1 6/4 Poles prototype SRM

Its inductance profile is a periodic function with respect to the rotor position and has symmetric structure. The Inductance (L) variation over one rotor pole pitch with constant current magnitude is shown in Figure 4.2. The phase inductance is maximum when the rotor pole is aligned with the stator pole and is minimum when the rotor pole is aligned with the inter polar axis of the stator.

\[
L(\theta) = \begin{cases} 
L_{\min} & \theta_0 \leq \theta < \theta_1 \\
L_{\min} + p\theta & \theta_1 \leq \theta < \beta_s \\
L_{\max} & \beta_s \leq \theta < \beta_r \\
L_{\max} - p(\theta - \beta_r) & \beta_r \leq \theta < \beta_s + \beta_r \\
L_{\min} & \theta_2 \leq \theta < \theta_3
\end{cases}
\]

(4.1)

where

- \( L(\theta) \) - Inductance variation over one rotor pole pitch
- \( L_{\min} \) - Unaligned Inductance (H)
- \( L_{\max} \) - Aligned Inductance (H)
- \( \beta_s \) - Stator pole arc (rad)
- \( \beta_r \) - Rotor pole arc (rad)

\( \beta_r > \beta_s \) and \( p = \frac{L_{\max} - L_{\min}}{\beta_s} \)
Figure 4.2  Inductance ($L$) Variation over one rotor pole pitch with constant current magnitude

The per phase equivalent circuit of the SRM can be drawn as shown in Figure 4.3.

![Per Phase Electrical equivalent circuit of SRM](image)

**Figure 4.3 Per Phase Electrical equivalent circuit of SRM**

\[ V(t) = Ri(t) + L(\theta, i) \frac{di}{dt} + i\omega \frac{dL(\theta, i)}{d\theta} \]  \hspace{1cm} (4.2)

where

- $v(t)$ - is the voltage applied to the phase (v)
- $R$ - is the phase resistance (ohm)
- $Ri(t)$ - is the resistive voltage drop (v)
- $L(\theta, i) \frac{di}{dt}$ - is the static voltage (v)
- $i\omega \frac{dL(\theta, i)}{d\theta}$ - is the rotational voltage or back emf, $E_b$ (v)
Equation (4.2) is valid only when the mutual inductances are neglected. From the above equation, it is understood that the motional voltage of each phase is proportional to the speed and rate of change of inductance with respect to rotor position.

The instantaneous torque (T) produced in the SRM is given by the formula

\[ T = \frac{1}{2} f \frac{dL}{d\theta} \]  \hspace{1cm} (4.3)

From the above equation, the positive torque is produced when the phase is switched on during the rising inductance, i.e. \( \frac{dL}{d\theta} > 0 \), negative torque is produced when the phase is switched on during the falling inductance, i.e. \( \frac{dL}{d\theta} < 0 \) and zero torque is produced when the phase is switched on during the constant inductance, i.e. \( \frac{dL}{d\theta} = 0 \). Hence it is required to energize the stator phase windings at proper rotor position. The exact choice of the turn-on and turn-off angles, determine the ultimate performance of the speed control of the SRM. In order to implement the control algorithm effectively, a high performance digital speed controller has to be implemented. The complete basic block diagram of feedback speed control system and SRM drive is shown in Figure 4.4.

![Figure 4.4 Basic speed feedback control system](image)
4.3 DIGITAL CONTROLLER IMPLEMENTATION

In general, digital controllers can be implemented as digital filters in the following form, where \( k \) is the current sample in time, for a given sample period \( T \)

\[
y(k) = \sum_{i=0}^{n} a_i x(k-i) - \sum_{i=1}^{n} b_i y(k-i)
\]  

(4.4)

in this form \( y(k) \) is the output, \( x(k) \) is the input, \( a_i \) and \( b_i \) are coefficients, or gains, of the controller.

These gains must be selected to produce the desired controller response for a given SRM to be controlled. The structure of this digital controller is illustrated in Figure 4.5. In the Figure 4.5, the \( Z^{-1} \) blocks represent delays of one sample period.

When \( n = 2 \), a second order filter is obtained which can be used to implement second order controllers. This representation for a 2\textsuperscript{nd} order discrete time controller in the sampled time domain is shown below

\[
y(k) = a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) - b_1 y(k-1) - b_2 y(k-2)
\]  

(4.5)

The \( z \)-transform of this gives the following transfer function

\[
D(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}
\]  

(4.6)
These kinds of 2\textsuperscript{nd} order controller are generally implemented using high-speed processors. In this chapter, speed control of SRM utilizes the P, PI and PID algorithm. The proportional control mode gives rapid closed-loop transient response to step changes in the speed reference and fast rejection of speed disturbances. The integral control mode assures that the final SRM speed will match the speed reference. Finally, the derivative control mode reduces transient overshoot and oscillation in the speed when the reference is changed from one value to another. The P, PI and PID algorithm can be expressed in the discrete-time domain as

(i) Discrete Proportional (P) Controller

\[ u(k) = K_p e(k) \]  

(ii) Discrete Proportional and Integral Controller (PI)

\[ u(k) = K_p e(k) + \frac{T}{K_i} S(k) \]
(iii) Proportional, Integral Controller and Derivative controller (PID)

\[ u(k) = K_p e(k) + \frac{T}{K_i} s(k) + K_d \frac{e(k) - e(k-1)}{T} \]  

(4.9)

To eliminate the need to calculate the full summation each in time step the summation is expressed as a running sum

\[ S(k) = S(k-1) + \frac{T}{2} [e(k) + e(k-1)] \]  

(4.10)

where

- \( u(k) \) is the control signal
- \( e(k) \) is the error signal
- \( T \) is the sample period
- \( K_p \) is the proportional mode control gain
- \( K_i \) is the integral mode control gain
- \( K_d \) is the derivative mode control gain

The \( z \)-transform of discrete PID controller gives the transfer function

\[ D(z) = K_p + \frac{K_i T}{2} \left( \frac{z + 1}{z - 1} \right) + \frac{K_d}{T} \left( \frac{z - 1}{z} \right) \]

\[ \left( K_p + K_i T / 2 + K_d / T \right) + \left( -K_p + K_i T / 2 - 2K_d / T \right) z^{-1} \]

\[ D(z) = \frac{\left( K_p z^{-1} + K_i T z^{-2} + K_d \right)}{\left( 1 - z^{-1} \right)} \]  

(4.11)

where

\[ a_0 = K_p + \frac{K_i T}{2} + \frac{K_d}{T}, a_1 = -K_p + \frac{K_i T}{2} - \frac{2K_d}{T} \]

\[ a_2 = \frac{K_p}{T}, b_1 = -1, b_2 = 0 \]

\( a_0, a_1, a_2, b_1 \) and \( b_2 \) are the gains of the digital controller.
4.4 EXPERIMENTAL SETUP

The SRM drive system under control consists of the components illustrated in Figure 4.6. The drive system is made up of several distinct subsystems: the 6/4 pole SRM, a personal computer (PC), the driving circuit, Classic bridge converter, sensing circuitry and the XC2S300E FPGA.

The motor is 1.2 kW, 3000 rpm prototype SRM. The SRM is equipped with a rotor position sensor. The three sensors are mounted on the shaft of the SRM, wherein a combination of infrared LED and photo transistor are used to sense the rotor position to indicate which of the three phases of the SRM is to be excited as the motor runs. One of the rotor position sensors output is also used to calculate the actual speed of the SRM. The controller is implemented by verilog coding and executed by an XC2S300E FPGA.

The control algorithm is written and loaded into the XC2S300E FPGA using the PC. The driver circuit is constructed using totem pole configuration, wherein NPN (3904) and PNP (2907) transistors are used. In order to maintain the current at desired value and adjust the controller coefficients, the actual phase current is measured through current sensor (LEM25-NP). The inputs to the FPGA are the rotor position information and dc bus current. The output of the switching logic section is a sequence of gating signals that are pulse width modulated (PWM) gating signals that are used to drive the classic bridge converter. The power converter is a classic bridge converter utilizing six MOSFET (IRFP360) and fast recovery diodes (MUR3060). The output of the power converter is chopped to get the desired voltage. The XC2S300E FPGA board is designed with 10 bit ADC to implement different control algorithms. Figure 4.7, shows the photograph of
experimental set up for speed control of the prototype 1.2 kw SRM drive using FPGA.

4.5 FPGA BASED CONTROLLER IMPLEMENTATION

Design criteria:

Operating frequency: 1 MHz = 1 \( \mu \)s

Maximum frequency of the signal produced from rotor position sensor is 100Hz.

The counter increments its value automatically with every positive edge of the clock.

![Diagram of experimental system block diagram of a speed controller for the SRM](image)

Figure 4.6 Experimental system block diagram of a speed controller for the SRM
No. of counts produced when sensor signal is high (i.e. for 5 ms): N
Time taken for N counts: N μs
Time for one rotation of the disc: 8N μs
In 60 seconds no of rotations: 60/8N μs
Speed in RPM: 7500000/N

The reference speed is defined in the software and the error calculation is done. The process repeats every 8 cycles of the sensor output. As the speed of the motor varies the on time of the sensor output also changes thereby varying the value of N. The difference between actual and reference speeds gives the error.

The digital controller consists of a sequence of blocks whose functionalities are realized by Verilog programming. The different parts of the controller are listed below.
- Rotor position signal receiver
- Registers
- Positive edge detector
- Counter for speed measurement
- Counter for PWM generation
- Speed comparator
- Commutation logic
- PWM output logic generator
- PID controller

### 4.5.1 Rotor Position Signal Receiver

It receives the position information of the motor and the same is given to the FPGA, then these values are stored in the registers.

```verilog
module (s1,out);
input s1;
output out;
reg out;
out =s1;
endmodule;
```

### 4.5.2 Registers

Registers are used to store the data and also the register acts as a buffer. In this chapter, the registers are used for temporary storage of data.

```verilog
module E_ff(q, data, enable, reset, clock);
output q;
input data, enable, reset, clock;
```
reg q;
always @(posedge clock) // whenever the clock makes a transition to 1
if (reset == 0)
    q = 1'b0;
else if (enable==1)
    q = data;
// implicitly : else q = q:
endmodule

4.5.3 Positive Edge Detector

Detects the positive edge of the sensor signal

input C, D;
output Q;
reg Q;
always @(posedge C)
begin
    Q = D;
end

4.5.4 Counter for Speed Measurement

The counter is used to measure the speed of the SRM and for rotor position calculations.

always@ (posedge clk)
begin
    if (a==0)
        count=0;
4.5.5 Counter for PWM Generation

The pwm is generated according to the duty ratio. The error is calculated from the reference speed and the actual speed. According to that error, we calculate the duty ratio and pwm is generated. For example PWM generated for the 75% duty cycle means the counter is generated the PWM like this

always@(posedge)
if (count < 750)
pwm = 1'b1;
else
pwm = 1'b0;

This pwm contains the ¾ of the on time and ¼ of the off time.

4.5.6 Speed Comparator

It compares the required speed and the actual speed. If the actual speed is less than the required speed then + error value will be generated and
actual speed is greater than the required speed then - error value will be generated. Then the error is fed to the PID controller.

\[
\text{always@}(\text{wac}) \\
\text{err} = \text{wref} - \text{wac}; \\
\text{if (wref} > \text{wac)} \\
\text{pwmout} = 1'b1; \\
\text{else} \\
\text{pwmout} = 1'b0;
\]

### 4.5.7 Commutation Logic

Three position sensor signals are directly taken in to the FPGA. The commutation logic for normal mode is generated with these following statements.

Phase 1 = \neg s2 and s1;  
Phase2 = s3 and \neg s1;  
Phase 3 = s2 and \neg s3;

### 4.5.8 PWM Generation Logic

The value of error is passed through the discretised equation for PID control action. This generates an actuating signal u(k). A fixed PWM at 2 Khz is generated and the value of u(k) will determine the ON time.

\[
\text{er} = \text{wref-wn}; \\
\text{err}=\text{er}/40; \\
\text{uk}=\text{uk}_{-1}+(\text{kp+ki+kd})*\text{err}-(\text{kp+2*kd})*\text{err}_{-1}\text{+kd*err}_{-2}; \\
\text{uk}_{-1}=\text{uk};
\]
err_2=err_1;
err_1=err;

4.5.9 PID Controller

A Proportional + Integral + Derivative controller is implemented using FPGA. \( P = K_p, I = K_i, D = K_d \)

The Process is
1) Set value for \( K_p, K_i, K_d \)
2) Calculate \( M_1, M_2, M_3 \)
3) Error = Reference speed – actual speed
4) \( P = K_p \times e(k); \)
   \( I = K_i \times [I(k-1) + e(k) + e(k-1)]; \)
   \( D = K_d \times [e(k) - e(k-1)]; \)
   \( U(k) = P + I + D; \)
5) \( U_{k-1} = U_k \)
   \( e_{k-1} = e_k \)
   \( e_{k-2} = e_{k-1} \)
6) Out the value of \( U(k) \)

where
\( U(k) \) is the control signal
\( e(k) \) is the error signal
\( e(k-1) \) is the previous index error signal
\( I(k-1) \) is the previous index integral term
\( k \) is the discrete-time domain index variable
\( K_p \) is the proportional mode control gain
\( K_i \) is the integral mode control gain
\( K_d \) is the derivative mode control gain
4.6 RESULTS AND DISCUSSION

The speed control systems were built and tested to evaluate the performance of discrete P, PI and PID control algorithm using FPGA. A prototype MOSFET based classic bridge converter is used to energize the 6/4-pole motor. Digital controller is implemented through FPGA. Figures 4.8 and 4.9 show the PWM and Commutation logic signals for 6/4 poles SRM by using PID controller. Figure 4.10, shows the gate level for PID controller.

Figures 4.11, 4.12 and 4.13 show the experimental results of actual speed, torque and measured phase voltage, phase current at 960 rpm in P, PI and PID control modes during steady state/transient operation under soft chopping operation. In all the cases the dwell angle is kept constant at 30 degrees. In the P control mode, results have shown that, a steady state error is always present on no load and loaded conditions. In the PI control mode, results have shown that, the steady state error is very less after 1.09 secs on load conditions.

![Figure 4.8](image)

**Figure 4.8** PWM and commutation logic signals for 6/4 pole SRM by using PID controller in hard chopping
Figure 4.9  PWM and commutation logic signals for 6/4 pole SRM by using PID controller during soft chopping

Figure 4.10  Gate level for PID control
In the PID control mode, results have shown that, the steady state error becomes less after 0.2 secs in loaded conditions. Also, from the Table 4.1, it is observed that the maximum speed RMSE is 106.3 RPM in P control mode, 9.8 RPM in PI control mode and 19.6 RPM in PID control mode. From the above discussion, it is understood that PI controller is more suitable in applications where very high speed accuracy is required and PID is suitable in applications where quick settling time is required. And it is also found the experimental result that in PID control mode large chattering effect is present.

Figure 4.11 Experimental results during P control. (a) Speed (top), torque (bottom), 0.4 p.u. load, 960 rpm (b) Phase voltage (upper trace), Phase current (lower trace)
Figure 4.12 Experimental results during PI control. (a) Speed (top), torque (bottom), 0.4 p.u. load, 960 rpm (b) Phase voltage (upper trace), Phase current (lower trace)
Figure 4.13 Experimental results during PID control. (a) Speed (top), torque (bottom), 0.4 p.u. load, 960 rpm (b) Phase voltage (upper trace), Phase current (lower trace)
<table>
<thead>
<tr>
<th>Speed in RPM</th>
<th>Speed RMSE in RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P Control</td>
</tr>
<tr>
<td>320</td>
<td>32.6</td>
</tr>
<tr>
<td>580</td>
<td>48.8</td>
</tr>
<tr>
<td>960</td>
<td>106.3</td>
</tr>
<tr>
<td>1200</td>
<td>49.1</td>
</tr>
<tr>
<td>1740</td>
<td>63.6</td>
</tr>
</tbody>
</table>

### 4.7 CONCLUSION

In this chapter, a robust speed controller for a switched Reluctance motor drive using a digital controller has been developed and implemented based on a three-phase classic bridge converter. The controller employs discrete P, PI and PID speed controllers. Experimental results in this chapter show that dc bus voltage can be used to successfully control a three-phase 6/4 pole switched reluctance motor drive system to obtain robust speed holding capability, even under unknown highly dynamic loads. Under load conditions the maximum speed RMSE error was only 9.8 rpm in PI control mode and 19.6 rpm in PID control mode. It is shown that almost perfect robust speed holding could be achieved if the input dc voltage is increased up to 160V (the rated voltage for the motor under test). Most significantly, the implementation of this high-performance speed controller needs only a high speed FPGA, a few logic IC’s, and three current sensors. It can be concluded that discrete PI and PID speed controllers are effective in dealing with the highly non-linear characteristics of the switched reluctance drive system.