HIGH PERFORMANCE DIRECT TORQUE CONTROL
INDUCTION MOTOR DRIVE UTILISING TMS320C31 DIGITAL SIGNAL PROCESSOR

Student: Nik Rumzi Nik Idris
Advising professor: Prof Dr. Abdul Halim Mohd Yatim
Fakulti Kejuruteraan Elektrik
Universiti Teknologi Malaysia
81310 UTM, Skudai, Johor, MALAYSIA

Abstract

Recently, digital signal processors (DSP's) are gaining more popularity in the area of ac motor control, in particular the induction motor drives. The fast number crunching capabilities of DSP's which are traditionally meant for data processing tasks such as image processing and speech recognition, are being utilised to perform on-line calculation of the ac motor variables, such as electromagnetic torque, fluxes, mechanical speed, etc. This project presents the implementation of a high performance induction motor control technique known as direct torque control (DTC). In DTC drives, the de-coupling of the torque and flux components is accomplished by using hysteresis comparators which compares the actual and estimated values of the electromagnetic torque and stator flux. The torque and flux estimated values are calculated based on the sampled motor's terminal variables, i.e. stator voltages and currents. Output of these comparators together with the stator flux orientation are used to select the appropriate voltage vectors of a three phase voltage source inverter. To ensure stability and high bandwidth operation, the sampling period must be made as small as possible which means that a fast calculation of stator flux and torque is required and this is achieved by utilising a DSP controller board, DS1102 from dSpace consisting of TMS320C31 at 60 MHz. To minimise the sampling period of the implemented drive system, a Xilinx FPGA (XC4005E) is used to perform some of the main tasks of DTC drives. The TMS320C31 was programmed in C-language with the motor's terminal variables sampled at 60µs. The drive system was constructed using an insulated gate bipolar transistor based voltage source inverter. Experimental results of this high performance drive system on a ¼ hp standard induction motor are as expected and agree with the theoretical work.

This document was an entry in the 1999 DSP Solutions Challenge, an annual contest organized by TI to encourage students from around the world to find innovative ways to use DSPs. For more information on the TI DSP Solutions Challenge, see TI's World Wide Web site at www.ti.com.
Table of Contents

INTRODUCTION ...........................................................................................................3
PRINCIPLES OF DTC OF IM................................................................................3
  Direct Flux Control ..................................................................................................4
  Direct Torque Control ............................................................................................4
STATOR FLUX AND ELECTROMAGNETIC TORQUE ESTIMATION ......................5
  Voltage model-based stator flux estimation ........................................................6
  Torque estimation ..................................................................................................7
SIMULATION RESULTS .........................................................................................7
IMPLEMENTATION OF DTC ..................................................................................8
  DS1102 controller board .......................................................................................8
  XILINX FPGA – XC4005E ..................................................................................9
  VSI and IM ..........................................................................................................9
EXPERIMENTAL RESULTS ..................................................................................9
CONCLUSION ..........................................................................................................11
APPENDICES ........................................................................................................11
  Appendix A – Induction machine equations .......................................................11
  Appendix B – Source Code ................................................................................12
  Appendix C – Xilinx FPGA schematic design ...................................................17
    Main design - Main.1 .........................................................................................17
    Blanking time – over.1 ......................................................................................18
REFERENCES .........................................................................................................18

Table of Figures

Figure 1 Basic block of DTC ..................................................................................3
Figure 2 Voltage vectors for 3-phase VSI .............................................................4
Figure 3 Six sectors of stator flux plane ...............................................................5
Figure 4 Simulation results with speed-loop .......................................................7
Figure 5 Simulation results with the speed-loop removed ..................................7
Figure 6 Block diagram of the proposed implementation of DTC .......................8
Figure 7 Experimental set-up ...............................................................................10
Figure 8 Experimental results with screen captured from trace plot window with the speed loop........10
Figure 9 Experimental results with screen captured from trace plot window without the speed loop.....11

Table of Tables

Table 1 Voltage vectors table. Top : counterclockwise rotation; bottom clockwise rotation..............5
Table 2 Induction machine parameters .....................................................................9
INTRODUCTION

Almost a century, induction machine (IM) has been the work-horse of industry due to its robustness, low cost, and less maintenance. Before power electronics was introduced, induction motors were mainly used for essentially constant speed applications because of the unavailability of the variable-frequency voltage or current supply. The advancement of power electronics has made it possible to vary the frequency of the voltage or current supplies relatively easy, thus has extended the used of induction motor in variable speed drive applications; but due to the inherent coupling of flux and torque components in IM, it could not provide the torque performance as good as the DC machine.

A couple of decades ago, field oriented control (FOC) of induction motor [1] was introduced that has open a new horizon to the induction motor applications. The method, which uses frame transformation, de-coupled the torque and flux components of the stator current therefore has transformed the performance of IM similar to that of the DC motor. The implementation of this system however is complicated and furthermore FOC, in particularly indirect method which is widely used, is well known to be highly sensitive to parameter variations due to the feed-forward structure of its control system [2]. Another IM control technique known as a direct torque control (DTC), which was introduced about a decade ago, has a relatively simple control structure yet performs at least as good as the FOC technique [3]. It is also known that DTC drive is less sensitive to parameter de-tuning. The DTC drive consists of DTC controller, torque and flux calculator, and a voltage source inverter (VSI) as depicted in Figure 1. The configuration is much simpler than the FOC system due to the absence of frame transformer, pulse width modulator and position encoder, which introduce delays and requires mechanical transducer. The implementation of DTC of IM, although simple in structure, requires a fast processor to perform on-line calculations of electromagnetic torque and stator flux based on sampled terminal variables [4].

Until quite recently, micro-controllers have been used extensively in scalar or vector control of AC machines. However, to obtain a high bandwidth control system and improved performance of AC drives, particularly in servo applications, the use of a fast processor is inevitable. In DTC drive for instance, a relatively slow processor will result in a large electromagnetic torque ripple and can even cause instability. The solution to this problem is to use a processor that can perform fast calculations of the estimated values, and this project will demonstrate how it can be implemented by using a floating point TI DSP TMS320C31. It is hope that the misconception of design engineers who tend to associate DSP’s with data processing tasks will be eliminated. With the current research trend in motion control, undoubtedly, the DSP-based system will dominate the market in years to come. This is particularly true, for example, with the newly introduced TI DSP TMS320C24x family and TI DSP controllers which are specifically designed for digital motor control applications.

Figure 1 Basic block of DTC

PRINCIPLES OF DTC OF IM

Theoretically, if a three phase VSI is connected to an IM, there can be eight possible configurations of six switching devices within the inverter. As a result, there are eight possible input voltage vectors to the IM. The eight voltage vectors, two of which are zero vectors, are depicted in Fig 3.

DTC utilizes the eight possible stator voltage vectors, two of which are zero vectors, to control the stator flux and torque to follow the reference values within the hysteresis bands. The voltage space vector of a three-phase system is given by:
\[ \nabla_s(t) = \frac{2}{\sqrt{3}} \left( v_{sA}(t) + a v_{sB}(t) + a^2 v_{sC}(t) \right), \quad \text{where } a = e^{j\frac{\pi}{3}} \]  
(1)

\( v_{sA}, v_{sB}, \) and \( v_{sC} \) are the instantaneous phase voltages.

For the switching VSI, it can be shown that for a DC link voltage of \( V_d \), the voltage space vector is given by:

\[ \nabla_s(t) = \frac{2}{\sqrt{3}} V_d \left( S_a(t) + S_b(t)a + S_c(t)a^2 \right), \quad \text{where } a = e^{j\frac{\pi}{3}} \]  
(2)

\( S_a(t), S_b(t) \) and \( S_c(t) \) are the switching functions of each leg of the VSI, such that,

\[ S_i = \begin{cases} 1 & \text{when upper switch is on} \\ 0 & \text{when lower switch is on} \end{cases} \quad i = a, b, c \]

**Figure 2. Voltage vectors for 3-phase VSI**

Direct Flux Control

The IM stator voltage equation is given by:

\[ \nabla_s = R_s i_s + \frac{d\Psi_s}{dt} \]  
(3)

Where \( \nabla_s, i_s, \) and \( \Psi_s \) are the stator voltage, current and stator flux space vectors respectively. According to equation (3), if the stator resistance is small and can be neglected, the change in stator flux, \( \Delta \Psi_s \), will follow the stator voltage, i.e.,

\[ \Delta \Psi_s = \nabla_s \Delta t \]  
(4)

This simply means that the tip of the stator flux will follow that of the stator voltage space vector multiplied by the small change in time. Hence if the stator flux space vector (magnitude and angle) is known, its locus can be controlled by selecting appropriate stator voltage vectors. In DTC the stator flux space vector is obtained by calculation utilizing the motor terminal variables (stator voltages and currents). The stator flux is forced to follow the reference value within a hysteresis band by selecting the appropriate stator voltage vector using the hysteresis comparator and selection table.

Direct Torque Control

As shown by Takahashi and Noguchi [3], under a condition of a constant mechanical frequency and stator flux magnitude, when a step increase in the stator angular frequency is applied at \( t=0 \), the rate of
change of torque at time $t=0$ is proportional to the slip frequency of the stator flux with respect to the rotor mechanical speed. Thus,

$$\left. \frac{dT}{dt} \right|_{t=0} \propto \omega_{sl} \bigg|_{t=0} \quad (5)$$

where $\omega_{sl}$ is the angular slip frequency of the stator flux with respect to the rotor mechanical frequency.

This means that the rate of change of torque can be made positive or negative regardless of whether the stator flux is increasing or decreasing. If the torque and stator flux is kept within their hysteresis bands by selecting appropriate voltage vectors, an independent control over the torque and stator flux is accomplished. If the stator flux space vector plane is divided into six sectors or segments (Figure 3), a set of table or rules of which voltage vector should be chosen in a particular sector (either to increase stator flux or to reduce stator flux and either to increase torque or to reduce torque) can be constructed; such table is given by Table 1.

![Figure 3 Six sectors of stator flux plane](image)

**Table 1 Voltage vectors table. Top: counterclockwise rotation; bottom: clockwise rotation**

<table>
<thead>
<tr>
<th></th>
<th>Counter clockwise</th>
<th>Sec I</th>
<th>Sec II</th>
<th>Sec III</th>
<th>Sec IV</th>
<th>Sec V</th>
<th>Sec VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inc Flux</td>
<td>Inc $T(01)$</td>
<td>100</td>
<td>110</td>
<td>010</td>
<td>011</td>
<td>001</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Dec $T(00)$</td>
<td>000</td>
<td>111</td>
<td>000</td>
<td>111</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>Dec Flux</td>
<td>Inc $T(01)$</td>
<td>110</td>
<td>010</td>
<td>011</td>
<td>011</td>
<td>101</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Dec $T(00)$</td>
<td>111</td>
<td>000</td>
<td>111</td>
<td>000</td>
<td>111</td>
<td>000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Clockwise</th>
<th>Sec I</th>
<th>Sec II</th>
<th>Sec III</th>
<th>Sec IV</th>
<th>Sec V</th>
<th>Sec VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inc Flux</td>
<td>Inc $T(10)$</td>
<td>001</td>
<td>101</td>
<td>100</td>
<td>110</td>
<td>010</td>
<td>011</td>
</tr>
<tr>
<td></td>
<td>Dec $T(00)$</td>
<td>000</td>
<td>111</td>
<td>000</td>
<td>111</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>Dec Flux</td>
<td>Inc $T(10)$</td>
<td>011</td>
<td>001</td>
<td>101</td>
<td>100</td>
<td>110</td>
<td>010</td>
</tr>
<tr>
<td></td>
<td>Dec $T(00)$</td>
<td>111</td>
<td>000</td>
<td>111</td>
<td>000</td>
<td>111</td>
<td>000</td>
</tr>
</tbody>
</table>

**STATOR FLUX AND ELECTROMAGNETIC TORQUE ESTIMATION**

To ensure a proper voltage vector selection by the DTC controller, the estimation of stator flux must be accurate. The calculation of the electromagnetic torque too, depends on the accuracy of stator flux estimation. A large deviation of the calculated torque and flux from the true values can result in instability of the drive system since wrong voltage vectors selection is made. Stator flux and torque estimations are based on the IM dynamic equations in the stationary stator reference frame as given in the appendix ((A1)-(A6)). Most of the stator flux calculation is based on voltage model, current model or the combination of both models [5]. The current model requires a precise knowledge of rotor position or
speed in order to accurately estimate the flux, which is not preferred in some industrial environment since mechanical incremental encoder is required to be installed to the IM. Voltage model-based estimator on the other hand only requires the knowledge of stator resistance to perform the estimation. In this report, the voltage model is used for the stator flux estimation.

**Voltage model-based stator flux estimation**

The voltage model-based stator flux is calculated using equation (6) which is obtained from equation (1).

$$\Psi_s = \int (i - i_{\text{Rs}}) dt$$  \hspace{1cm} (6)

In practice, the integrator of equation (6) poses problem of integration drift or sometime referred as ‘integration wind-up, due to the present of even small dc offset in the back Electro Motive Force (emf) of the motor [6]. This is commonly overcome by replacing the integrator with a low-pass filter with appropriate cut-off frequency. Let the $x$ equals to $(v_s - i_s R_s)$ and $y$ equals to stator flux, then, from the LP filter frequency domain transfer function,

$$y = \frac{1}{s + \omega_{\text{cutoff}}} x$$

$$\Rightarrow y(s + \omega_{\text{cutoff}}) = x$$  \hspace{1cm} (7)

Transforming (7) into discrete time domain form,

$$\frac{y_n - y_{n-1}}{T} + y_n \omega_{\text{cutoff}} = x_n$$

$$\Rightarrow y_n = \frac{x_n T + y_{n-1}}{1 + T \omega_{\text{cutoff}}}$$  \hspace{1cm} (8)

Equation (8) shows that the LP filter implementation in discrete time domain is very simple as it is only an integration implementation multiplied by $1/(1 + T \omega_{\text{cutoff}})$, where $T$ is the sampling period. In practical implementation, the variables are separated into their d and q components. In general, the transformation from three phases to 2 phases (d-q) is given by (9). $f_d$ and $f_q$ represents the d and q components while $f_s$, $f_b$ and $f_c$ are the three phases components. This transformation is performed on the 3-phase terminal variables, i.e. stator voltage and current.

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = \begin{bmatrix} 1 & -0.5 \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} f_s \\ f_b \\ f_c \end{bmatrix}$$  \hspace{1cm} (9)

The estimated stator flux d and q components based on equation (8) is given by equation (10)

$$d \text{ component} \quad - \quad \Psi_{sd,n} = \left( (v_{sd,n} - i_{sd,n} R_s) I_{ts} + \Psi_{sd,n-1} \right) \frac{1}{1 + T_s \omega_{\text{cutoff}}}$$

$$q \text{ component} \quad - \quad \Psi_{sq,n} = \left( (v_{sq,n} - i_{sq,n} R_s) I_{ts} + \Psi_{sq,n-1} \right) \frac{1}{1 + T_s \omega_{\text{cutoff}}}$$  \hspace{1cm} (10)
Torque estimation

The torque is calculated based on equation (11) with all quantities referred to the stationary reference frame. \( p \) is the number of pole pairs.

\[
T = 1.5p \left( \Psi_s \cdot (-j\tilde{i}) \right) \tag{11}
\]

In terms of \( d \) and \( q \) components, equation (11) can be written as (12)

\[
T = 1.5p (\Psi_{sd}i_d - \Psi_{sq}i_q) \tag{12}
\]

SIMULATION RESULTS

A simulation on the DTC drive is carried out using Matlab/Simulink simulation package. The parameters of the motor are extracted from the real IM motor used in the experiment (see Table 2). Figure 4 shows the response when the square wave speed reference at 0.45 Hz with an amplitude of 70 rad/s is applied to the drive. Figure 5 shows the response with the speed loop removed and a 2.2 Hz square wave torque command with an amplitude of 0.9 Nm is applied.

Figure 4  Simulation results with speed-loop.

Figure 5  Simulation results with the speed-loop removed
IMPLEMENTATION OF DTC

The DTC control algorithm is perform utilising a DSP controller board DS1102 from dSpace. The board consist of TMS320C31 as the main processor and TMS320P14 as the co-processor. The board provides 4 analogue to digital converter (ADC) and 4 digital to analogue converter (DAC) channels. There are 16 bit programmable digital i/o’s as well as two incremental encoder interface channels. The full specification of the board is given in the appendix. The optimal switching pattern which are selected based on the flux and torque status are stored in a look-up table implemented using a Xilinx FPGA. Figure 6 shows the proposed implementation of the DTC drive.

Figure 6  Block diagram of the proposed implementation of DTC

The following will briefly describe the elements shown in Figure 6.

DS1102 controller board

The DS1102 controller board is built around the Texas Instrument (TI) TMS320C31 floating point DSP. It is interfaced to the host by a block of four 16-bit i/o ports and three 8-bit i/o ports. It contains 128K x 32-bit zero wait state memory that can be accessed by the host even when the DSP is running. The on-board hardware is designed to implement functions which are normally performed by software thus reducing the program overhead [7]. Tasks perform by the board is summarized as follows:

- To obtain the speed, torque and/or stator flux command from the user via the on-board ADCs. To sample the stator current and switching pattern of the 3-phase voltage source inverter thus to construct the stator current and voltage vectors. In this implementation, it is assumed that the dc link voltage is constant. The sampling period for this implementation is 60\(\mu\)s.

- To estimate the stator flux and electromagnetic torque based on the sampled terminal variables. To compare between the actual and estimated stator flux and torque values using three and two level hysterisis comparators. The output of these comparators are passed to the Xilinx FPGA.
• To determine the sector in which the instantaneous stator flux is oriented from the d and q components of the estimated stator flux.

• To obtain the speed of the motor via the on-board incremental encoder interface channel. Since the sampled speed is used only for the closed-loop speed control purpose and the incremental encoder used has a very low resolution, the speed is sampled at a larger sampling period of 10ms.

**XILINX FPGA – XC4005E**

The XILINX FPGA is incorporated in the implementation to reduce the program overhead thus allowing a smaller sampling period of the terminal variables. The tasks performed by the FPGA can be summarized as follows:

• To select the appropriate voltage vectors based on the stator flux and torque error passed by the DS1102 board

• To generate the blanking time required for the upper and lower switching device of a VSI leg.

• To shut-up the system if an over-current is detected.

**VSI and IM**

The VSI including the snubber circuit is constructed using an IGBT-based module rated at 600V and 10A. The switching signals of the 6 devices generated by the FPGA is fed to the driving circuits for conditioning and isolation before they are fed to the gate of the IGBT’s. The DTC is performed to a ¼ hp, 50 Hz, standard induction IM. The machine parameter is given in table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stator resistance</td>
<td>$R_s$</td>
<td>10.9 Ω</td>
</tr>
<tr>
<td>Rotor resistance</td>
<td>$R_r$</td>
<td>9.588 Ω</td>
</tr>
<tr>
<td>Stator self inductance</td>
<td>$L_s$</td>
<td>0.857 H</td>
</tr>
<tr>
<td>Rotor self inductance</td>
<td>$L_r$</td>
<td>0.857 H</td>
</tr>
<tr>
<td>Mutual inductance</td>
<td>$L_m$</td>
<td>0.828 H</td>
</tr>
<tr>
<td>Nominal speed</td>
<td>$\omega_0$</td>
<td>2820 rpm</td>
</tr>
<tr>
<td>Nominal voltage (phase)</td>
<td>$V_s$</td>
<td>138 V</td>
</tr>
<tr>
<td>No of pole pairs</td>
<td>$P$</td>
<td>1</td>
</tr>
</tbody>
</table>

To sense two of the three stator phase currents, the Hall effect current sensors with anti-alias filtering are employed. A 200 pulse per revolution (ppr) incremental encoder is coupled to the end of the IM shaft for speed monitoring purposes.

**EXPERIMENTAL RESULTS**

A standard PC is used as a host to the software development environment. The software used for the system development is:

• TI floating point DSP C-compiler
- TRACE31 (from dSpace), a tool which enable the control variable software to be visualized on the PC screen in real-time.

- COCKPIT31 (from dSpace), a program that can be used to display or update software parameter values stored in the DSP’s memory.

The overall experimental set-up is shown in Figure 7.

**Figure 7 Experimental set-up**

![Experimental set-up diagram]

**Figure 8 Experimental results with screen captured from trace plot window with the speed loop.**

![Experimental results screenshot]
The drive was given a square wave speed reference of 0.45 Hz with amplitude of 70 rad/s. Figure 8 shows the speed and speed reference (speed_ref), estimated torque (T), d-axis estimated flux (flxde), and d-axis current (id) captured from the TRACE31 plot window of the PC screen. The speed loop was next removed, and a square wave torque reference of 2.2 Hz with an amplitude of 0.9 Nm was applied. The response was captured and depicted in Figure 9. Both figures show that excellent torque and speed responses are accomplished.

**Figure 9** Experimental results with screen captured from trace plot window without the speed loop.

**CONCLUSION**

The Direct Torque Control of IM, although has a relatively simple structure, requires the on-line calculations of the estimated torque and stator flux which can be made possible by using a fast digital signal processor. The project has demonstrated the implementation of this high performance direct torque control technique utilizing the floating point TI DSP, TMS320C31. The DTC algorithm is implemented using a high level C-language with a sampling period of 60µs. The experimental results show that an excellent torque response is achieved and agree with the theoretical and simulation results. Based on the current research trend in ac motor control, it is anticipated that the DSP’s will be used extensively in ac motor control in years to come.

**APPENDICES**

**Appendix A – Induction machine equations**

Stator voltage equation

\[ \nabla_s = R_s \vec{i}_s + \frac{d\varphi_s}{dt} \]  \hspace{1cm} (A1)

Rotor voltage equation

\[ 0 = R_r \vec{i}_r + j\omega_r \varphi_r + \frac{d\varphi_r}{dt} \]  \hspace{1cm} (A2)

Where,

\[ \varphi_s = L_s \vec{i}_s + L_m \vec{i}_r \]  \hspace{1cm} (A3)
\[ \vec{\Psi}_r = L_r \hat{\vec{I}}_r + L_m \hat{\vec{I}}_s \]  \hspace{1cm} (A4)

Electromagnetic Torque
\[ t_{elec} = \frac{3}{2} L_m \hat{\vec{I}}_r \times \hat{\vec{I}}_r \]  \hspace{1cm} (A5)

Mechanical dynamic
\[ \frac{2}{p} \frac{d\alpha_r}{dt} = t_{elec} - t_{load} \]  \hspace{1cm} (A6)

where,
\[ \vec{v}_r, \vec{i}_r \]  Stator's voltage and current space phasors in stationary reference frame
\[ \hat{\vec{I}}_r \]  Rotor's current space phasor in stationary reference frame
\[ \vec{\Psi}_r, \vec{\Psi}_s \]  Stator and rotor flux linkage space phasors in stationary reference frame

T  Electromagnetic torque
\( L_r, L_i \)  Stator and rotor self inductance
\( R_s, R_i \)  Stator and rotor resistance
\( L_m \)  Mutual Inductance
\( p \)  No. of poles
\( \omega_r \)  Rotor frequency, rad/s
\( \omega_e \)  Synchronous frequency, rad/s
\( s \)  slip
\( J \)  Moment of inertia

Appendix B – Source Code

```c
#include <brtenv.h>           /* basic real-time environment */
#include <math.h>
#include <dtc_3.h>

/*-------------------------------------------------------------*/
#define DT 55e-6              /* sampling step size */
#define DT2 10e-3             /* sampling step size for speed*/

/* error flag for CHKERR at last dual-port memory location */
volatile int *error = (int *) (DP_MEM_BASE + DP_MEM_SIZE - 1);

/* timer0 interrupt service routine */
void isr_t0()
{
    int I, angint, angint1, mask3, mask5;
    float upl=0.08;  /* upper band for torque */
    float uplf=0.0173;  /* upper band for flux */
```
float upll=0; /* upper band for torque */
float lol=0; /* lower band for torque */
float lolf=-0.0173; /* lower band for flux */
float loll=-0.08; /* lower band for flux */
float ie_count=0;
unsigned int IO_port_read; /* I/O-port status */

begin_isr_t0(*error); /* overload check */
service_mtrace("0"); /* Starting mtrace for DS1102 board */

input(u);

/* Offsets for the sensed phase current */
u3= u3 - offs1;
u4 = u4 - offs2;
u5= (-u3 - u4);

/* 3-phase to 2-phase transformation*/
id = (0.6666667*u4)-(0.3333333*(u5+u3));
iq = (0.5773503*(u5-u3));

/* read I/O-port of the switching pattern from Xilinx*/
IO_port_read = ds1102_p14_pin_io_in();
for(i=0; i<=2; i++){
    /* evaluate I/O-port pin status */
    port_pin_read[i] = ( (IO_port_read & (0x2000 << i)) != 0 );
}

/* Stator d-q voltages with DC link of 120V DC */
v6 = (80*port_pin_read[0])-(40*(port_pin_read[1]+port_pin_read[2]));
vq = 69.282*(port_pin_read[1]-port_pin_read[2]);

/* Stator flux estimation using the voltage model with LP filter */
flxd = (flxd + (vd - id*10.9)*DT)*0.9989;
flxq = (flxq + (vq - iq*10.9)*DT)*0.9989;

/* The following is to estimate the operating frequency, wsp */
vxd = (vd - id*10.9);
vxq = (vq - iq*10.9);
ws1=(vxq*flxdk-vxd*flxqk);
ws2=Flx*Flx;
ws=ws1/ws2;

/* Low pass filtering to smooth out wsp */
wsp = (wsp + 2.5*ws*DT)* 0.99986;

/* Check for the flag to initiate the compensation */
if(flag>5){
    flag=7;
    flxdk = delflxd; /* Initiate the compensation */
    flxqk = delflxq;
} else {
    flag=0;
    flxdk = flxd; /* No compensation */
    flxqk = flxq;
}
/* Torque calculation */
T = 1.5*(flxde*iq-flxqe*id);

/* Calculates the flux magnitude*/
Flx = sqrt(flxde*flxde + flxqe*flxqe);

/* Determine the stator flux orientation to be passed to pin 17, 18
and 19 of Xilinx FPGA*/
flx_ = 0.57735027*flxde;

if(flxde >= 0){
    if(flxqe >= 0){
        if(flxqe >= flx_){
            angint = 0xfe02; /* sector=3 */
        } else angint=0xfe01; /* sector=2 */
    } else {
        if(flxqe <= -flx_){
            angint = 0xfe00; /* sector=1 */
        } else angint=0xfe01; /* sector=2 */
    }
} else {
    if(flxqe >=0){
        if(flxqe >= -flx_){
            angint = 0xfe03; /* sector=4 */
        } else angint=0xfe04; /* sector=5 */
    } else {
        if(flxqe <= flx_){
            angint=0xfe05; /* sector=6 */
        } else angint=0xfe04; /* sector=5 */
    }
}

angint = angint | 0xfe00;

/* Closed loop speed control system */
speed_error = speed_ref-speed;
Propd = (speed_error)*Kp;

speed_error_int = speed_error_int + (speed_error)*DT*Ki;
Tstat = Propd + speed_error_int; /* PI controller output */

if(Tstat>1)Tstat=1; /* To limit the torque reference to 1 N-m */
else if(Tstat<-1)Tstat=-1;
else Tstat=Tstat;

/* Torque error */
Terr_old = Terr;
Terr = Tstat - T;

******** Three-level torque hysterisis: */

/* Error increasing and 1*/
if((Terr_old < Terr)&&(mask1==0xfbff))
    {mask1=0xfbff; flagm=1;}
/* Error increasing and 0 */
if((Terr_old < Terr)&&(mask1==0xf9ff))
    {mask1=0xf9ff; flagm=0;}

HIGH PERFORMANCE DIRECT TORQUE CONTROL INDUCTION MOTOR DRIVE UTILISING TMS320C31 DIGITAL SIGNAL PROCESSOR
if(Terr >= upl){mask1=0xfbff;flagm=1;
else {mask1=0xf9ff;flagm=0;}}

/* Error decreasing and 0 */
if((Terr_old > Terr) && (mask1==0xf9ff))
    {mask1=0xf9ff;flagm=0;}
/* Error decreasing and 1 */
if((Terr_old > Terr) && (mask1==0xfbff))
    if(Terr >= lol){mask1=0xfbff;flagm=1;
    else {mask1=0xf9ff;flagm=0;}}
/* Negative error */

/* Error increasinga and 0 */
if((Terr_old < Terr) && (mask2==0xf9ff))
    {mask2=0xf9ff;flagm1=0;}
/* Error increasing and -1 */
if((Terr_old < Terr) && (mask2==0xfdff)){
    if(Terr >= upll){mask2=0xf9ff;flagm1=0;
    else {mask2=0xfdff;flagm1=-1;}}
/* Error decreasing and -1 */
if((Terr_old > Terr) && (mask2==0xfdff))
    {mask2=0xfdff;flagm1=-1;}
/* Error decreasing and 0 */
if((Terr_old > Terr) && (mask2==0xf9ff)){
    if(Terr >= loll){mask2=0xf9ff;flagm1=0;
    else {mask2=0xfdff;flagm1=-1;}}

if(Terr_old == Terr){mask2=mask2;
    mask1=mask1;}

/***********************************************************/
mask3 = mask1 | mask2;
flagm2 = flagm1 + flagm;

fluxerr_old=fluxerr;
fluxerr = Flx - fluxref;

/******** Two-level flux hysterisis: **********************
if((fluxerr_old < fluxerr) && (mask4==0xffff))
    mask4=0xffff;
if((fluxerr_old < fluxerr) && (mask4==0xf7ff)){
    if(fluxerr > uplf)mask4=0xffff;
    else mask4=0xf7ff;}
if((fluxerr_old > fluxerr) && (mask4==0xf7ff))
    mask4=0xf7ff;
if((fluxerr_old > fluxerr) && (mask4==0xffff)){
    if(fluxerr > lolf)mask4=0xffff;
    else mask4=0xf7ff;}
/**************************************************************/
mask5 = mask3 & mask4;

angint1 = angint & mask5;
/*writing to pin 17.18 &19 of XC4005E */
ds1102_p14_pin_io_write(angint1);
/*output to DAC (oscilloscope)*/
output1; output2; output3;output4;

end_isr_t0();    /***************************************************************************/

void isr_t1()
{% begin_isr_t1(*error);    /* overload check  */

/* the count from ds1102_inc() is 4 times */
   ie_count=(ds1102_inc(1)*8388608);
/* speed in rad/s of the 200 p.p.r. */
speed=ie_count*(3.1415926)/4;
ds1102_inc_clear_counter(1);    /* clearing the counter */
}
end_isr_t1();    /***************************************************************************/

main()
{% int i;
   float u3t=0;
   float u4t=0;
   float speedt=0;
   init();
   *error = NO_ERROR;    /* initialize error flag */

ds1102_p14_pin_io_init(0x1fff);
/* initialise switch status array */
for(i=0; i<=2; i++){
   port_pin_read[i] = 0;
}

/**** Calculating the offsets for the currents **********************/
for(i=0; i<=100000; i++){
   ds1102_ad_start();
   u3 = ds1102_ad(3)*4.1667;
   u4 = ds1102_ad(4)*4.1667;
   u5=(-u3 - u4);
   u3t = u3t + u3;
   u4t = u4t + u4;
}
offs1 = u3t/100001; offs2 = u4t/100001;
/*************************************************************************/
start_isr_t0(DT);    /* initialize sampling clock timer1 */
start_isr_t1(DT2);  /* initialize sampling clock timer2 */

while (*error == NO_ERROR)
    /* background process */
    service_cockpit(); /* call COCKPIT code */

Appendix C – Xilinx FPGA schematic design

Main design - Main.1
REFERENCES


TI is a trademark of Texas Instruments Incorporated.
DSpace is a trademark of dSpace GmbH, Germany.
MATLAB and SIMULINK is a trade mark of The Mathworks, Inc.

HIGH PERFORMANCE DIRECT TORQUE CONTROL INDUCTION MOTOR DRIVE UTILISING TMS320C31 DIGITAL SIGNAL PROCESSOR