FPGA Implementation Vector Control of Tandem Converter Fed Induction Machine

József Vásárhelyi[1], Mária Imecs[2], Csaba Szabó[2], Ioan Iov Incze[2]

[1] Egyetemváros, University of Miskolc, H-3515 Miskolc, Hungary
vajo@mazsola.iit.uni-miskolc.hu
[2] Technical University of Cluj-Napoca, P. O Box 99, Cluj-Napoca, Romania
{csaba.szabo, ioan.incze, imecs}@edr.utcluj.ro

Abstract: The paper focuses on the implementation in Field Programmable Gate Arrays (FPGA) of vector control systems of the induction motor supplied from the tandem (hybrid) static-frequency converter. The paper tries to give a synthesis of the tandem vector control structure implementation. The implementation was made using the library elements for rapid prototyping of vector control systems for induction motor. The implementation analysis is presented.

Keywords: FPGA, embedded system, vector control, tandem inverted, motion control.

1 Introduction

There are two types of digital signal processing algorithms: filtering algorithms and signal analysis algorithms. These algorithms are based on either differential equations (recursive and non-recursive), or discrete Fourier transformation (DFT), and can be implemented in any of the following forms: hardware, firmware, and software.

In the software approach, the algorithm is implemented as a computer program on a general-purpose computer such as a workstation, a minicomputer, a personal computer, a programmable DSP chip, or a microcontroller or other hardware support.

In the hardware approach, the algorithm is implemented using digital circuitry, such as the shift register to provide the delaying operation, the digital multiplier, and the digital adder (such as Field Programmable Gate Arrays - FPGA). Alternatively, a special purpose VLSI chip may be designed and fabricated to implement a specific filtering algorithm (i.e. Application Specific Integrated Circuit - ASIC).

Finally, in the firmware approach, the algorithm is implemented using both hardware and software solutions. Additional control circuitries, and storage registers, are usually needed in the final hardware or firmware realization.

This paper presents the implementation aspects and some results of digital signal processing (DSP) algorithms for vector control for the tandem inverter.

1.1 Vector Control of the Tandem Inverter

An alternative solution for medium- and high-power AC drives is the "tandem" static frequency converter (SFC) fed induction motor. This configuration is a hybrid SFC, which combines the advantages of two, parallel working, different types and different power ranges DC-link converters. A large power Current Source Inverter (CSI), operating in Pulse Amplitude Modulation (PAM) converts the active power, and a small power Voltage Source Inverter (VSI) working in Pulse Width Modulation (PWM) and supplies the reactive power required for improving the quality of the motor currents [9], [11].

To obtain the best dynamic behaviour the control of the tandem-converter-fed induction motor can be achieved using conventional vector-control structures. The tandem converter needs different control strategies depending on the character of the working component-converter and on the modulation procedure used for the VSI [11].

The tandem converter needs different control strategies depending on the type of the PWM procedure used for the VSI. The selected PWM procedure can change the source character of the VSI and of the tandem converter, too [12]. The open-loop voltage-control PWM procedures, i.e. carrier wave or Space-Vector Modulation (SVM), keep the voltage-source character of the VSI, but using closed-loop current-control PWM procedures (e.g. the common bang-bang current control) the behaviour of the VSI becomes of current-source character.

1.1.1 Current Controlled PWM-VSI-Fed Induction Motor with Rotor-Field Orientation

Due to the voltage-source character of the tandem converter, the motor absorbs freely its stator currents. Consequently, the **VSI** will be the actuator ensuring the vector control of the induction motor drive. It is possible to apply the common PWM procedures (voltage- or current controlled ones) characteristic to the **VSI**.

Applying to the **VSI** current-controlled-PWM, in manner of the "bang-bang" converter, the tandem-converter-fed motor will be controlled in fact in current. Constant switching frequency is obtained using synchronized on-off switching controllers. The above-mentioned procedures are appropriate for field-orientation-based tandem-fed drives.

In Figure 1 the induction motor operates supplied from the both converters in tandem mode. Because of the difficulties encountered by direct measurement of the modulated-voltage waves, the stator voltage is identified in block ***V_sId*** using the measured DC-link voltage and the state of inverter switches according to the PWM logic taking into account the voltage losses on semiconductor devices, too.

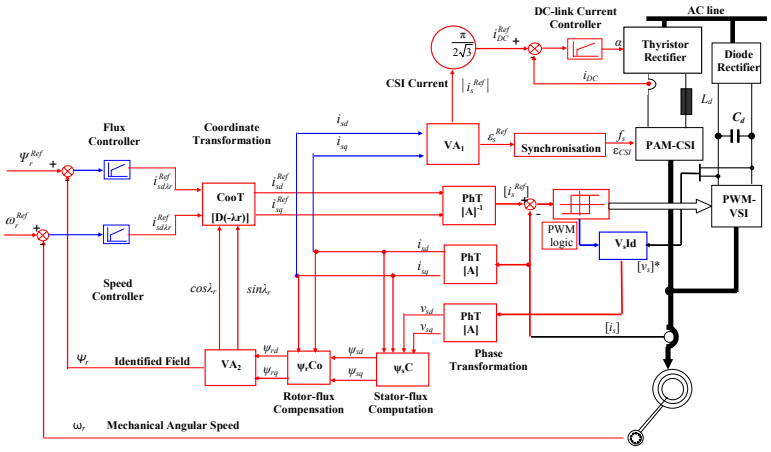

Figure 1

Rotor-field Oriented Vector Control System with Current Feedback Modulation for the Tandem Converter-Fed Induction Motor [13]

Based on the stator-voltage and current components transformed in *d-q* reference frame, the block ***$\Psi_s C$*** integrates the natural stator-voltage equations yielding at its outputs the stator-flux *d-q* components. In order to obtain the orientation flux, the block ***$\Psi_r Co$*** compensates the stator flux. The vector-analyzer ***VA_2*** computes the amplitude and the angular position of the orientation field. The reference values of the stator-current space-phasor components are obtained from the flux- and speed-control loops. After the coordinate transformation, they will be transformed to the three-phase references of the hysteresis dead band current-controllers [9] and [13].

2 Vector Control Structure Analyses

The creation of a library for modelling and rapid prototyping of vector control system for AC drives was motivated by the fact that the FPGA abundant resources allow the implementations in low priced FPGA chips. This possibility gave the idea to implement a simulation and also an implementation library (using Matlab Simulink), which is completely parametrical and any change on the vector control

system's structure can be applied very fast and easy in the implementation hardware. The elements of the library are the most common modules of vector control systems, and each present a standalone unit in the library.

The analysis of the vector control schemes and especially the vector control structures from the point of view of modularity were presented in detail in [10].

The equations of vector control schemes can be decomposed in elementary mathematical operations [10]. What is more, these elementary operations can be combined in the most used DSP function "multiply and accumulate":

$$c = \sum_{i=1}^{k} (a_i \times b_i) \tag{1}$$

The difference between the DSP and FPGA implementation of the 'multiply and accumulate' (MAC) is that in the case of latter one the operations from equation (1) are executed in parallel and not sequentially. In such a way, the execution time is reduced by the parallel computation. Using this feature of the FPGAs, we can make a decomposition of the vector control system in MAC elementary functions. [10] Taking into account these circumstances, one can use different types of implementation topologies: sequential, parallel and the combination of the both types.

The *parallel implementation* of the algorithm results in very fast execution speed. For this reason, the sampling period can be decreased until the technology and the PID controllers allow it. The parallel computation of equations control equations give a significant improvement compared to the DSP sequential computation. The parallel implementation method disadvantages could be the intensive hardware resource consuming and the price paid for a chip.

3 IP Core Implementation

There are some major advantages of using a pre-designed parametrical model when implementation is targeted. These advantages are:

- The implementation time of the simulation model is short, as the simulation model is the implementation itself. This can be done with the translation of the cores in configuration data (using the Xilinx development environment).
- The computation speed increase. This results from the parallel implementation of computation algorithm of both components (d, q) and the parallel computation of each IP core. This is a significant advantage compared to the DSP sequential implementations.

- The parameters of each IP core element can be adjusted easily to any AC motor characteristics. Even the data format can be modified if necessary.
- Flexibility in implementation: each IP core can be translated separately and the vector control system can be translated as a whole.
- The targeted device can be changed if necessary.
- The optimisation of the IP elements is made for speed or/and area, which are characteristic to FPGA implementations.

In the following, we will present some of the implemented IP library elements in details, while the implementation results will be presented in Table 1.

All the equations in the following sections are referred as per unit equations. The implementation characteristics of the implemented library elements are characterised from point of view such as time delay introduced by the module in the control loop and hardware resources occupied in the FPGA.

The characteristic modules of the vector control system will be presented first. These modules are the direct and reverse phase transformation, vector analyser, and coordinate transformation.

3.1 Direct Phase Transformation Block PHT[A]

The so-called direct Park transformations execute the change of variable from the three-phase quantities g_a, g_b, g_c to another three ones – g_d, g_q, g_0. They are the two components of the space phasor in the complex plane and the corresponding zero-sequence component. The implementation is based on equation:

$$\underline{g} = k_{ph}(g_a + \underline{a} \cdot g_b + \underline{a}^2 \cdot g_c), \tag{2}$$

$$k_{Ph} = \frac{2}{3} \tag{3}$$

$$\begin{cases} g_d = g_a - g_0; \\ g_q = \frac{1}{\sqrt{3}}(g_b - g_c); \end{cases} \tag{4}$$

$$g_0 = \frac{g_a + g_b + g_c}{3}; \tag{5}$$

To keep generality of the implementation equations (3) and (5) were realised. Figure shows the implementation.

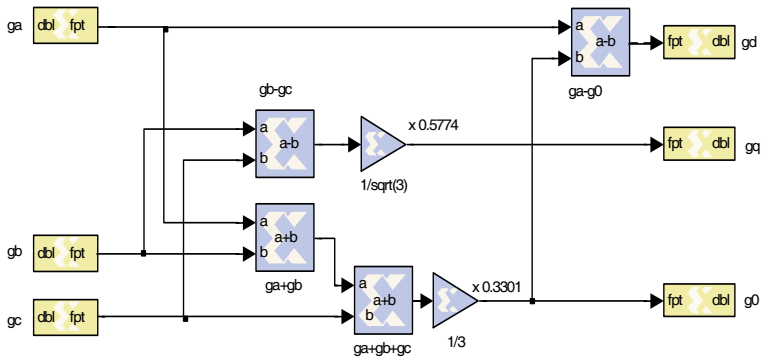

Figure 2
System Generator Implementation of the PHT[A]

The analyses of the PHT[A] block is made after the translation of the block in VHDL language and after the implementation. As result of these steps, we obtained the following implementation report (see Figure 3).

From the implementation report presented in Figure 3 it results the resources used for the implementation of the PHT[A] module. The total number of slices used for the implementation is 152, which is only 5% of the chip targeted. The number of LUT used as 4 inputs LUT is 277. The amount of LUT used to implement combinatorial functions is 253 and 24 LUT are used for routing.

Design Summary

Number of errors:	0		
Number of Slices:	152 out of	3,008	5%
Number of Slices containing unrelated logic:	0 out of	152	0%
Total Number 4 input LUTs:	277 out of	6,016	4%
Number used as LUTs:	253		
Number used as a route-thru:	24		

Figure 3
Design implementation summary of IP core PHT[A]

The ‘pad to pad’ delay (Figure 4), which is the module ‘input to output’ delay, is 27ns. This delay is composed from the delay introduced by the implementation logic (15.7ns) and the delay introduced by the routing nets (11.3ns).

```
--------------------------------------------------------------------------------
Release 5.1i - Trace F.23
Copyright (c) 1995-2002 Xilinx, Inc.  All rights reserved.
Design file:              pht.ncd
Physical constraint file: pht.pcf
Device,speed:             xc2vp4,-5 (ADVANCED 1.66 2002-07-03)
--------------------------------------------------------------------------------
All values displayed in nanoseconds (ns)
Pad to Pad
---------------+---------------+---------+
Source Pad     |Destination Pad|  Delay  |
---------------+---------------+---------+
ga<0>          |g0<0>          |   24.871|
ga<0>          |gd<7>          |   29.030|
ga<11>         |g0<2>          |   22.971|
ga<12>         |gd<15>         |   28.114|
ga<14>         |g0<0>          |   22.302|
gb<10>         |gq<1>          |   14.269|
---------------+---------------+---------+
Analysis completed Thu Jan 30 17:07:15 2003
--------------------------------------------------------------------------------
Total                  27.060ns (15.744ns logic, 11.316ns route)
                                (58.2% logic, 41.8% route)
```

Figure 4

Post Place and route static timing report of module PHT[A]

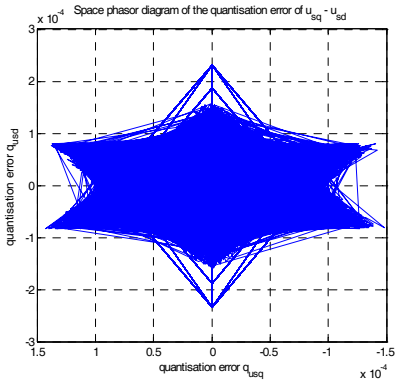

Figure 5

Quantisation error of the block PHT[A] for the variables u_{sd} - u_{sq}

From the implementation results the quantisation error introduced by this module:

$$\Delta\varepsilon_d \in \left[-1.5\cdot10^{-4}, 1.5\cdot10^{-4}\right] \text{ and } \Delta\varepsilon_q \in \left[-1.5\cdot10^{-4}, 3\cdot10^{-4}\right] \quad (6)$$

where $\Delta\varepsilon_{d/q}$ is the quantisation error. We can say that comparing the input value range of the voltage/current value is acceptable. In the simplified PHT[A] module implementation we have considered the zero sequence component g_0=0. Since this

implementation omits the zero sequence components, it is much simple, and for this reason, the consumed hardware resources are lower. The implementation contains only a subtraction module and a constant multiplier.

3.2 Vector Analyzer Module

While all the other modules were implemented using System Generator library elements, the implementation of the VA module is implemented using Xilinx Core Generator and VHDL. The operations were implemented in VHDL code and then the code was imported for simulation/implementation into the Simulink model. The VA module was implemented in accordance with equations:

$$\left|\underline{g}\right| = \sqrt{g_d^2 + g_q^2}; \text{ and } \sin\alpha = \frac{g_q}{\left|\underline{g}\right|}; \quad \cos\alpha = \frac{g_d}{\left|\underline{g}\right|}. \tag{7}$$

In Figure 6 there are two multiplexers, who allow implement/simulate VHDL code in Simulink.

The square operation is made by the multipliers Mult and Mult1, and the addition of the two squares are made by the block AddSub. The module finally is computed by the square root (sqrt) operation and implemented in VHDL code.

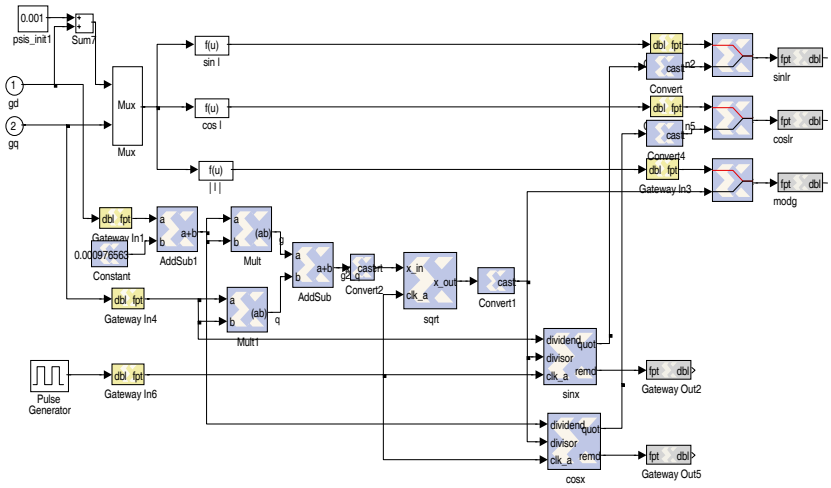

Figure 6
VA module implementation and simulation model

The run-time computation of the sine and cosine functions computed in accordance with equation (8) by the blocks sin(α) and cos(α). Table 1 presents the IP module implementation results (see next section).

4 Implementation of Rotor-field Oriented Vector Control System

The implementations of the reconfigurable vector control system contain three major elements that can clearly separate in three independent boards. These three elements are the configuration supervisor and/or AD control card, the 6 channel A/D converter and the control-system implementation FPGA card(s).

The interaction between these cards is based on the vector control system structure and reconfiguration algorithms. The six-channel A/D converter is a DEM ADS7864 Texas Instruments board, with six simultaneous sampling channels, 2μs total throughput per channel, 1MHz effective sampling period and 12 bit accuracy. The A/D-converter sampling period limits the sampling period of the control system. Even if the FPGA allow faster computing times, the A/D converter limits the sampling period to 12μs. This time results from the fact that the converted data are transmitted serially. The FPGA boards implements the vector control system. The board is composed of a Xilinx Spartan 3 XC3S-5FT256 chip. Table 1 presents the IP library elements implementation.

The table shows the LC (Logic Cell) slices consumed by each block, the worth path delay introduced by the block. There is shown the quantisation error where the module can be characterised by this parameter. Another characteristic of the modules is the maximum working frequency if sequential circuits implement the module.

The table shows, that the VA consumed the maximum number of logic cells. The maximum delay introduced is around 48ns. The lowest working frequency is 48MHz. These values were obtained when each module was independently implemented. In the above table, n.a. means not available, such as for the combinatorial modules the maximum clock frequency is unavailable.

The most simple vector control structure is the vector control system with current feedback modulation VSI-fed induction motor with rotor-FOC. We considered this vector control structure for implementation. Table 1 shows also the implementation results of the mentioned structure. The introduced time delay is 48ns and allows a maximum working frequency of 38MHz.

We can say that the 4496 slices consumed for the implementation allow the implementation in the Spartan 3 chip.

Conclusions

This paper presented the implementation in FPGA of vector control system for tandem converted fed FOC system. It presented an intellectual property library, which allows the simulation and rapid prototyping of any vector control system. The library is extensible to sensorless and/or intelligent control systems, but this was not subject of this paper.

Table 1
Characteristics of the implemented vector control library modules

Module name	Slices needed for implementation	Worth path delay introduced td[ns]	Quantisation error	Max. working f[MHz]
Direct Phase Transformation Block	152	27.00	q_e<1.5*10-4	n.a
Reverse Phase Transformation Block	217	4.90	~0	n.a
Stator+Rotor-flux Compensation	1000	41.70	-0.02<q_e<0.1	42,00
Vector Analyser	1995	16.60	n.a	166,90
Coordinate Transformation	25	10.00	<1. 10-4	n.a
Space Vector Modulation	27	3.06	n.a.	n.a
Current feedback Modulation	77	31.20	n.a	224.15
Flux Controller	24	4.00	~0.6*10-4	n.a
Speed Controller	24	4.00	~0.6*10-4	n.a
Flux+Speed Controller	135	13.73	qeflux ≅ -6*10-3 -0.1<qespeed<0.16	128.18
DC-link Current Controller	298	24.00		86.70
Reconfiguration Multiplexer	5	3.12	0	n.a
CSI current constant multiplier	79	16.99	-2*10-5<q_e<5*10-5	n.a
Total Estimated resources	4496 slices 1058 FF 9334 LUT	48.29	n.a	38.45

From the implementation results of each element one can estimate the hardware needed to implement the vector control system, the maximum delay introduced by the system, which influence the sampling period of the control system, and can analyse the quantisation error of each module, which influence the computation accuracy.

We did not intend to compare the performances of the implemented control system compared to other implementations. We concentrated only on the reconfigurable vector control implementation. We analysed also the conditions they are able to implement vector control and their possible disadvantages.

The control system presents modularity and this modularity can be used to create a module library. The modularity can help the reconfiguration process.

References

[1] Kelemen A., Imecs M., *Vector Control of AC Drives*, OMIKK Publisher Budapest, ISBN 963 593 140 9, Budapest, Hungary, 1992

[2] Aubépart F., Poure P., Braun F., *Contribution to System-on-Chip in motion control: VLSI design of a digital controller for an induction machine,* PCIM 2001 Power Electronics Intelligent Motion Power Quality, June 19-21, 2001 Nuremberg, Germany, pp. 211-216

[3] Beierke S., *Rapid Implementation of a Field-Oriented Control Method for Fixed-Point DSP Controlled Asynchronous Servo Drives,* European Power Electronics Chapter Symposium on Electric Drive Design and Applications, organized by EPFL Lausanne, Switzerland; 19-20 Oct. 1994, pp. 361-365

[4] Belmimoun M. H., Monmasson E., Sambuis E., *Modularity in Code Development for DTSFC Algorithms Implementation on a Fixed-Point DSP,* PCIM 2002 Power Electronics, Intelligent Motion, Power Quality, May 12-16, 2002 Nuremberg, Germany, pp.129-135

[5] Cirstea M., McCormick M., *Intelligent ASIC control of power electronics,* Proceedings of PCIM '98 – Conference on Power Conversion and Intelligent Motion, Nuremberg, Germany, May 1998, pp. 95-99

[6] Cirstea M., Aounis A., McCormick M., *Rapid Prototyping of Induction Motor Vector Control System Based on Reusable VHDL Digital Architectures and FPGA Implementation*, PCIM 2002 Power Electronics Intelligent Motion Power Quality, May 14-16, 2002 Nuremberg, Germany, pp. 199-203

[7] Imecs M., Bikfalvi P, Nedevschi S., Vásárhelyi J.: *Implementation of a Configurable Controller for an AC Drive Control a Case Study*, Proceedings of IEEE Symposium on FCCM 2000, Napa, California, USA, 2000 pp. 323-324

[8] Imecs M., Bikfalvi P, Nedevschi S., Vásárhelyi J.*: Implementation of a Reconfigurable Controller for an AC Drive Control,* on the CD of the Proceeding of Control 2000 Conference CD-ROM and Book of Abstracts Control 2000, Oxford, 2000. pp. 199

[9] Imecs Maria, Incze I. I., Vásárhelyi J., Szabó Cs.: *Tandem Converter Fed Induction Motor Drive Controlled With Re-Configurable Vector Control System,* PCIM 2001 Power Electronics Intelligent Motion Power Quality, June 19-21, 2001 Nuremberg, Germany, pp. 341-346 and CD-ROM

[10] Imecs M., Vásárhelyi J., Incze J. J., Szabó Cs.: *Vector Control Of Tandem Converter Fed Induction Motor Drive Using Configurable System On A Chip,* INES 2001 IEEE International Conference on Intelligent Engineering Systems, September 16-18, 2001, Helsinki-Stockholm-Helsinki, Finland-Sweden, pp. 489-495

[11] Imecs M., Incze I. I., Szabó Cs.: *Control Strategies of Induction Motor Fed by a Tandem DC Link Frequency Converter,* European Conference on

Power Electronics and Application, EPE 2001, Graz, 27-29 Aug. 2001, Book of Abstracts p. L1b-7 and CD-ROM

[12] Imecs M., Trzynadlowski A. M., Patriciu N., Radian Kreiszer M.: *About Performances of Current Controlled SVM-VSI and Tandem Inverter Used in Induction Motor Drives*, SPEEDAM 2000, Ischia, Italy, pp C4-7 – C4-12, 2000

[13] Imecs M., Incze J. J., Szabó CS., Vásárhelyi J.: *Simple Approach For Induction Motor Control using Reconfigurable Hardware*, Proceedings of the National Conference on Automation CNAE 2002, ISBN 973-8352-83-5, Galati, Romania, pp. 158-164, 2002

[14] Makimoto T.: *The Rising Wave of Field Programmability*, Proceedings of Field Programmable Applications FPL2000, Villach, Austria, August 2000, pp.1-6

[15] Monmasson, E., Hapiot J. C., Grandpierre M.: *A Digital Control System Based on Field Programmable Gate Array for AC Drives*. EPE Journal, Vol. 3, no. 4 (1993), pp. 227-233

[16] Poure P., Aubépart F., Braun F., *ICs for real time Motion Control: A design methodology for rapid prototyping,* PCIM 2001 Power Electronics Intelligent Motion Power Quality, June 19-21, 2001 Nuremberg, Germany, pp. 211-216

[17] Vásárhelyi J., Imecs M., Incze J. J., Szabó Cs.: *Module Library for Rapid Prototyping and Hardware Implementation of Vector Control Systems,* INES 2002 IEEE International Conference on Intelligent Engineering Systems, May 26-28, 2002, Opatija, Croatia, ISBN 953-6071-17-7, pp. 447-452

[18] Vásárhelyi J.: *Reconfigurable Architectures for Vector Control Structures of Induction Motor Drives,* PhD thesis, Technical University of Cluj, Romania, April 4, 2004, pp.180

[19] Imecs M., Trzynadlowski A. M., Incze I. I., Szabo Cs.: *Vector control schemes for tandem-converter fed induction motor drives*, IEEE Transactions in Power Electronics. Vol. 20, no. 2, 2005, pp. 493-501