CHAPTER EIGHT

CONTROL IMPLEMENTATION

8.1 Introduction:

The performance of an AC induction motor is strongly dependant on its control. The recent advances of powerful microcontrollers with DSP functions has enhanced complex and real time algorithms. In particular, the use of a powerful microcontroller brings the following:

- system cost reduction by an efficient control and right dimensioning power devices as well.
- the removal of speed or position sensors by the implementation of sensorless algorithms that need higher complexity calculations.
- a reduction of current harmonics using enhanced algorithms
- a reduction in the number of look-up tables which reduces the amount of memory required.
- real-time generation of torque and flux profiles and move trajectories, resulting in better-performance.

Thanks to the capability of such modern microcontrollers it is possible to implement sophisticated controls like Vector Control.

Vector control refers not only to the magnitude but also to the phase of variables. Matrix and vectors are used to represent the control quantities. This method takes into account not only successive steady-states but real mathematical equations that describe the motor itself, so that the obtained results have a better dynamic for torque variations in a wider speed range.

The Field Oriented Control (FOC) offers a solution to circumvent the need to solve high order equations with a large number of variables and nonlinearities and achieve an efficient control with high dynamic.

This approach needs more calculations than other standard control schemes and has the following advantages:

- full motor torque capability at low speed.
- better dynamic behavior.
- higher efficiency for each operation point in a wide speed range.
- decoupled control of torque and flux.
- short term overload capability.
- four quadrant operation.

8.2 General Design Methodology:

The discussions given in chapter two section (2.11) will give guidance to readers for the formulation of an FL application for a certain problem and its implementation. In summary, the general design procedure for fuzzy control can be given as follows:

1. First. analyze whether the problem has sufficient elements to warrant a FL application: otherwise, apply a conventional method. For example, in a linear feedback system where the mathematical model is known and there is no parameter variation or load disturbance problem, FL has little advantage.
2. Get all the information from the operator of the plant to be controlled. Get information about the design and operational characteristics of the plant from the plant designer, if possible.
3. If a plant model is available, develop a simulation program with conventional control and study the performance characteristics.
4. Identify the functional elements where FL can be applied.
5. Identify the input and output variables of each fuzzy system.
6. Define the universe of discourse of the variables and convert to corresponding per unit variables as necessary.
7. Formulate the fuzzy sets and select the corresponding MF shape of each. For a sensitive variable, more fuzzy sets are needed.

8. Formulate the rule table. (This step and the previous one are the main design steps, which require intuition and experience about the process.)

9. If the mathematical plant model is available, simulate the system with the fuzzy controller. Iterate the fuzzy sets and rule table until the performance is optimized. For a plant without a model, the fuzzy system must be designed conservatively and then fine-tuned by the test results on the operating plant.

10. Implement the control in real time and further iterate to improve performance.

By flowing the above steps the fuzzy logic controllers are designed, and their surface viewers can be presented as shown in the figures below:

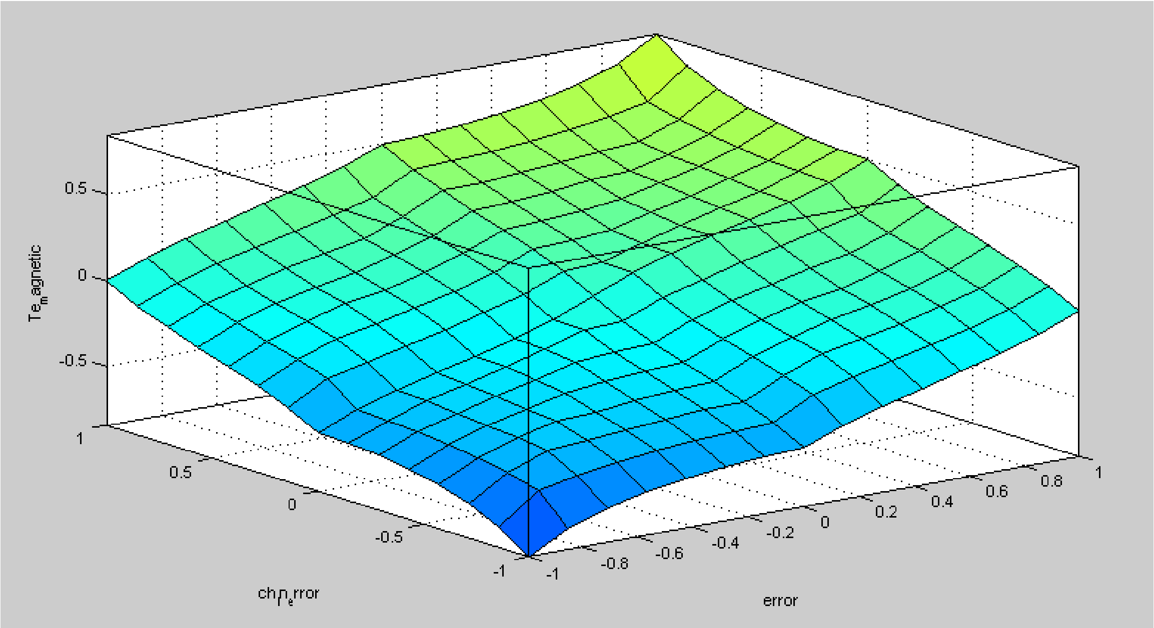

Fig. 8.1 Surface viewer fuzzy rules (Trian.3*3*5)

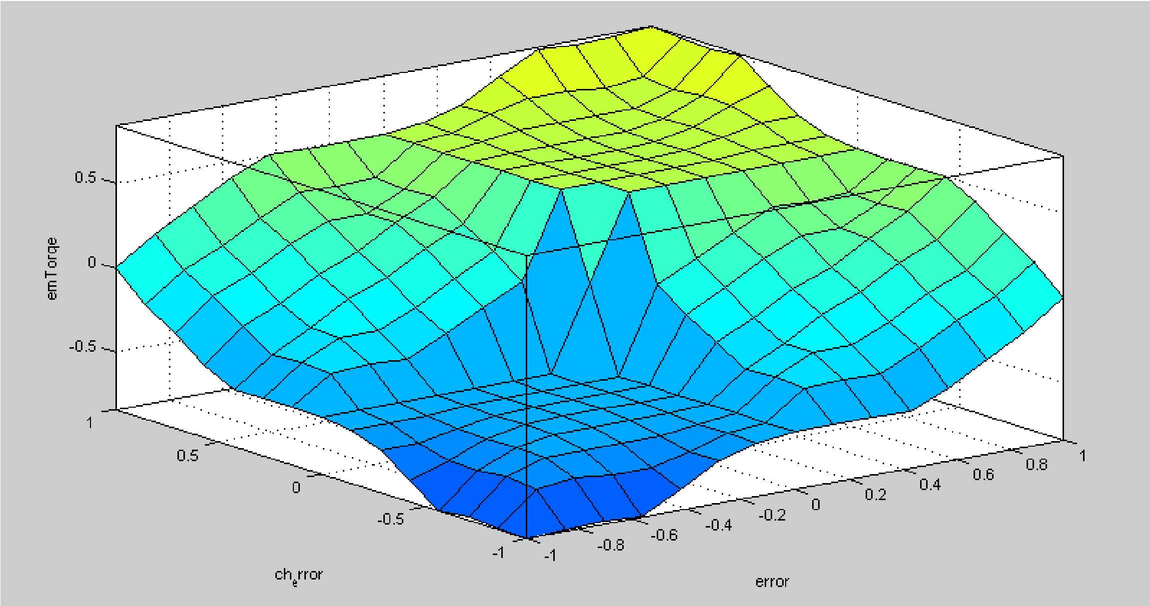

Fig. 8.2 Surface viewer fuzzy rules (Trian.5*5*5)

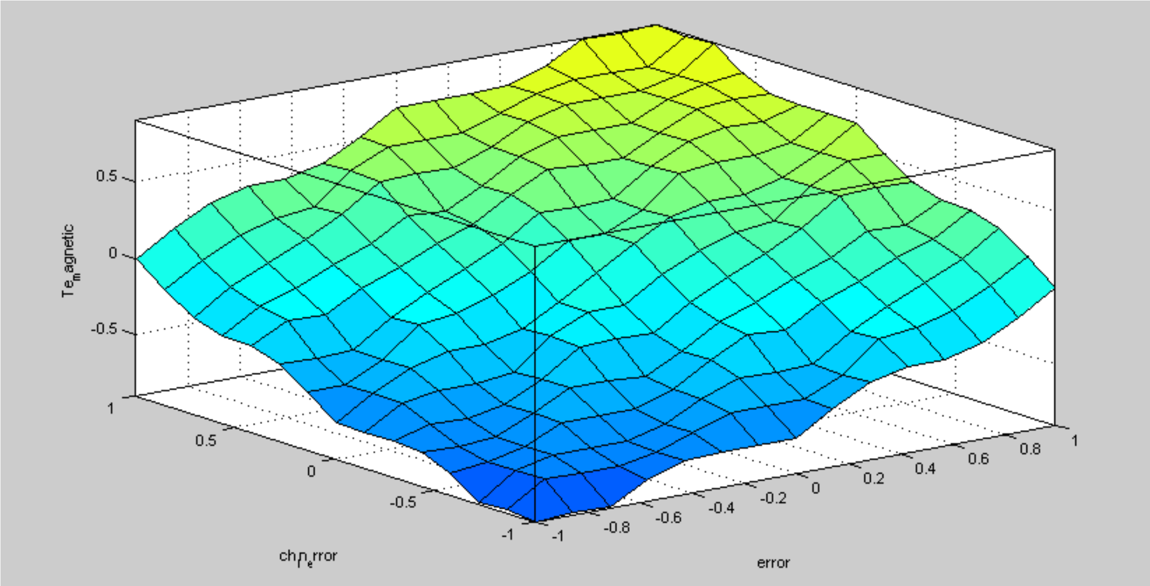

Fig. 8.3 Surface viewer fuzzy rules (Trian.7*7*7)

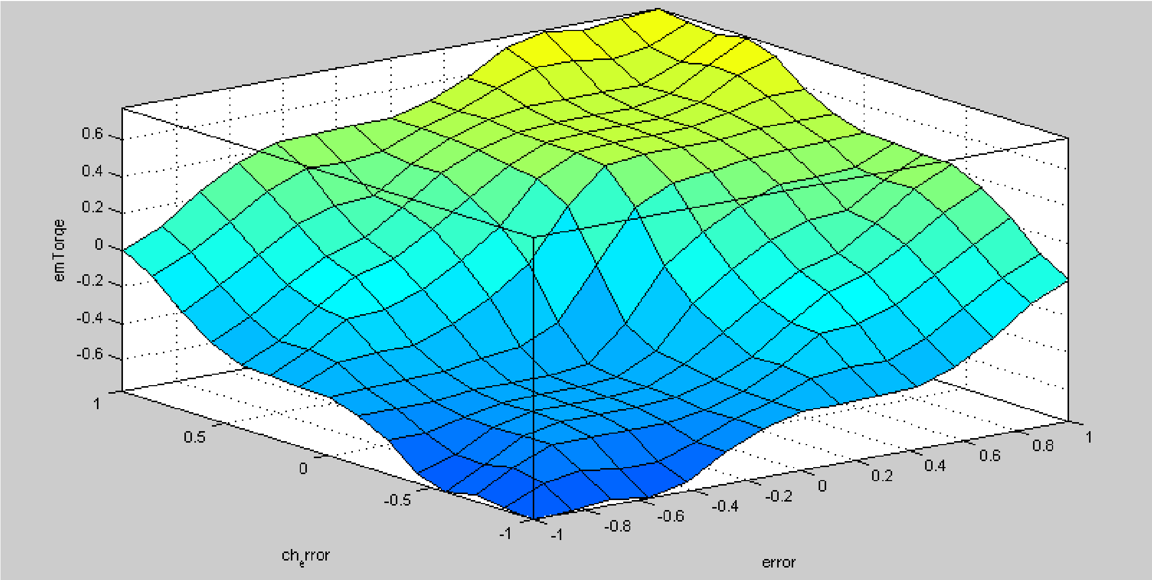

Fig. 8.4 Surface viewer fuzzy rules (Gaus.5*5*5)

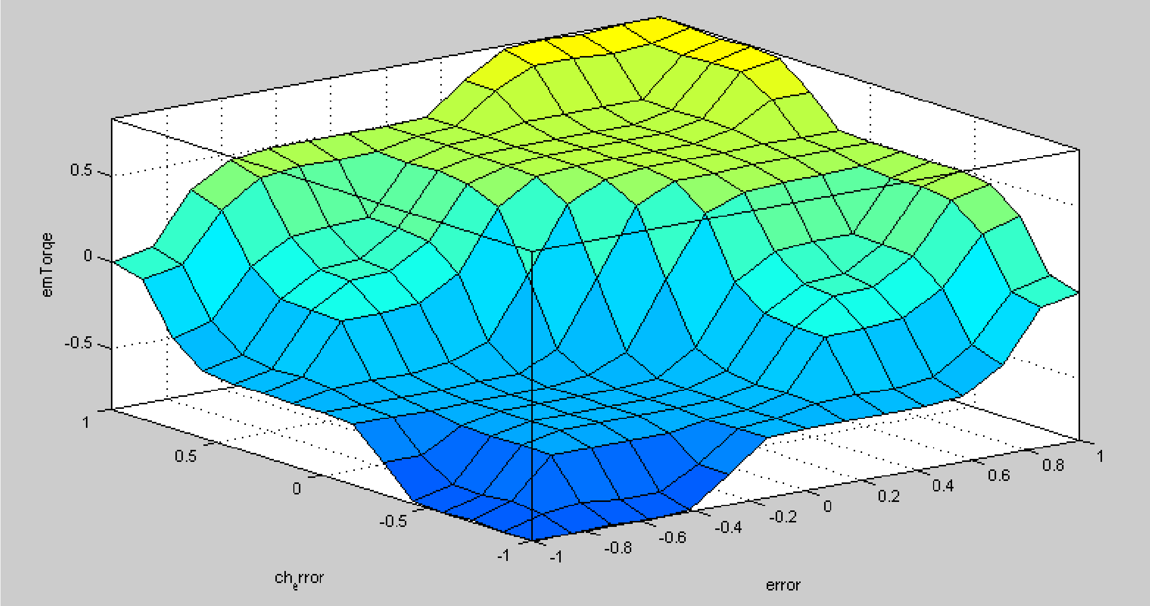

Fig. 8.5 Surface viewer fuzzy rules (Bell5*5*5)

Once a fuzzy controllers program is developed, its performance can be tested by embedding it in the SIMULINK simulation of the system for fine-tuning. And then, the final stand-alone C program (with the fuzzy inference engine) can be generated, compiled, and downloaded to a DSP for real-time implementation.

8.3 Control Implementation:

There are essentially two methods for implementation of fuzzy control. The first involves rigorous mathematical computation for Fuzzification, evaluation of control rules, and defuzzification in real time. This is the generally accepted method. An efficient C program is normally developed with the help of a FL tool. Such as the Fuzzy Logic Tool box in the MATLAB (Math Works. Inc.) environment. The program is compiled and the object program is loaded in a DSP (digital signal processor) for execution.

Commercial (digital or analog) ASIC chips are also available for implementation. The second method is the look-up table method, where all the input/output static mapping computation (Fuzzification, evaluation of control rules, and defuzzification) is done ahead of time and stored in the form of a large look-up table for real-time implementation.

8.4 Possible Methods of Implementation:

Using such DSPs the hardware required for realizing a real time controller is reduced leading to improvement in reliability. In this implementation, the external hardware electronics consists of the IGBT based VSI bridge, current sensing circuit, speed sensing circuit, the gate driver circuits, interfacing circuits and control software. The three phase AC induction motor must be with shaft encoder, as shown in figure 8.6.

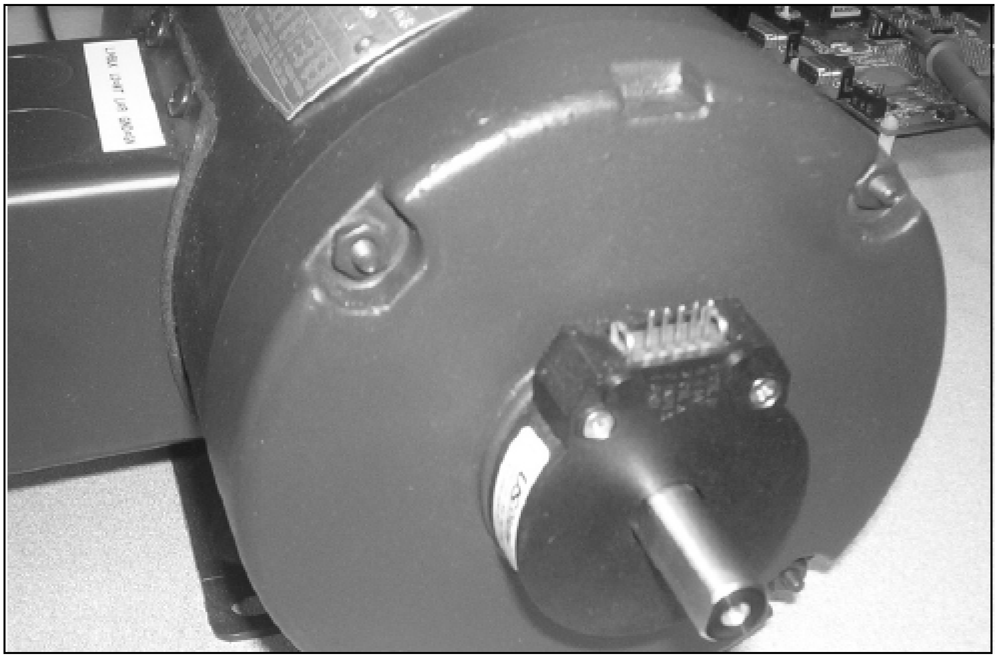

Fig. 8.6 Motor with encoder

Figure 8.7 shows the basic building block diagram of the proposed drive system.

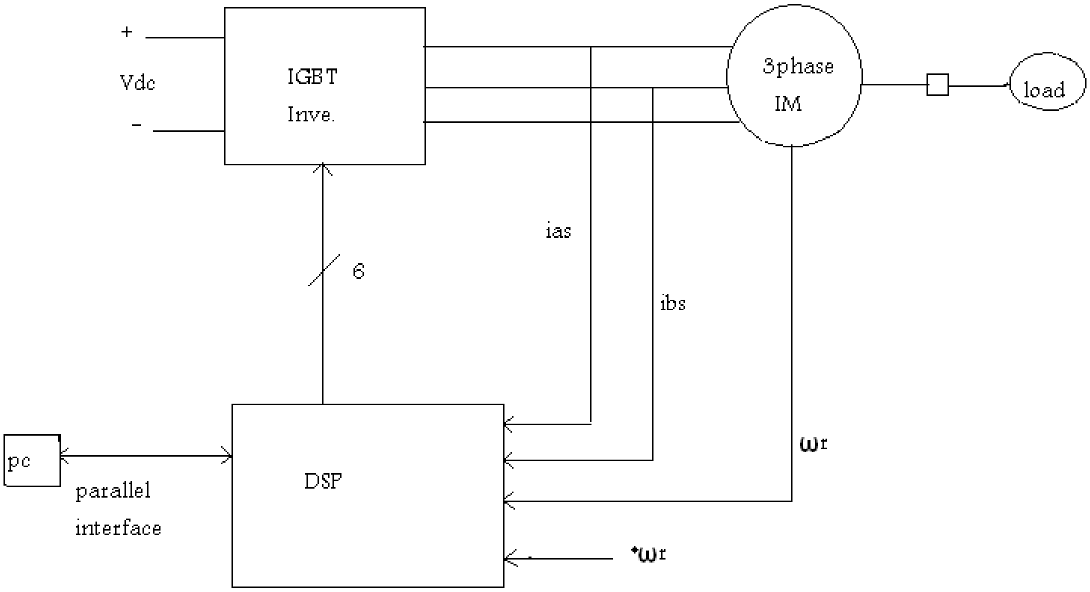

Fig. 8.7 Proposed block diagram of vector controlled induction motor drive