REAL TIME SIMULATION OF ELECTRICAL MOTORS USING SYSTEM-ON-CHIP APPROACH

T. X. Mei* and Y. J. Zhou

School of Electronic and Electrical Engineering
The University of Leeds, Leeds, LS2 9JT, UK
*Tel: 0113 343 2066, Fax: 0113 343 2032, and E-mail: t.x.mei@leeds.ac.uk

Keywords: Real-time simulation, electrical motor, control, FPGA, Soft processing.

Abstract

This paper presents a novel solution for the real-time simulation of electrical motors based on the latest system-on-chip concept, with the aim to provide a cheap and flexible platform for testing and rapid prototyping of motor control systems. Generalised models for different types of electrical motors are presented. Differential equation solvers are discussed and implementation issues for real time simulations are studied. The study is primarily concerned with the optimisation of simulation parameters and the development of appropriate processing architecture, but the hardware design issues are also addressed.

1 Introduction

The control of electrical motors for high performance drive systems often involves complex algorithms and control structures, especially where the load characteristics are complex and uncertain. Computer simulations have played a key role in the development and design of motor control systems in both research and in product design, but those are carried out mainly at early stages of a development/design cycle and largely off-line. Testing and commissioning of real time control systems in full operation conditions are extremely difficult to achieve in laboratories. There are hardware-in-loop systems available, but those tend to be expensive and require great computing power to ensure the execution of simulation algorithms in real time. In one such a development, three Texas TMS320C40 DSPs were used in order to achieve desired real time performances [1]. The complexity of the models has been identified as the main cause of the high demand, but the nature of the largely sequential processing of software algorithms also contributes significantly to the problem.

Recent development and advances in the programmable system-on-chip technology has enabled the implementation of large and complex processing tasks on a single chip (hardware) without the need for the traditional CPUs and associated peripherals. In the area of motor drives, this new approach has been used to realise fairly complex control algorithms [2,3], and to process signals from encoder measurements [4].

This paper studies a novel solution for the real time simulation of electrical motors and corresponding load characteristics, based on the latest FPGA devices. The use of a programmable soft-processing approach will fully explore advantage of parallel processing and easy re-configuration for different industrial applications. The main aim of the study is to provide a low cost and flexible platform for rapid development, testing and commissioning of real time control algorithms and functionalities, which will be suitable and adaptable for use at any stage of the development and design.

The paper is organised as follows. After a brief introduction in section 1, the modelling of electrical motors is discussed in section 2. Section 3 studies simulation algorithms for solving the differential equations, and investigates the requirements for word-length and step-size in the real time implementation. Practical implementation including the associated hardware design and results from the real time system is presented in section 4. Conclusions and further work are discussed in section 5.

2 Modelling of Electrical Motors

There are many different types of electrical motors, but the general principals of those machines are the same in the sense that they all rely on the interaction between electric current and magnetic flux to produce the force. Therefore the mathematical models of any electrical machine can be expressed in a standard format as given in equations (1) – (3).

$$\mathbf{V} = \mathbf{R} \cdot \mathbf{I} + \dot{\boldsymbol{\lambda}} \tag{1}$$

$$T_m = k \cdot \mathbf{I} \times \boldsymbol{\lambda} \tag{2}$$

$$J \cdot \dot{\omega} = T_m - T_{load} \tag{3}$$

For standard dc motors, the vectors of the supplied **V**, current **I** and flux λ are of a dimension of 2x1 for the stator and rotor windings. For an AC induction motor, however, those are of a dimension of 6x1 as there are three separate windings in the stator and rotor. The load T_{load} may vary significantly from application to application. It can be largely static in some cases and therefore straight forward to model. However in many high performance drives, the load characteristics are very complex and sometimes nonlinear and therefore it would be necessary to include the models for the system which the motor is used to drive.

This generalised form for modelling motors indicates that simulation algorithms and hence main architectures of a FPGA can be easily adopted to simulate different types of motors. However, a separately excited dc motor is used in this study, the mathematical model of which is given in equations 4-6 is used. The motor has a power rating of 70*kW*. The rated armature voltage and current are 700*V* and 100*A*, and rated field current is 60*A*. All relevant motor parameters are defined in the nomenclature.

$$v_a = L_a \cdot \dot{i}_a + R_a \cdot i_a + K_e \cdot \omega \tag{4}$$

$$v_f = L_f \cdot \dot{i}_f + R_f \cdot i_f \tag{5}$$

$$J \cdot \dot{\omega} = K_t \cdot i_a - F_m \cdot \omega - T_L \tag{6}$$

where motor electrical and mechanical constants are defined as

$$K_e = K_t = K_{const} \cdot \lambda$$

The model is in fact a non-linear one as both the flux and the inductance of the field winding are non-linear functions of the field current. The non-linearity is caused by the flux entering the saturation region, which is shown in Figure 1.

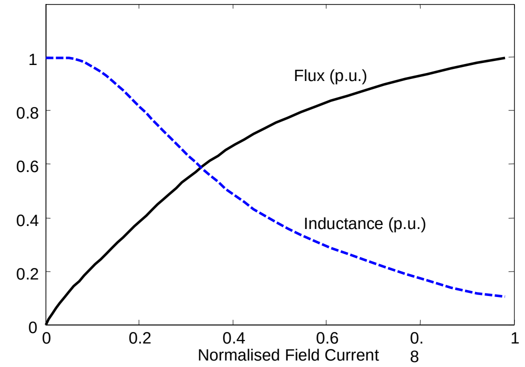

Figure 1. Non-linear characteristics of the field flux and (small signal) inductance.

The work for real time implementation is carried out in two stages. The first stage assumes a slow changing field flux and therefore uses a linearised model with a constant flux. The second stage includes the full non-linear model and full load characteristics. Equations 4-6 can be converted to a set of first order differential equations can be derived and shown in equation 7, where relevant coefficients of matrices **A** and **B** are varied with the field current/flux.

$$\mathbf{F}(t, \mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{7}$$

3 Simulation Algorithms and Implementation Requirements

The selection of a solver for the differential equations is a trade-off between the accuracy requirement and the algorithm complexity. This study has investigated and compared a number of different options, including the Euler, the Heun and the Runge-Kutta methods [5, 6]. It is concluded that a fourth-order R-K solver as defined in equation (8) achieves sufficient accuracy and will also be feasible for the hardware implementation.

As part of standard checks on the simulation accuracy, two major contributing factors of errors are investigated. The first is the truncation error related to the step-size used by the differential equation solver. The selection of a smaller step-size reduces the truncation error in general, but it requires FPGAs to performance more iterations of calculation and to run faster in order to ensure real time performances. The level of truncation error is examined in this study using high precision arithmetic, e.g. floating-point double-precision, with which other errors are likely to be negligible with such calculations.

The second factor is the round-off error, which is associated with the limited word length used for processing variables and parameters. The word-length is constrained, because it is directly related to requirements for the memory allocation and the number of logic gates on a FPGA device. Also integer arithmetic is used to reduce the complexity of FPGA architectures in the real time implementation. The conversion from the floating point to the integer is carried out by scaling all variables and gains. Each term of the equation needs to be multiplied by a specific scaling number and rounded to the nearest integer. The greater scaling number is used, the more accuracy is retained for that term, but at the same time the more memory (for larger word-length) is used to store data.

$$\left.\begin{aligned}
\mathbf{K}_1 &= h \cdot \mathbf{F}(t_n, \mathbf{x}_n) \\
\mathbf{K}_2 &= h \cdot \mathbf{F}(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{1}{2}\mathbf{K}_1) \\
\mathbf{K}_3 &= h \cdot \mathbf{F}(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{1}{2}\mathbf{K}_2) \\
\mathbf{K}_4 &= h \cdot \mathbf{F}(t_n + h, \mathbf{x}_n + \mathbf{K}_3) \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{1}{6}\left(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4\right)
\end{aligned}\right\} \tag{8}$$

Clearly, the overall simulation error is reduced by using a large word length and small step-size. But the desired word-length and step-size must be selected by also taking into account of the issues of hardware complexity and associated costs, and a compromise must be made. Based on the differential equation solver given in equation (7), two sets of different (off-line) simulations are carried out to examine the two different errors. The simulation condition is that the motor is applied with an input voltage of ramp signal and a constant load torque.

Figures 2 and 3 show the simulation errors for the stator current and rotor speed with different word lengths (or different scaling factors) whilst the step-size is fixed to 0.001 seconds. It is clear from the figures that the total error is reduced and converges to the truncation error as the scaling factor (or the word length) is increased. At $S=10^6$ (for speed) or $S=2*10^7$ (for armature current), the round-off error due to the limited word length becomes negligible compared with the truncation error. Any further increase of the word-length is no longer necessary.

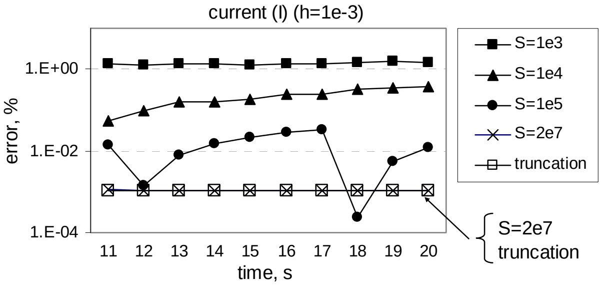

Figure 2. Motor current at different word-length

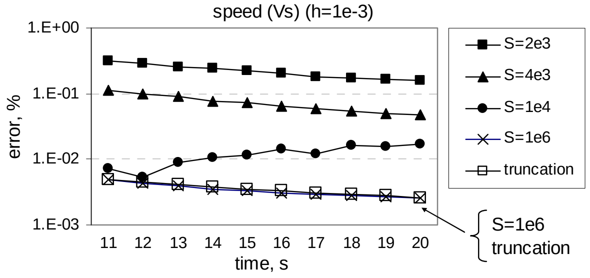

Figure 3. Rotor speed at different word-length

Figures 4 and 5 give the simulated stator current and rotor speed with different step-sizes. An adequate step-size can only be determined by examining the two different errors at the same time, as a decrease in step-size tends to reduce the truncation error, but to increase the round-off error. It is evident from the figures that there is an optimal step size for a given word-length where the total simulation error is minimised.

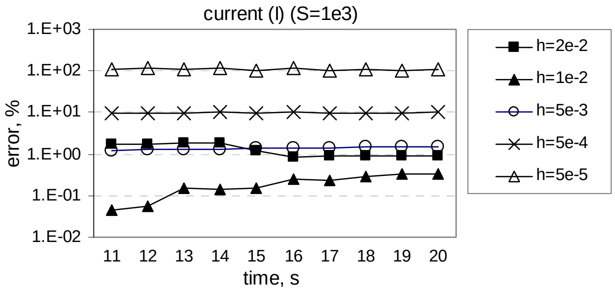

Figure 4. Motor current with different step-size

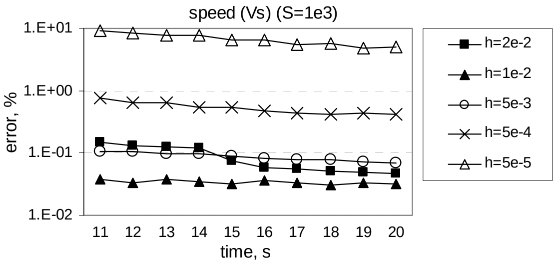

Figure 5. Rotor speed with different step size.

For the dc motor model used in the initial study, a step-size of 0.005s and a word-length of 34 bits are selected.

4 Implementation and Results

The real time simulation system is developed based on a FPGA chip (Altera FLEX 10K®EPF10K70). Figure 6 gives a block diagram of the overall design.

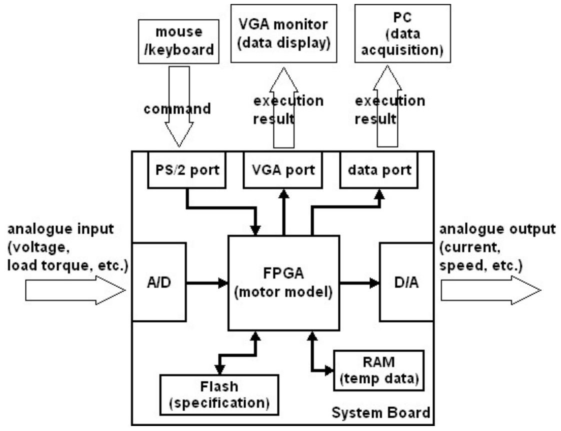

Figure 6. Block diagram of the real time simulation system

In addition to the FPGA chip, some peripherals are necessary to communicate with external sources. A/D converters are used to input the voltage applied to the motor model, whereas DC converters are used to output simulation results such as the stator current and the speed and angular position of the rotor. Some memory space is included, which may be used to store simulation parameters, look-up tables and other temporary data. For downloading and monitoring purposes, interfaces with other devices such as PC parallel/series ports or a VGA monitor can also be included.

On the FPGA chip, the core element is for the implementation of the 4th order R-K differential equation solver, but a number of other features are also included as shown in Figure 7.

A relatively small part of the FPGA is assigned for the control of A/D and D/A converters, which will be executed separately from the main part of the algorithms. The memory control and other peripheral controls are implemented in a similar manner. In the design, the models are also included to simulate the dynamics and other key features of sensors. This will be useful for testing motor controllers that require those measurements, but without the need for real sensors.

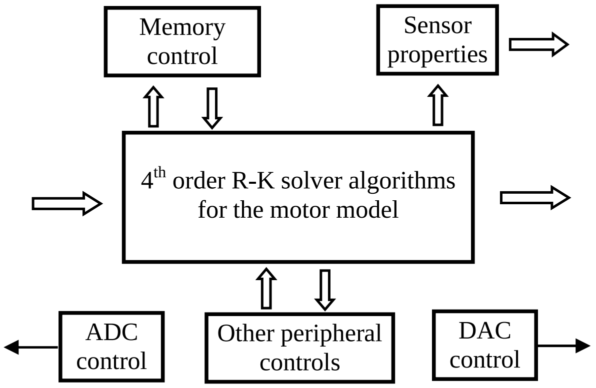

Figure 7. Block diagram of the FPGA design

The internal design of the FPGA is carried out using a combination of VHDL and schematic diagram methods. The R-K algorithms have been re-arranged so that most calculations are performed in parallel – an approach that reduces the computing time (approximately 3.7 times faster compared with the original algorithms) and at the same time does not require additional capacity of the FPGA chip. The total computing time for the main algorithms is about 140ns per sampling step, significantly less than many advanced DSP can deliver. The coefficients in the algorithms are scaled such that all dividers are chosen to be the power of 2. This offers the obvious advantage of not needing the use of complex divisions, as they are replaced by the shift of any bits in a single clock cycle.

Some results from the real time implementation are presented in Figures 8-12. The flux is assumed to be constant at this stage. An input voltage as shown in Figure 8 is applied to the motor armature, which represents different conditions such as accelerating, coasting, and decelerating etc. A digitalisation process reflecting the effect of the A/D converter is carried out to obtain the input voltage in digital form. The load torque of the motor is treated as a constant.

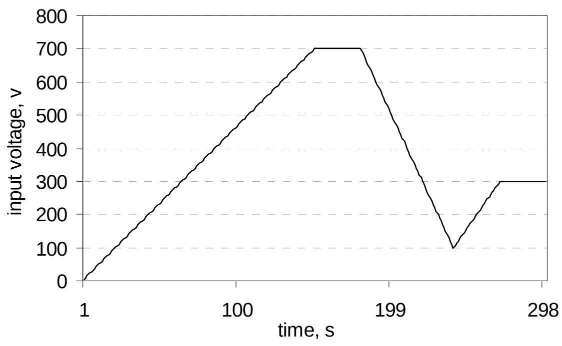

Figure 8. Input voltage of the motor simulator

Figures 9 and 10 show the simulated current and speed responses of the dc motor. Figures 11 and 12 give the corresponding simulation errors (in percentage). The armature current is fairly high and increases at the beginning as the voltage is increased, which drives the motor speed up. The current drops off to a level to overcome the running resistance when the supplied voltage becomes constant. When the voltage is reduced, the motor decelerates and the moment of motor inertia cancels part of the load, resulting in a very low current level. This pattern repeats when the input voltage is repeated.

The simulation of the motor models can be achieved in real time with a total error around 0.1% or less as clearly demonstrated in figures 11-12.

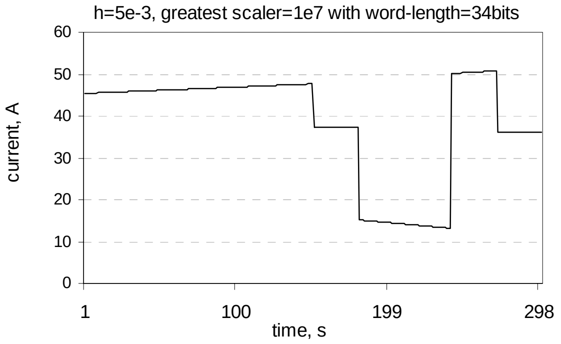

Figure 9. Simulated current response

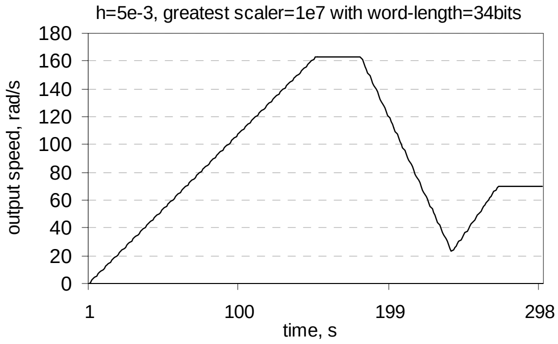

Figure 10. Simulated speed response

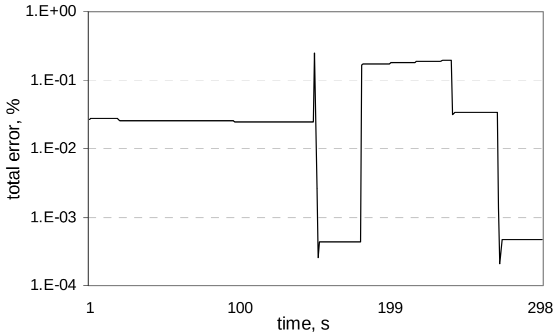

Figure 11. Simulation error (current)

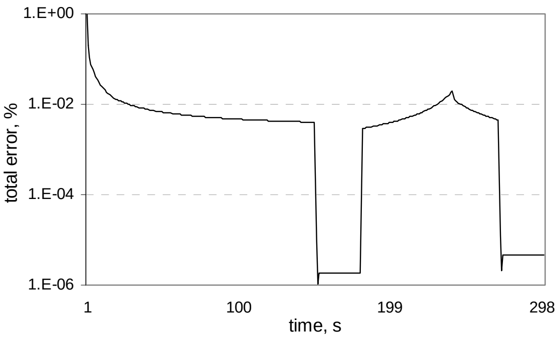

Figure 12. Simulation error (speed)

5 Conclusions and Further Work

This paper has presented the development of a real time simulation system for modelling electrical motors based on a single FPGA chip, without the use of a conventional microprocessor or a DSP. Generalised models for the electrical motors have been presented, and the fourth order Runge-Kutta method has been selected as the differential equation solver. Practicalities and hardware design of real time implementation have been discussed and preliminary results given.

It has been shown in the study that the use of an appropriate word length and step-size is essential to minimise computing errors in the real time simulations. On the other hand, the study has revealed that the requirements for the word-length and step size are not overly demanding. It has been demonstrated in the paper that the simulation of a typical dc motor can be achieved in real time with a total error around 0.1% or less, which is acceptable in most practical applications.

The main contributions of this paper are two folds. Not only it has demonstrated the feasibility of implementing a real time simulation of an electrical motor using a fairly simple FPGA chip and therefore offers a more cost effective solution for testing and commissioning real time motor controllers, but also it has directly proved that the use of the system-on-chip approach can provide a much faster solver than many advanced DSP devices and hence enable the simulation of even more complex and fast dynamic systems in real time.

The further work includes the physical implementation of the full non-linear model and the load characteristics. Also the work can be extended to model ac motors which are dynamically and structurally more complex.

References

[1] Jack, A G; Atkinson, D J and Slater H J. "Real-time emulation for power equipment development. Part 1: Real-time simulation", *IEE Proceedings – Electric Power Applications*, **volume 145**, No 2, pp. 92-97, (1998).

[2] Kim, S J; Lee, H J; Kim, S K and Kwon Y A. "ASIC design for DTC based speed control of induction motor", IEEE ISIE2001, pp 956 – 961, (2001)

[3] Ricci, F and Le-Huy, H "An FPGA based rapid prototyping platform for variable-speed drives", IEEE IECON'02, pp. 1156-1161, (2002)

[4] Tsai, M F and Chen C P "Design of a quadrature decoder/counter interface IC for motor control using CPLD", IECON'02, pp. 1936-1941, (2002)

[5] Paul W. Davis, "Differential Equations for Mathematics, Science, and Engineering", Prentice Hall, Englewood Cliffs, New Jersey, (1992)

[6] Leon Lapidus, John H. Seinfeld, "Numerical Solution of Ordinary Differential Equations", Academic Press, New York and London, (1971)

Nomenclature

$\mathbf{A}$	State matrix
$\mathbf{B}$	Input matrix
$\mathbf{I}$	Motor current vector
$\mathbf{J}$	Moment of inertia of the rotor
$\mathbf{R}$	Motor resistance matrix
$\mathbf{V}$	Motor voltage vector
$\mathbf{x}$	State vector
$\mathbf{u}$	Input vector
$\boldsymbol{\lambda}$	Motor flux vector
h	step-size
I_a	Armature current
I_f	Field current
J	Moment of inertia of the rotor
K_{const}	Motor constant
K_e	Motor electrical constant
K_t	Motor mechanical constant
L_a	Armature inductance
L_f	Field inductance
R_a	Armature resistance
R_f	Field resistance
T_m	Motor torque
T_L, T_{load}	Motor load
V_a	Armature voltage
V_f	Field voltage
λ	Motor flux
ω	Motor speed