

Design of an optimized SRM control architecture based on a hardware/software partitioning

Mickaël Hilairet, *member, IEEE*, Hala Hannoun, Claude Marchand, *member, IEEE*
 Laboratoire de Génie Electrique de Paris (LGE) / SPEE-Labs,
 CNRS UMR 8507; SUPELEC; Université Pierre et Marie Curie P6; Université Paris-Sud 11;
 11 rue Curie, Plateau de Moulon F91192 Gif sur Yvette CEDEX
 Email : mickael.hilairet@lgep.supelec.fr

Abstract—This paper presents an effective digital speed control implementation for a switched reluctance machine. An optimized architecture is proposed based on a hardware/software partitioning in order to implement it on a System on Programmable Chip. This solution leads to a user-friendly development solution without over pending for performance or sacrificing features. Finally, the proposed strategy achieves lower current and torque ripples in a large speed range compared to a software implementation.

Index Terms—Switched reluctance machine, speed control, continuous and discontinuous conduction mode, hardware/software partitioning, FPGA, System on Programmable Chip.

I. INTRODUCTION

NOWADAYS, speed variation is considered in various applications going from robotics until transport domain. The industrial requirements can vary between cost, size, user's comfort, efficiency ... depending on the application type. Switched Reluctance Motor (SRM) drives are under consideration in various applications requiring high performances such as electrical or hybrid vehicles. This is certainly due to its numerous advantages such as simple and robust construction, high speed and high-temperature performance, low costs, and fault tolerance control capabilities.

In this paper, a particular attention is dedicated to the SRM controller implementation. The objective is to design a control strategy with optimized control parameters for every operating point in the speed/torque plane and choose the component (processor or programmable chip) that best fits the proposed application.

From the point of view of driving system equipment, the software based controller is generally used for the speed control. The software development is user-friendly nowadays, so that development cycle is shortened. However, the software control has some limitations as the computational time principally. Therefore, hardware devices as CPLD, FPGA and ASIC are used for torque control of AC machines due to the high speed computation. Especially, FPGA are rewritable, high performance and low cost devices, so systems using FPGA are increasing since 10 years [1]–[3].

The system based on chip approach (ASIC technology) allows high performance and integration, but :

- it is poorly adapted to the system evolution,
- it is reserved for a large production scale,

- manufacturing and testing steps are costly.

System on Programmable Chip (SoPC) is an alternative solution and solves some of these problems :

- it allows short development and rapid prototyping,
- a component reconfiguration in a few ms.

However, this solution have a high consumption and a low performances. In this work, this second solution is adopted and leads to an user-friendly development solution without over pending for performance or sacrificing features.

This paper is organized into two sections as follows : In section two, an average torque control is described so that the machine operates in a wide speed operating range. In the second section, an optimized SRM architecture is detailed. Afterward, the proposed algorithm is simulated with a fixed-point data representation in order to implement it on a FPGA.

II. SRM CONTROL STRATEGY

Current or torque control constitutes the main control block in drives to obtain the desired high bandwidth in torque and speed responses. Two different methods exist for torque control of switched reluctance machine. The first one is called “instantaneous torque control” where the phase current references are computed at each sample time according to the desired phase's torque and the rotor position. The second one is called “average torque control” where square wave current references are imposed over one excitation period. This strategy is easily implementable but leads to important torque ripples compared to the first solution. At low speeds torque ripples could cause significant speed fluctuations and oscillations. This is the main drawback of this second control strategy [4].

A source of limitation for the instantaneous torque control is the condition imposed by the turn on angle ψ^1 for positive instantaneous torque production. Advanced angle commutation ($\psi < 0$), necessary for torque maximization at a given speed, is not possible with instantaneous torque control strategy. Consequently, the capabilities of the motor are not fully utilized at every rotor speed unless the algorithm commutates to the average torque control strategy. Therefore, the second methods is adopted in this work so that the SRM can operate in

¹The turn on angle ψ represents the starting magnetization moment considered relatively to the minimum inductance position.

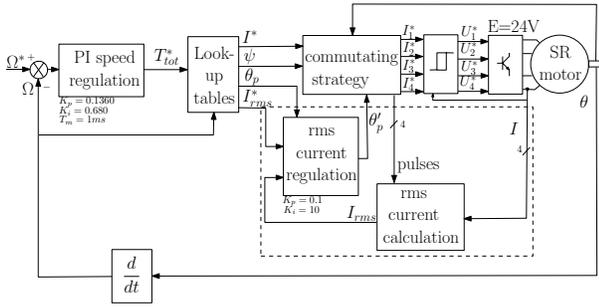


Fig. 1. Block diagram of the SRM drive.

a wide speed operating range without switching from different control strategies [5].

The conventional method for SRM operation may be called “discontinuous conduction mode (DCM)”: the current in each motor winding starts at zero and returns to zero during each stroke. As speed increases, the period for fluxing the machine is reduced and the back-emf increases more rapidly to such a point that the phase voltage is saturated and the phase current is naturally limited. Consequently, torque and power production fall and the constant power region is limited [6]–[8].

In order to maintain the amplitude of the flux, and hence produce torque during high speed operation, the inverter commutation angles are advanced such that turn on angle ψ occurs well before the phase inductance starts to increase. This early magnetization enables the current to rise to a reasonable level before the back-emf dominates the dc-link voltage [9]–[11]. Additionally, the conduction angle θ_p ² can be extended providing a greater time period over which to flux the machine. With an extended conduction period, phase current does not extinguish anymore at each stroke and the starting current is not zero. The machine operates in the “continuous conduction mode (CCM)” where the phase is not completely demagnetized before the next conducting period is reached.

Fig. 1 shows the block diagram of the SRM drive so that the machine operates in continuous conduction mode. To compensate the imprecision of the model used to build the look up tables, the conduction angle θ_p is re-adjusted with an angle $\Delta\theta_p$ as in [7], [12] so that the rms phase current I_{rms} is equal to its reference I_{rms}^* . The corrected angle θ_p' is then sent to the commutating strategy. The turn on angle ψ , conduction angle θ_p and rms phase current reference I_{rms}^* are tabulated in look-up tables. These tables are constructed off-line using a SRM simulation software [13] in such a way to maximize the torque for a given rms current when operating in CCM.

The rms current loop based on a classical PI controller is only activated in CCM. This controller compensates the imprecision of the model used to build the look up tables and serves as a system protection in case of parameter variations like resistance increase with temperature.

Fig. 2 shows the rms phase current response for the 50A step at the nominal speed (3000 rpm), with a sampling period for the controller equal to $1\mu s$. It shows also the evolutions

²The conduction angle θ_p is the duration where positive voltage is applied to the phase.

of the conduction angle, the phase current and the phase flux respectively. Here, the turn on angle ψ is fixed to -110° . The rms phase current is estimated at the end of each conducting period then used to correct the conduction angle. The new corrected value of θ_p is then applied at the next conducting period.

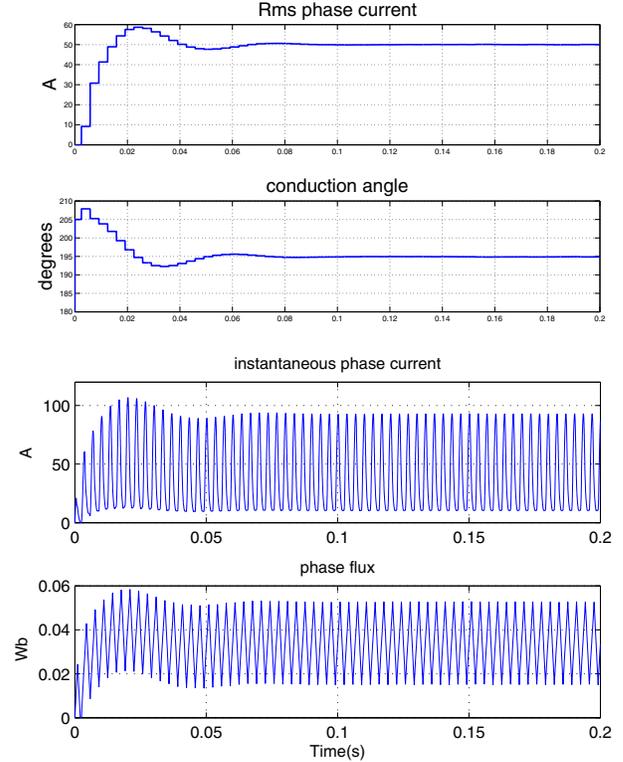


Fig. 2. Simulation results during continuous conduction mode.

The control strategy was first validated by simulation and then on an experimental test bench. In practice, a speed controller is implemented so as to avoid any problem. The speed is regulated through a discrete PI controller whose output is the total reference torque T_{tot}^* as shown on Fig. 1. The proposed controller is implemented on the test bench using a DSPACE DS1103 board. The speed is regulated with a sampling period equal to 1 ms, while the phase voltage is updated every $50\mu s$ due to the hardware limitation. The evolution of the phase current is plotted on Fig. 3.a. A zoom on the current (Fig. 3.b), around the operating switching mode from discontinuous to continuous conduction, shows clearly the effect of the new conduction mode: the current increases gradually until it enters in continuous conduction.

However, this algorithm is sensitive to the mechanical position measurement, and this sensitivity gets worse when the speed increases. Fig. 4 shows CCM simulations with 3 different sampling time (50, 10 and $1\mu s$) of the mechanical position. We can observe that the torque oscillates around its reference, and that low frequency oscillation appears as the sampling time increase. In practice, these oscillations appear in the currents as shown in Fig. 3. Although the closed loop system is yet stable in practice with a sampling time equal to $50\mu s$, these oscillations could be reduced by the use of fast programmable chip as FPGA.

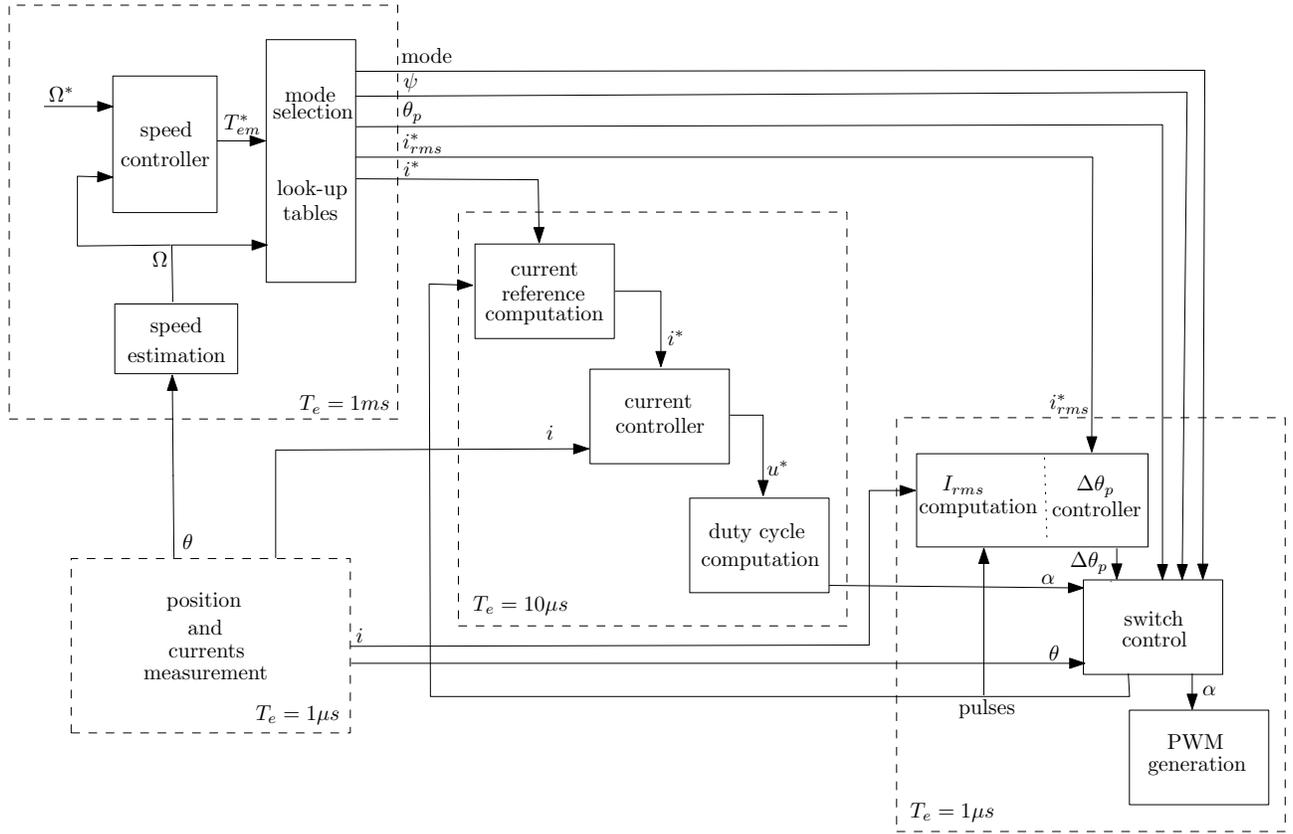


Fig. 5. Proposed SRM controller architecture.

III. OPTIMIZED HARDWARE/SOFTWARE PARTITIONING

A. SRM control architecture

It is well known that current control need lower sampling time value than the mechanical speed controller update. Therefore, the controller has been decomposed into 4 blocks with its own sampling time as shown in Fig. 5, namely :

- The first block computes current references I^* or I_{rms}^* , angles ψ and θ_p , and operating mode (continuous or discontinuous) in order to control the mechanical speed. Here, the sample time is equal to 1ms, because lower values do not appear to improve the speed control performance. Furthermore, the mechanical speed is computed in this block based on a Kalman filter from the mechanical position measurement [14].
- The second block computes the phase voltages in order to control the current phases. Therefore, the block outputs are the duty cycles of the half-bridge inverter. For our SRM prototype, the current controller is computed at a sampling time between $20\mu s$ and $50\mu s$. Here, $50\mu s$ seems to be an upper value that should not exceeded. In fact, an experimental test with an hysteresis controller computed with a sampling time equal to $50\mu s$ shows that the current exceeds the limits allowed as shown in Fig. 6.
- The third block is composed of 3 sub-systems : the switching logic, the RMS current controller for the continuous conduction mode and the converter control. Simulation and experimental results (see Fig. 4) show

that currents and torque ripples increase with the sampling time, and that $50\mu s$ seems to be the limit, where the stability is still ensure. Therefore, a sampling time between $1\mu s$ to $10\mu s$ seems to be a good value in order to ensure that the SRM is well controlled at high speed.

- The last block represents the input data : currents and position. The rotor position and angular speed are obtained from an incremental pulse encoder which is mounted on the rotor axis, as shown in Fig. 7. The rotor position is easily obtained using an up/down counter and glue logic.

B. Hardware/Software partitioning

The proposed SRM strategy could be implemented by software [5], [7], however simulation and experimental results (see 4) show that currents and torque are not well controlled due to the board computational limitation that is achieved. Thus, in this work, an FPGA implementation is adopted, that is a viable solution nowadays due to user-friendly tools. Moreover, in order to design custom algorithm more quickly without over pending for performance or sacrificing features, the control algorithm is split into 2 parts : a software part running on a processor for the speed control (slow dynamics), and a hardware part for current control (fast dynamics) [15].

a) *Software part*: In order to reduce the development time, the speed estimation and the speed controller are implemented with a floating-point representation. The advantage of floating-point representation over fixed-point (and integer)

representation is that it can support a much wider range of values. Because fixed point operations can produce results that have more bits than the operands, there is opportunity for information loss. If any integer bits are lost, the value will be radically inaccurate. This is considered to be an overflow, and needs to be avoided in embedded calculations. It is recommended that simulation tool such as Matlab/Simulink or VisSim be used to detect and avoid such overflows by use of appropriate result word size and radix point, proper scaling gains, and magnitude limiting of intermediate results.

Therefore, in this work, floating-point representation is adopted for the speed estimation and speed controller in order to reduce development time. Here, we proposed to implement the slow dynamics part on the Altera NIOS II processor.

b) Hardware part: This part is composed of the current controllers, the switching logic, the RMS current controller (Proportional Integral) for the continuous conduction mode and the converter control. Consequently, the designer selects the number of bits necessary to compute values appearing in the PI controller algorithm flow and data such as ψ and θ_p , while avoiding overflow and performance damages.

Therefore, the integer and decimal part size should not decrease the controller performances and should be optimized. To define the specific fixed-point format, 3 major parameters are taken into account :

- the maximal operation values,
- the minimal operation values,
- estimated control accuracy.

To choose specific binary format, simulation analysis have been carried out in the Matlab/Simulink environment. Here, we opt to use data with 10 bits to store look-up table such as ψ , θ_p , I^* and I_{rms}^* . These data have been scaled in order to use all the range between zero and the upper bound ($2^{10} - 1 = 1023$). Therefore, the switched controller have been modified according to the scaling factor. Finally, the $\Delta\theta_p$ controller data used fixed-point representation with signed and unsigned numbers. These intermediate variables use integer numbers from 10 to 32 bits so that the performances are not degraded.

Fig. 8 represents some important values and parameters

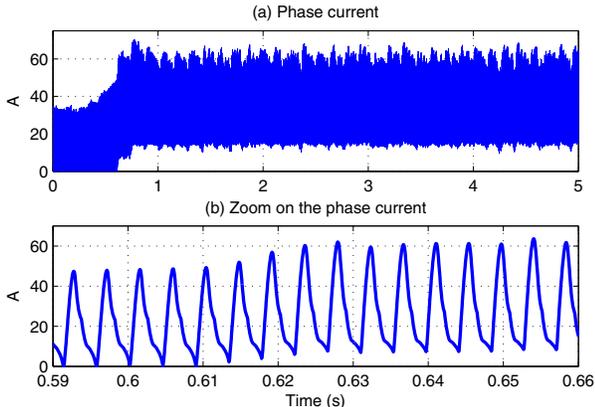


Fig. 3. Experimental phase current evolution (a) with a zoom around the operating switching mode (b).

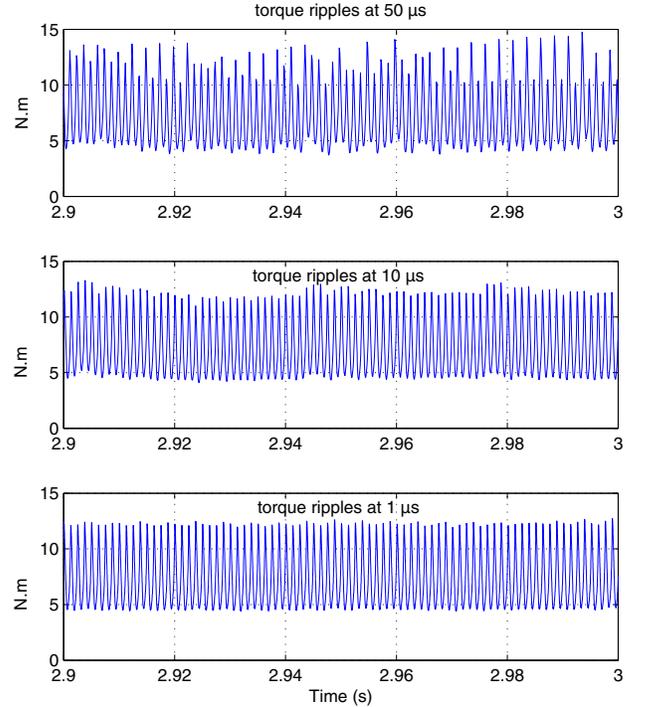


Fig. 4. Sampling time influence (50, 10 and 1 μ s) on the torque ripples.

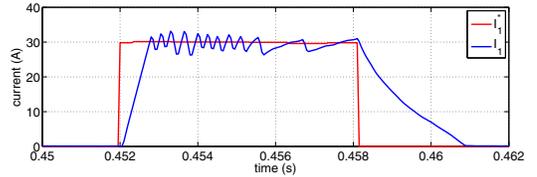


Fig. 6. Hysteresis current control.

of the proposed algorithm with a fixed-point simulation. The obtained results are similar to a floating-point simulation. In fact, the differences are nearly zero as shown in Fig. 9. It means that the specific binary format is judiciously chosen and the digital adaptation of the algorithm did not damage the performances.

IV. CONCLUSION

This paper presents an effective digital speed control implementation for a switched reluctance machine in order to improve the drive performances during continuous conduction mode, as well during discontinuous conduction operation. The optimized architecture is based on an algorithm partitioning, with specific sampling time adapted to the desired performances. The proposed strategy achieves lower current and torque ripples in all the speed range.

Finally, we proposed an FPGA implementation associated with a hardware/software partitioning, where some algorithm parts are implemented on a System on Programmable Chip (SoPC) and others in a hardware part, so that the development time is reduced. Simulation results show that fixed-point simulation does not damage the algorithm performances. Thereby,

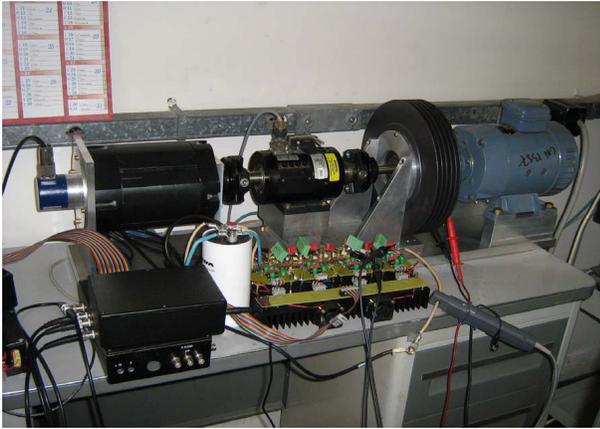


Fig. 7. The experimental test bench.

this work represents a first step toward a full FPGA-based controller implementation for SRM drive.

REFERENCES

- [1] R. Dubey, "Programmable logic devices for motion control - A review," *IEEE Transactions on Industrial Electronics*, Vol. 54, N° 1, pp 559-566, Feb. 2007.
- [2] M.W. Naour et al, "FPGA-based current controllers for AC machine drives - A review," *IEEE Transactions on Industrial Electronics*, Vol. 54, N° 4, pp 1907-1925, Aug. 2007.
- [3] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems - A review," *IEEE Transactions on Industrial Electronics*, Vol. 54, N° 4, pp 1824-1842, Aug. 2007.
- [4] H. Hannoun, M. Hilaret, C. Marchand, "Comparison of instantaneous and average torque control for a Switched Reluctance Motor," *IEEE International Symposium on Industrial Electronics ISIE*, Cambridge UK, 2008.
- [5] H. Hannoun, M. Hilaret, C. Marchand, "Experimental validation of a SRM operating in continuous conduction mode," *submitted at IEEE Transactions on Industrial Electronics in 2009*.
- [6] N. Schofield, S. A. Long, D. Howe, M. McClelland, "Design of a switched reluctance machine for extended speed operation," *IEEE Transactions on Industry Applications*, Vol. 45, N° 1, pp 116-122, Jan-Feb 2009.
- [7] H. Hannoun, M. Hilaret, C. Marchand, "Speed control of a switched reluctance machine in continuous conduction mode," *IEEE International Conference on Electrical Machines ICEM08*, Vilamoura, Portugal, Sept 2008.
- [8] M. Rekik, M. Besbes, C. Marchand, B. Multon, S. Loudot, D. Lhotelier, "High-speed-range enhancement of switched reluctance motor with continuous mode for automotive applications," *European Transactions on Electrical Power*, Vol. 18, pp 674-693, 2008.
- [9] J.S. Lawler, J.M. Bailey, J.W. McKeever, P.J. Otaduy, "Impact of continuous conduction on the constant power speed range of the switched reluctance motor," *IEEE International Conference on Electric Machines and Drives (IEMDC'05)*, pp 1285-1292, May. 2005.
- [10] J.M. Stephenson, W.F. Ray, "Control of switched reluctance machines," *European patent, EP 0 534 761 A1*, 1993.
- [11] I. Jordison et al, "Control of switched reluctance machines," *European patent, EP 1 385 263 A2*, 2005.
- [12] S. Loudot, "Method for controlling a heat engine vehicle driving assembly," *International patent, WO 2006/05028 A2*, 2006.
- [13] M. Besbes and B. Multon, "MRVSIM Logiciel de simulation et d'aide à la conception de Machines à réluctance variable à double saillance à alimentation électronique," *Deposit APP CNRS n.IDDN.FR.001.430010.000.S.C.2004.000.30645*, 2004
- [14] M. Hilaret, F. Auger, M. Ait-Ahmed, M.F. Benkhoris, "Speed and position estimation from an absolute position encoder," *Electrimacs 99*, pp 217-222, Lisbonne Portugal, Sept. 1999.
- [15] X. Shao, D. Sun, "Development of a new robot controller architecture with FPGA-based IC design for improved high-speed performance," *IEEE Transactions on Industrial Informatics*, Vol. 3, N° 4, pp 312-321, Nov. 2007.

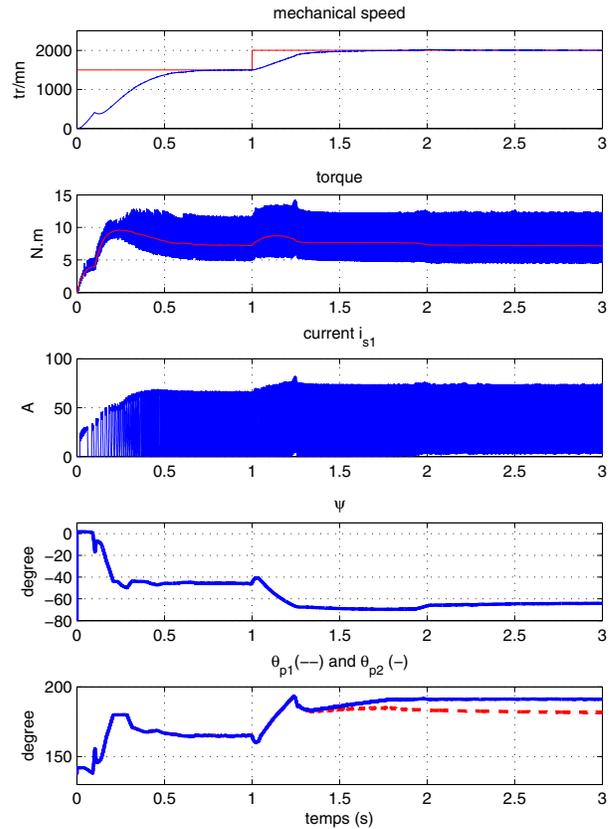


Fig. 8. Continuous conduction mode control with fixed-point representation.

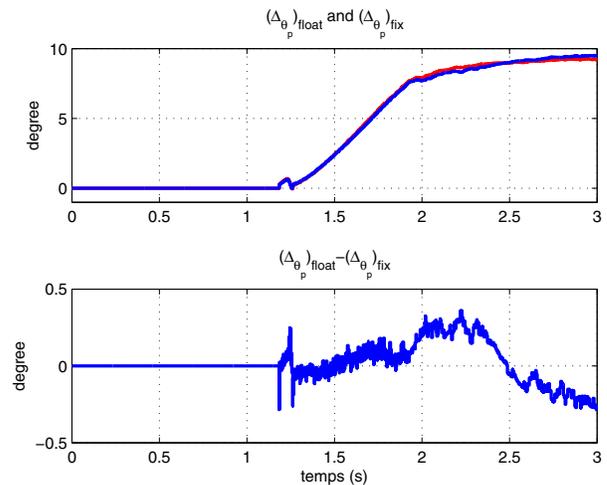


Fig. 9. I_{rms} controller output with floating-point and fixed-point representation.