Implementation of a Motion Control IC for DC Motor Device

Cheng Xuan[*1], Yang Xiangdong[2], Cao Diansong[3]

[1,2]Department of Mechanical Engineering, Tsinghua University Haidian District, Beijing, China

[3]Corporate Technology, Siemens China Ltd. Chaoyang District, Beijing, China

[1]chengxuan2002@163.com; [2]yangxd@mail.tsinghua.edu.cn; [3]diansong.cao@siemens.com

***Abstract*-The new generation FPGA technology enables an embedded processor with application of intellectual property (IP) cores into a system-on-programmable-chip (SoPC) development environment. Therefore, in this paper, a study presents a motion control integrated circuit (IC) for DC motor devices under SoPC technology. Firstly, the architecture of the control IC is described, including several customized application IP cores and a Nios II embedded processor. The designed applications IPs are utilized to implement the motion trajectory algorithm, an improved PID controller, PWM signals generation and position decoder in hardware owing to the low control latency, high reliability, and high sampling frequency control (sample interval is 128μs). The Nios II processor is used to synchronize the state of each part and communicate with host processor in software. All of them connect to each other via Avalon bus. Finally, a verification system is set up and some experimental results are demonstrated. The contribution of this work is to implement a motor control system in a single chip combing low control latency, high sampling frequency control with flexibility and scalability based on SoPC technology, compared with traditional LM629 or other ASIC-based and DSP-based controllers.**

Keywords- Motion Control; DC Motor Device; SoPC; Trajectory Generation; Improved PID Controller; IP Cores; Embedded Processor

I. INTRODUCTION

DC motor has been widely used in industry area and healthcare area for a long time, because it has many advantages of simple principle model, high response performance, high efficiency and being easier to be linear control etc [1, 2]. Generally speaking, the position and speed characteristics are essential in controlling a DC motor. It is a basic property that the speed of DC motor can be adjusted by the terminal voltage [1]. Despite many intelligent control methods have appeared in the literature for optimization, such as fuzzy logic control, neural network and genetic algorithm etc. PID control algorithm and its improved algorithms are also the most widely-used control approaches over the past several years [3], due to its simplicity, good robustness, fast response, and very large application range. It is estimated that PID controllers are still employed in more than 95% of industrial processes [4]. In general, DC motor control system based on digital PID controller is a significant tool in digital servo field, especially in real-time digital control. Furthermore, more and more researchers pay attention to optimization methods for tuning PID parameters [5-7].

Nowadays, the control systems for DC motor are often implemented in three approaches: computer-based system [8], DSP/- micro-processor-based system [9] and specialized chips [10, 11], such as LM629 chip [11] etc. The first two are based on software, and the third one belongs to hardware. It is noted that the system designed based on specialized chip is very popular, owing to the simplification of hardware and software design and improvement in stability compared with software implementation. However, this scheme is usually limited by low sample rate, non-multichannel capacity and non-flexible configuration. Also, the problem of communication and synchronization with extra control unit and other components should be considered. Therefore, a novel solution including an overall hardware/software design using a single chip is a better way to build a whole control system.

In this paper, a motion control IC is introduced for DC motor device based on FPGA. The motion control IC is designed as the whole control system using SOPC technology [12]. This IC not only contains embedded processor and closed loop controller [13-15], but also includes trajectory generation, interface, control signal denoise and drive signal generation. All of these components are packaged as IP (Intellectual Property) cores and a command control program can be auto-load and running in the embedded processor. It is competitive with a high sampling rate host interface including efficient FIFOs, assuring long term availability and ease of modification due to the open FPGA based design. Furthermore, because of the embedded processor running in FPGA, the controller designed is more flexible and scalable for designers, and control parameters and variables can be monitored easily. Therefore, it is an ideal replacement for LM629 and other ASIC based motion controllers.

This paper can be divided into following parts. Section II describes the details about the proposed motor servo control system, containing the system architecture, trajectory generation, PID controller, decoder, drive signal for motor and host interfaces etc. Section III elaborates verification result, and then section IV brings the finding of this work and further improvement.

II. SYSTEM DESCRIPTION AND DESIGN METHOD

The architecture of the proposed motor control system is shown in Fig. 1. Trajectory generation module and host interface which are used for communication with extra devices are implemented by software running in the Nios II embedded processor. PID controller, motor drive signal generation and decoder module for decoding motor position signals are achieved by hardware as IP cores in FPGA, which are integrated with Nois II processor through Avalon on-chip bus [16]. The power consumption of the system is about 900mW (not considering the efficiency of power supply), which is much less than DSP/microprocessor-based system, and is similar with specialized chips, such as LM629.

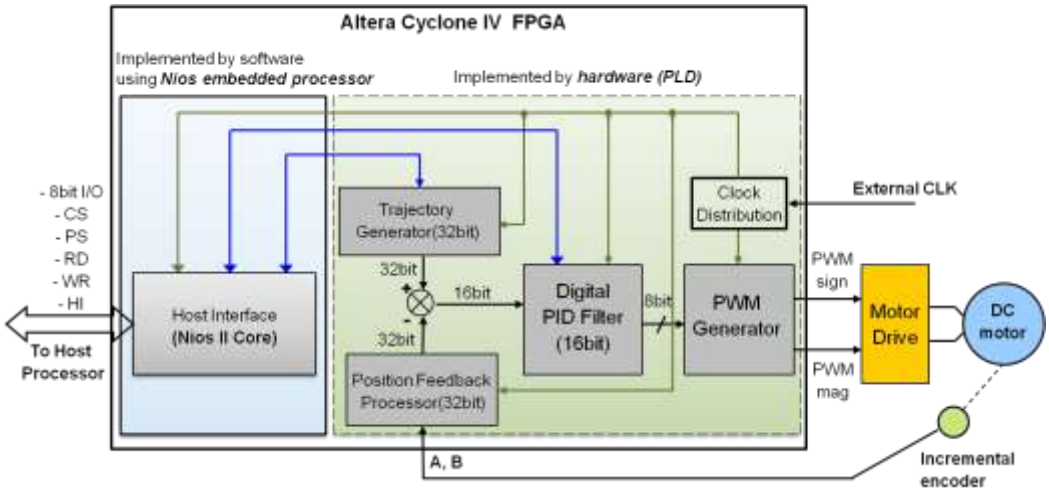

Fig. 1 Architecture of the SoC-based motor control system

A. *Trajectory Generation*

Desired motor motion trajectory is provided by an independent motion generation algorithm. In this scheme, the trajectory profile gets information from input commands via host interface and computes a new current position for each sample interval. For finishing the full computation, the target position, the maximum velocity, the constant acceleration and deceleration should be used as input parameters. Because of position feedback coming from the digital encoder, the dimension of calculation adopts sample counts. Therefore, based on the trajectory profile, the control of instantaneous position at each sample interval can be determined as follows.

1. Calculate the transient trajectory in the acceleration stage

$$P_{pre}(t) = V_{init} * t + \frac{1}{2} A_+ * t^2 \quad t \in [0, t_{pre}] \tag{1}$$

Where $P(t)$ is a transient position in every sample interval t. A_+ represents the acceleration, and V_{init} is the initial velocity. t_{pre} notes the time interval in acceleration stage, and it can be expressed as:

$$t_{pre} = (V_{max} - V_{init})/A_+ \tag{2}$$

2. Compute the transient trajectory in deceleration stage

In deceleration stage, the computation is similar to step 1, deceleration A_- should replace A_+, and the transient position in this stage is defined by

$$P_{post}(t) = V_{max} * t - \frac{1}{2} A_- * t^2 \quad t \in [0, t_{post}] \tag{3}$$

Also, time interval of deceleration stage t_{post} can be expressed by

$$t_{post} = (V_{max} - V_{end})/A_- \tag{4}$$

3. Determine whether the whole trajectory has a constant speed stage

Although $P_{pre}(t)$ and $P_{post}(t)$ can be calculation based on input trajectory parameters, the motion control may have three conditions according to the target position valid input.

$$\begin{cases} P_{pre} + P_{post} < P_{target} & (condition\ 1) \\ P_{pre} + P_{post} = P_{target} & (condition\ 2) \\ P_{pre} + P_{post} > P_{target} & (condition\ 3) \end{cases} \tag{5}$$

In condition 1, the whole trajectory has a constant speed region, and related transient position can be calculated as expression (5). In condition 2, constant speed region is needless, and an error should be caused with A_+, A_-, V_{max} or P_{target} parameters' modification in condition 3.

$$Condition\ 1: \quad P_{mid}(t) = V_{max} * t \quad t \in [0, t_{mid}] \tag{6}$$

$$t_{mid} = (P_{target} - P_{pre} - P_{post})/V_{max} \tag{7}$$

$$Condition\ 2: \quad P_{mid}(t) = 0 \tag{8}$$

4. Obtain the entire trajectory generation

According to above-mentioned analysis, the entire trajectory should be generated

$$P_{target}(t) = P_{pre}(t) + P_{mid}(t) + P_{post}(t) \tag{9}$$

Due to the linear operation in trajectory generation algorithm, the motion trajectory is designed by hardware logic in FPGA. Furthermore, the algorithm implemented by hardware is good for efficiency comparing with software running in Nios II processor. Fig. 2 illustrates a motion trajectory generated with input parameters: $V_{init} = 0\ counts/sample$, $V_{max} = 7\ counts/sample$, $A_+ = A_- = 2.33e-4\ counts/sample^2$, $P_{target} = 200000\ counts$. After implementation with VHDL (hardware description language), a functional simulation has been realized in ModelSim environment, shown in Fig. 3.

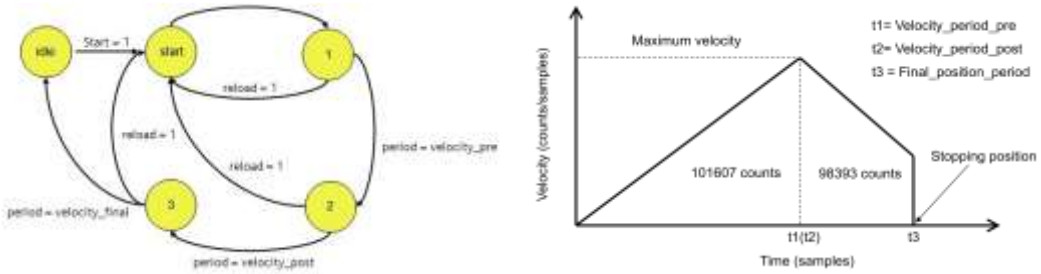

Fig. 2 Motion trajectory diagram

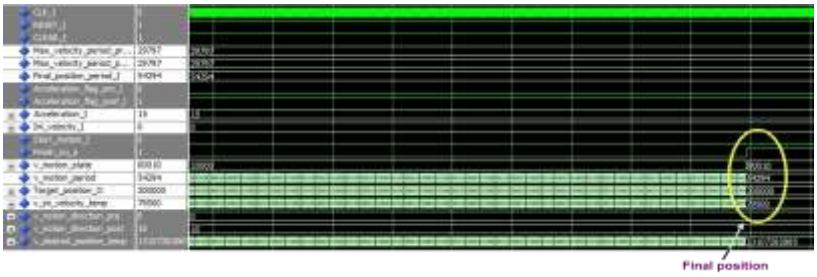

Fig. 3 Functional simulation of trajectory generation

B. Improved PID Controller

On the basis of motor's mathematical model in dynamic performance, the purpose of PID controller should get the closed-loop control in the best response by means of proportion and differential and integral adjustment. In traditional condition, the PID algorithm is usually realized by software method in microprocessors. However, when this method is adopted, some bad influences may affect the control performance under a special condition, such as program fleet and computer malfunction. To overcome the factors mentioned above, a PID controller is adopted by hardware logic in FPGA. Moreover, the higher arithmetic speed and higher reliability can also be achieved based on VLSI technology [17].

It is known that the integral link in PID module aims to remove static error, but easy to induce system shock. Many modified PID algorithms have been introduced [18, 19]. In order to overcome the deficiencies, the PID controller is improved based on integral-separation technology [18]. When the overshoot is too large, it cancels integration action until the overshoot is lower than the separation threshold value. Replacing the differential item as backward difference, classical digital PID control algorithm can be expressed as:

$$u(n) = K_p\{e(n) + \alpha \frac{T_{is}}{T_i} \sum_{k=1}^{n} e(k) + \frac{T_d}{T_{ds}}[e(n) - e(n-1)]\} \tag{10}$$

Considering the separation threshold value, the condition coefficient α should be added [12]

$$\alpha = \begin{cases} 1, |e(n)| \leq \varepsilon \\ 0, |e(n)| > \varepsilon \end{cases} \tag{11}$$

Where $u(n)$ is the output motor control signal from PID controller, $e(n)$ is the error signal between input position and motor encoder feedback signals, T_{is}, T_{ds}, T_i, T_d and α are the sample interval, the derivative sample interval, integral time constant, differential time constant and artificial separation threshold, respectively. When $\alpha = 1$, the algorithm is PID control, while $\alpha = 0$, it changes into PD control.

The detail of improved PID controller is shown in Fig. 4. Z^{-1} is a back-shift operator, while T_i and T_d are generated by PLL module from clock distribution network of the system, and the related multiplication is implemented using hardware multipliers in FPGA for extremely improving operating speed and saving logical resources. Fig. 5 shows the functional simulation result for a certain trajectory. It is seen that after certain transition, when the position error $e(n)$ is close to zero, the real position can reach the desire position.

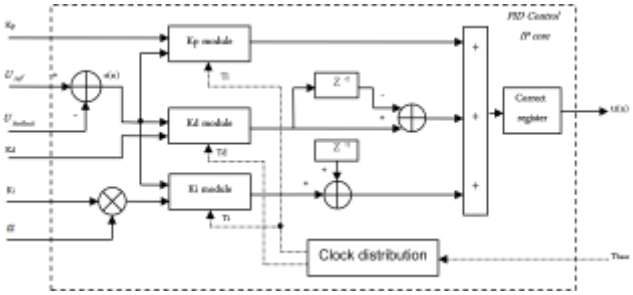

Fig. 4 Schematic diagram of PID IP core

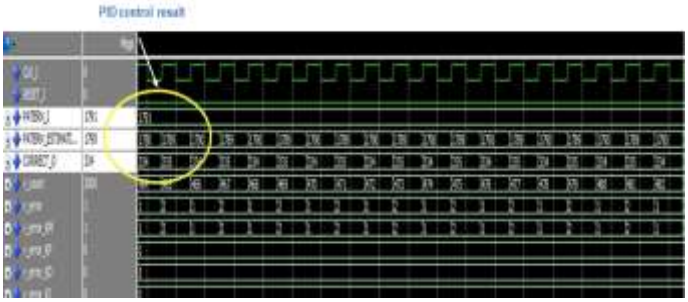

Fig. 5 Functional simulation of PID IP core

C. PWM and Position Decoder Module

Similar to traditional motor control system, PWM signals here are exciting sources for motor drives [20], and the format of input signals in decoder module matches incremental encoder. PWM module admits an input 8-bit signal, which includes a duty cycle signal in 0~7 bit region and a direction signal using the most significant bit. The architecture of PWM module consists of three software accessible registers: a duty cycle input register, a PWM direction output register and a PWM magnitude output register, which are all connected to the avalon bus for communication with Nios II processor.

Position decoder module is crucial for motion control as an input parameter of PID controller. Therefore, signal quality and condition monitoring should be considered. This module can be described in Fig. 6, which includes digital filter, actual counter, control logic, and four registers, such as enable register, control register, status register and data register.

The digital filter is a flip-flop filter, which is responsible for rejecting noise on the incoming quadrature signals from incremental encoder. And it is based on Schmitt-trigger inputs and a three-clock cycle delay filter combining to reject low level noise and large, short duration noise spikes that typically occur in the motor system application.

Actual counter is used to decode the encoder signals synchronized by control logic, and then latch the decoding signals in data register. Meanwhile, the control commands, such as several enabled signals of counting, index interrupting, overflow interrupting, errors, and count latch, etc., and are written into control registers which provide control logic with the basis for the determination. In the working process, error status and overflow information can be obtained in the status register, and the output signals, connecting to Nios II processor, can be switched by enabling the register independently. Timing diagram of functional simulation (Fig. 7) clearly shows the decoder signals in decimal format.

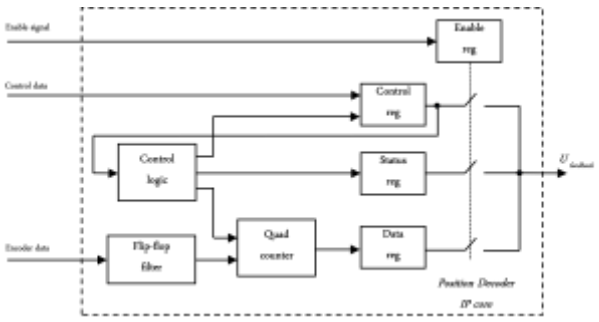

Fig. 6 Schematic diagram of Position Decoder IP core

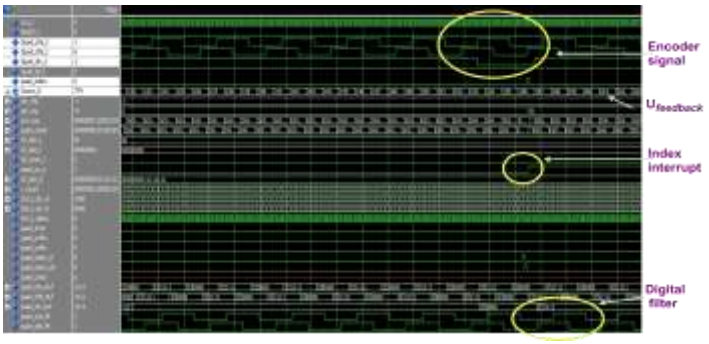

Fig. 7 Functional simulation of Position Decoder IP core

D. Input / Output Elements (IOE) in the Closed-control Loop

As a SoC-based motor control IC, I/O interface is necessary to communicate with host processor and users. It is provided by a Nios II embedded processor. A control of the processor is needed in order to synchronize the state of system and to avoid undesired situations during the working process (Command operation, Interrupt generation, registers control, state management, etc.). The interface has five control lines and one 8-bit data bus. This IC can be controlled by the *RD*, *WR*, *CS* and *PS* lines and by use of the busy flag – *HI* of the status.

E. Proposed Motion Control System for DC Motor Device

As the internal architecture shown in Fig. 1, the FPGA in this paper adopts Altera Cyclone VI EP4CE15F17, which has 15,408 LEs, maximum 166 user IOs, 112 multipliers, total 516,096 RAM bits, and a Nios II embedded processor, running in SDRAM with 100MHz rate. As in Fig. 1, there are two parts in the whole motor IC. Part-I performs the functions of sequential control and communication interface mentioned in section D. These functions are implemented in the Nios II system by software, which is shown in Fig. 8. Its flow chart of service routine for motor control is plotted and shown in Fig. 9. Part-II is implemented by hardware using PLD due to the need of fast-response and simple multiplications and additions computation. All of the hardware parts are encapsulated as IP cores for modularity and reusability. Furthermore, all parts in part I and part II connect to avalon bus for communication with each other, which is a 32-bit on-chip bus designed by Altera corporation. Every module in this part has been described above. In order to set up integrated motion control system for motor device, motor drive circuit and a brushed DC motor with incremental encoder have been involved. Only two PWM output signals (direction and margin) within motor IC connect to the motor drive circuit, which is comprised of h-bridge EFTs and relevant driver chip HIP4081A roughly. For close loop control, motor position signals coming from incremental encoder should be monitored and sent to decoder module in motor IC as feedback signals. After that, the difference between current position signal and desire position signal from trajectory generation module should be sent to PID filter module as input parameters for control. The purpose of following PID filter and PWM generation modules is to generate proper PWM signals for driving motor accurately. Therefore, the type of closed control composed by hardware IP cores (position decoder, trajectory generation, PID filter and PWM generation) contributes significantly to dropping down the control latency, compared with DSP/microprocessor-based system using software. The resource utilization of the overall circuit includes a Nios II embedded processor IP and application IP cores, use 15,240 LEs (13%), 1,325,678(33%) RAM bits, and four 9-bit multipliers.

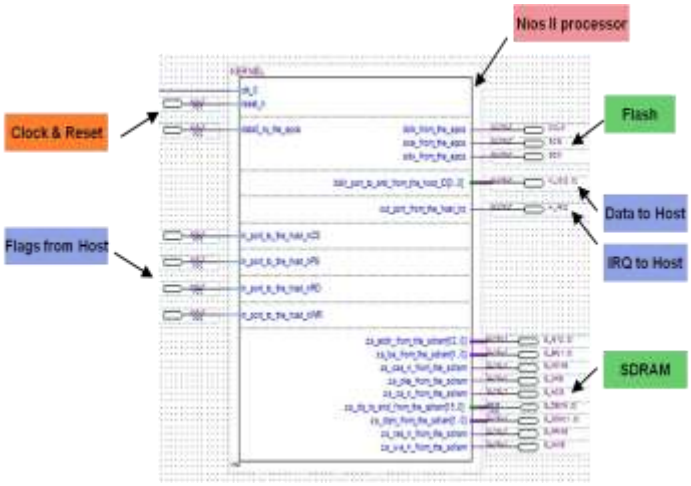

Fig. 8 Nios II system generation in Quartus software

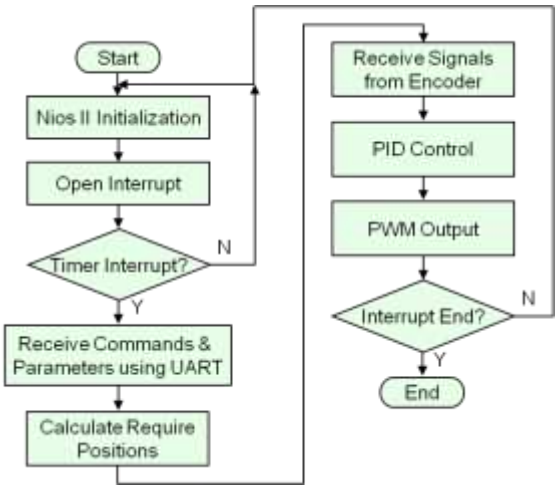

Fig. 9 Service routine flow in Nios II processor

Based on SoPC technology, another point needs to be mentioned here. Not only these modules mentioned above, but also any extra function modules can be designed to modify or improve the control IC. Benefiting from on-chip bus technology, each module can be packaged as IP core to connect with others flexibility. Generally, this control IC can be classed as "programmable system" or "reconfigurable system".

III. VERIFICATION

In order to evaluate the proposed motor IC's performance, a real verification system has been designed shown in Fig. 10. It includes a motion control IC implemented in Altera FPGA (Cyclone IV EP4CE15F17) board, a motor drive circuit board mentioned above, and a 24V DC brushed motor with 16-bit optical quadrature incremental encoder as the position sensor. Host board is used to simulate host processor, and control board is a carrier for mounting the motion control IC. The sample interval is $128\mu s$ for a system clock frequency of 16MHz. This sampling frequency can be modified flexibly according to the requirement by configuring PLL. Three motion cases have been planned for test. The purpose of first case is to test velocity control. The motor velocity can be changed in real-time condition. In case 2, motor should always run in the forward direction with specific velocity and stop in the specific position. However, in case 3, target position can be changed in motion state.

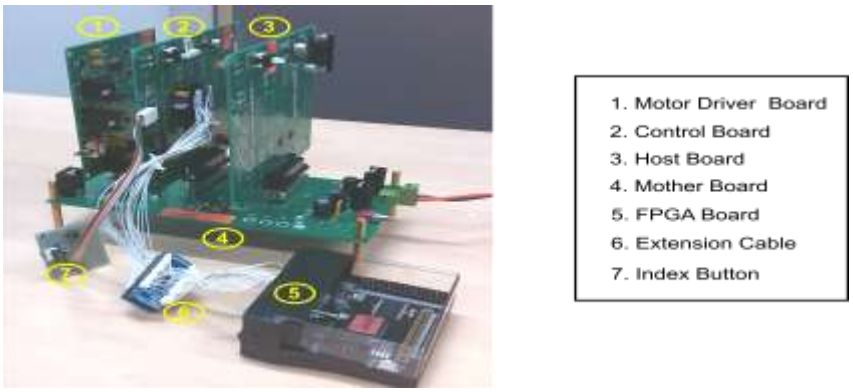

Fig. 10 The real verification system setup

Under the condition of case 1, motor velocity has been changed four times in the whole motion processing. Fig. 11 shows the monitor of motor velocity at each sampling point, and the motion trajectory has been recorded in real time shown in Fig. 12. The slope of trajectory curve increases gradually following the velocity modification in real time, while, desired position and real position curves are complete overlapping.

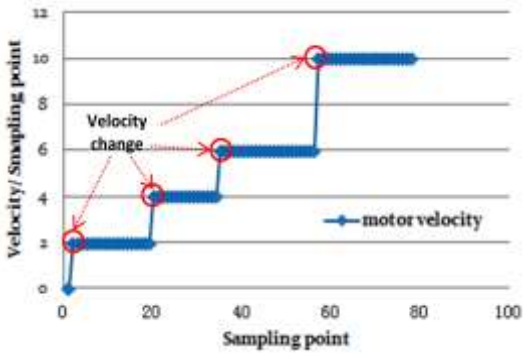

Fig. 11 Motor velocity modification in real time (Case-1)

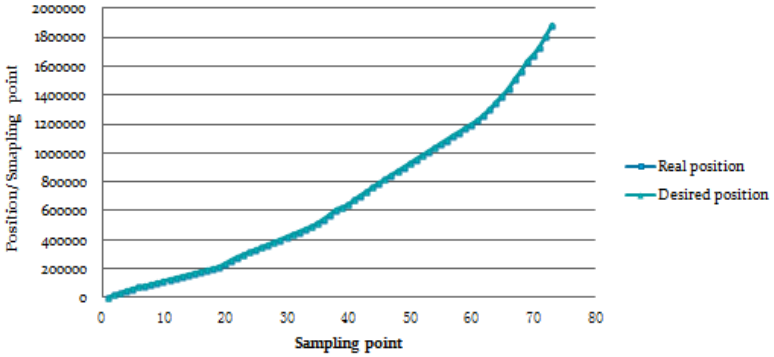

Fig. 12 Motion trajectory monitor in real time (Case-1)

The target position parameter about case-2 is 800,000 counts with 2 counts/sample velocity. The test condition is summarized as following: *acceleration*, *constant velocity*, *deceleration* and *stop*. The verification system records the real position in the decoder module of motion control IC during the specific interval, and the desired position could be calculated in each sampling point. Fig. 13 shows the comparison between desired position and real position. The average error between them is about 1.7%. Case 3 seems more complex, in which the requirement can be described as: *acceleration*, *deceleration*, *reverse acceleration*, *reverse deceleration*, *acceleration*, *constant velocity*, *deceleration*, and *stop*. The result is presented by Fig. 14, which shows the performance of close-loop motion control such as velocity control, trajectory control seems well-tested.

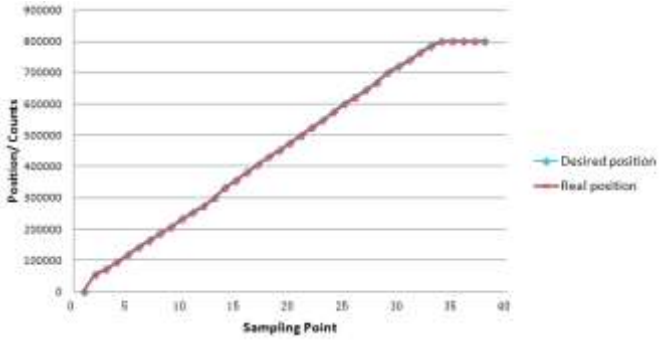

Fig. 13 Motor trajectory monitoring in real time (Case-2)

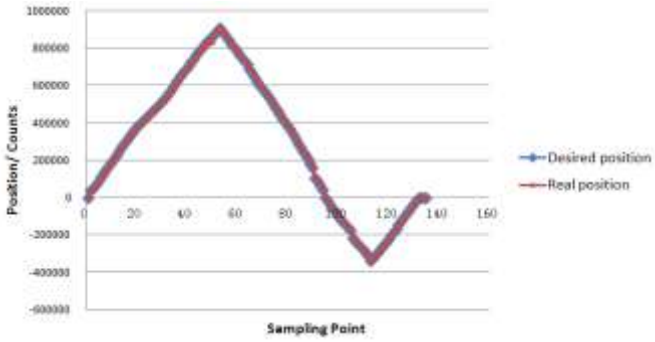

Fig. 14 Motion trajectory monitoring in real time (Case-3)

IV. CONCLUSIONS

In this paper, a motion control IC implementation for motor devices based on SOPC technology was proposed. The aim of this work was to present a novel integrated solution for motor control. Except for low consumption, low cost and high reliability, compared with traditional ASIC or DSP solutions, it still has several new characteristics of low control latency, well scalability and parameters monitoring easily in motion processing. The superior performance should be achieved by hardware

IP cores (FPGA matrix) and embedded processor in high level, in which motion trajectory algorithm plays the most import role. In order to validate the performance of motion control IC purposed, a verification platform has been set up, including host computer for communication, motor drive circuit and brushed DC motor with optical encoder as feedback input in control loop. Finally, the experiment results of the closed-loop motion control for trajectory, velocity and acceleration of DC motor. In the test, compared with desired parameters, trajectory algorithm and improved PID controller are used to generate proper PWM signals in real time, which are demonstrated to be a good result. Moreover, the result also benefits from the parameters monitoring ability of the motion control IC.

This paper focuses on the prototype design of the motion control IC, some crucial technologies for application have not been considered. In the future, research work will concentrate on the optimization algorithms about tuning PID controller intelligently, considering classical methods and meta-heuristic. And another import work is further verification on real devices using the DC motor in real-time control application, such as computed tomography.

REFERENCES

[1] Salim J. Ohri and Naveen, "Speed control of a DC motor using fuzzy logic based on LabVIEW," *International Journal of Scientific and Research Publications*, vol. 3, no. 6, pp. 1-5, 2013.

[2] A. Bature, M. Muhammad, and A. M. Abdullahi, "Identification and Real Time Control of a DC Motor," *IOSR Journal of Electrical and Electronics Engineering*, vol. 7, no. 4, pp. 54-58, 2013.

[3] S. Sonoli and K. N. Raju, "Implementation of FPGA based PID controller for DC motor speed control system," *Proc. WCECS'10*, vol. II, 2010.

[4] K. J. Astrom and T. H. Hagglund, "New tuning far PID controllers," *Proc. of 3rd European Conference*, pp. 2456-2462, 1995.

[5] M. Dashti, K. Shojaee, S.M.H. Seyedkashi, and M. Behnam, "Tuning of Digital PID Controller using Particle Swarm Optimization," *Proc. of 29th Chinese Control Conference*, pp. 3383-3389, 2010.

[6] Y. Shang and R. Bouffanais, "Consensus reaching in swarms ruled by a hybrid metric-topological distance," *European Physical Journal B*, vol. 87, pp. 294-296, 2014.

[7] N.J. Cheung, X.M. Ding, and H.B. Shen, "OptiFel: a convergent heterogeneous particle swarm optimization algorithm for Takagi-Sugeno fuzzy modeling," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 919-933, 2013.

[8] T. Asfour, K. Regenstein, and P. Azad, "ARMAR-III: An integrated humanoid platform for sensory-motor control," *Proc. of 6th IEEE-RAS*, pp. 169-175, 2006.

[9] Y. K. Hou, H. J. Xu, and Q. H. Tong, "The design of DC motor servo control system based on ARM and LM629," *Modern Manufacturing Engineering*, vol. 4, pp. 124-127, 2010.

[10] J. J. Fang, H. Zhang, and S. L. Liu, "Study on control system based on LM629 for mobile robot," *Machinery Manufacturing Engineer*, vol. 3, pp. 106-107, 2005.

[11] Zhang Chengjun, Wu Xiaoguang, and Zuo Xiaoyan, "FPGA soft-core based step motor driving," *Proc. ICECE'10*, pp. 1035-1038, 2010.

[12] Y. S. Kung, K. H. Tseng, and F. Y. Tai, "FPGA-based servo control IC for X-Y table," *Proc. IEEE Int. Conf. Ind. Technol.*, pp. 2913-2918, 2006.

[13] L. X. Wang, "A course in fuzzy systems and control," *Englewood Cliffs*, NJ: Prentice-Hall, 1997.

[14] I. Takahashi and T. Noguchi, "A new quick response and high efficiency control strategy of an induction motor," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 5, pp. 820-827, 1986.

[15] *Nios II Processor Reference Handbook*, Altera, 2013.

[16] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes, "Features, design tools, and application domains of FPGAs," *IEEE Trans. On Industrial Electronics*, vol. 54, no. 4, pp. 1810-1823, 2007.

[17] J. T. Han, S. H. Jiao, and Z. Y. Jiang, "Study on integral-separate PID improved predictive control for electro-hydraulic control system of strip steel CPC," *Advanced Materials Research*, vol. 422, pp. 250-256, 2012.

[18] K. Xuan, X. Bao, and C. Li, "Application of integral-separated PID algorithm in orbit feedback," *Proc. ICALEPCS'11*, pp. 171-173, 2011.

[19] W. B. Wang, X. D. Wang, and D. J. Liu, "Incomplete differential PID neural network temperature controller for thermal process," *Proc. IFOST'11*, vol. 2, pp. 920-923, 2011.

[20] Z. Zhou, T. Li, T. Takahahi, and E. Ho, "Design of a universal space vector PWM controller based on FPGA," *Proc. Appl. Power Electron. Conf.*, pp. 1698-1072, 2004.

Xuan Cheng received the B. S. degree in electronic engineering form Sichuan University of China in 2005, and the Ph. D degree in precise instrument from the college of Instruments Sciences and Technology in Beihang University of China, in 2012.

He has a long research experience on digital electronic system and optoelectronic system design and simulation. His current job is doing post-doc research on electro-mechanical system simulation and related algorithms and software design. He has published in academic journals and conferences, in the areas of SoC design, optical-electronic imaging technology and electronic CAD tool design.

Dr. Xuan is a research associate in Mechanical Engineering of Tsinghua University of China. He is a reviewer of the Journal of Chinese Optics Letters, which is indexed by SCI & EI.

Xiangdong Yang received the B. S. degree in mechanical engineering from Tsinghua University of China in 1989, and the Ph. D degree in

mechanical engineering from the Department of Precision Instrument and Mechanics in Tsinghua University of China, in 1995. Dr. Yang is a professor in the Department of Precision Instrument and Mechanics of Tsinghua University. He is also a doctoral supervisor in Tsinghua University of China.

He has a long research experience on robotics, motion planning and smart control for robot application.

Dianssong Cao received the M.S. degree in control engineering from Beihang University in 2000 and B.S. degree in computer science from Northwestern Polytechnical University in 1992. Since 2007 he has worked as senior research scientist at Siemens Limited China for various innovation topics. Before that he had worked at GE Healthcare as a senior R&D engineer. The fields of research interested cover the new technologies in computer science and control engineering.