J Intell Robot Syst (2008) 53:247–261
DOI 10.1007/s10846-008-9240-7

DSP/FPGA-based Controller Architecture for Flexible Joint Robot with Enhanced Impedance Performance

J. B. Huang · Z. W. Xie · H. Liu · K. Sun · Y. C. Liu · Z. N. Jiang

Received: 14 June 2007 / Accepted: 22 April 2008 /
Published online: 23 May 2008

Abstract Some practical issues associated with enhancing the Cartesian impedance performance of flexible joint manipulator are investigated. A digital signal processing/field programmable gate array (DSP/FPGA) structure is proposed to realize the singular perturbation based impedance controller. To increase the bandwidth of torque control and minimize the joint torque ripple, boundary layer system and field-oriented control (FOC) are fully implemented in a FPGA of each joint. The kernel of the hardware system is a peripheral component interface (PCI)-based high speed floating-point DSP for the Cartesian level control, and FPGA for high speed (200 us cycle time) multipoint low-voltage differential signaling (M-LVDS) serial data bus communication between robot Cartesian level and joint level. Experimental results with a four-degree-of-freedom flexible-joint manipulator under constrained-motion task, demonstrate that the controller architecture can enhance the robot impedance performance effectively.

Keywords Impedance control · DSP · FPGA · M-LVDS serial data bus · Torque ripple · Flexible joint

1 Introduction

In the past few years, the robotics community evolved growing interest in dexterous robots, which are designed for operation in space or hazardous environment. A

J. B. Huang (✉) · Z. W. Xie · H. Liu · K. Sun · Y. C. Liu · Z. N. Jiang
Institute of Robotics, Harbin Institute of Technology, 150001,
Harbin, People's Republic of China
e-mail: Huang_jianbin@sina.com

H. Liu
Institute of Robotics and Mechatronics, German Aerospace Center (DLR),
Oberpfaffenhofen, Germany
e-mail: Hong.liu@dlr.de

main focus of this kind of robots is the ability to perform compliant manipulation when contacting with an unknown environment. One possible approach to achieve compliant behavior is impedance control, which was extensively theorized by Hogan [1] and experimentally applied by Kazerooni et al. [2]. The basic idea of impedance law is defined as a mass–spring–damp system, where the interacting force is designed to be a linear combination of the position error (spring), first derivative of the position (damping), and second derivative of the position (inertia).

The equations of a manipulator's motion are usually derived in terms of joint coordinates, which is in "joint space". However, the task is described in end-point coordinates, the "Cartesian space" of the manipulator. The goal of Cartesian impedance control, as developed by Khatib [3], is to realize a desired dynamical relationship between the motion of end-effector and the external torques. Various approaches of impedance control were developed for robotic systems based on rigid body models, which neglected the effects of joint elasticity. In this paper, the joint elasticity is considered with gravity compensation.

The effective method of the Cartesian impedance control for a flexible joint robot is based on a singular perturbation analysis of the flexible joint model [4]. From this perspective an impedance controller may be designed in the same manner as for a robot with rigid joints. The flexibility of the joints is then treated in a sufficiently fast inner torque control loop. The stability of the system has been proven by Ott et al. [5]. This approach is very attractive at a first glance due to the simplicity of the resulting controllers, but it has some strict structural and soft requirements for a robot to realize better impedance characteristics, such as high-speed inner control, data communication requirement between inner and outer control loop, and torque dynamics requirement in the joint control. Only little work has been done on these requirements for flexible joints' robot so far.

With development of the IC technology, field programmable gate array (FPGA) has become the mainstream in complex logic circuit design due to its flexibility, ease to use, and short time to market. Some researchers have incorporated control algorithms into FPGA to improve the performance of the servo control system. The system developed by Takahashi and Goetz [6] could run a current control algorithm with a FPGA to increase the bandwidth of the current loop control; Tzou and Kuo [7] have performed the vector and velocity controls of a PMAC servo motor by using FPGA technology successfully; FPGA-based motion control has been utilized in other works [8, 9].

This paper focuses on enhancing manipulator's impedance performance, which includes following three aspects: (1) DSP and FPGA are used to design ideal hardware architecture to implement the Cartesian impedance control. (2) Multipoint low-voltage differential signaling (M-LVDS) serial data bus is designed with 200us cycle time by the hardware architecture. (3) Space vector pulse width modulation (SVPWM) algorithm is implemented in FPGA to minimize torque ripple. In particular, item (2) is contrasted with similar approaches of serial data bus, and item (3) is compared with the square wave control. All the design aspects are realized in a 4-degree-of-freedom (DOF) flexible joint robot, as shown in Fig. 1.

This paper will be organized as follows: Section 2 gives Cartesian impedance control architecture of the flexible joint manipulator; Section 3 describes three important requirements to implement the control algorithm, including the hardware

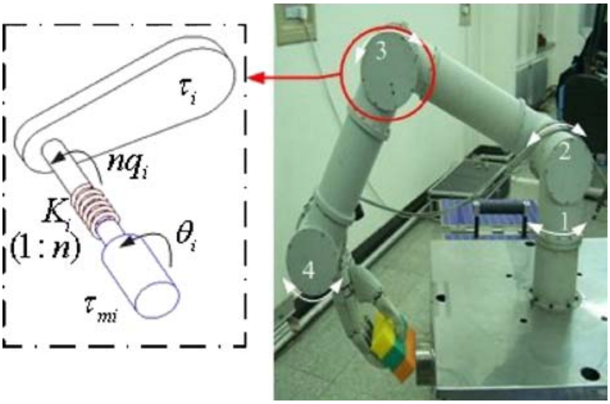

Fig. 1 Model of a flexible joint robot and the 4-DOF HIT robot

architecture, M-LVDS serial data bus, and joint FPGA-based servo system; Section 4 presents the experiments and the conclusion of this paper are drawn in Section 5.

2 Cartesian Impedance Control Architecture

The joint of the HIT robot is powered by electro-mechanical drives, which consists of brushless direct-current (DC) motor and light harmonic drive gear. The main sources of flexibility are the harmonic drive gear and the torque sensor. Therefore, a flexible joint robot model [10] will be considered as follows.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau + \tau_{\mathrm{ext}} \tag{1}$$

$$B\ddot{\theta} + \tau/n = \tau_m \tag{2}$$

In this equation, $M(q)$ represents the inertia matrix, $C(q,\dot{q})\dot{q}$ is the centrifugal/ Coriolis term, and $g(q)$ is the gravity term. The vector of the joint torques is given by $\tau = K(\theta - nq)$, where θ, q indicate the vector of the motor angle and the link side joint angle, respectively, n is the transmission ratio, and K, B are diagonal matrices, which contain the joint stiffness and the motor inertias (see Fig. 1). In addition, τ_{ext} are the external torques and τ_{m} are the generalized motor torques, which is regarded as input variables for the control design.

In the following, it is assumed that the position and orientation of the end effector's can be described by a set of local coordinates $x \in R^m$, and the forward kinematics $x = f(q)$ is known. With the Jacobian $J(q) = \partial f(q)/\partial q$, Cartesian velocities and accelerations can be written as

$$\dot{x} = J(q)\dot{q} \tag{3}$$

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{4}$$

Notice that in this paper only the nonsingular case is treated, thus it is assumed that the manipulator's Jacobian $J(q)$ has full row rank in the considered region of the workspace.

The external torques τ_{ext} shall be related to the vector of generalized external forces F_{ext} via $\tau_{\text{ext}} = J(q)^T F_{\text{ext}}$. To specify the desired impedance behavior, the position error $\tilde{x} = x - x_{\text{d}}$, which is between real position x and a virtual equilibrium position x_{d} is introduced. Then the control objective is to achieve a dynamical relationship of the form

$$\Lambda_{\text{d}}\ddot{\tilde{x}} + D_{\text{d}}\dot{\tilde{x}} + K_{\text{d}}\tilde{x} = F_{\text{ext}} \tag{5}$$

between $\tilde{x}$ and F_{ext}, where Λ_{d}, D_{d}, and K_{d} are the symmetric and positive definite matrices of the desired inertia, damping and stiffness, respectively.

From Eqs. 1, 3 and 4, the relationship between the Cartesian coordinates x and the joint torques τ can now be written as

$$\Lambda(x)\,\ddot{x} + \mu(x,\dot{x})\,\dot{x} + J(q)^{-T} g(q) = J(q)^{-T}\tau + F_{\text{ext}} \tag{6}$$

where the matrices $\Lambda(x)$ and $\mu(x,\dot{x})$ are given by

$$\Lambda(x) = J(q)^{-T} M(q)\, J(q)^{-1} \tag{7}$$

$$\mu(x,\dot{x}) = J(q)^{-T}\left(C(q,\dot{q}) - M(q)\, J(q)^{-1}\dot{J}(q)\right) J(q)^{-1}. \tag{8}$$

Similar to the external torques, the gravity torques $g(q)$ and the joint torque τ can be rewritten in form of the equivalent task space gravity forces $F_g(x)$ and the new input vector F_τ. Therefore, the system equations finally have the form

$$\Lambda(x)\,\ddot{x} + \mu(x,\dot{x})\,\dot{x} + F_g(x) = F_\tau + F_{\text{ext}}. \tag{9}$$

The matrices $\Lambda(x)$ and $\mu(x,\dot{x})$ are the inertia matrix and the Coriolis/centrifugal matrix with respect to the coordinates x.

The classical impedance control law can be directly computed from Eq. 9. The control input F_τ, which leads to the desired closed loop system (Eq. 5) is given by

$$\begin{aligned} F_\tau &= \Lambda(x)\,\ddot{x} + \mu(x,\dot{x})\,\dot{x} + F_g(x) - \Lambda(x)\,\Lambda_{\text{d}}^{-1}\left(D_{\text{d}}\dot{\tilde{x}} + K_{\text{d}}\tilde{x}\right) \\ &\quad + \left(\Lambda(x)\,\Lambda_{\text{d}}^{-1} - I\right) F_{\text{ext}} \end{aligned} \tag{10}$$

The feedback of external forces F_{ext} can be avoided when the desired inertia Λ_d is identical to the robot inertia $\Lambda(x)$. This Cartesian impedance controller is then actually implemented via the joint torques τ as $\tau = J(q)^T F_\tau$.

Considering the flexibility of the joint and on the assumption that the manipulator is working at steady state, the gravity compensation can be obtained on line when we introduce a new variable named "gravity-biased" motor position.

$$\tilde{\theta} = \theta - K^{-1} g(q_{\text{d}}) \tag{11}$$

One common approach to implement the Cartesian impedance control (Eq. 10) is the singular perturbation approach in which the flexible joint model is virtually split up into a fast subsystem for the joint torques τ and a slow subsystem for the link positions q. Based on the subsystems it is possible to design an inner joint torque τ loop controller as the boundary layer system, and an outer link position q loop controller as the quasi-steady state system. In the non-redundant and non-singular case the complete controller is finally given by

$$\tau_{\mathrm{m}} = \left(\tau_{\mathrm{d}} + K_{\tau P}\left(\tau_{\mathrm{d}} - \tau\right) - K_{\tau D}\dot{\tau}\right)/n \tag{12}$$

$$\tau_d = g\left(\tilde{\theta}\right) - J\left(\theta\right)^T \left(K_{\mathrm{d}}\left(x\left(\theta\right) - x_{\mathrm{d}}\right) + D_{\mathrm{d}}\dot{x}\left(\theta\right) - \Lambda\left(x\right)\ddot{x}_{\mathrm{d}} - \mu\left(x,\dot{x}\right)\dot{x}_{\mathrm{d}}\right) \tag{13}$$

3 Control Requirements

The quasi-steady state system (Eq. 13) includes abundant mathematics operation and will be implemented in the Cartesian level. Meanwhile, the boundary layer system (Eq. 12) should be controlled in a high frequency and should update the desired torque from the Cartesian level into the joint level. So, the Cartesian impedance control presents three important requirements, which should be fulfilled to enhance the robot compliant performance:

- R1. The hardware system consists of Cartesian control level and joint control level. Complex control algorithm and fast computation should be implemented in the Cartesian level; servo control should be carried out fast in the joint level.
- R2. To realize precise real time control, the ideal minimum cycle time for data transmission between the Cartesian controller and joint controller is necessary.
- R3. In the joint level, a good dynamic torque response with small ripples should be obtained.

In the following we will put forward solutions to achieve the R1, R2, and R3 at length.

3.1 R1: DSP/FPGA–FPGAs System

Success of the robot control system depends not only on the control algorithm, but also on the hardware controller structure. Figure 2 illustrates a schematic of the proposed control architecture.

In the joint control level, a Cyclone™ FPGA EP1C20 with densities 20060 logic elements (LEs) and 288 Kbits of RAM was chosen, which can achieve a more flexible implementation of the joint controller with a high control rate and a small-sized joint electronics. Furthermore, by means of flexible DSP structures and integrated processing units, modern FPGAs enable the efficient implementation of algorithms (Eq. 12) directly in the joints. Diverse IO standards support the connection of different hardware components (e.g. motors, brakes, sensors). It is more advantageous over conventional digital controllers, such as microcontrollers or DSP.

In the proposed control architecture, the peripheral component interface (PCI)-based DSP/FPGA board is used for the Cartesian level. A Texas instruments (TI) floating-point digital signal processor (DSP) TMS320C6713 [11] with maximum 1350 MFLOPS is selected to carry out the complex algorithms (Eq. 13) more efficiently. The PCI board exchanges data with PC via PCI bridge controller, and the FPGA converts serial signals from the joint-level FPGAs to parallel signals and transmits them to the DSP via the parallel interface, and vice versa.

Based on the DSP/FPGA–FPGA control architecture, Cartesian impedance control will be implemented efficiently, so the R1 requirement is fulfilled.

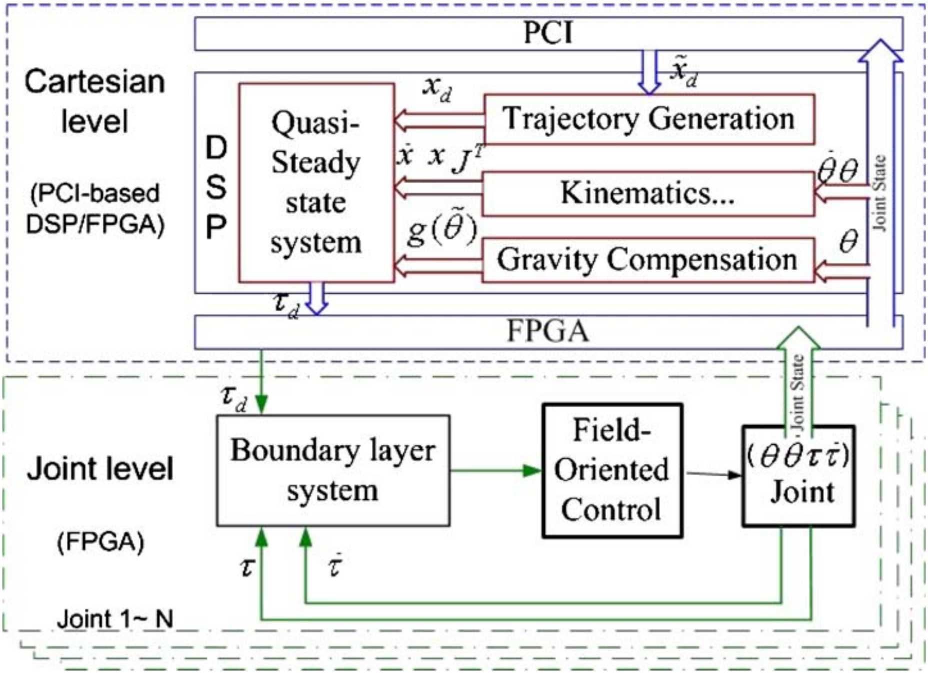

Fig. 2 Diagram of the DSP/FPGA-FPGA control architecture

3.2 R2: M-LVDS Serial Bus

To implement the real-time feedback control of the robot, the Cartesian level should get the feedback of positions and velocities of the joints rapidly to calculate the quasi-steady state system. At the same time, the joint level should update the input data instantly especially in the fast-moving state. So a high speed data bus is needed in our control architecture as the R2 illustrates.

Multipoint low-voltage differential signaling (M-LVDS) allows multipoint configurations of LVDS and it is defined by the ANSI/TIA/EIA-899 M-LVDS standard, which recommends a maximum data rate of 500 Mbps. Because of its low noise,

Table 1 KPP comparison of differential standards

Parameter	RS-422	RS-485	ECL	M-LVDS
Bus	Point-to-point Multidrop	Point-to-point Multidrop Multipoint	Point-to-point Multidrop Multipoint	Point-to-point Multidrop Multipoint
Speed	< 10 Mbps	< 10 Mbps	< 2 Gbps	< 500 Mbps
Power	Moderate	Moderate	High	Low
Signal swing	2 V Min	1.5 V Min	0.8 V to 1 V	480 to 680 mV
Threshold	±200 mV	±200 mV	±200 mV	±50 mV
Common mode range	±7 V	−7 V to +10 V	Varies	±2 V

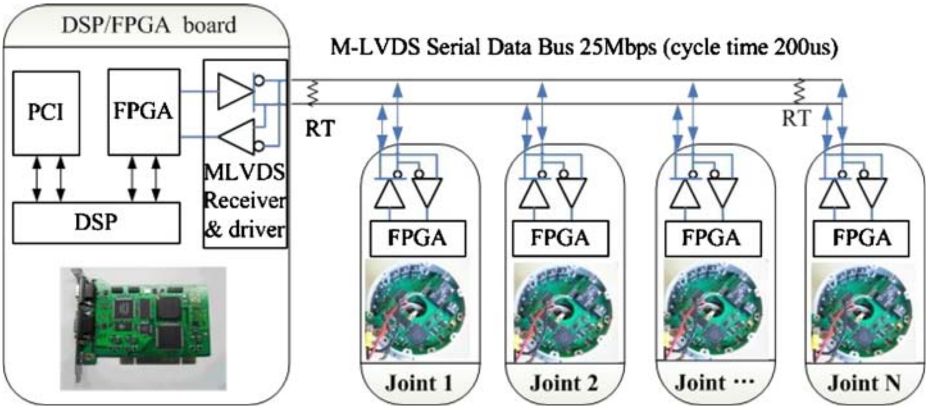

Fig. 3 M-LVDS serial data bus

low power, high-immunity interface, and ease in implementing in FPGA, by means of low-voltage differential line drivers and receivers, a M-LVDS serial bus system was designed for the data communication (half-duplex) between the joints and PCI-based DSP/FPGA board. The M-LVDS serial bus operates on 100-Ω media with a 100-Ω terminal R_T at the driver and the receiver, as shown in Fig. 3. Table 1 below, compares the key performance parameters (KPP) of M-LVDS with other differential serial standards. It is obvious that M-LVDS has great advantages when it comes to power, signal swing, and common-mode range while still providing speed as emitter-coupled logic (ECL) can.

In the software design, M-LVDS serial bus has the following characteristics.

1) M-LVDS serial bus uses a 16-bit cyclical redundancy check (CRC) checksum to detect the majority of communication errors.

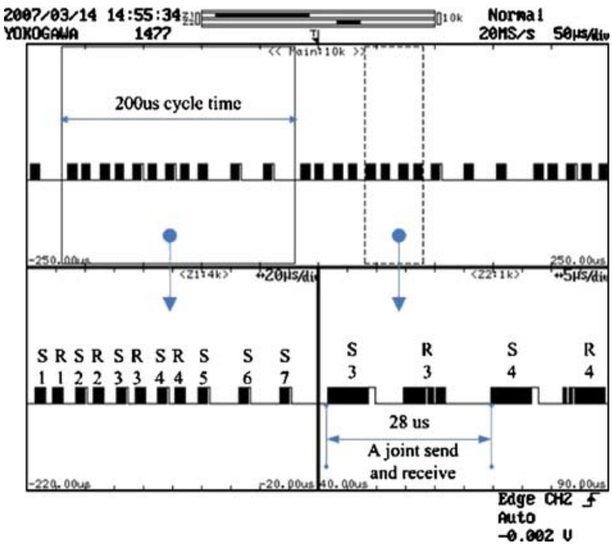

Fig. 4 Four-DOF robot communication by M-LVDS serial bus. S, R represent the Cartesian level sends and receives data, respectively

2) NRZI data encoding and automatic bit stuffing/stripping are used for data encoding.
3) Variable baud rate, super-high communication rate will drag down the speed of the processor, so a bandwidth of 25 Mbps is enough.

Multipoint low-voltage differential signaling (M-LVDS) serial bus is designed for seven joints communication, and cycle time less than 200 us is achieved. The serial bus is embedded into 4-DOF robot systems successfully. All communication and other control programs for all FPGAs are written in VHDL and run in FPGAs. For a 4-DOF robot under working condition, Fig. 4 illustrates that the DSP/FPGA board sends the joint data sequentially per 28 us and the relative joint sends the feedback data immediately.

3.3 R3: Joint FPGA-based Servo System

To control permant magnet synchronous motor (PMSM) efficiently, the field-oriented control (FOC) is often applied. FOC is an excellent control algorithm that is used to control space vectors of magnetic flux, current and voltage. Actually, FOC relies on the SVPWM control strategy. Taking advantage of SVPWM, it is possible to set up the coordinate system to decompose the vectors into a magnetic field-generating axis and a torque-generating axis. So by utilizing SVPWM, the PMSM magnetic flux and torque can be controlled separately.

The motion control of the joint level includes the boundary layer system, the current vector control, M-LVDS serial bus TXD and RXD module, SVPWM and sensors detector. As described in Fig. 5, the boundary layer system and the current vector control are implemented by software using Nios embedded processor, and the other modules are implemented by hardware programmable logic device (PLD). Therefore, all of the functions for the joint motion control are integrated and realized

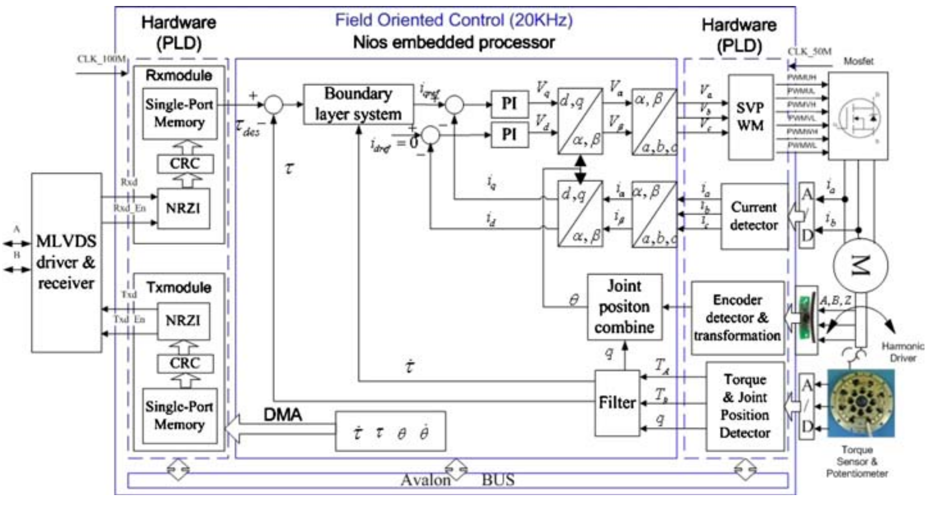

Fig. 5 Internal architecture of FPGA-based servo controller

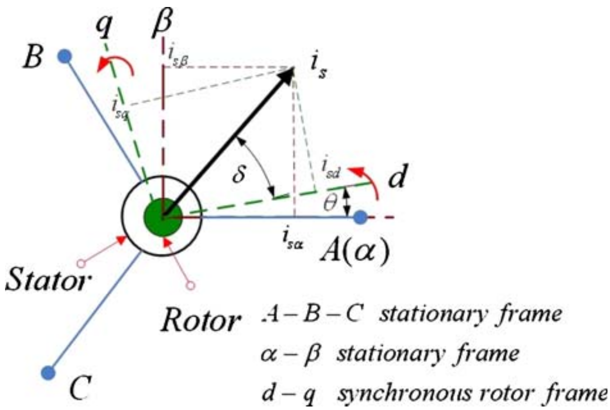

Fig. 6 Reference frames for vector control

in the single system on a programmable chip (SOPC), and the control frequency of the FOC can up to 20 KHz.

As shown in Figs. 5 and 6, FOC transforms the three-phase stator current i_a, i_b, i_c to the torque feedback current i_{sq} and field flux feedback current i_{sd} via Clarke and Park matrix transformations in the synchronous rotor frame d–q. Then the i_{sq} and i_{sd} are compared against the reference value respectively; the results are transferred to two PI controllers to produce the required voltage commands, namely u_{sq} and u_{sd}, which are transformed back to the two-phase stationary-frame α–β for controlling the SVPWM module. In field-oriented control, the torque current and exciting current are decoupled, so the independent torque and flux can be controlled in permanent magnet synchronous motor. In other words, SVPWM makes the phase current approach to the ideal sinusoidal wave, and the principle of SVPWM can be referred to [12, 13].

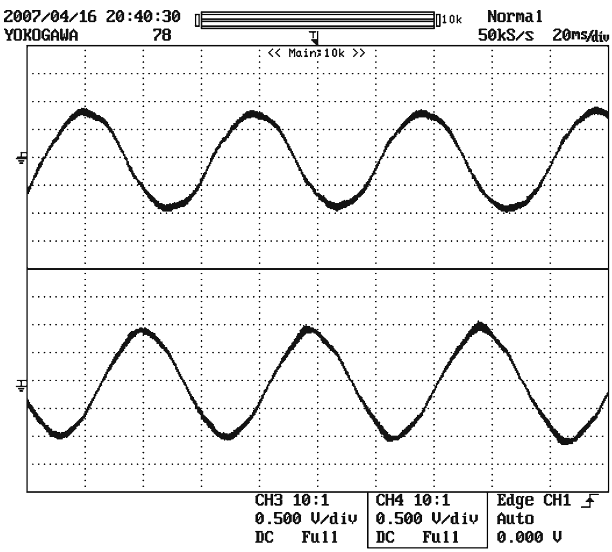

Fig. 7 Two phase current in SVPWM control mode

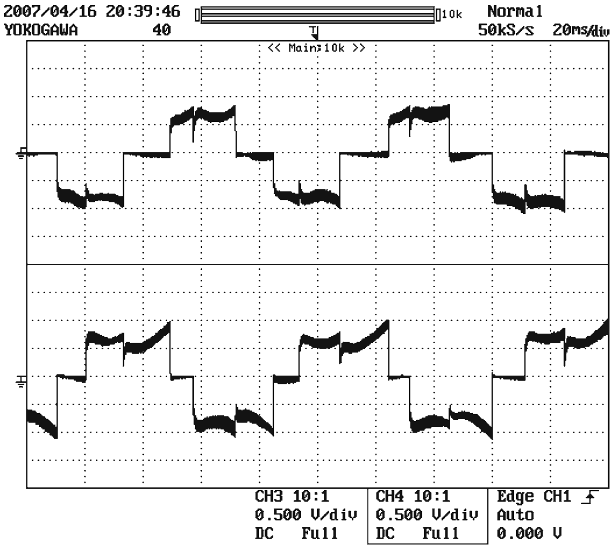

Fig. 8 Two phase current in Square wave drive mode

Figures 7 and 8 illustrate two-phase current i_a, i_b waves in SVPWM control mode and in square wave drive mode, respectively; they are implemented fully in the FPGA of the joint. The results demonstrate the phase current is regular sine wave in the SVPWM control mode, and testify that SVPWM can get more smooth torque than square wave drive.

4 Experiments

Experiments are performed on the 4-DOF arm. All four joints are identical in the macro structure, which are driven by brushless DC motor and harmonic drive gear (gear ratio 1:160). A potentiometer and a magnetic encoder are equipped to measure the absolute angular position of the joint and the motor, respectively. The motor phase current i_a and i_b are measured by two linear Hall sensors. The joint torque sensor is designed based on shear strain theory. Eight strain gauges are fixed crossly to the output shaft of the harmonic drive gear to construct two full bridges that measure the torques TA and TB.

In the first experiment, the third joint tracks a three-order curve (Fig. 9) with the previous impedance control methods (12) and (13). By adjusting the baud rate of M-LVDS serial bus and optimally choosing the impedance characteristics K_d and D_d, three kinds of cycle time are applied to the robot, as shown in Fig. 10. When the cycle time is 5 ms ($K_\mathrm{d} = 2170$, $D_\mathrm{d} = 2$) the static error of joint is 0.021° and the dynamic error is 0.03°. While the cycle time reduces to 1 ms (K_d=2100, D_d = 2, dashed line), the static and dynamic errors of joint are 0.01° and 0.02°, respectively. As the cycle time is 200us (K_d = 2000, D_d = 2, solid line), the static error of the joint is less than 0.007° and the dynamic error is 0.015°. Accordingly, when the transmitting rate goes faster, the robot system tracking accuracy is higher, and 200us cycle time is enough for our system.

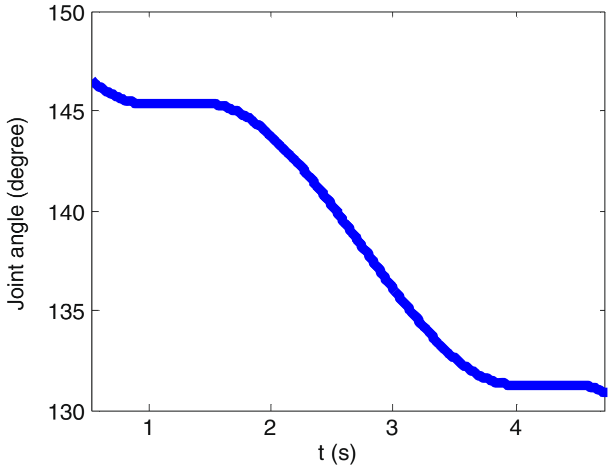

Fig. 9 The desired trajectory of three-order curve

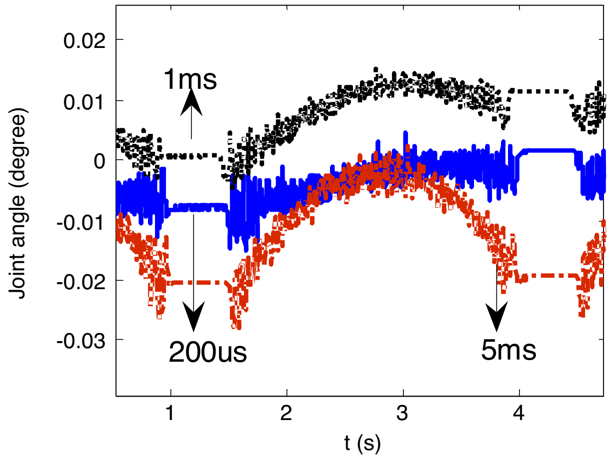

Fig. 10 The position tracking errors while the serial bus cycle time is changed

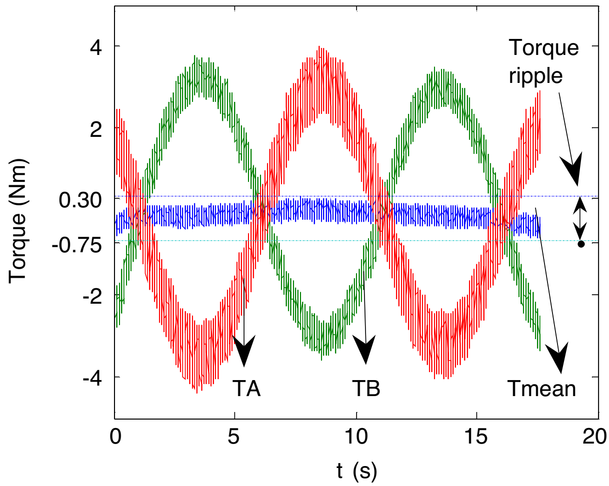

Fig. 11 Torque measured in the joint use of field-oriented control

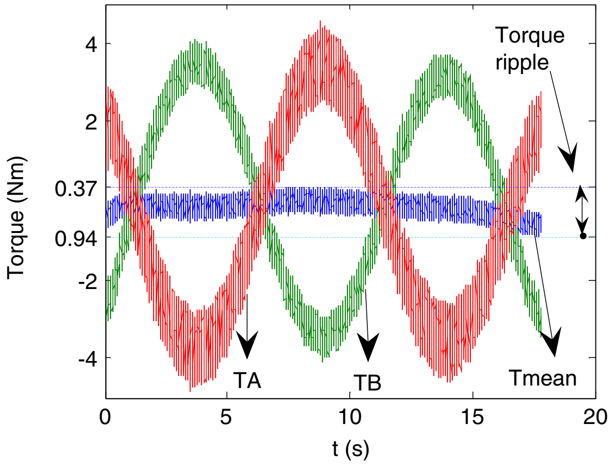

Fig. 12 Torque measured in the joint use of square wave drive

In the second experiment, a joint is driven in the free space without any force by the field- orient control and square wave drive. Figures 11 and 12 show the torques of joint measured during the two control modes above. Due to the inherent characteristic of Harmonic drive gear [14], TA and TB vary as sine wave and phase symmetrical. The medial line in Figs. 11 and 12 is the mean torque τ of the joint. Torque ripple has been reduced from 1.31 Nm in Fig. 12 to 1.05 Nm in Fig. 11. So, the torque ripple has been reduced evidently by use of the FOC.

In a further experiment, a single joint is used to demonstrate the impedance performance (K_d = 1,000, D_d = 10) while using Cartesian impedance control architecture. As shown in Fig. 13, the joint tracks the sine position trajectory (dashed line) and contacts a rigid environment where the joint angle under 1.85°, and real tracking curve is shown as the solid line. Corresponding to the desired trajectory tracking

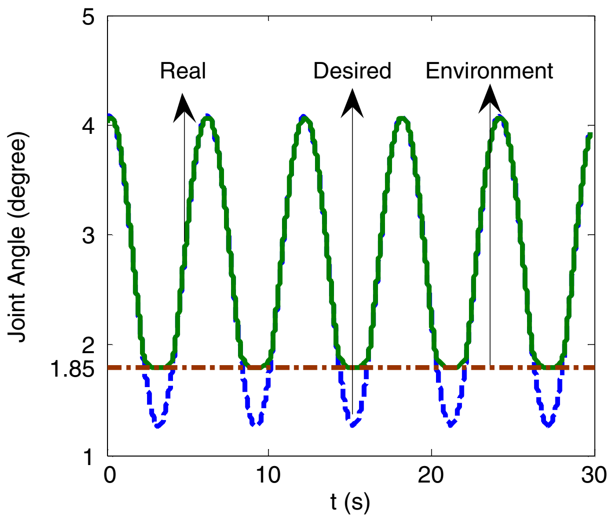

Fig. 13 Position tracking with environment contact by using impedance control

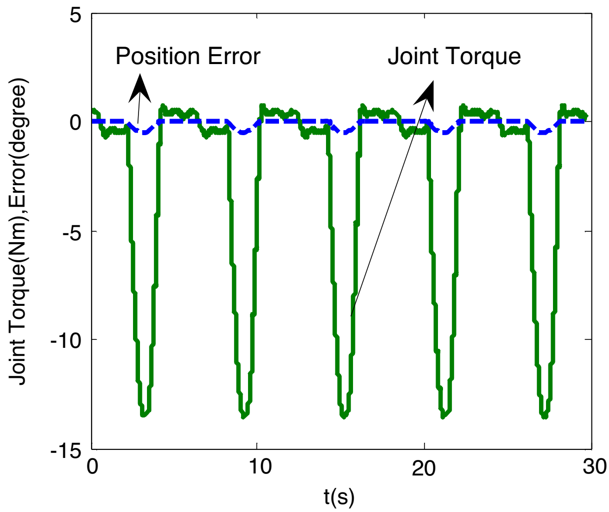

Fig. 14 The position-tracking error and the corresponding joint torque by using impedance control

curve in Fig. 13, the position-tracking error (dashed line) and filtered joint torque (solid line) is shown in Fig. 14. The experimental results show that the joint can follow the desired trajectory ideally in the free space, and the joint torque increases stably while it makes contact with the environment.

Finally, Cartesian impedance experiment has been made in 4-DOF robot. The robot at first stays put in the desired Cartesian origin $C_D = [0,0,0]$. Then, we pull the robot in a different direction as shown in Fig. 15. Finally, the robot will overcome the gravity and return to the C_D as soon as we release force. Figure 16 shows the corresponding Cartesian forces along with the Cartesian position. It can be concluded that the ideal Cartesian impedance behavior is successfully achieved, as theoretically predicted.

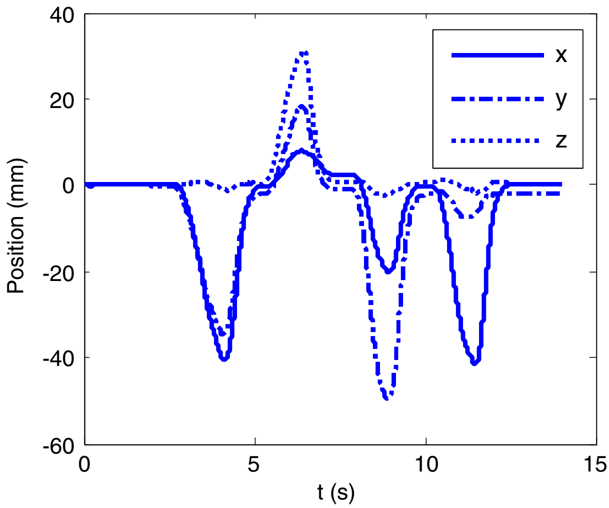

Fig. 15 Cartesian coordinate varies in Cartesian impedance control

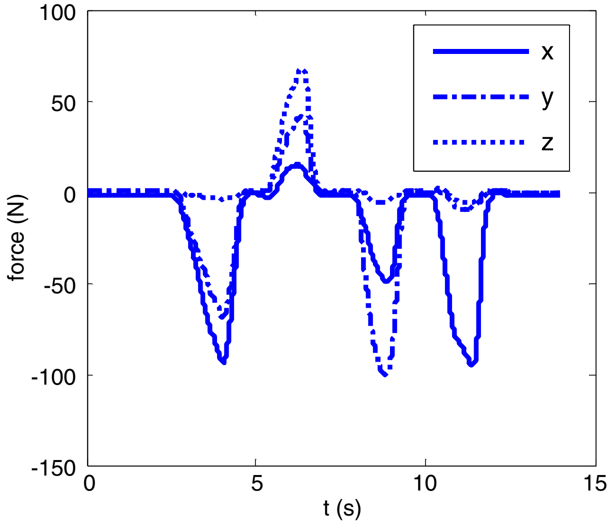

Fig. 16 Cartesian torque varies in Cartesian impedance control

5 Conclusion

The singular perturbation-based Cartesian impedance control plus gravity compensation for flexible joint manipulator was studied in this paper. To realize the control algorithm ideally, a new DSP/FPGA–FPGA system was designed. The joint motion controller with FOC control algorithm was verified and implemented on one-chip FPGA for high-performance servo drives of the permanent magnet synchronous motor. A PCI-based DSP/FPGA board was designed to realize the Cartesian level control, and M-LVDS serial data bus was designed with 200us cycle time to communicate between Cartesian level and joint level by using of the FPGA. Experimental results, obtained on a 4-DOF flexible robot, were verified that the DSP/FPGA-based control architecture have enhanced the robot impedance characteristics.

Acknowledgements This project is supported by the National High Technology Research and Development Program of China (grant no. 2006AA04Z228). The authors wish to thank Prof Jiang Li and Dr G. L. Xiong for their patient help in proofreading this paper.

References

1. Hogan, N.: Impedance control: an approach to manipulation: theory (part 1); implementation (part 2); applications (part 3). J. Dyn. Syst. Meas. Control **107**, 1–24 (1985)
2. Kazerooni, H., Sheridan, T.B., Houpt, P.K.: Robust compliant motion for manipulators: the fundamental concepts of compliant motion (part I); design method (part II). IEEE J. Robot. Autom. **RA-2**(2), 83–105 (1986)
3. Khatib, O.: A unified approach for motion and force control of robotic manipulators. The operational space formulation. IEEE J. Robot. Autom. **RA3**(1), 43–53 (1987)
4. Spong, M.: Adaptive control of flexible joint manipulators: comments on two papers. Automatica **31**(4), 585–590 (1995)
5. Ott, Ch., Albu-Schäffer, A., Kugi, A., Stramigioli, S., Hirzinger, G.: A passivity based Cartesian impedance controller – part I: torque feedback and gravity compensation. In: 2004 IEEE Int. Conf. Robot. Autom. **3**, 2659–2665 (2004)

6. Takahashi, T., Goetz, J.: Implementation of complete AC servo control in a low cost FPGA and subsequent ASSP conversion. Appl. Power Electron. Conference Exposition **1**, 565–570 (2004)
7. Tzou, Y.Y., Kuo, T.S.: Design and implementation of an FPGA based motor control IC for permanent magnet AC servo motors. In: 23rd International Conference on Industrial Electronics, Control and Instrumentation, vol. 2, pp. 943–947 (1997)
8. Paramasivam, S., Aumugam, R., Vijayan, S., etc: Ingenious digital controller for switched reluctance motor using Verilog (HDL). In: Conference on Convergent Technologies for Asia-Pacific Region, vol. 3, pp. 993–997 (2003)
9. Dubey, R., Agarwal, P., Vasantha, M.K.: FPGA based PMAC motor control for system-on-chip applications. In: Proceedings of First International Conference on Power Electronics Systems and Application, pp. 194–200 (2004)
10. Spong, M.: Modeling and control of elastic joint robots. J. Dyn. Syst. Meas. Control **109**(4), 310–319 (1987)
11. TI Corporation: TMS320C6711, TMS320C6711B, TMS320C6711C, Floating-point digital signal processors. The technology document of TI Company (1999)
12. Vas, P.: Vector Control of AC Machines [M]. New York: Oxford University Press (1990)
13. Zhou, Z., Li, T., Takahashi, T.: FPGA realization of a high-performance servo controller for PMSM. Anaheim: APEC2004, 1604–1609 (2004)
14. Taghirad, H.D., Belanger, P.R.: Intelligent built-in torque sensor for harmonic drive systems. IEEE Trans. Instrum. Meas. **48**(6), 1201–1207 (1999)