International Journal of Control and Automation
Vol. 10, No. 2 (2017), pp.161-172
http//dx.doi.org/10.14257/ijca.2017.10.2.14

A Study on High-Speed and High Precision Motion Control Application of NURBS Interpolation Based on FPGA

Xiaotong Yao[1], Yawen Li[1*] and Yu Gao[1]

[1.]*School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070, Gansu, China*
wydxxzj@163.com

Abstract

On the basis of studying the existing interpolation control algorithm in CNC technology in this paper, we compared the advantages and disadvantages of line, arc, Bezier curves, B-spline curves and NURBS (Non-Uniform Rational B-Spline) interpolation principle. Then focus on the NURBS interpolation algorithm study, and simulate this algorithm by using MATLAB Software. Quartus II development environment is used for the hardware interpolation system design and implement it with FPGA (Field Programmable Gate Array). For hardware description language design, we chose Modelsim-altera simulator to do the gate-level simulation. At last, we design PC control interface, and complete the whole interpolation system. After comparing the results of the final hardware debugging and software simulation, it confirms the correctness and practicability of the interpolation system.

Keywords*: NURBS interpolation, high-speed, motion control, FPGA, CNC (Computer Numerical Control)*

1. Introduction

In CNC machining, the contour of machined workpiece has various shapes. The traditional CNC interpolation system has simple function such as linear and circular interpolation, but in some special cases in the CNC machining, such as an airplane wing, streamlined car cover, mold cavity, and turbine blades, *etc*. The processing of many parts with complex profile shape, needing a new interpolation algorithm [1]. From an implementation point of view, interpolation technology is divided into software and hardware interpolation. As the implement of interpolation algorithm by using computer only, the calculation speed of interpolation points can be guaranteed, but because the limitation of data transmission rate between the upper computer and lower hardware, it is difficult to ensure the performance of real-time between calculation and lower PC interpolation, which make the actual interpolation speed limited. Due to the limitations of its functions traditional numerical control system during processing complex Workpiece, it requires a lot of lines or arcs to approximate the contour, and it is easy to generate velocity shocks while accelerating and decelerating frequently. When do this, it is not only need to write plenty of G-code, but also directly affect the quality of the machining workpiece [2].

NURBS (Non-Uniform Rational B-Spline) curve, because of its flexible design, algorithm stability, *etc*, has become one of the hot research issues of CNC interpolation algorithm [3]. Wang Tianmiao *et al* [4]. Proposed a real-time interpolation method based on de-Boor NURBS curve recursive algorithm, but it is lack of speed control elaboration.

* Corresponding Author： Yawen Li.
Email: wydxxzj@163.com.

ISSN: 2005-4297 IJCA

For NURBS curve interpolation, feed rate control is the guarantee of interpolation precision. Bedi *et al* [5]. Proposed an isoparametric method which uses a variable increment of constant parameter to realize NURBS interpolation, but this method cannot guarantee the consistency of each segment approximation error to original curve. Tsai *et al* [6]. studied the velocity planning method of feeding speed with the "forward" mechanism, but forward-looking algorithm is complex which increases the workload of real-time computing, so the performance of real-time decreased. In this paper, by using the limitation of chord error to achieve the adjustment of feed rate on the basis of de-boor NURBS curve interpolation algorithm [7]. The simulation proved the validity and correctness of this motion control system design.

2. Study and Comparison of CNC Major Interpolation Algorithm

Bezier curve is a smooth curve which based on four coordinates in any position. According to known curve parametric equations to determine the four points, the people who initially studied Bezier curve using the design ideas to get this vector curve drawing method. First of all in 1962, the French mathematicians Pierre Bézier study this method of vector drawing curves, and gave a detailed calculation formula [8].

In 1074, Gordon and Riesenfeld using B-spline basis function instead of Bernstein basis function construct the B-spline curve. B-spline curves are divided into segments, and each segment of the parameter t interval is [0, 1]. This method overcomes the Bezier curve shortcomings that change any control point of Bezier curve. All points on the curve are transformed [9].

NURBS is the abbreviation of Non-Uniform Rational B-Spline, NURBS was proposed by Versprille in his doctoral research. In 1992, the International Organization for Standardization added NURBS into international standard PHIGS (Programmer's Hierarchical Interactive Graphics system) as the PHIGS Plus expansion section. Bezier, rational Bezier, B-spline uniform and non-uniform B-spline are unified into NURBS.

Each control point of Bezier curve can affect the whole shape of the curve, but B-spline control point only affects a part of the curve. B-spline overcomes some disadvantages of Bezier curve. In other words, each control point of B-spline curve affects only a part of the curve parameters, so as to do partial modification. B-spline provides more flexibility apparently. Bezier and B-spline curves are polynomial curves, which cannot express some of basic curve such as arc, so we introduce the NURBS curve to solve this problem [10].

In summary, Bezier curve is a special case of B-spline, and B-spline is a special case of NURBS. Comparing with the traditional interpolation methods using large number of small line to approach the curve, NURBS curve interpolation has many obvious advantages such as small data storage, information integrity and speed smoothing, *etc*. It is easy to achieve high-speed and high-precision interpolation, and supported by a growing number of CNC systems.

3. Design Proposal and Selection

By using MCU (Micro Controller Unit) to realize motion control interpolation, as its own architecture, memory and speed performance, and its accuracy and scalability are not well. It can only put in some situations which require the simple control track. This scheme cannot meet the modern CNC motion control requirements [11]. If using ARM (Acorn RISC Machine) processors to realize complex interpolation algorithm, the performance - to - price ratio is not high to achieve in interpolation system due to the limitations of their speed and parallel processing capabilities. ARM is not suitable for the cybernetics core of interpolation motion control system.

DSP (Digital Signal Processor) is proper tool for digital processing. It is most suitable for processing algorithm but interpolation control algorithm require more processing

speed, and has to obtain the interpolation points as soon as possible, rather than the complexity of an algorithm. Compared to FPGA, not only the reconfigurable characteristics of internal resources, but also the parallel processing capacity and the processing speed of interpolation algorithm, DSP reflects a lack of their own. In this paper, after the comparison of different properties of processor, we chose Cyclone IV FPGA (chip: EP4CE22F17C8) by Altera Corporation.

4. Design of Hardware System

In this paper, the system consists of two main components as shown in Figure.1: upper PC-controlled module and lower FPGA hardware interpolation module. The system implements the NURBS curve interpolation function and speed control in interpolation processing, taking into account of the situation in the interpolation process like stop emergency processor, rotary encoder detect the motor position information in current, zero switches, and limit switches to restrict the motor. The system also comes with a 6-channel pulse output individually, and 6-channel PWM output function.

The main function of PC-controlled module is to send control instruction to FPGA and make adjustments according to the current operation based on the feedback of FPGA (including motor position, speed and a variety of unexpected situations). Another feature of upper computer is pretreatment for NURBS curve interpolation, and conveys the middle parameters to FPGA by USB2.0 interface. The main function of lower FPGA hardware interpolation module is to complete the NURBS interpolation algorithm in hardware, and send the information of the current interpolation process status back to the upper computer to ensure the successful completion of the whole interpolation process.

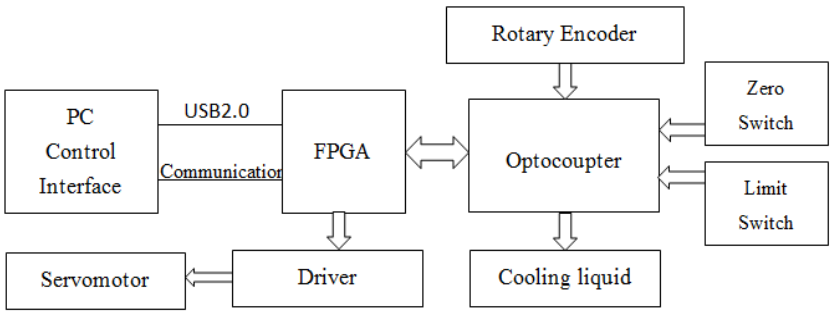


Figure 1. The Construction of Interpolation System

5. NURBS Interpolation Theory and Algorithm Simulation

The interpolation system adopts an algorithm [12] of adaptive interpolation algorithm based on de Boor NURBS curve. A K times NURBS curve can have a rational polynomial piecewise function expression:

$$P(\mathrm{u}) = \frac{\sum_{i=0}^{n} \omega_i d_i N_{i,k}(u)}{\sum_{i=0}^{n} \omega_i N_{i,k}(u)} \tag{1}$$

In this equation, ωi(i=0,1,…,n) is called weighter factor, and continuous k weighter factor are not zero simultaneous, di (i = 0,1, ..., n) are the control vertices, Ni, k (u) are the basis function of standard k times B-spline determined by knot vector U = [u0, u1, ..., un + k + 1].

In practical applications the algorithm uses cubic polynomial. In this paper, we calculate it by using k=3, so Equation (1) can be written as:

$$P(u) = \frac{\sum_{i=0}^{n} d_i \omega_i N_{i,3}(u)}{\sum_{i=0}^{n} \omega_i N_{i,3}(u)} \tag{2}$$

Let $t=(u-u_{i+k})/(u_{i+k+1}-u_{i+k})=(u-u_{i+k})/\Delta_{i+k}$, then:

$$P_i(t) = \frac{[1\ t\ t^2\ t^3]M_i \begin{bmatrix} \omega_i d_i \\ \omega_{i+1} d_{i+1} \\ \omega_{i+2} d_{i+2} \\ \omega_{i+3} d_{i+3} \end{bmatrix}}{[1\ t\ t^2\ t^3]M_i \begin{bmatrix} \omega_i \\ \omega_{i+1} \\ \omega_{i+2} \\ \omega_{i+3} \end{bmatrix}} \tag{3}$$

For the expressions of M_i:

$$M_i = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \tag{4}$$

In this system, following are some pretreatment parameters that used in NURBS interpolation algorithm:

```
m(4, 4) = 0;m(3, 4) = 0;m(2, 4) = 0;
m(1, 1) = -(U(i+1) - U(i)).^2. / (U(i+1) - U(i-1)). / (U(i+1) - U(i-2));
m(1, 4) = (U(i+1) - U(i)).^2. / (U(i+2) - U(i)). / (U(i+3) - U(i));
m(2, 3) = 3.*(U(i+1) - U(i)).^2. / (U(i+1) - U(i-1)). / (U(i+2) - U(i-1));
m(3, 3) = 3.*(U(i+1) - U(i)).*(U(i) - U(i-1)). / (U(i+1) - U(i-1)). / (U(i+2) - U(i-1));
m(4, 3) = (U(i) - U(i-1)).^2. / (U(i+1) - U(i-1)). / (U(i+2) - U(i-1));
m(1, 3) = -m(2, 3). / 3 - m(1, 4) - (U(i+1) - U(i)).^2. / (U(i+2) - U(i)). / (U(i+2) - U(i-1));
m(1, 2) = -m(1, 1) - m(1, 3) - m(1, 4);
m(2, 1) = -3.*m(1, 1);
m(2, 2) = 3.*m(1, 1) - m(2, 3);
m(3, 1) = 3.*m(1, 1);
m(3, 2) = -3.*m(1, 1) - m(3, 3);
m(4, 1) = -m(1, 1);
m(4, 2) = 1 + m(1, 1) - m(4, 3);
temp1_x = M*[W(i-3)*X(i-3);W(i-2)*X(i-2);W(i-1)*X(i-1);W(i)*X(i)];
temp1_y = M*[W(i-3)*Y(i-3);W(i-2)*Y(i-2);W(i-1)*Y(i-1);W(i)*Y(i)];
temp1_z = M*[W(i-3)*Z(i-3);W(i-2)*Z(i-2);W(i-1)*Z(i-1);W(i)*Z(i)];
```

In this paper, the interpolation algorithm was implemented by using NURBS interpolation algorithm, and specific interpolation process is done in FPGA. The upper computer calculates the parameters of temp1_x, temp1_y, and temp1_z *etc*. based on the input of weighter factor, control vertex and knot vector. Then send these parameters to FPGA via the USB interface. After receiving these parameters, FPGA begins interpolation for NURBS curves until completion.

To verify the correctness of such a pre-treatment process, we used MATLAB for the simulation of this process. In order to make easier measurement and analysis of the actual hardware interpolation, set all Z axis inputs to 0. Weighting factor inputs are [1 1 1 1 1 1], the knot vectors are [0 0 0 0 0.3 0.7 1 1 1 1], the control vertex are: x = [0 10 10 20 20 30]; y = [0 0 10 10 0 0], and z = [0 0 0 0 0 0]. The simulation results are shown in Figure 2.

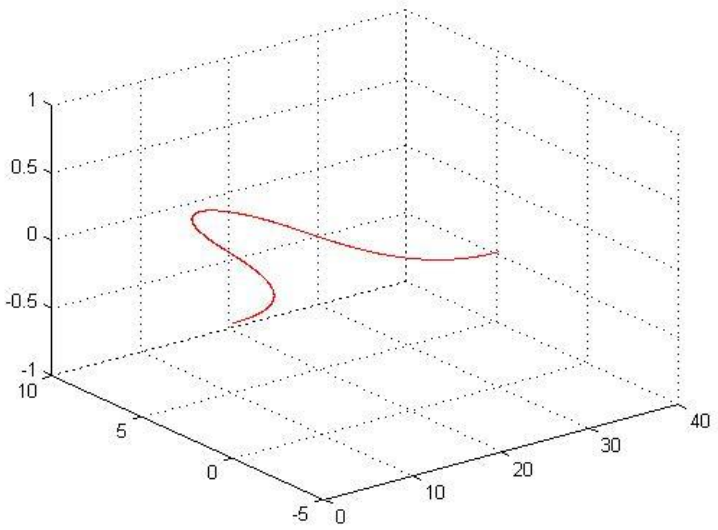


Figure 2. MATLAB Simulation

In the case to meet interpolation precision 1um, the interpolation completed a total number of 1001 times, the output results are shown in Table 1:

Table 1. The Interpolation Points of NURBS Curve

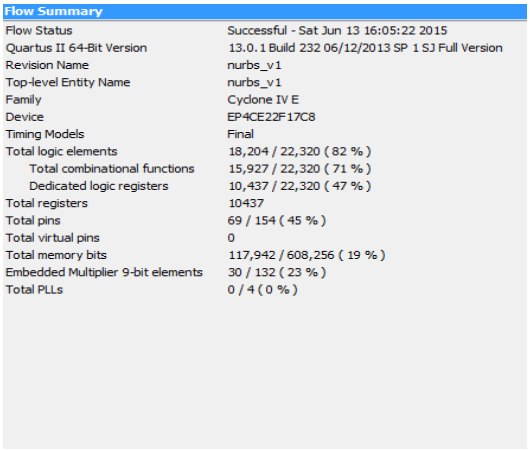
Step	Increment	x	y	z
cnt=1	0	0	0	0
cnt=2	0.001	0.09667	0.04261	0
cnt=3	0.002	0.19867	0.05696	0
cnt=4	0.003	0.29701	0.01279	0
cnt=5	0.004	0.39469	0.02271	0
cnt=6	0.005	0.49172	0.03543	0
cnt=7	0.006	0.58809	0.05093	0
cnt=8	0.007	0.68381	0.06922	0
cnt=9	0.008	0.77888	0.09026	0
…	…	…	…	…
cnt=1001	1	30.00000	0.00000	0

6. Design in FPGA and Hardware Debugging

In this paper, hardware interpolation was completed by using Altera's FPGA chip EP4CE22F17C8. The entire design consist of: system modules (pulse output module, read axis status module, speed control module, an individual pulse output module, and 6-channel PWM output module, *etc*.), zero switches, limit switches module, logical location module, rotary encoder modules, and NURBS interpolation module.

6.1. NURBS Interpolation Design in FPGA

This design was completed in QuartusII (version: 13.0) development environment, and implement the whole NURBS interpolation algorithm in Cyclone IV FPGA (chip: EP4CE22F17C8). This chip has 22320 Les (logic Elements), and 154 I/Os. From the compiler report in Figure 3, we found that the design uses a total number of 18,204 Les, and chip resource utilization up to 82%. Whether from a design point of view, or performance price ratio, both meet the requirements, and this design is compatible For example, if we need to add more control function and existing Les resources are not sufficient then we can directly replace a logic chip which has more resources.



Flow Summary	
Flow Status	Successful - Sat Jun 13 16:05:22 2015
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	nurbs_v1
Top-level Entity Name	nurbs_v1
Family	Cyclone IV E
Device	EP4CE22F17C8
Timing Models	Final
Total logic elements	18,204 / 22,320 (82 %)
Total combinational functions	15,927 / 22,320 (71 %)
Dedicated logic registers	10,437 / 22,320 (47 %)
Total registers	10437
Total pins	69 / 154 (45 %)
Total virtual pins	0
Total memory bits	117,942 / 608,256 (19 %)
Embedded Multiplier 9-bit elements	30 / 132 (23 %)
Total PLLs	0 / 4 (0 %)

Figure 3. Compiler Report

The Figure 4 shows the RTL view of NURBS curve interpolation algorithm implemented in FPGA. The whole circuit layout system and the realized function can be seen from the figure. In the figure, Pulse_mode_i is individual pulse output mode signal which use in some individually motor control situation through Set_single_num signal to output the number of pulses required. Set_single_speed is of the motor speed control signal at this time. Axis_state [5: 0] is used for 6-axis NURBS interpolation state. Nurbs_seg signal is the number of NURBS curve segments, which is 8-bit integer. The formula for Nurbs_seg signal: the number of NURBS curve segment= the number of control vertices-3. Nurbs_step_u is the step of interpolation signal that is related with the accuracy of NURBS interpolation. It is restricted by chord error, reduce its value can improve interpolation precision, but increase the amount of calculation at the same time, can reduce the interpolation speed. Nurbs_fifo_data is the NURBS data signal which followed by the address written in NURBS interpolation, and parameters were issued by the host computer. When these parameters send completely, FPGA send (nurbs_inte_req) NURBS interpolation request signal activate (set_cb_nurbs_ack)acknowledge signal to start interpolation, then follow the speed control signal (nurbs_speed) to complete interpolation process, and then wait for the answer signal (nurbs_inte_ack) to display NURBS interpolation is completed.

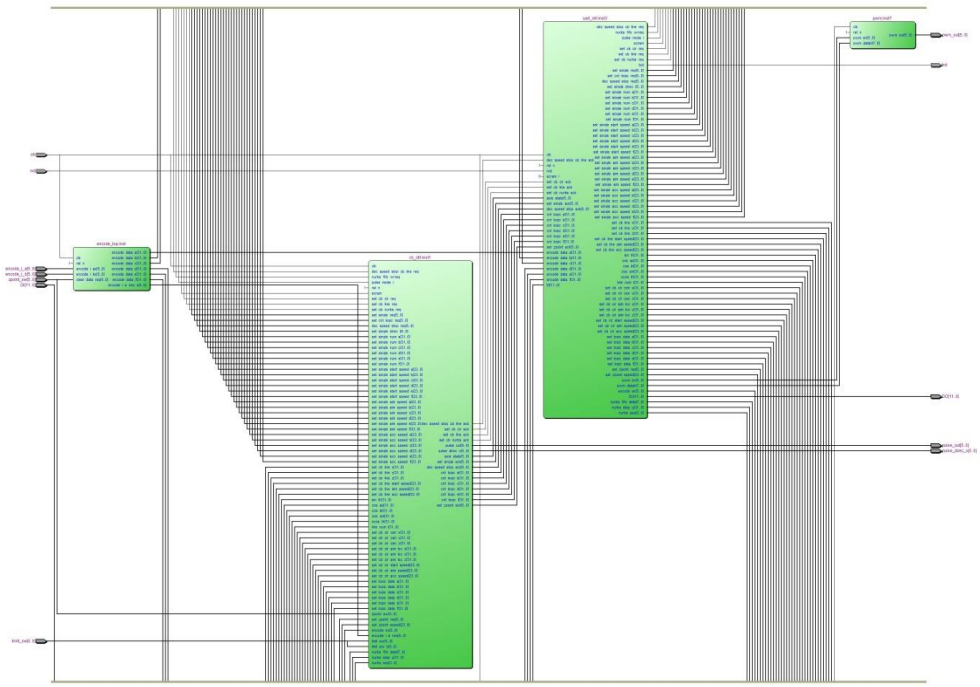


Figure 4. RTL Viewer in Quartus II

6.2. Simulation in Modelsim-altera Simulator

In order to test the validity of above design, we have prepared a simulation Test bench in Modelsim-altera simulation software. In accordance with the timing requirements of the input signal, let the input signal in Table 2 as follow: clk input is 50MHz, set nurbs_fifo_wrreq (require signal of NURBS written in FIFO) signal is high, manually write the data (calculated data by upper computer as shown in table 2) into FIFO, then set nurbs_inte_req (NURBS interpolation request signal) to high, and start NURBS interpolation simulation.

Table 2. Middle Parameter Calculated by PC

PC Parameter name	output data	PC Parameter name	output data
Men[0]	8'h00	Men[205]	8'h00
Men[1]	8'h00	Men[206]	8'h00
Men[2]	8'h3F	Men[207]	8'h00
Men[3]	8'h80	Men[208]	8'h00
Men[4]	8'h00	Men[209]	8'h00
Men[5]	8'h00	Men[210]	8'h00
Men[6]	8'h41	Men[211]	8'h00
Men[7]	8'h34	Men[212]	8'h3F
Men[8]	8'h93	Men[213]	8'h80
Men[9]	8'h75	Men[214]	8'h00
Men[10]	8'Hc1	Men[215]	8'h00
…	…	Men[216]	8'h00

In the Figure 5, three signals result_int_x, result_int_y, and result_int_z are interpolation results, as can be seen from the simulation figure, X, Y, Z axis outputs can perfectly and correctly complete the interpolation process.

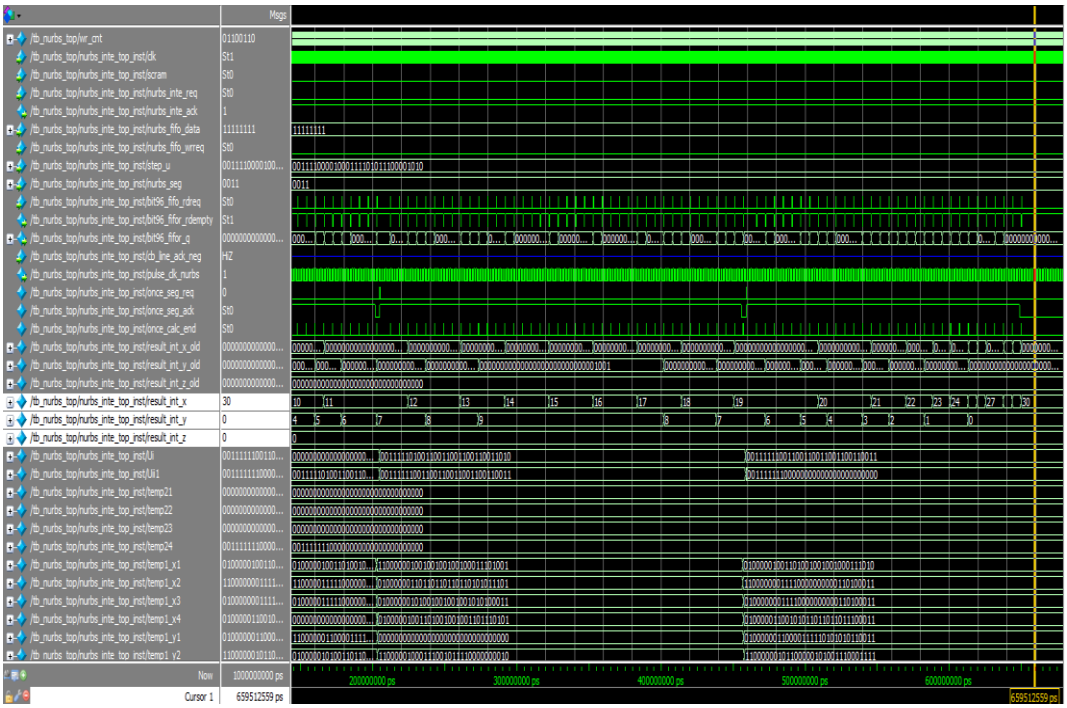


Figure 5. Modelsim Simulation

6.3. Hardware Design and Testing

According to the requirements of NURBS interpolation algorithm, we design the motion controller hardware circuit completely. As can be seen from Figure 6, J9 is the interface section of inputs such as zero switches, limit switches and encoder signals. We isolate the external inputs signal and the FPGA chip by optocoupler. We gave power supply to optocoupler by using B0505S-1W power module to protect FPGA chip from being damaged due to external wrong connection or high current. J10 interface is the output section, we added 74HCT14 chip in this output port, which has a pulse wave shaping features like rectangular pulse, distortion in the transmission, and rising and falling situation. Using this chip we can get more accurate output pulses. J1 is USB2.0 input interface which has two functions, one is the power supply for whole board and second one is the data communication between upper PC and FPGA. J6 is the JTAG interface which helps to download program into FPGA. J8 interface are FPGA I / Os that can be used for large number of signals. The board hardware oscillator is 50MHz. After testing, the motion control system will be stable and operate accurately.

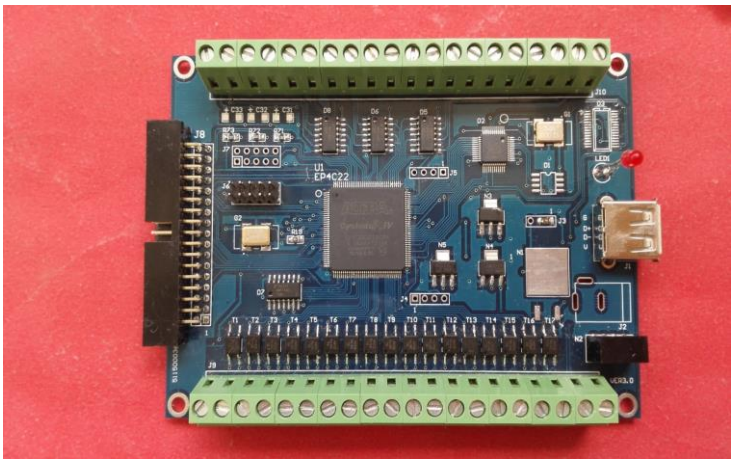


Figure 6. Hardware Design

7. Control Interface Design

In order to normal operation of the whole system, we have designed a user interface for NURBS interpolation algorithm control as shown in Figure 7. We connect the hardware board with PC by using USB2.0 interface. First we click the key of OpenDevice and then click ZeroPoint to make sure that the motor is in initial state. In NURBS interpolation module: sequentially input weighter factor, knot vector, control vertex, and the relationship between these parameters is as follows: when the number of input control vertex is n, the number of knot vector = n + k + 1; K = 3 (As formula (2)), the number of weighter factor = the number of control vertex. In the figure, each time you enter a point and one time click input button until all inputs are finished. When all setup is complete then click the Start button to start the interpolation preprocessing by using upper computer. The USB2.0 interface is used to send intermediate processing parameters to FPGA and then NURBS interpolation process begin. During interpolation, FPGA feedback timely six axis logical position, the switch signal and the axis status signal to the upper computer, so that the user can make appropriate action on time.

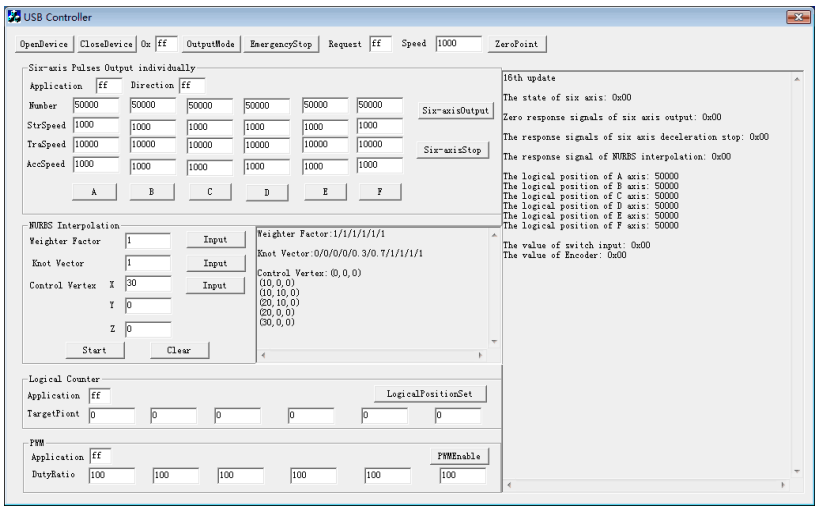


Figure 7. PC Interface

8. Conclusions

Based on the study, the advantages and disadvantages of existing motion control CNC interpolation algorithm, and an adaptive interpolation algorithm for NURBS interpolation speed control algorithm was finally determined. This algorithm solves the problems of speed and accuracy of the system and verifies the feasibility of the algorithm by MATLAB simulation and physical experiments.

The system realizes NURBS interpolation algorithm in FPGA hardware. This design has the advantage of both timeliness and accuracy. After many experiments, the system has the high interpolation speed, high precision controlled, interpolation acceleration and deceleration, and smooth transition performance. From the user point of view, the algorithm does not need to line approximate curve, simplifying the job of writing the amount for G-code, and easy to implement.

Acknowledgements

This work is supported by science and technology of Lanzhou, Gansu province, China (Grant no. 213188).

References

[1] J. Tong, H. Min and Y. Chen, "Design and Implementation of NURBS Interpolation with FPGA Device", Machine Tool & Hydraulics, vol.41 no.1, **(2013)**, pp.111-113.
[2] M. Nie, X. Jiang and G. Li, "Embedded platform of NURBS interpolation based on ARM-FPGA", Computer Engineering and Applications, vol.6, no.50, **(2014)**, pp.261-264.
[3] G. Huang, P. Zhang and S. Zhou, "Research on Real-time NURBS Curve Interpolation System Based on DSP", Machine Tool & Hydraulics, vol.41 no.23, **(2013)**, pp.41-44.
[4] T. Wang, Y. Cao, Y. Chen and H. Wei, "NURBS Interpolation and Feedrate Adaptive Control Based on de Boor Algorithm", China Mechanical Engineering, vol.18, no.21, **(2007)**, pp.2608-2613.
[5] S. Bedi, I. Ali and N. Quan, "Advanced interpolation techniques for N.C.machines", Journal of Engineering for Industry, no. 115, **(1993)**, pp.329-336.
[6] M. S. Tsai, H. W. Nien and H. T. Yau, "Debelopment of integrated acceleration/deceleration look-ahead nterpolation technique for multi-blocks NURBS curves", Int J Adv Manuf Technol, vol.56, **(2011)**, pp.601-618.
[7] Y.-G. Zhao, W. Li, F. Guo, Y.-F. Li and C.-X. Liu, "The Random Degree NURBS Curve Interpolation and Simulation Based on Cox-de Boor Algorithm", Modular Machine Tool & Automatic Manufacturing Technique, no.6, **(2012)**, pp.45-48.
[8] Min Ren, "The Bezier Curve Expression Algorithm Research", Modern Machinery, no.1, **(2007)**, pp.65-76.
[9] W.-H. Chen and T. Zhang, "The study of cubic uniform rational B-spline interpolation algorithm", Machinery Design & Manufacture, no.8, **(2010)**, pp.3-5.
[10] J. Zhao, L. Li and G. Wang, "Research of NURBS Curve Real-time Interpolation Feed Speed Planning", Tool Engineering, vol.49, no.3, **(2015)**, pp.7-11.
[11] W. Luo, "The Research of Industrial Robot Motion Controller Based On DSP and FPGA", Guangzhou, South China University of Technology, **(2011)**, pp.11-13.
[12] P. Wu, "The Research on NURBS Curve Interpolation Algorithm Of 6-DOF Industrial Robot", Zhejiang, Zhejiang University of Technology, **(2013)**, pp.40-46.

Authors

Xiaotong Yao, he was born in 1976, Gansu province, China. He is a PhD candidate in Lanzhou Jiaotong University, Lanzhou, China .His interested areas are intelligent application development, automated remote control application development, embedded systems, computer control, networking and application development, *etc*.

Yawen Li, he was born in 1988, Shanxi province, China. He is currently a postgraduate student at School of Electronic and Information Engineering, Lanzhou Jiaotong University. His main research areas include application of the robot, CNC interpolation controller, FPGA applications and instrumentation.

Yu Gao, he was born in 1990, Henan province, China. He is currently a postgraduate student at School of Electronic and Information Engineering, Lanzhou Jiaotong University. His main research areas include computer applications and control automation.