The Research On the Control System of the Minimally Invasive Surgical Robot

Da Liu [α] LiNan Cong [α] Jingmeng Liu [β] Dong Xu [α]

[α] Robotics Institute
Beijing University of Aeronautics and Astronautics
Beijing, China
Andy@me.buaa.edu.cn

[β] School of Automation
Beijing University of Aeronautics and Astronautics
Beijing, China
Ljmbuaa110@163.com

***Abstract*—This paper presents a robotics control system of the parallel serial robot for the minimally invasive surgery. Respectively, for the serial part, which is a 6 degree of freedom mechanical arm, this system presents a control unit based on the Field Programmable Gate Array and a transmission unit based on the Universal Serial Bus; for the parallel part, which is a 4 degree of freedom structure, this paper discusses a method based on the PID parameters debugging of the Programmable Multi Axes Controller. This control system with the parallel serial robot is the newest member of a growing family of minimally invasive surgical robots under development.**

***Keywords*—Control System, FPGA, USB, PMAC, PID, Minimally Invasive Surgical Robot**

I. Introduction

Using robot to operate the minimal invasive surgery is presently performed in the operation room. Especially, when the surgical environment is limited and radiant, such as in a CT gantry, the working procedure of the robot becomes more challenging: it requires the extensive precision due to both the lack of doctor's nearby operation and the very limited surgical environment.

To overcome this problem, several researchers investigated the control method of the robotics systems to assist in needle precision placement. Potamianos and Davies proposed a stereo-pair of two x-ray views registered to a common fiducially system with a five degree of freedom (DOF) passive linkage equipped with position encoders to position a passive needle guide[1,2]. Bzostek et al. used an active robot (LARS [3]) for similar purposes[4]. These systems successfully addressed difficult issues of needle precision placement. In their present state of development, however, these robotics systems are expensive and their size makes them cumbersome for routine use in the limited operating environment.

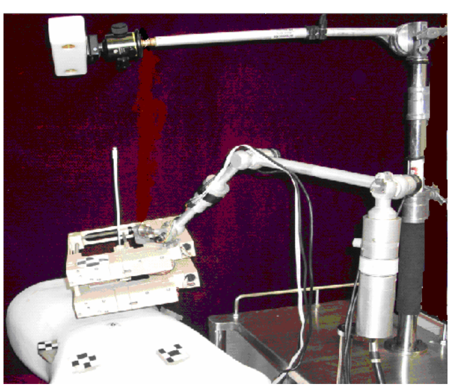

Figure 1. Parallel serial minimally invasive surgical robot

In contrast, our group recently reported the development of a simple minimally invasive surgical robot, iSurgery (Figure 1), including a serial mechanical arm and a parallel structure. To ensure its working precision, we designed a control system, which based on Field Programmable Gate Array (FPGA), Universal Serial Bus (USB) and Programmable Multi Axes Controller (PMAC). iSurgery's serial mechanical arm was used to orientation and lockup, so that it could hold the ending parallel structure. The parallel structure was designed to lead the surgical needle to its accurate access. Considering the usage of the two parts, we installed one encoder on each joint of the mechanical arm to calculate its position and pose, used one DC motor to perform movement on each DOF of the parallel structure.

In the attempt to get the feedback information of the encoders exactly and transform to the surgical control centre effectively, we made the FPGA based calculation module and the USB based transformation module. For the parallel structure, we use the PMAC and adjust its PID parameter to control the movement of the DC motors with veracity and reliability. A general presentation of this control system and adjusting method are outlined next.

II. Joint Information Collection System of the 6 DOF Serial Mechanical Arm

A. Hardware Architecture

The 6 DOF Serial Mechanical Arm is used to orientation and lockup, so that it could hold the ending parallel structure. In the attempt to get the feedback information of the encoders exactly and transform to the surgical control centre effectively, we made the FPGA based calculation module and the USB based transformation module.

So that we can calculate the exact position and pose of the arm according to the feedback information of the encoders and transform the information to the surgical control centre effectively through the USB.

B. The Analysis of the Encoder 4 Times Frequency

Usually, there are two kinds of encoders, one is the absolute encoder, the other is the incremental encoder. The output of the absolute encoder is the digital angle according to the rotary axis position. This kind of encoder does not need the 4 times frequency. Differently, the incremental encoder gives out a series of pulse according to the rotary angle of the axis, and then, sums up the total pulse through the counting circuit to get the relative angular displacement. Because of one absolute encoder can only test 0～360° rotary angle, we have to use more than one absolute encoders if we want to test more than 360° rotary angle. In order to make the whole system simple, we chose the incremental encoder.

We used the 4 times frequency method to increase the precision of the encoders. Ordinarily, the 4 times frequency circuit and the orientation circuit could be combined; it is called the 4 times frequency and orientation circuit. In our practice, however, we found that due to some of the 4 times frequency circuit were lack of precision, the capability of the system declined instead of increasing. To solve this problem, we made a reliable and accurate 4 times frequency and orientation circuit to get the encoders feedback information exactly.

The typical outputs of the encoders were two square wave signals A and B whose phase difference was 90° and zero pulse Z (Figure 2).

The number of the A and B pulse defined the rotary angle and the phase relationship of A and B defined the rotary

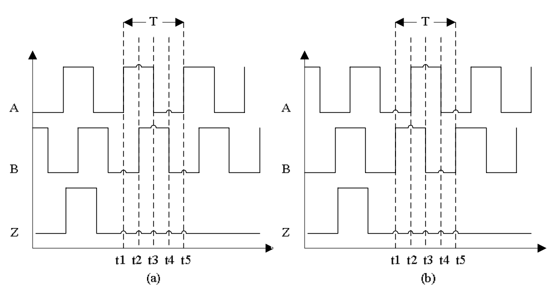

Figure 2. The output of the incremental encoder

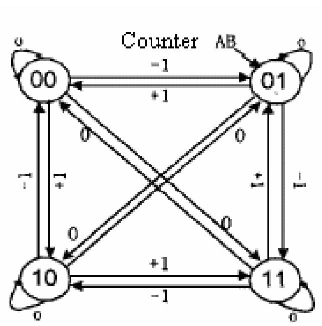

Figure 3. FPGA state turning diagram

orientation. When A phase is 90° in front of B, it means the encoder is turning positively (Figure 2a), otherwise, the encoder is rollback (Figure 2b). If we define the pulse cycle of the encoder as T and the corresponding angle displacement as θ, so the quantization error was θ/2. If we can make a 4 times frequency of A and B pulse, the pulse cycle would decline to T/4, the quantization error would decline to θ/8, which could increase the encoder rotary precision 4 times. Because of the encoder rotary speed was unpredictable, the pulse cycle T was uncertain and we couldn't use regular PLL frequency method. Observing Figure 2, we could found that A and B signal changed 4 times such as the rising edge on t1, t2 and the falling edge on t3, t4. In spite of T is uncertain, since the phase relationship of pulse A and B is determinate, the 4 times changing are uniformly distributing on the phase. If we could use the 4 times changing to make 4 times frequency signal, we would increase the precision of the encoders.

The 4 times frequency signal can't be translated into comparative position until get through the counting module. There are 2 kinds of method for counting: one is implemented by the inner counter of the micro controller, the other is by reversible counter. If the encoders are fewer, the former method costs fewer components and has simple structure, it makes sense. While the encoders are more, we choose the latter, using Complex Programmable Logic Device (CPLD) would make our design simpler. According to the above ideas, we used the FPGA to design the encoder feedback information collection interface circuit on one chip, including the 4 times frequency circuit, the orientation circuit and the reversible counter circuit.

Firstly, we analyze the encoder signal as following:

1) *When the encoder is turning positively, there are 4 times changing of A and B signals:* 00->10->11->01 (Figure 2a). If the reversible counter pluses 1 once it changes, the counter would plus 4 times in one cycle, so we can make the 4 times frequency when turning positively.

2) *When the encoder is rolling back, there are 4 times changing of A and B signals as well:* 00->01->11->10 (Figure 2b). So we can also make the 4 times frequency when rolling back as above.

3) *When the circuit is under trouble or disturbing:* it will appear other state changing, but the counter won't work.

According to the analysis above, we could get the FPGA module state timing diagram (Figure 4).

We can found not only the design procedure, but the circuit structure becomes much simpler if we use FPGA as the encoder feedback information collection system. By the way,

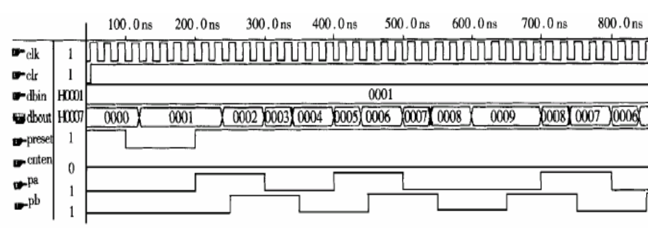

Figure 4. FPGA module state timing diagram

we should pay attention to that the timing cycle of FPGA should shorter than one quarter of the encoder pulse, usually, the timing cycle of FPGA is much less than the encoder pulse cycle.

C. USB Based Transformation Module

The encoder feedback information should be transformed to the surgical control centre for the surgery planning. This function was performed by a 51 serial micro controller, however, the serial port of the 51 controller could only worked under the speed of 19.2Kbps, which can't satisfy the high-speed environment, so the speed became a bottle-neck in the serial port transformation of 51 controller.

The USB interface deals quite well with the questions above. Data can be transformed under the speed of 1.5Mb/s (low speed), 12Mb/s (complete speed) and as high as 480Mb/s (under USB2.0 criterion) through USB. Except for the data transformed of BUS state, control, error testing and etc., USB could transform under the speed of 1.2Mb/s, 9.6Mb/s and even more. Thus, if we introduce USB to transformation module of the 51 micro controllers, the transform speed will be increased consumedly.

In this paper, we choose the ISP1581 serial USB chip, which is made by PHILIPS in the year 2002, based on the USB2.0 protocol. Except for the features other chips have, there are some characteristics of ISP1581 such as soft connection, low frequency crystal, much architecture of DMA module and etc. ISP1581 can connect different kinds of DSP, ASIC and other micro controller to complete the high speed transformation.

During the design of hardware, we use the 8051 micro controller to initialize, enumerate and even transform to the ISP1581. There are two modes of configure of ISP1581, we can choose the mode by setting the BUS_CONF pin. In this system, we choose the general processor mode (BUS_CONF=1). In this mode, the chip has absolute 8 bits address bus, 16 bits data bus, which can connect different kinds of micro controller. Besides, it also has the reading and writing control choices. The maximum speed of this mode is 225Mb/s. The connecting circuit is illustrated in the Figure 5.

The firmware of ISP1581 can be categorized to three kinds: main program, interrupt service and request management. The main program is an infinite loop to test some flags such as request management, the parameter to fill in and others. Once the flags become validate, corresponding sub program will be transferred. The hard interrupt should be dealt with by the interrupt service program, at the same time the program will set or clear the different flags according to the different interrupt. The request management program deals with the standard request during the period of enumerating and the received manufacturer request during the period of usually working.

The work to be done in the main program is illustrated as follows: initialize the micro controller, initialize and configure the ISP1581, begin the testing loop for the flags. Figure 6 is the flow chart of the main program. According to the different micro controller and the hardware design, the difference of the main program firmware lies in the method of register address mapping and visiting, the method of interrupt engendering and

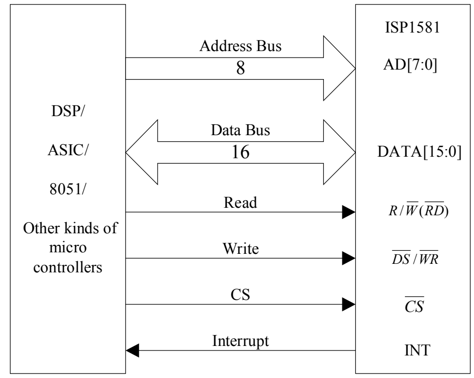

Figure 5. Connecting circuit in general controller mode

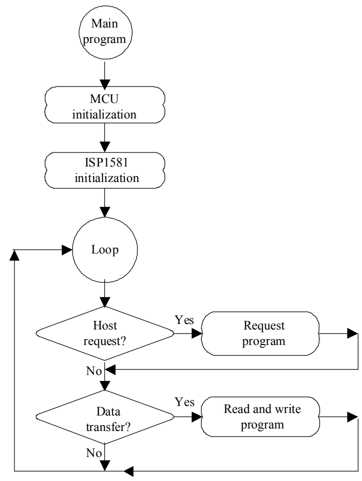

Figure 6. Main program flow chart

management and the initialization of different micro controller. After enabling the soft connection, ISP1581 could receive the reset signal of the host controller and then get into the reset interrupt. If the register address is mapped to the I/O space, the read and write of the register should be changed into I/O read and write mode. The interrupt spring mode of ISP1581 can be set as level trigger or edge trigger and the polarity can be set as well, which provide it a more convenient method to configure with different interrupt modes of different micro controllers. The mode of interrupt management can be both the hardware interrupt and the request interrupt. According to our experience, as long as the interrupt could be dealt with timely, the two modes both can work well.

USB device controller is complete interrupt driven, the whole information is transformed by interrupt and all of the transformations, except for the remote arousing, are started by the host, the devices are only waiting for responding. Thus, the main functions of the USB devices are completed by the host, which declines the cost of USB devices.

The interrupt service program completes all kinds of interrupts. Figure 7 is the flow diagram. After getting into the interrupt service program, the program read the interrupt register of ISP1581, and then transfers the corresponding sub program to deal with it. The reasons of the interrupt can be

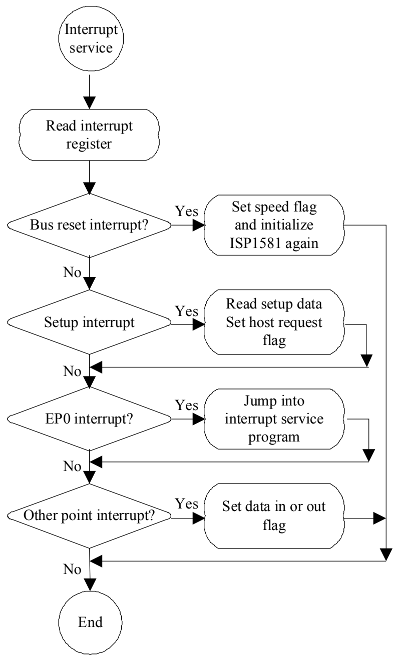

Figure 7. Interrupt service program flow chart

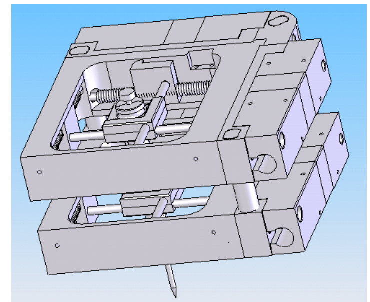

Figure 8. Structure of parallel robot

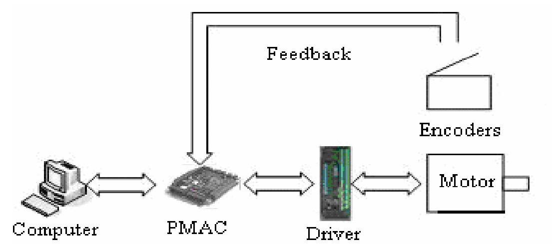

Figure 9. Structure of control system for paralled robot

summarized as follows: setup interrupt, hang, arouse, SOF, test high speed state, point 0 management and other interrupts. The interrupts illustrated in the flow diagram are the necessary interrupts to be dealt with, while other interrupts should be dealt with according to the different instance. Once the program receives the interrupt information, it firstly read out the 8 bytes of the setup pack, and then, sets the flags, when the main program tests out the flags, the interrupt request will be executed. Once receives the point 0 interrupt, except for the read and write operation, the interrupt program sends acknowledge signal to the host. Once receives the interrupt from other points, the program sets the corresponding flags, when the main program tests out the flag, it will read out data from the point or write data to it.

Request management deals with the standard request which the host sends to the devices during enumerating period and the manufacture request which the host sends to the devices during ordinary working. The standard request includes getting the device descriptor, setting address, getting configuring descriptor and etc. This part of program can be completed both in the interrupt disposal program and in the main program. For the former, this program is after the setup interrupt disposal sub program. For the latter, this program sets the interrupt flags waiting for the main program to test, once tested, the request management program will be executed and the main program transfers the corresponding sub program to deal with it.

III. THE CONTROL SYSTEM OF PARALLEL 4 DOF ROBOT

A. *Hardware Architecture*

We designed the parallel robot (Figure 8) to lead the needle to the precise access of the surgery. The movements of the parallel robot were driven by four DC motors on each degree of freedom. According to this feature, we chose its control system based on the computer and the movement control card. The movement control card is in charge of calculating the servo movement, while the computer can perform other operation related tasks such as the image mapping, surgical path planning and etc. The two parts are connected by parallel bus, together with the double ports RAM, the process become much speedier, so that this architecture could perform complicated arithmetic such as the multi-spindle and the track interpolation, what is more, the capability of the network communication is quite effective and more reliable, which can satisfy all the requests of this robot. The architecture is illustrated in Figure 9.

B. *The PID Adjusting of PMAC*

In this paper, we chose the Programmable Multi Axes Controller (PMAC) which is designed by Delta Tau, the type of this PMAC is PMAC2A-PC/104. We used the PMAC to control the four DC motors successfully, the following part we will emphasize the PID adjusting method of the PMAC.

In the field of industry control system, PID is one of the earliest and most widely used control method. There are many features of PID method such as simple principle, agile adjusting, better flexibility, even in the tough environment it can keep robust and reliability; it is widely used both in machine, electric, chemistry and other industrial field and in robot control, aeronautics, astronautics and other frontier field. The standard PMAC provide PID adjusting, servo loop filter, notch filter and other functions which developed dynamic capability of the control system.

PID controller is a device of proportional, integral and differential functions. The proportional part provides the system's rigidity; it determines the speed of responding. The integral part could eliminate the state error. The differential part provides the damp to the system to develop the dynamic characteristic of the system. Introducing both the feedforward and the feedback control structure could decline the error by

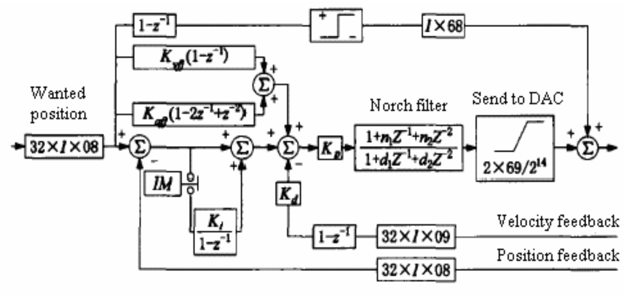

Figure 10. Structure of PID and norch filter

feedback and could compensate the error before the disturbing. This architecture plus a disturbing feedforward control on the base of feedback control. Since the whole compensatory condition doesn't change, the reliability of the system isn't affected. The feedforward controller doesn't destroy the reliability of the system, only compensates the adjective quantity, so that the PMAC used the two feedforwards to decline the track error of the servo control system, the velocity feedforward and the accelerate feedforward. The follow error caused by the inertial of the controlled object has a positive relation with the acceleration, which can be compensated by the accelerate feedforward. The velocity feedforward can decline the differential or the motor caused error. Adding these two feedforwards to the standard PID controller of PMAC, we can get the PID servo filter as illustrated in Figure 10.

PMAC provides powerful PID adjusting functions. Through the PEWIN TUNING software we could adjust the PID parameter conveniently and get the perfect control capability. There are two kinds of adjusting method of the PEWIN TUNING, one is manual adjusting, the other is auto adjusting. On one hand, for the system with better rigid such as the connection of the motor and load is rigid, we use the auto adjusting. On the other hand, for the system with a high frequency of resonance such as the connection of the motor and load is long axis, we use the manual adjusting method. PMAC can calculate the responding parameter automatically and draw the responding curve by the settings.

Step signal is usually used to reflect the capability of the filter. If we send the servo motor a step signal and observe the response of the system, we could get the exact steady-state and dynamic quality of the system by the curves drew by the PEWIN TUNING. We mainly adjust the PMAC parameter Kp (proportion, Ix30), Kd (differential, Ix31) and Ki (integral, Ix33) to set the feedback filter.

Firstly, we set the proportion Ix30 to 1000, differential Ix31 to 0, integral Ix33 to 0, the velocity feedforward and the accelerate feedforward to 0, and we will get the curves in Figure 11. From the curves the responding rise time is slow and the over shoot is zero, the position error is too much, so we need to adjust it again. On base of Figure 11, we increase the Kp and set Kd to 500, keep other parameters as before, we will get Figure 12. Comparing with the Figure 11, the rise time became shorter, the over shoot is still zero and the responding speed became faster.

In order to observe the function of Kp in the adjusting, we increase Kp to 50000 and get Figure 13. Apparently, the curve

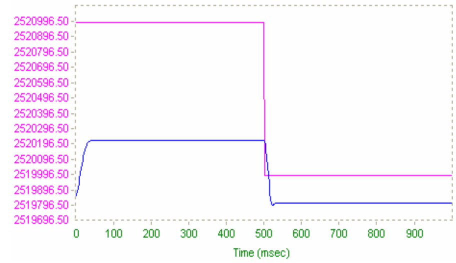

Figure 11. Motor step response curve

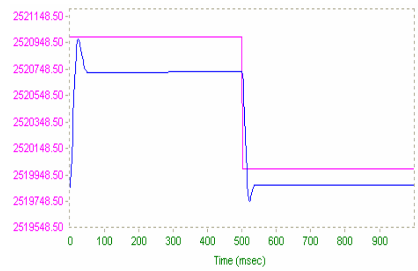

Figure 12. Motor step response curve

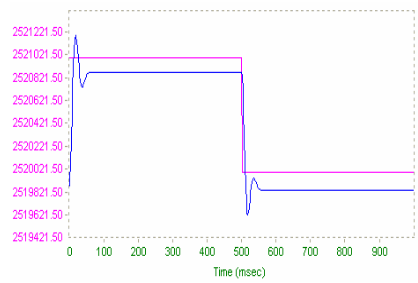

Figure 13. Motor step response curve

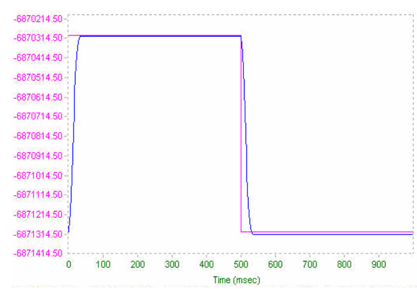

Figure 14. Motor step response curve

surges a lot, the over shoot is 31.7%, of course, that's not eligible. But the rise time became shorter, only 0.012s, the responding speed is developed. Through changing Kp, Kd and Ki, finally, we got the perfect responding curve of step signal (Figure 14). The rise time is 0.018s, peak time is 0.037s, damping is 1.0, and over shoot is 0, which could satisfy our requests of system.

For the servo system without the feedforward control, the follow error is always positive with the velocity and acceleration. When we introduce the velocity and accelerate feedforward, we could decline the follow error through adjusting the velocity and accelerate feedforward.

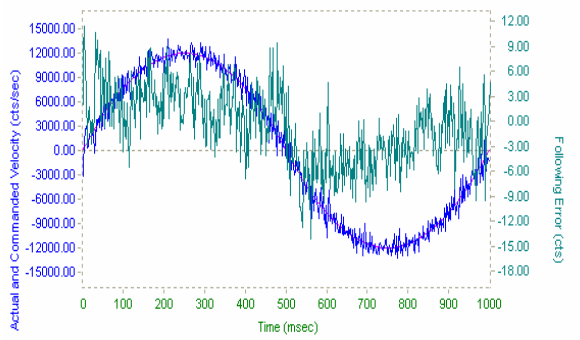

Figure 15. Motor parabola response curve

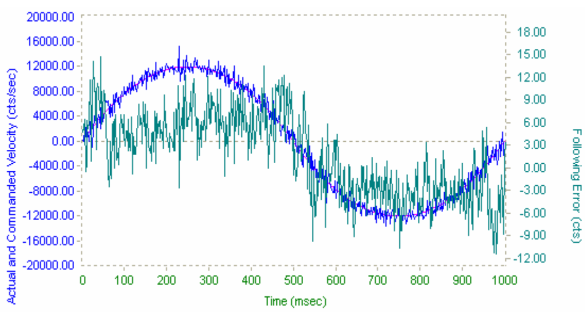

Figure 16. Motor parabola response curve

Firstly, we adjust the feedforward and run a serial of parabola movement to observe the result in order to decline the follow error and correlation modulus. We increase the feedforward (firstly for velocity and then for accelerate) from zero, till the ratio is near to zero. If the correlation modulus is a huge number, there must be some overshoots at the end of the movement.

Figure 15 to Figure 18 are the responding parabola curves. In the figure, the smooth curve is the wanted velocity curve; on this curve, the small wavy curve is the fact curve we get; and the big wavy one is the follow error curve. The vertical coordinate on the left is for the fact and the wanted, the vertical coordinate on the right is the follow error.

From Figure 15, we can find that when the proportion and the differential stay steady, velocity feedforward is 2500, the follow error is too large to satisfy our request. When the motor stands still, we set the integral mode Ix34 to 1 to let the integral make sense, Ix33 is 10000, the curve became better (Figure 16). If we increase the velocity feedforward to 3000, the curve of follow error tends to cover the curve of velocity and then turns out to be the opposite of the velocity curve (Figure 17), so we'd better not increase the velocity feedforward. If we set the accelerate feedforward to 10000, we will get Figure 18, the follow error waves on the wanted curve and distributes uniformity, that's the curves we wanted and the motor runs in a very good state.

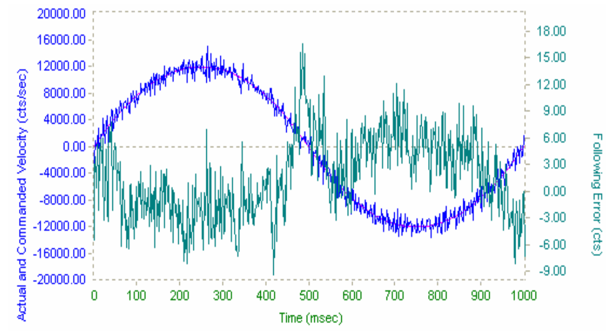

Figure 17. Motor parabola response curve

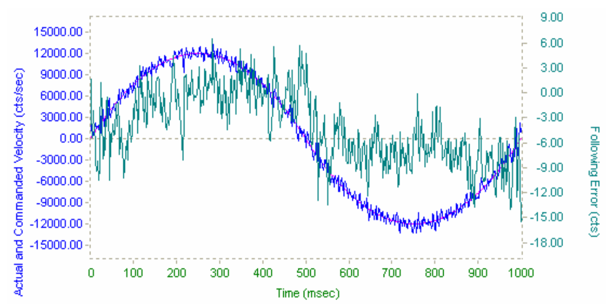

Figure 18. Motor parabola response curve

IV. CONCLUSION

We developed the position testing accuracy of the serial robot by the design of FPGA based 4 times frequency and orientation circuit. At the same, we introduced the USB based transformation module to highly increase the speed of the data transformation. In addition, through the adjusting of PID parameter based on PMAC, the parallel robot hit the design target of fast response and high precision. The whole robot system could work under perfect state.

REFERENCES

[1] Potamianos, P., Davies, B.L., and Hibberd, R.D. "Intra-operative imaging guidance for keyhole surgery methodology and calibration". Proc. First Int. Symposium on Medical Robotics and Computer Assisted Surgery, Pittsburgh, PA. p. 98-104, 1994.

[2] Potamianos, P., Davies, B.L., and Hibberd, R.D. "Intra-operative imaging guidance for percutaneous surgery". Proc. Second Int. Symposium on Medical Robotics and Computer Assisted Surgery, Baltimore, MD. p. 156-164, 1995.

[3] Taylor R.H., Funda J., Eldridge B., Gruben K., LaRose D., Gomory S., Talamini M., Kavoussi L.R., Anderson J. "A Telerobotic Assistant for Laparoscopic Surgery", IEEE Engineering in Medicine and Biology Magazine, Vol. 14, pp. 279-287, 1995.

[4] Bzostek, A., Schreiner, S., Barnes, A.C., Cadeddu, J.A., Roberts, W., Anderson, J.H., Taylor, R.H.,Kavoussi, L.R.. "An automated system for precise percutaneous access of the renal collecting system". Lecture Notes in Computer Science, Spring-Verlag, Vol. 1205, pp.299-308, 1997.