2017 IEEE 3rd International Conference on Control Science and Systems Engineering

A Precise High-Speed Tracking and Pointing Control System of Camera Based on FPGA

Closed loop feedback control system to control the remote sensing camera lens moving

Zhao Liting, Lin Zhe
Camera Control Technology Laboratory
Beijing Institute of Space Mechanics and Electricity
Beijing, China
e-mail: 403812355@qq.com, zhe_lin_hit@yahoo.com

Abstract—**This paper designed a precise high-speed tracking and pointing Control system of Camera based on FPGA. The control system can control the remote sensing camera lens moving accurately and flexibly. The system had the function of control angle position、control speed and stable speed regulation. We used the step motor to be actuator. The acceleration and deceleration process was stable、effective and high precision. The system adopted the closed loop feedback method. Using the FPGA simplifies the structure of motor control system, reduced the system cost. The modular system can be widely used in various fields.**

Keywords-point control system; remote sensing camera; FPGA; step motor

I. INTRODUCTION

Remote sensing cameras change the particular direction and the position to satisfy the meet of imaging. Camera pointing machines are important parts of camera. Camera pointing machines follow up the scent of a special area accurately during the satellites flying on the orbit. The lens focuses the light onto the focal plane in a fixed period. The Exact Control systems of Step Motor drive the direction structures to direct at target accurately, and achieve the scheduled function.

Step motors are small in size and have high positioning accuracy, they have high operating frequency and good dynamic characteristic, they have no accumulative error [1]; step motors are easy to be controlled reliably, theirs location lie on the total number of instructions pulse, theirs speed is proportional to the instructions frequency, so it is easy to be controlled accurately. This paper introduces the application of FPGA in a precise point Control system of Camera. The point control system uses the step motor as the actuator, uses FPGA (Field Programmable Gate Array) to calculate the drive signals of the step motor. The point accuracy for the system achieves $\pm 0.02°$. The range of steering angle is from -35° to +57.5° . The most moving speed of the step motor is 1075pps (Pulses per second).

II. HARDWARE STRUCTURE OF THE CONTROL SYSTEM

The relationship between the point control system and the rest equipment is expressed as in Fig. 1:

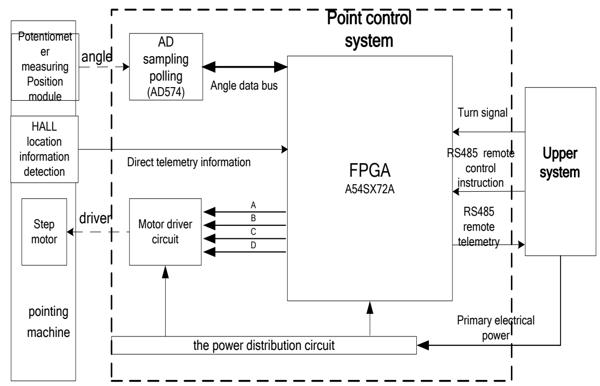


Figure 1. Hardware components connection diagram of the control system.

FPGA receives remote control instructions of upper system by RS485 communication; FPGA send information to upper system by RS485 communication of telemetry; FPGA receives real time position information from Potentiometer measuring Position module, AD sampling polls the position information of motor; HALL location information detection send the HALL effective information to FPGA. FPGA carries locomotion control arithmetic to calculate ABCD four-phase frequency conversion signals, then exports the signals to the Motor driver circuit to drive the motor; the power distribution circuit receives the Primary electrical power from the upper system, then transform the voltage to supply FPGA and Motor driver circuit with electric power.

The FPGA acts as the control chip of the motor, it achieves the functions of control pulse output、direction and exciting way, etc. So the FPGA does not require a mass of programmable logic units. A general FPGA can satisfy the design needs. We chose FPGA A54SX72A-1CQ208M model from the ACTEL Company. Speed Grade -1, 108000 System Gates, Package PQFP, clock frequency 20MHz, power consumption less than 1w, good security; we used HALL sensor CS3040 to collect location information; Potentiometer measuring Position module was angle feedback component [2]; AD sampling polling (AD574) converted the analog signals of Potentiometer into the digital signals. FPGA read the digital signals to obtain the real time angle information.

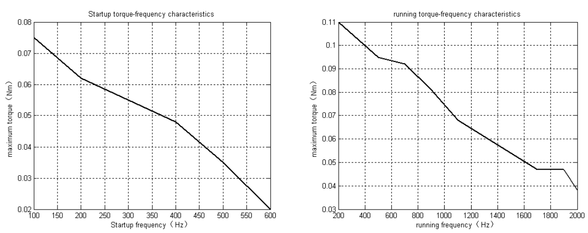
978-1-5386-0484-7/17/$31.00 ©2017 IEEE

This paper chose J47BH001 type hybrid stepping motor and a harmonic reducer, the main technical parameters of the stepping motor are shown in Table I below:

TABLE I. MAIN TECHNICAL PARAMETERS OF THE STEP MOTOR

Type	J47BH001
Number of phase	four
voltage	24v
step angle	1.8°
Work temperature	-20°C~+100°C
total quality	≤550g
Phase resistance	（5 ± 0.5）Ω
Phase inductance	（18 ± 5）mH
Position torque	≥5 mNm
Remain torque	≥100 mNm

The torque-frequency characteristics of stepping motor are shown in Fig. 2 below:



(a) Startup torque-frequency characteristics (b) running torque-frequency characteristics

Figure 2. The torque-frequency characteristics of stepping motor.

III. ACCELERATING AND DECELERATING MOTION DESIGN FOR MOTOR

This paper used a harmonic reducer with 50 decelerate ratio. So the maximum error of the point is the half step angle between the stepping motor and the target angle. The step angle is 1.8°, therefore the error is 1.8°/50/2=0.018°, it satisfy the precision request ±0.04°; the maximum move speed is 1075pps; the transmission efficiency⩾65%.

The calculate formula [3] of the accelerate torque for step motor is in (1):

$$J\frac{d^2\theta}{dt^2} + D\frac{d\theta}{dt} + T_L = T_M \qquad (1)$$

In the formula, T_L is the load torque; J is the rotary inertia of motor shaft load; θ is angle location of motor; D is velocity proportional coefficient; ignore the velocity $D\frac{d\theta}{dt}$ because D is so small; T_M is electromagnetic torque which the motor can generate, $T_M = T_L + T_a$, T_a is inertia accelerate torque. The accelerate torque is in (2):

$$T_a = J\left(\frac{\pi\theta_S}{180}\right)\frac{(f_2 - f_1)}{t_a} \qquad (2)$$

In the formula, f_1 is the frequency before the acceleration; f_2 is the frequency after the acceleration; t_a is accelerating times; θ_S is step angle of the motor.

In this paper the rotary inertia of the whole point machine is $J= 2.8\times10^{-2}/K^2$ (kgm^2); axis of rotation had 50 decelerate ratio; the length of drive pulse is $c_1=f/f_1$, $c_2=f/f_2$; f_1 is the previous pulse counting; f_2 is the next pulse counting; f is the running frequency of FPGA.

The motor drives the load to startup using the expected speed. It will not startup if the drive speed is faster than the maximum pulse frequency of the motor. We used slow acceleration and deceleration control method to increase speed linearly. The design principle of the method is: firstly using the Startup and running torque-frequency characteristics in figure (2) to program the accelerate torque curve; then using the accelerate torque curve to calculate the drive pulse of motor [4].

The maximum running speed of motor in the pointing control system is 1075 pps, it is equivalent to the Startup torque-frequency being 1075Hz. The motor cannot start up with the 1075pps, so we designed the acceleration and deceleration process as Fig. 3. At this time the required torque for motor is calculated as Fig. 4. The acceleration curve should meet the technical requirements; the drive torque margin should be greater than 3.

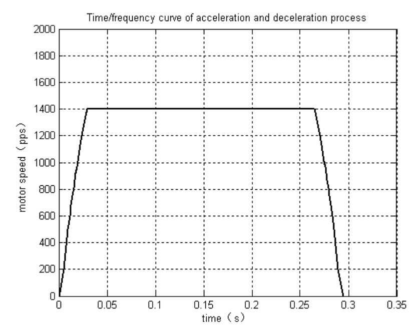


Figure 3. Time/frequency curve of acceleration and deceleration process.

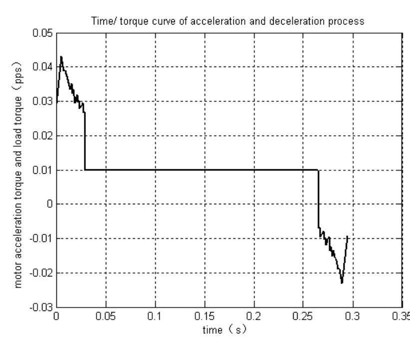
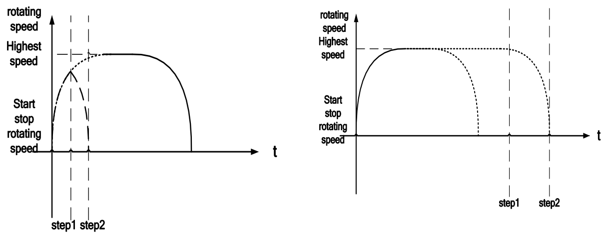


Figure 4. Time/ torque curve of acceleration and deceleration process.

The rotation angle of the motor is random when the each work mode of the point controller. We designed an acceleration and deceleration curve for the maximum running speed. The acceleration process matched the deceleration process; the frequency change rules and step numbers are the same. The angle switch used the same curve as Fig. 3 in different situations. The key to programming acceleration and deceleration curve is how we differentiated the deceleration time of different steps. The angle switch pattern in software is shown in Fig. 5.



(a) small angle switch sketch map (b) big angle switch sketch map

Figure 5. Any angle switch sketch of motor.

IV. FPGA SOFTWARE DESIGN

The workflow of the FPGA software are described below: power FPGA up, initialize and reset FPGA; forbid motor-enabled to lock the motor; receive the RS485 remote control instruction from the upper system; parse command to get the angle command of motor; receive the real time angle information from the AD sampling polling and HALL location information detection; send angle information and movement step to calculate module of motor control signals; calculate the pulse signals according to the acceleration and deceleration curve; generate and sent driver signals of motor; sent three lines of telemetry to upper system which containing the angle information.

The calculate flow of control signals is shown in Fig. 6.

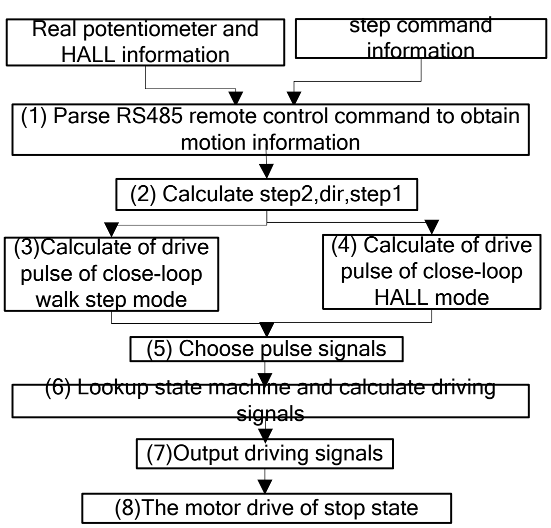


Figure 6. Calculation process of control signals.

FPGA software algorithm is shown as below:

A. We Used RS485 for Bus Communication. Authors and Affiliations

The length of the unit was 20 bit. Bit stream included 1 bit start bit 1 bit flag bit 16bit information bit 1bit parity bit 1bit stop bit. Two types were instruction word and data word. Synchronization head of instruction word was 00. Data word head was 01. The bus is always high in all free time. Baud rate of 485 communications is 1 Mbps.

Read remote control instruction by 20MHz. Judged center position of the falling edge. Reading the parity bit and doing check of odd parity and accumulation.

B. FPGA Reads Real Time Angle Information

FPGA parses the step command information from the RS485 of remote control; FPGA uses the angle information and step command to calculate step number step2 as (3)：

$$step2=|step\text{-}angle|*K\,/\,\theta s \tag{3}$$

In the formula, *step* is the step number of command; angle is the real time position of motor; θs is the step angle of motor, if the step motor adopted the control mode of four-phase eight-beat, $\theta s=1.8°/2$; K is the reduction ratio.

FPGA should realize the multiplication of decimals, we designed arithmetic as bellow: used 16bit binary number to represent the angle information; integer is 11bit because the max angle command is below 1000°, used complementation to represent minus; decimal fraction is 5bit, so the precision satisfied 0.02°; because the reciprocal of step angle $1/\theta s \approx 0.5556$, used 14bit to represent the decimal fraction, then the precision satisfied 0.0001; Used Mega Wizard Plug In Manager in quartus to build a lpm_mult 16bit*14bit component to do the multiplication.

Calculated the movement direction *dir* as in (4):

$$dir = \begin{cases} 1 \text{ when } step - angle > 0 \\ 0 \text{ when } step - angle \le 0 \end{cases} \tag{4}$$

Designed the step numbers of acceleration and deceleration process are both N. Calculated the total number *step*1 of step before slowing down as in (5):

$$step1 = \begin{cases} \mathrm{INT}\left(step2\,/\,2\right) & \text{when } 0 < step2 < 2N \\ step2 - N & \text{when } \quad step2 \ge 2N \end{cases} \tag{5}$$

INT was the meaning of rounding up to an integer.

C. Calculated the Drive Pulse Signals When Close-Loop Walk Step Mode

Calculated the drive pulse signals based on *step*1 、*step*2 and direction information from the forward communication module; The count value of pulse width was stored in the ROM. The lpm_rom component was selected from FPGA LPM library. The data width of ROM was 16bit; the data depth of ROM was N; the address width of the ROM was INT $[\log_2(N)]$. The algorithm of pulse width counting is shown as below:

if now_*step*<*step*1 and *now_step*<*N*, then *address* <= *address*+1;

elsif *now_step*<*step*1 and *now_step*>*N*, then *address* <= *address*;

elsif *now_step*>*step*1 and *now_step*<*step*2, then *address* <= *address*-1;

elsif *now_step*>*step*2, then *address* <= *address*; *now_step* is the count number of the current step, the initial value *now_step*=0; *now_step*+1 when a pulse data has been generated correctly. *now_step*=0 when *now_step*>*step*2.

D. Calculated the Drive Pulse Signals When Close-Loop HALL in Place Mode

The motion control flow chose Hall information when the close-loop HALL in place mode is valid. We used HALL in place information and current steps to Calculate drive pulse signals:

if now_step <*N* and hall = '0' then *address* <= *address*+1;

elsif now_step >= *N* and hall = '0' then *address* <= *address*;

elsif hall = '1' and *address* > 0 then *address* <= *address* - 1;

elsif hall = '1' and *address* = 0 then *address* <= *address*;

Produced the last drive pulse and set *now_step* zero when HALL in place and the *address* = 0; *address* = 0 mean the motor was slow down to the lowest; The counter was reset and drive pulse was made when the accumulation addition was equal to the corresponding data of *address.*

E. Chose Control Process

Chose control process according to Remote control information. This paper designed three drive control mode: PDE (pulse\direction\enable) external input drive mode; close-loop walk step mode; close-loop HALL in place mode.

F. The State Machine

Looked up the current state of the state machine, and then generated the pulse signals for corresponding pins: the step motor adopted the control mode of four-phase eight-beat; the movement divided into the positive and negative directions. The order of stator electricity is A – AB – B – BC – C – CD – D - DA when positive movement, it corresponded eight states of the state machine, so the A/B/C/D output of the drive pulse was "1000"-"1100"-"0100"-"0110"-"0010"-"0011"-"0001"-"1001". "1" presented power-up state, and "0" presented power-off state. The initial state and the final state were "0000".

G. Output Drive Pulse

FPGA take count of the pulse width after looking up the state, and then outputted the drive pulse for A/B/C/D.

H. The Motor Drive of Stop State

The motor drive of stop state: Our system designed the contain power-on mode and contain power-off mode after one control process finish. Software detect the last drive state to guarantee the contain power-on mode being single phase power up. Four-phase eight-beat drive mode has the possibility of double phase power up and it is large current and low efficiency. We need record last status value to avoid out of step accumulation.

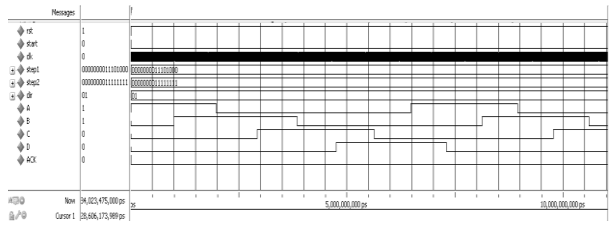


Figure 7. Four-phase drive signals of motor.

We used modelsim to write testbench for step motor control module. The input motion angle was 609°, the control module calculated that *step1*=654，*step*2=677。The pulse width of drive signals just liked the pre-set width curse as shown in Fig. 7.

V. VERIFICATION RESULTS OF TRACKING AND POINTING CONTROL SYSTEM

The tracking angle command condition of the motor is shown as Fig. 8. The experiment results showed that the control performance can satisfy the design requirements. The angle position command 667° could be responded by 1253pps>1075pps. The tracking time was less than 0.3s, and the tracking step number was 376. The accelerate process was not out of step and running smoothly. The driving torque margin was above 3. So the tracking and pointing control system on-orbit was high reliability and security. The power consumption for FPGA was less than 1 watt. The control system was successfully used on the lens orientation movement of space remote-sensing camera.

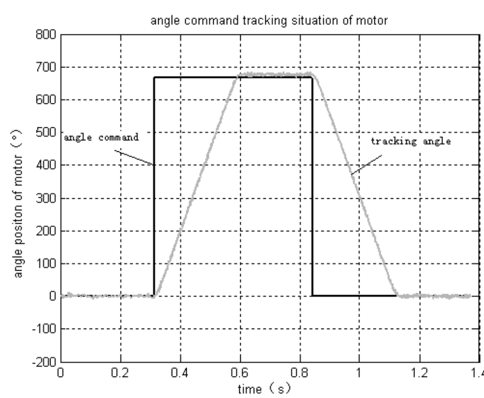


Figure 8. Angle command tracking situation of motor.

VI. CONCLUSION

This paper designed a precise high-speed tracking and pointing Control system of Camera based on FPGA. The control system can control the remote sensing camera lens moving accurately and flexibly. The system had the function of control angle position and stable speed regulation. We used the step motor to be actuator. The acceleration and deceleration process was stable、 effective and high precision. The system adopted the closed loop feedback method by Rotation Transformer resolver, so the motor was complete synchronization and no stuck, etc. Using the FPGA 54 series chip simplifies the structure of motor control system, reduced the system cost. The modular system can be widely used in various fields.

REFERENCES

[1] Gong Ya 1i , Wang Wen ming, "Application of FPGA in Control of Step Motor", electronics technology, Vol.36, No.11,pp.11-12,2009.

[2] Wang ai yuan,Control motor and its application, Shanghai jiaotong university press, Shanghai, China,2013. pp.115-120.

[3] Sakamoto Tekisuto. Stepper motor application technology ,Sciences Press,Beijing China, pp. 94-95,2010.

[4] Xiao long, Tang ensheng. The Optimizition of Genetic Algorithms for the PID Parameters of Motor Controller System , spacecraft recovery & remote sensing ,Beijing China, VOL.27, No.2, pp.34-37,2006.

[5] Zhang Gui—rong, Li Guang—can, PAN ji—wen, "Conversion of pulse control signal to frequency control word in stepping motor", Modern Electronics Technique, vol.35, No.17,pp. 177-179,2012.

[6] Zhang heng-hao,Yan jian-hong,Li zhu-jun,Zhou yong, "Design and Implmentation of FPGA in Space Electromotor Control System",Space Electronic Technology, Vol.12, No.2,pp. 44-50,2012.

[7] Liang Chen,HE Huio, "High precision Controller for Stepper Motor with AutoSpeed Regulation Based On FPGA", Chemical automation & instrument, Vol.12, No.12, pp. 1481-1484, 2011.