Fast and Accurate Motion Control Based on Good Gain Method

Gao Qingji, Li Zheng*
Robotics Institute, Civil Aviation University of China, No.2898 Jinbei Road, Tianjin 300300, P.R.China
*Corresponding author, email: arex1987@live.cn

Abstract

There are some drawbacks exist in multi-method intelligence control algorithms which are popular researched in motion control. Many of the most significant ones are high algorithm complexity, long time-consuming and not easily applied in industrial production. According to the disadvantages mentioned above, and also take the conflicts between control accuracy and response speed into consideration, a "Good Gain" PID tuning method is studied on motion control. Firstly this PID tuning method is described and the processes of finding proper PID gains by analyzing the outputs of PID controller are designed. Secondly the method is simulated on a DC motor model and finally acquires a proper balance between response speed and control accuracy. Finally the method is tested on real DC motor and control devices. After software simulation and real experiments, the effectiveness of this PID tuning method is proved.

Keywords: *Good Gain, Motion Control, PID tuning, LabVIEW, DC motor*

1. Introduction

The large inertia characteristics lead DC motor velocity to a hard control which is mainly reflected in the equilibrium between fast response and control accuracy. Now, many kinds of motion control algorithms are under research, and one of the most popular fields is multi-method intelligent control and self-tuning such as Fuzzy-PID control, ANN-PID control, Rough-PID control and so on [1-6]. These control algorithms can automatically provide reliable control process and require a stable velocity curve, but their disadvantages are also very significant. Firstly, their complex tuning and control progresses cost such a long time. Secondly, even the results of these algorithms are stable, but mostly their control responses can hardly meet the running expectation of real systems, changing their control parameters is also a hard work.

For the reasons above, the motion control based on "Good Gain" PID tuning method is studied. Taking DC motor velocity control into consideration, by tuning PID parameters, this method ensures the fast response of large inertia part. Firstly the whole process is simulated in LabVIEW environment. Then its control effect is tested and verified through real experiments on DC motor cooperated with CompactRIO control system.

2. "Good Gain" PID Tuning Method

"Good Gain" PID tuning method was developed by Finn Haugen in 2010 [5]. It is a method that based on both experience and theoretical calculation, and it also belongs to an effective human involvement linear tuning method. Its methodology flow is shown in Figure 1.

The way of calculating PID parameters K_p, T_i and T_d from observed K_{PGG} and T_{ou} are:

$$K_p = n_p K_{PGG} \, , T_i = n_i T_{ou} \, , T_d = n_d T_i \qquad (1)$$

In equation (1), the value of proportional gain K_p is slightly less than K_{PGG}, so n_p can take the value about 0.8; the integral time is longer than T_{ou}, so n_i can be the reference value

Received October 5, 2012; Revised January 8, 2013; Accepted January 19, 2013

1.5; differential time is proportional to the integral time according to classic Z-N rule, so n_d is 0.25.

The differential and integral gain K_i and K_d can be calculated from T_i and T_d cooperate with the sampling period T_s:

$$K_i = \frac{K_p{}' \, T_s}{T_i}, K_d = \frac{K_p{}' \, T_d}{T_s} \tag{2}$$

The unit of all the time variable T_i, T_d and T_s using in equation (1) and (2) are second(s).

Since the values of K_{PGG} and T_{ou} in this method are initially acquired by observation, so the PID gains calculated from them may not be proper values that can be used in the control system. Therefore, after the PID gains have been calculated, some manual adjustment is still needed in order to find acceptable control result.

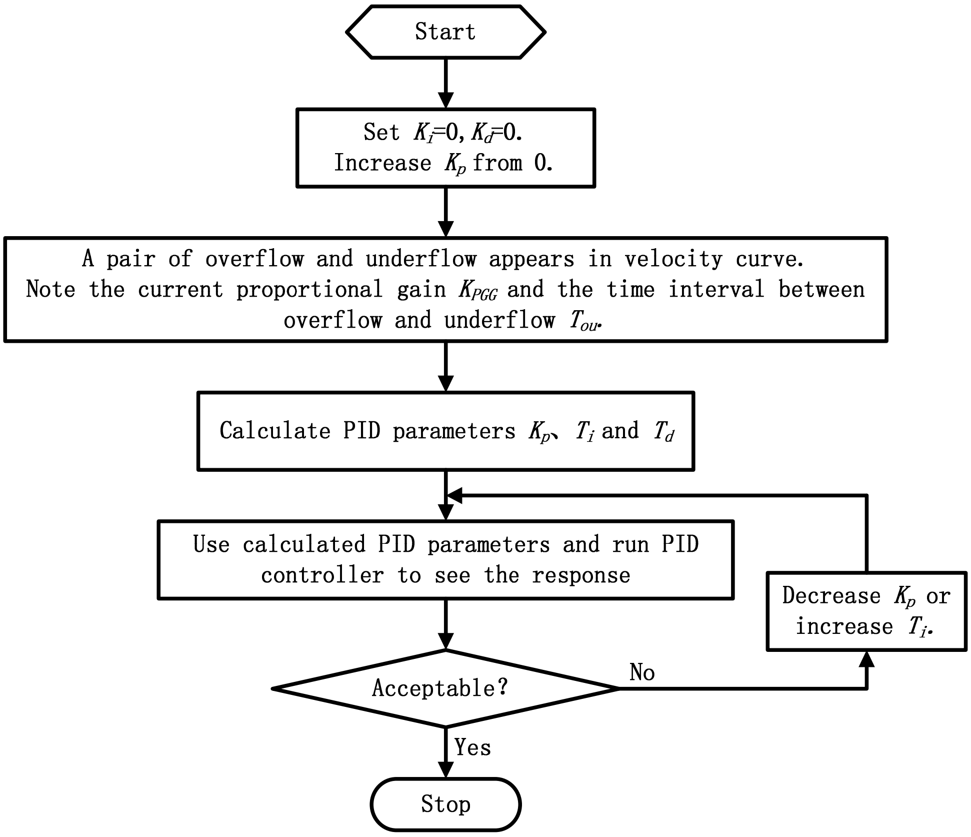
Start
Set K_i=0, K_d=0.
Increase K_p from 0.
A pair of overflow and underflow appears in velocity curve.
Note the current proportional gain K_{PGG} and the time interval between overflow and underflow T_{ou}.
Calculate PID parameters K_p、 T_i and T_d
Use calculated PID parameters and run PID controller to see the response
Decrease K_p or increase T_i.
Acceptable?
No
Yes
Stop

Figure 1. Methodology flow of “Good Gain” method

3. DC Motor Model and Software Simulation

3.1. DC Motor Model

The internal structure of the DC motor and the external load, and the related various parameters are shown in Figure 2 [5].

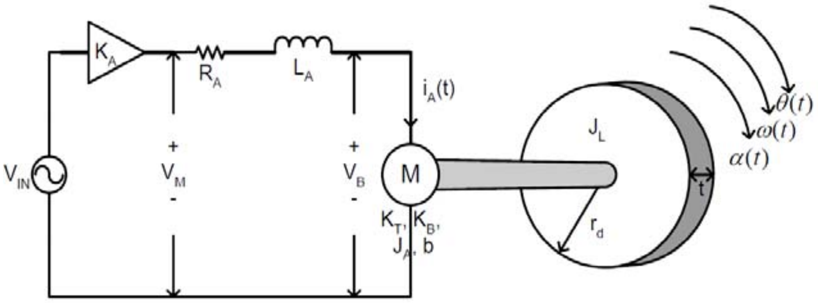

Figure 2. DC motor model

The mathematical model of DC motor's angular acceleration and internal current can be drawn from the mechanical properties and electrical characteristics:

$$\frac{d^2\theta(t)}{dt^2}=\frac{1}{J_T}\left[K_T i_A(t)-b\frac{d\theta(t)}{dt}\right] \tag{3}$$

$$\frac{di_A(t)}{dt}=\frac{1}{L_A}\left[-Ri_A(t)+V_M-K_B\frac{d\theta(t)}{dt}\right] \tag{4}$$

The value of J_T in equation (3) is the total inertia of motor and load disk.

$$J_T=J_A+J_L \tag{5}$$

According to the DC motor model mentioned above, a simulation program is developed by using Control Design & Simulation module in LabVIEW environment. Under the condition of open loop and no load, the result of simulation is shown in Figure 3.

The curve on Figure 3 shows the simulation result is consistent with the motor chosen for real experiment, including start-up current, maximum velocity and the linear relationship of PWM duty cycle and motor velocity. So this motor model is proved and the further simulations and analyses of the control system can be started on it.

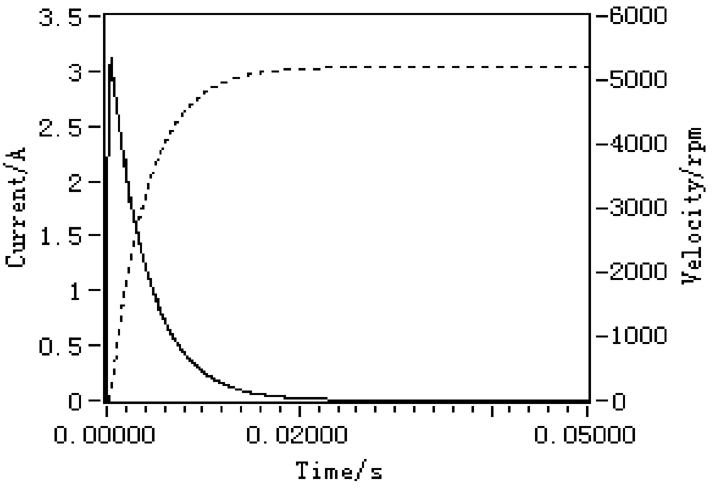

Figure 3. No-load motor velocity simulation (100% Duty cycle)

3.2. PID Tuning

Simulation program takes PID.vi provided by LabVIEW as the controller, uses PID algorithm to control motor velocity by adjusting PWM duty cycle. Set the time of PID control cycle to 0.01ms. The simulation of control system is shown in Figure 4.

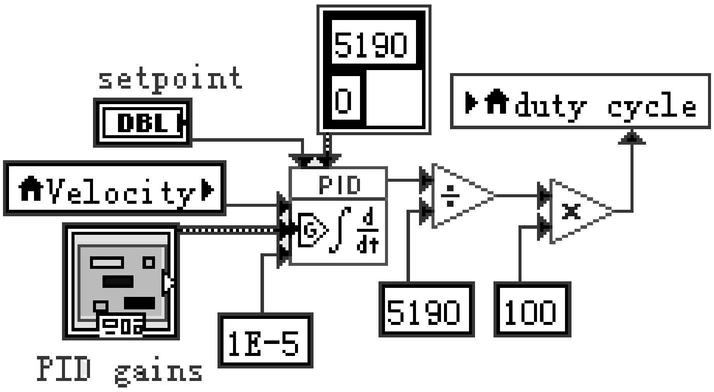

Figure 4. PID control simulation program in LabVIEW environment

The inputs of PID controller are motor velocity, setpoint, PID gains and sample time. After essential calculation, the output of PID controller is transferred to PWM duty cycle.

The relationships between the inputs and outputs of this PID control simulation are:

$$u(k)=u_p(k)+u_i(k)+u_d(k) \quad , \quad e(k)=setpoint(k)-PV(k)$$

$$u_p(k)=K_pe(k)\ ,\ u_i(k)=K_i\sum_{i=1}^{k}\left[\frac{e(i)+e(i-1)}{2}\right]\ ,\ u_d(k)=-K_d\left[PV(k)-PV(k-1)\right]$$

In the equations above:

k : Sample times;

$u(k)$: PID output;

$e(k)$: Error, the difference between process value and setpoint.

$PV(k)$: Process value, in this controller is motor velocity.

$u_p(k), u_i(k), u_d(k)$: Outputs of proportional, integral and differential controller.

K_p, K_i, K_d : Proportional, integral and differential gain.

Follow the methodology flow which is shown in Figure 1 to find initial PID gains by using “Good Gain” tuning method. Firstly, ensure that the controller is a P controller with the values of integral and differential gains are 0, increase proportional gain value from 0 until you observe some overshoot and a barely observable undershoot, like what is shown in Figure 5. The proportional gain value at this time is denoted K_{PGG}. By analyzing the response curve, find the time interval between overshoot and undershoot T_{ou}. So $K_{PGG}=25$, $T_{ou}=0.0006$.

According to equation (1), the proportional gain, integral time and differential time can be calculated from K_{PGG} and T_{ou}:

$$K_p=20\ , T_i=9e-4\ , T_d=2.25e-4$$

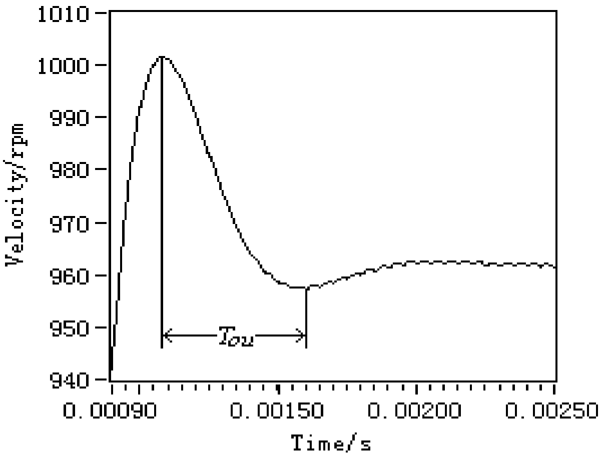

Figure 5. Getting K_{PGG} and T_{ou} (setpoint =1000)

Using these values in PID controller and the curve of motor no-load velocity response simulation is shown in Figure 6. From the simulation result, the velocity response runs well in high speed, but some oscillations appear in low speed. So the PID gains need manual tuning like what is shown in the methodology of Figure 1. To solve this problem and get more acceptable response, the possible ways are to increase integral time or decrease proportional gain [5].

The final PID gains after adjustment are: $K_p = 20$, $T_i = 4.8e - 3$, $T_d = 1.2e - 3$.

The PID control velocity response under adjusted parameters is shown in Figure 7. Increasing the velocity of DC motor from 0rpm to 1000rpm and 5190rpm (maximum velocity) respectively take about 2ms and 19ms. Both of the control processes have stable running at the set velocity with no obvious oscillation, and steady-state error is less than 1%.

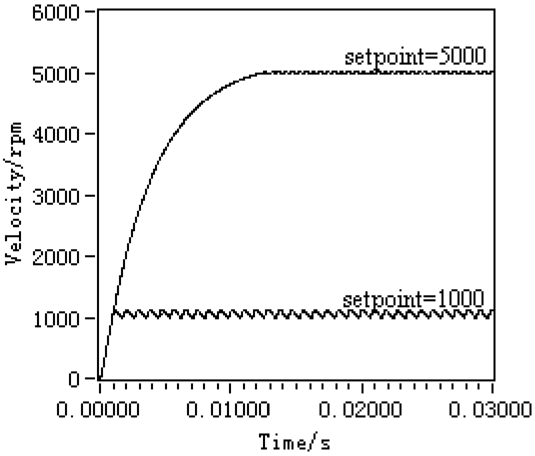

Figure 6. Simulation on initial PID gains

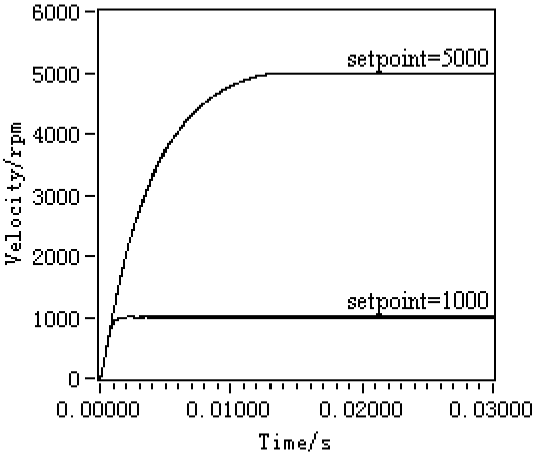

Figure 7. Simulation after manual PID gains adjustment

However, in real control system, DC motor often runs with load. So in our experiment we use an inertia disk like what is shown in Figure 2. Its inertia can be calculated from equation (6) below:

$$J_T = \frac{1}{2} M r_d^{\ 2} = \frac{1}{2} r\, p\, r_d^{\ 4} t \qquad (6)$$

In equation (6):

r_d : Radius of inertia disk;

r : The density of the material of inertia disk;

t : Thickness of inertia disk.

From equation (5) and (6), total inertia of the DC motor using in real experiments can be calculated as 2.55×10^{-6}kg·m^2. The control simulation under this condition is shown in Figure 8.

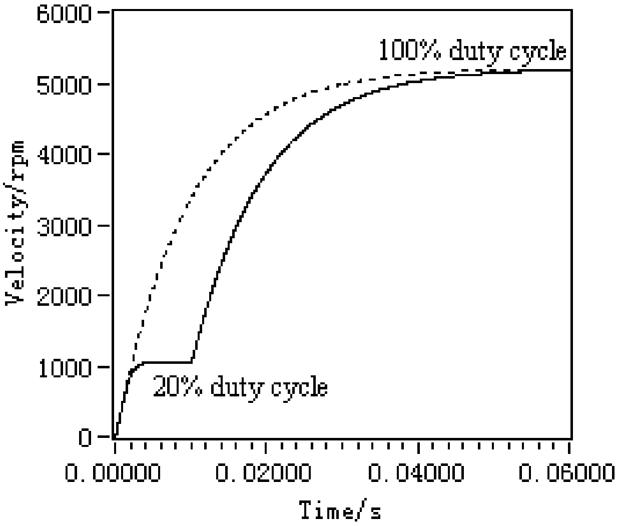

Figure 8. Simulation with inertia disk

Figure 8 draws that the motor starts running from 0rpm to maximum speed takes about 55ms; from 0rpm to 1038rpm (20% PWM duty cycle) then accelerates to maximum velocity at the time of 10ms takes about 5ms and 44ms respectively.

3.3. Compare with Other Tuning Methods

When compare with self-tuning methods, “Good Gain” method has its own superiorities. Firstly, after manual adjustment, “Good Gain” method can provide a stable response fits for the corresponding system. Secondly, the algorithm of “Good Gain” method is easier than self-tuning ones to understand, and also its simple algorithm leads to a proper response. For these reasons, “Good Gain” manual PID tuning method can be better used for real system.

When compare with classic experience based manual tuning methods Ziegler-Nichols’ rule, “Good Gain” method takes less time reaching the desired goal. “Good Gain” tuning method has many similarities with the classic Ziegler-Nichols’ rule. But the latter one has a serious drawback, that it requires the control loop to be brought to the limit of stability during the tuning, while the Good Gain method requires a stable loop during the tuning. In Z-N rule, it is hard to find the oscillation limit, which leads to more slight adjustment after theoretical calculation. But in “Good Gain”, it is easy to find the parameter T_{ou} and quickly calculate the result which is closer to the reasonable one. Figure 9 shows the different velocity control response after first step PID tuning. So, it is obviously that “Good Gain” method can lead to a more acceptable result which needs less slight adjustment for PID gains.

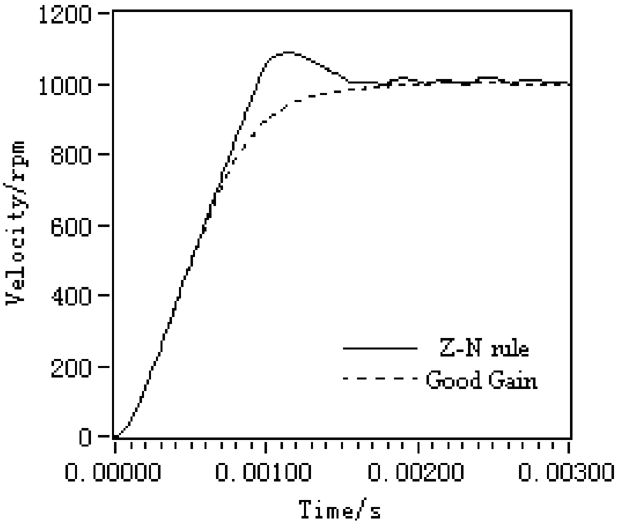

Figure 9. Comparison between Z-N rule & "Good Gain" (setpoint=1000)

4. Experiments and Result Analyses

4.1. Experiments

The composition of the whole control system and workflow is shown in Figure 9. The core program of control system is composed in LabVIEW environment, consists of the following four parts [8].

- PID loop: Implement PID algorithm using the sampled data, the core part of the whole program.
- Encoder loop: Acquire the feedback from encoder and calculate motor velocity from it.
- PWM generate loop: Generate PWM wave based on given duty cycle.
- Surveillance loop: Monitor the running status of every part of the control system and operate the switches of every function module.

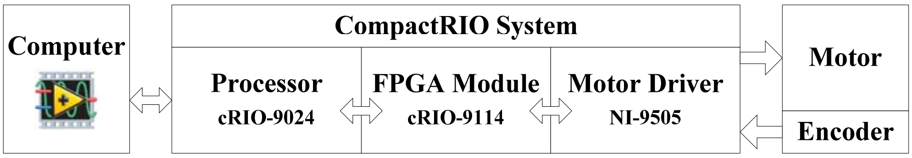

Figure 10. System composition & workflow

Set up the PID gains of the controller with the PID tuning results of simulation and run the program. Then some manual adjustments need to be done according to the status of motor velocity curve in order to get a fast response and stable running status of the motor. The 2 diagrams in Figure 10 indicate the velocity control processes in low speed status (from 800rpm to 1800rpm) and high speed status (from 2500rpm to 3500rpm).

4.2. Result and Analyses

The result of real experiment indicates that increasing the motor velocity by 1000rpm from one stable running velocity to another takes about 260 steps in average. Every step represents a control cycle which cost about 0.026ms. Therefore the whole process cycle takes about 7ms makes the value of motor velocity from one to another. The more velocity increases, the longer time the control process costs. Driving the motor from stationary status to 5000rpm takes about 40ms in average. At the same time, the control accuracy and its control response speed reach an acceptable equilibrium. Figure 11 takes the motor running status at 1500rpm as an example to indicate its proper control accuracy with a steady-state error less than 0.8%.

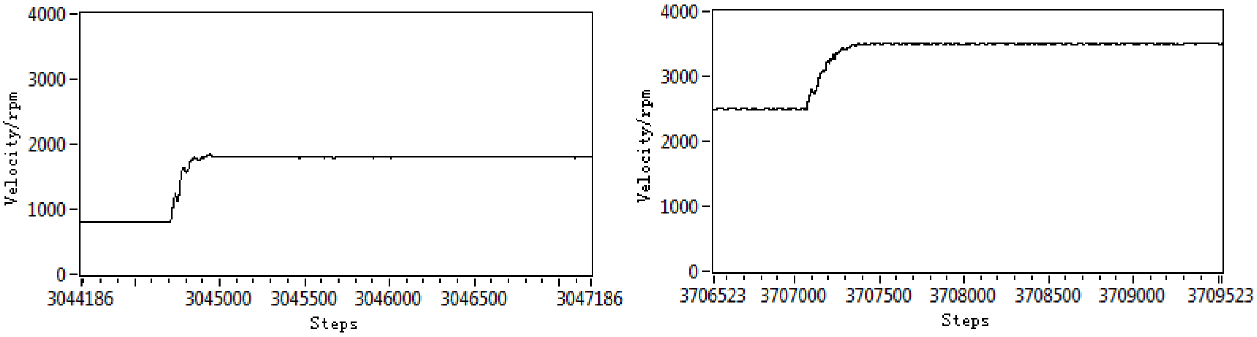

Figure 11. Motor velocity PID response on real devices

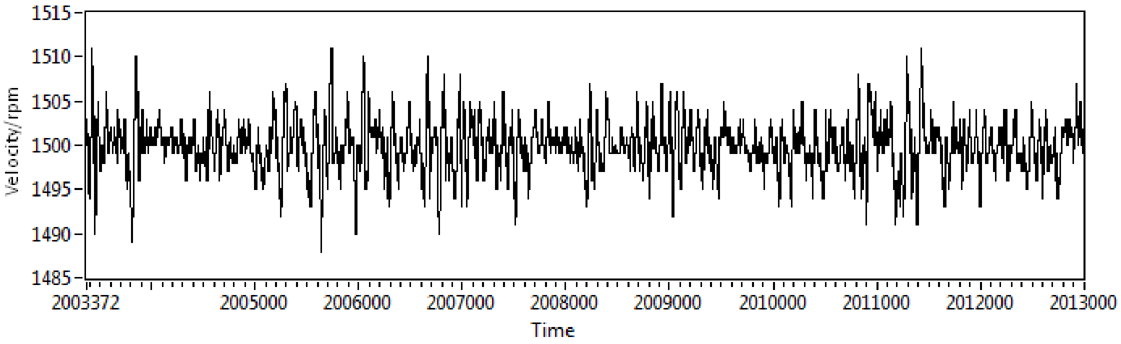

Figure 12. Motor running status at the velocity of 1500rpm

5. Conclusion

In order to make a fast motion control response with an acceptable accuracy, a control process which takes "Good Gain" PID tuning method as the core idea is studied. Based on "Good Gain" method, the motor model and PID control workflow are simulated in LabVIEW environment to get theoretical PID gains and control performance. Experiments result on real devices indicate that the PID gains calculated from this "Good Gain" tuning method can make a fast response of the motion control. During the control process and, its stationary-state error is less than 0.8% which is an acceptable accuracy. So, the fast response and stability characteristics of this method are proved.

References

[1] Zhang Lihong, Zang Xiaohui. Analysis and Simulation of DC Motor Control System Based on Fuzzy-PID Control. *Computer Simulation*. 2009; 26(1): 154-161.

[2] Xu Fan, Fanlin Meng. Research of Brushless DC Motor Simulation System Based On RBF-PID Algorithm. *Second International Symposium on Knowledge Acquisition and Modeling*. 2009; 277-280.

[3] Triwiyatno Aris, Nuh Mohammad, Santoso Ari. Engine torque control of spark ignition engine using fuzzy gain scheduling. *Telkomnika*. 2012; 10(1): 83-90.

[4] Chen, Shaomian, Zhao, Jun, Qian, Jixin. A design method of bang-bang and PID integrated controller based on rough set. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*. 2007; 2: 529-533.

[5] Li, Jiejia, Liu, DaiYan, Qu, Rui. The strategy of construction equipment energy-saving control. *Telkomnika*. 2012; 10(4): 706-712.

[6] Tan Guanzheng, Hu Shengyuan. Parameter on-line self-tuning fuzzy controller based on genetic algorithm. *Journal of CENT South University Technology*. 2003; 34(3): 285-289.

[7] Finn Haugen. Basic Dynamics and Control. Porsgrunn: TeachTech. 2010.

[8] Erick L. DC Motor with Inertia Disk Model Development, Proportional Controller, and State Feedback Controller with Full State Estimator. Madison: Oberstar Consulting. 2002.

[9] Ziegler JG, Nichols NB. Optimum Settings for Automatic Controllers. *ASME*. 1942; 64: 759-768

[10] Fakhrulddin H Ali, Mohammed Mahmood Hussein, Sinan M.B. Ismael. LabVIEW FPGA Implementation of a PID Controller for D.C. Motor Speed Control. *Electrical and Electronic Engineering*. 2010; 2(6): 139-144.

[11] XU Hui, Xiao Gonghai. Brushless DC motor control system based on CompactRIO. *Electronic Design Engineering*. 2011; 19(11): 128-131.