工学硕士学位论文

基于数据驱动方法的金线焊线机的控制器参数自整定方法

Controller Parameters Tuning for Wire Bonder based on Data-driven Method

姜海蓉

哈尔滨工业大学

2010 年 12 月

国内图书分类号：TP273

国际图书分类号：621

工学硕士学位论文

基于数据驱动方法的金线焊线机的控制器参数自整定方法

硕 士 研究生： 姜海蓉

导 师： 李泽湘 教授

申 请 学 位： 工学硕士

学 科 、 专 业： 控制科学与工程

所 在 单 位： 深圳研究生院

答 辩 日 期： 2010 年 12 月

授予学位单位： 哈尔滨工业大学

Classified Index: TP273

U.D.C: 621

Dissertation for the Master Degree of Engineering

Controller Parameters Tuning for Wire Bonder based on Data-driven Method

Candidate:	Hairong Jiang
Supervisor:	Prof. Zexiang Li
Academic Degree Applied for:	Master of Engineering
Specialty:	Control Science and Engineering
Affiliation:	Shenzhen Graduate School
Date of Defence:	December, 2010
Degree-Conferring-Institution:	Harbin Institute of Technology

摘要

运动控制系统广泛应用于工业中的各个领域，如半导体封装，数控机床，工业机器人等。为了满足对运动控制系统高速高精度的要求，路径优化，插补算法，控制算法以及机械装配等方面的研究都至关重要。其中，针对控制算法的研究尤其关键。

本文首先针对一类运动控制系统进行了机理建模和辨识，为控制器的设计提供了必要的信息。针对辨识中存在的输入激励不够的问题，本文采取一种有效的开关机制予以解决。PID 控制器广泛应用于工业领域，但人工整定参数不仅效率低下，而且在控制对象发生变化的情况下不能得到良好的控制性能。本文基于极点配置和零极点相消的原理，提出一种自整定的 PID 控制器，改善系统性能的同时提高了工作效率。另外，为了能够进一步改善系统的跟踪性能，本文在前馈控制器设计的部分采用了自整定 ZPETC。

最后，论文中提到的各种算法都在实际的三轴雕铣床上进行了验证和应用。实验结果表明，这些算法在改善系统动态性能和跟踪性能的同时，大大提高了工作效率，而且由于算法的灵活性，可以很好地应用和推广到实际的工业控制领域中。

关键词　运动控制；系统辨识；自整定控制；PID 控制器；ZPETC。

ABSTRACT

Motion control systems have been applied into almost every field in industry and they play an important role in the development of a country's economy. To satisfy the requirement of high speed and high precision in these systems, many aspects should be considered, among which control part is most flexible with application of control algorithms.

Details of the modeling and identification for a class of motion control systems are given, which offer necessary information for the controller design. In particular, a mechanism is proposed to deal with the low-excitation problem in the identification for self-tuning controller design. Then a self-tuning PID controller based on pole placement and zero-pole cancellation is proposed to improve the autonomous performance and work efficiency because it keeps simplicity of a general PID controller and at the same time avoids problems brought by manual tuning. Furthermore, to improve the tracking performance, which is an important index in motion control systems, a self-tuning ZPETC is utilized for the feedforward part.

At last, all of these algorithms mentioned above are verified in a real milling machine and experiments results indicate their validity. The proposed algorithms can be easily applied into practical motion control systems and improve their performance and work efficiency.

Keywords: Motion control; Identification; Self-tuning control; PID controller; ZPETC

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Research Background and Meaning

Automatic control is the scientific discipline that employs methods from mathematics and engineering in order to force dynamical systems to be in a desired fashion. The first application of feedback control goes as far back as the invention of a float regulator by Greek Ktesibos in times antiquity [1]. In recent years, the increase in the demand for higher accuracy, economical benefits, increased safety, reduced energy consumption, and so on, has made feedback loops inevitable in almost every part of our daily lives. Typical examples of systems that use feedback loops are vehicles, consumer electronics, aircraft, power plants and chemical processes.

At the turn of the century, competitiveness in the global economy remains the same and the need for rapid and flexible manufacturing has become standard practice. Currently (1996), there are about 4×10^{12} wires bonded per year on the planet. Most are used in the approximately 40 to 50 billion ICs produced, but many more are in transistors, LEDs, etc. The infrastructure is so extensive that no other chip-interconnection method can displace wire bonds in the foreseeable future [2].

The key components of the wire bonding machine is the XY-table and the control of the Z direction, however, all these components are controlled using the PID controller or adding the velocity and the acceleration feed-forward. It has long been recognized that industrial control is one of the key technologies to make existing processes economically competitive. In theory, sophisticated control strategies-supervisor, adaptive, model predictive control-should be the norm of industrial practice in modern plants. Unfortunately, a recent survey, by Desborough and Miller has shown otherwise. This indicates that 97% of regulatory controllers are of the proportional-integral-derivative (PID) type and only 32% of the loops show “excellent” or “good” performance [3].

The first systematic methods based on the use of Bode and Nyquist plots were developed and applied to the design of amplifiers in the 1930s and 1940s [4-5]. In 1960, Kalman published his seminal papers that introduced state-space methods along with the design equations for the linear quadratic regulator and the discrete Kalman filter [6-7]. These papers set the stage for an extraordinary development of model-based control design methods. A few years later, in the field of system identification, the paper [8] appeared that sparked the development of the prediction error (PE) framework [9]. These identification techniques provided reliable models allowing the applicability of model-based control design to a wide range of dynamical systems and processes. For a long time, identification and control design were considered separately within the framework of model-based control. The dominant idea was to identify the best possible model and then design the controller based on the basis of that model. Then, the paradigm “goal-oriented identification” emerged in the system identification community [10-11]. However, the identified model is just an approximation of the “true system” with the model uncertainty. The controller based on this model may not get a good performance. Based on this consideration, a new research “iterative identification and control” started to develop around 1990. Adaptive and iterative control algorithm based on explicit criterion minimization begins their way and the iterative feedback tuning (IFT), is present [12]. The data-driven methods have been brought back into focus in the mid-1990s. Data-driven methods include Virtual Reference Feedback Tuning (VRFT) [13], Iterative Feedback Tuning and Correlation based Tuning (CBT) [14-15]. Usually it uses the Simultaneous Perturbation Stochastic Approximation control appeared in this context. It belongs to the class of stochastic approximation algorithm that is typically used for finding roots in the presence of noisy measurements.

1.2 Features of golden wire bonding system

The golden wire bonding machine is a typical motion control system [16], and a motion control system usually should meet the requirement of high precision, high speed and high reliability. The feature of a golden wire bonding system is shown below:

- The systems' inherent time delays limit their achievable bandwidth;
- The systems' internal limits (such as current and voltage) mean they are highly non-linear;
- The required tracking error is extremely low;
- To meet modern manufacturing requirements systems must exhibit a high degree of repeatability and reliability.

A motion control system usually has three loops, the current loop, the velocity loop and the position loop. Fig.1.1 shows the three-loop control architecture. The current loop of the system has already been tuned by the driver, we can not tune it and the velocity loop of the system is tuned using the driver, it is not too hard to do this work. When it comes to the position loop, the work becomes a little bit hard, we should find some algorithm to tune the parameters of the controller, that is also what we are concerning about.

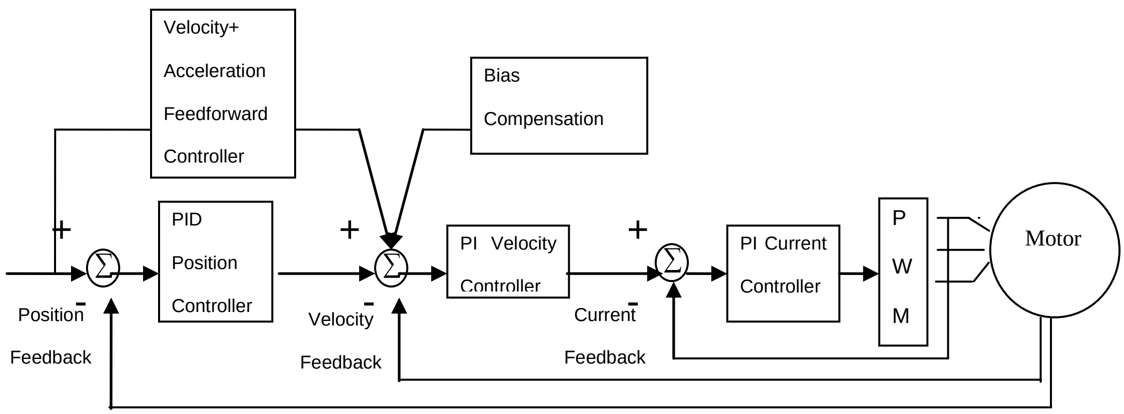


Fig. 1.1 Three-loop control architecture

1.3 Control in industry

Industrial feedback motion controllers are tuned using annual loop shaping, according to certain design rules [17]. It is interesting to note that more than half of the industrial controllers in use today utilize PID or modified PID control schemes. Because most PID controllers are adjusted on-site, many different types of tuning rules have been proposed in the literature. Using these tuning rules, delicate and fine tuning rules have been proposed in the literature. Using these tuning rules, delicate and fine tuning of PID controllers can be made on-site. Also automatic tuning methods have been developed and some of the PID controllers may possess on-line automatic tuning capabilities. Modified forms of PID control, such as I-PD control and two-degrees-of-less switching (from manual operation to automatic operation) and gain scheduling are commercially available.

The usefulness of PID controls lies in their general applicability to most control systems. In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used, PID controls prove to be most useful. In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control [18].

1.4 Motivation and objective

As mentioned above, the traditional PID + VA feedforward controller is applied in about 90% loops in industry for its simplicity [19] and it offers yet most efficient implemented in real time. However, they usually have specific constraints and this defect limits their application.

There are a lot of algorithms creating for tuning the parameters of the loops of the motion control system. Even though one of the main purposes of feedback is to make the closed loop system insensitive to the process, some process knowledge is required to be able to design a control system which guarantees stability and also provides good performance. Hence, modeling and means to map experimental information into the controller (such as identification followed by model based control design) are essential ingredients in any feedback control design procedure. An iterative is used to solve this problem by means of separate identification and control design. In each identification step, the previously designed controller is used to obtain new data from the plant. Then, the controller is designed on the basis of the model obtained in the identification step [20].

1.5 Content of the Research

This dissertation is organized as follows:

- The data-driven methods will be introduced in Chapter 2. In particular, VRFT, NICBT, ICBT, IFT and the control criteria based on data-driven methods are used to tune the parameters of the controller. Then the simulation results of these methods are shown in this chapter.

- The system composition of the golden wire bonding system is introduced in Chapter 3. The golden wire bonding system is a very complex system. My focus is only on the XY table and Z axis. The actuator of XY table is linear motor, while the actuator of Z axis is a linear motor. It is driven by the drive and controlled by the motion control card.
- In Chapter 4. I use the data-driven methods to tune the controller parameters of the actual system. The input signals are carefully selected to meet the requirements of the data-driven methods. The tuning results are shown in this chapter.
- At last, conclusions will follow in Chapter 5, which include some results and the future work.

CHAPTER 2

DATA-DRIVEN METHODS

In many practical control applications, before designing the controller, a mathematical description of the plant should be available, and the controller has to be designed on the basis of measurements. The classical model-based approach suggests a three-step procedure: (1) identify a plant model and evaluate it using the data (2) compute a high-order controller that minimizes the criterion and stabilizes the plant model (3) reduce the controller order by standard controller order reduction techniques. However, it is not easy to get an accurate model of the system so as to get an optimal controller. Data-driven controller tuning approaches try to lump these three steps and present a direct "data-to-control" algorithm. Consequently, in this thesis, data-driven method is used to tune the controller parameters for the wire bonder.

2.1 Virtual Reference Feedback Tuning Method

Virtual Reference Feedback Tuning (VRFT) is yet a data-driven method that appeared recently. The original concept behind VRFT is introduced in [21]. This concept is further developed in [22-27], where implementation aspects have been addressed and a data pre-filter has been designed in order to match the VRFT and model-following criteria. This approach aims at solving a model reference control problem without iteration, i.e. by using a single set of measurements.

The idea of VRFT is to let a set of input-output data, say (u_m, y_m), be measured on a noise-free system (it does not matter whether data are collected in open-loop or closed-loop operation) and assume that a reference model is defined. A reference signal r_m would give the measured output y_m if applied to the reference model can be constructed. This reference signal is called "virtual" because it is not used in the actual generation of the output y_m. Moreover, it is possible to compute the virtual tracking error defined as the difference between the virtual reference and y_m. The idea is to compute a controller that generates u_m when fed by the virtual tracking error. By suitable filtering of the virtual tracking error and u_m, the transfer function of the feedback loop consisting of the calculated controller and the unknown plant is equal to the given reference model, provided that the controller is of appropriate order. Observe that the task of calculating the controller reduces to a simple identification problem. VRFT uses an instrumental variable method to counteract the effect of the noise. An interesting application of this one-shot method consists of providing initial controllers for iterative algorithms intended to perform the “fine-tuning” of the controllers

2.1.1 Algorithm of the VRFT method

The main idea of virtual reference feedback tuning method can be implemented by 3-step algorithm. Given a set of measured I/O data $\{u(t), y(t)\}_{t=1,\ldots N}$, the 3-step algorithm is shown below:

1) Calculate the virtual reference $\overline{r(t)}$, since we suppose that

$$\overline{r(t)} = M\left(z\right)^{-1} y\left(t\right) \tag{2-1}$$

where M(z) is the reference model. Calculate the corresponding tracking error

$$e\left(t\right) = \overline{r(t)} - y\left(t\right) \tag{2-2}$$

(suppose $M(z) \neq 1$, otherwise e(t)=0);

2) Filter the signals e(t) and u(t) with a suitable filter L(z):

$$e_L(t) = L(z)e(t) \tag{2-3}$$

$$u_L(t) = L(z)u(t) \tag{2-4}$$

3) Select the controller parameter vector; say that minimizes the following criterion:

$$J_{VR}^{N}(\theta)=\frac{1}{N}\sum_{t=1}^{N}(u_L(t)-C(z;\theta)e_L(t))^2 \tag{2-5}$$

Note that when $C(z;\theta)=\beta^T(z)\theta$, criterion can be given by the form :

$$J_{VR}^{N}(\theta)=\frac{1}{N}\sum_{t=1}^{N}\left(u_L(t)-\varphi_L^T(t)\theta\right)^2 \tag{2-6}$$

$$\varphi_L(t)=\beta(z)e_L(t) \tag{2-7}$$

And the parameter vector is given by

$$\hat{\theta}_N=\left[\sum_{t=1}^{N}\varphi_L(t)\varphi_L(t)^T\right]^{-1}\sum_{t=1}^{N}\varphi_L(t)u_L(t) \tag{2-8}$$

4) A suitable selection of the pre-filter L(z) is :

$$|L|^2=\frac{|1-M|^2|M|^2|W|^2}{\Phi_u} \quad \forall\omega\in[-\pi;\pi] \tag{2-9}$$

Where Φ_u can be estimated using many different techniques, among which a high-order AR or ARX model

2.1.2 Apply least square method with constrain to VRFT method

Least Square method has already been used to the virtual reference feedback tuning method by M. C. Campi, A. Lecchini, and S.M. Savaresi as shown in (2-8). However, in the practical application, we may get the parameters we don't need when applying the least square method without constrain to virtual reference feedback tuning method. For example, we may get negative parameters when we tune the parameters of the PID controller using virtual reference feedback tuning method by least square method without constrain. These parameters may be applicable in simulation, but dose not practical in the industrial application. In addition, when it comes to the physical world, there must be a limit for most of the parameters, that is why a least square method with constrain is needed to the VRFT method.

$$\min_x f(x) \tag{2-10}$$

Subject to:

$$c(x) \le 0 \tag{2-11}$$

$$ceq(x) = 0 \tag{2-12}$$

$$A \cdot x \le b \tag{2-13}$$

$$Aeq \cdot x = beq \tag{2-14}$$

$$lb \le x \le ub \tag{2-15}$$

Where x, b, beq, lb, and ub are vectors, A and Aeq are matrices, c(x) and ceq(x) are functions that return vectors, and f(x) is a function that returns a scalar. f(x), c(x), and ceq(x) can be nonlinear functions.

2.1.3 Simulation results of VRFT method

The simulation is implemented in MATLAB. The program is separated into three parts: the application program, the core program and the subprogram for calculation. In the simulation, I will show the comparison between the VRFT method without constrain and the VRFT method with constrain, and also the VRFT method without filter and the VRFT method with filter. Suppose we tune the PID controller for the system we choose and the input signal is chosen as 9 order PRBS signal. In the practical industrial application, we know that the three parameters of PID are always chosen to be positive numbers. So in the simulation, we constrain the three parameters to be equal to or bigger than zero, ie. $Kp \ge 0, Ki \ge 0, Kd \ge 0$.

(1) Choose the model of the system as:

$$G(s) = \frac{1}{s^2 + 5s + 4} \tag{2-16}$$

The discrete transfer function of the desired model is shown below:

$$\mathrm{M}(z^{-1})=\frac{0.2162+0.4324z^{-1}+0.2162z^{-2}}{1-0.4182z^{-1}+0.283z^{-2}} \tag{2-17}$$

In this example, Choose the settling time 0.5s and the overshoot 10%, the sampling time is 0.1s. VRFT method without constrain and VRFT method with constrain get the same result, because the PID parameters are all positive. The results are shown below:

(a) Filter not used (with or without constrain):

$$K_p=55.2912,\ K_i=4.8242,\ K_d=82.7656$$

(b) Filter used (with or without constrain):

$$K_p=62.2541,\ K_i=3.6815,\ K_d=76.9414$$

In Fig.2.1 and Fig.2.2, the left picture shows the step response and bode diagram of the desired system and the actual system when the optimal filter is not used, while the right picture shows the step response and bode diagram of the desired system and the actual system when the optimal filter is used.

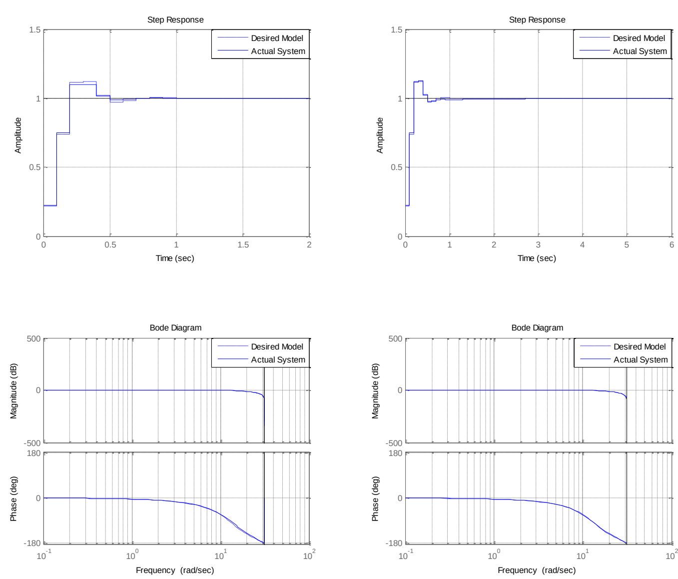


Fig. 2.1 Least square method without constrain

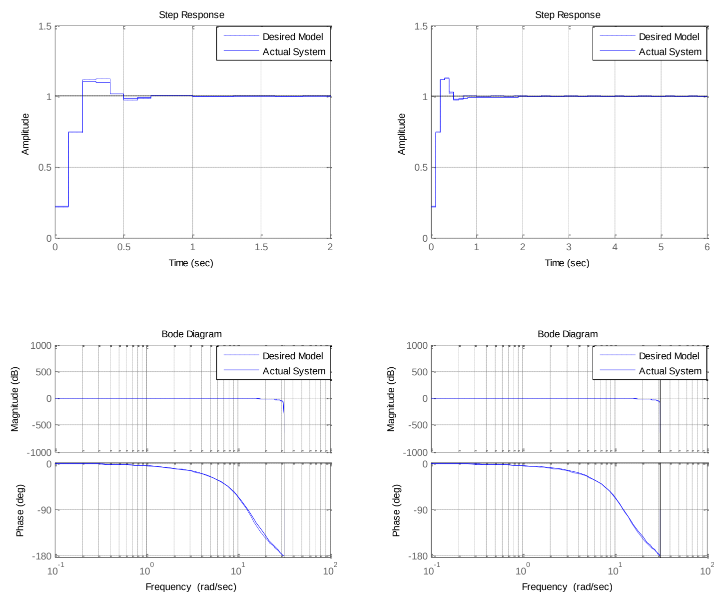


Fig. 2.2 Least square method with constrain

(2) Choose the model of the system as:

$$G(s) = \frac{s+2}{s^2+5s+4} \tag{2-18}$$

The discrete transfer function of the desired model:

$$\mathrm{M}(z^{-1}) = \frac{0.2162+0.4324z^{-1}+0.2162z^{-2}}{1-0.4182z^{-1}+0.283z^{-2}} \tag{2-19}$$

In this simulation, we still choose the settling time 0.5s and the overshoot 10%, the sampling time is 0.1s. VRFT method without constrain and VRFT method with constrain get different results, because the PID parameters are not all positive. The results are shown below:

(a) Optimal filter not used (shown in the left picture in Fig.2.3):

K_p = 12.013, K_i= 2.4153, K_d= -6.6112

(b) optimal filter used(shown in the right picture in Fig.2.3):

K_p=7.5368, K_i=3.7625, K_d= -5.0762

The step response and the bode diagram are shown in Fig.2.3. We can see that K_d is negative, though it is meaningful in the simulation, we can not apply this parameter to the practical industrial application. So we apply the least square method with constrain $Kp \geq 0, Ki \geq 0, Kd \geq 0$ to the VRFT method, and get the result

(c) optimal filter not used (shown in the left picture in Fig.2.4):

$$K_p=0,\ K_i=2.3886,\ K_d=0$$

(d) optimal filter used (shown in the right picture in Fig.2.4. The parameters we get are meaningful to the practical system) :

$$K_p=1.4892,\ K_i=8.3737,\ K_d=0$$

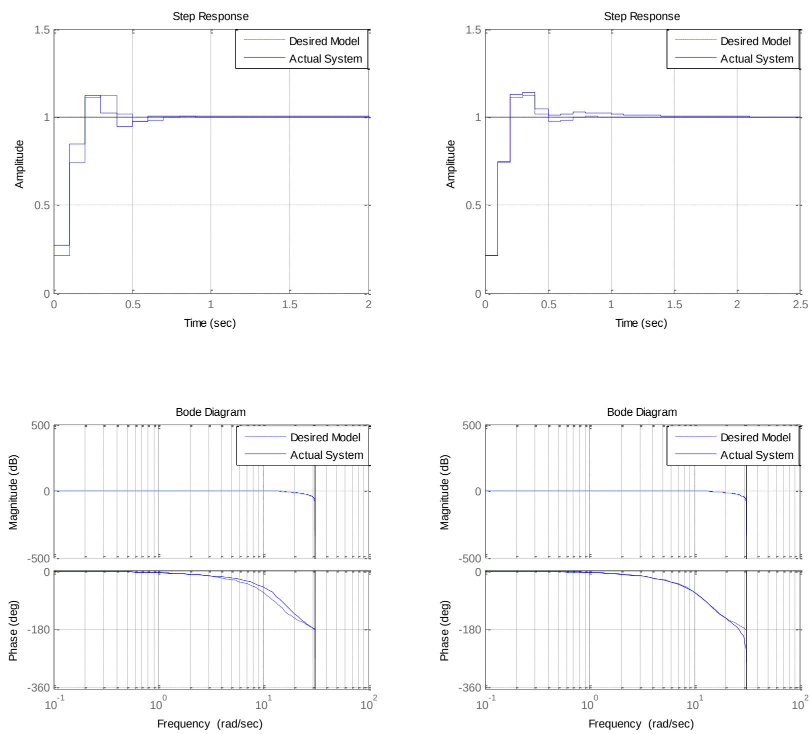


Fig. 2.3 Least square method without constrain

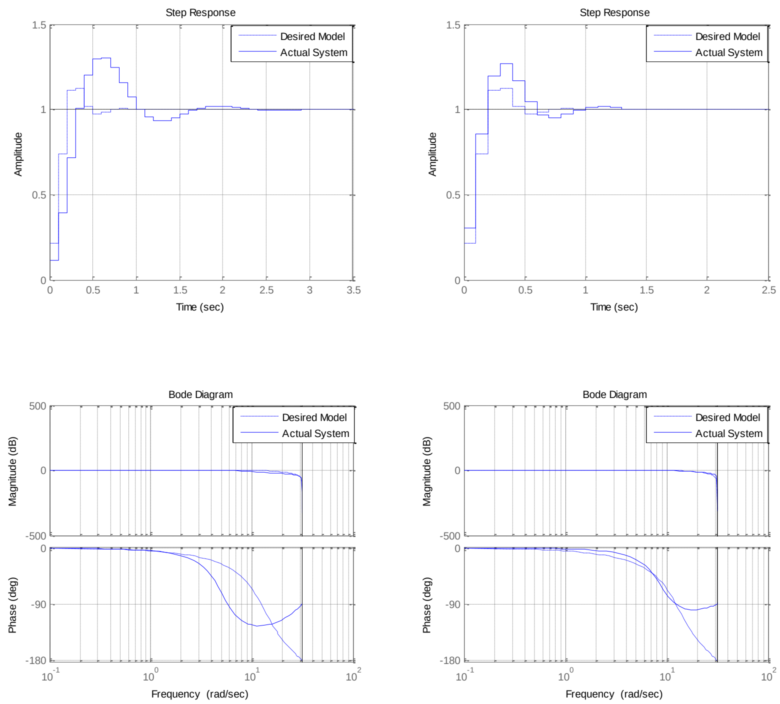


Fig 2.4 Least square method with constrain

From Fig.2.1, Fig.2.2, Fig.2.3 and Fig.2.4, we can see that the tracking effect of the VRFT methods without constrain is better than the VRFT method with constrain. However, to meet the requirement in the practical industrial application, we should apply the least square method with constrain to the VRFT method to get the parameters we want.

2.2 Non-iterative correlation tuning method

The method is based on the correlation approach, uses a single set of input/output data from open loop or closed loop operation. A specific choice of instrumental variables makes the correlation criterion an approximation of the model reference control criterion. The controller parameters and the correlation criterion are asymptotically not affected by noise. In addition, based on the small gain theorem a sufficient condition for the stability of the closed-loop system is given in terms of the infinity norm of a transfer function. An unbiased estimate of this infinity norm can be obtained as the solution to a convex optimization problem using an infinite number of noise-free data. It is also shown that, for noisy data, the use of the correlation approach can improve significantly the estimate.

2.2.1 Algorithm of the Non-iterative correlation tuning method

Non-iterative correlation-based tuning method is introduced in details in [28]. This method is used to solve the model reference control problem; the diagram is shown in Fig.2.5.

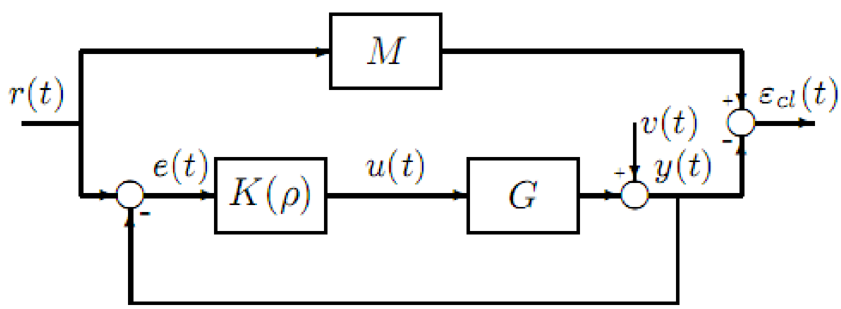


Fig. 2.5 Model reference control problem

The control criterion is defined by:

$$J_{mr}(\rho)=\left\|M-\frac{K(\rho)G}{1+K(\rho)G}\right\|_2^2 \tag{2-20}$$

Let the reference model $M(q^{-1})$ be expressed as:

$$M(q^{-1})=\frac{K^*(q^{-1})G(q^{-1})}{1+K^*(q^{-1})G(q^{-1})} \tag{2-21}$$

Where $K^*(q^{-1})$ is the ideal controller. Then (23) can be approximated by:

$$J(\rho)=\left\|W[M-K(\rho)G(1-M)]\right\|_2^2 \tag{2-22}$$

Where W is an approximate weighting filter. By this equation the diagram of the model reference control problem becomes Fig.2.6.

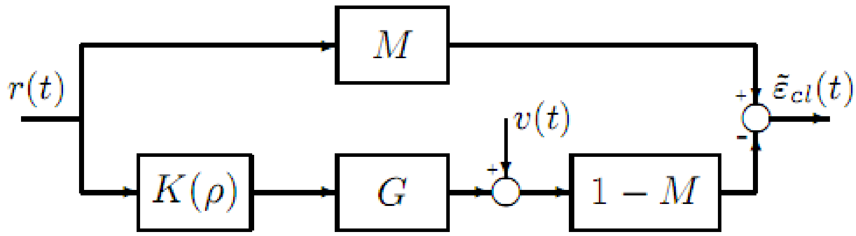


Fig. 2.6 Approximation of model reference control problem

Non-iterative correlation-based tuning in [28] minimizes the control criterion through correlation approach. Let the correlation function be defined as:

$$f(\rho) = E\{\zeta_{\omega}(t)\varepsilon(\rho,t)\} \tag{2-23}$$

$$\zeta_{\omega} = \left[r_{\omega}(t+l), \quad r_{\omega}(t+l-1), \quad \dots \quad r_{\omega}(t), \quad r_{\omega}(t-1), \quad \dots \quad , r_{\omega}(t-l)\right]^{T} \tag{2-24}$$

with

$$r_{\omega}(t) = W(q^{-1})r(t) \tag{2-25}$$

and l is a sufficiently large integer. The optimal controller parameters minimize the correlation criterion defined as the two-norm of the correlation function:

$$\hat{\rho} = \arg\min_{\rho} J_c(\rho) = f^T(\rho)\, f(\rho) \tag{2-26}$$

For a finite number of data N, the correlation function can be estimated as:

$$\hat{f}(\rho) = \frac{1}{N}\sum_{t=1}^{N}\zeta_{\omega}(t)\varepsilon(t) = \frac{1}{N}\sum_{t=1}^{N}\zeta_{\omega}(t)\left[y_d(t) - \phi^T(t)\rho\right] \tag{2-27}$$

The objective function can be written as:

$$J_c(\rho) = f^T(\rho)\, f(\rho) \tag{2-28}$$

Thus, the optimal controller parameters minimizing the correlation criterion can be obtained by the standard-least squares algorithm:

$$\hat{\rho} = \left(Q^T Q\right)^{-1} Q^T Z \tag{32}$$

Where

$$Q = \frac{1}{N}\sum_{t=1}^{N} \zeta_{\omega}(t)\phi^T(t) \tag{2-29}$$

$$Z = \frac{1}{N}\sum_{t=1}^{N} \zeta_{\omega}(t) y_d(t) \tag{2-30}$$

and Q^TQ is nonsingular.

The filter is given by:

$$W(e^{-j\omega}) = \frac{1 - M(e^{-j\omega})}{\Phi_{ur}(\omega)} \tag{2-31}$$

2.2.2 Simulation results of non-iterative correlation tuning method

The simulation is implemented in MATLAB. The program is separated into three parts: the application program, the core program and the subprogram for calculation.

(1) Choose the model of the system:

$$G(s) = \frac{1}{s^2 + 4s + 5} \tag{2-32}$$

The discrete transfer function of the desired model:

$$\mathrm{M}(z^{-1}) = \frac{0.08207 + 0.1641z^{-1} + 0.08207z^{-2}}{1 - 1.127z^{-1} + 0.4556z^{-2}} \tag{2-33}$$

The overshoot is 10%, the settling time is 10 times the sampling time.

We get the PID parameters of the controller by the non-iterative correlation-based tuning method. Kp= 8.4612, Ki=1.5077, Kd=6.9021. The step response and bode diagram of the desired model and the actual system are shown in Fig.2.7.

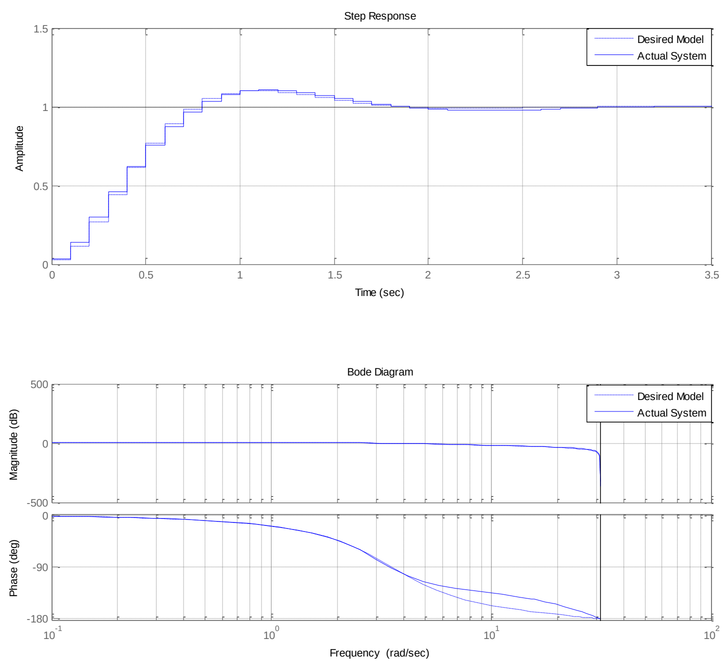


Fig. 2.7 Non-iterative correlation-based tuning method

(2) Choose the model of the system:

$$G(s) = \frac{s+2}{s^2+4s+5} \tag{2-34}$$

The discrete transfer function of the desired model:

$$\mathrm{M}(z^{-1}) = \frac{0.08207+0.1641z^{-1}+0.08207z^{-2}}{1-1.127z^{-1}+0.4556z^{-2}} \tag{2-35}$$

The overshoot is 10%, the settling time is 10 times the sampling time.

We get the PID parameters of the controller by the non-iterative correlation-based tuning method. Kp= 0.4069, Ki=0.7539, Kd=-0.5988. The step response and bode diagram of the desired model and the actual system are shown in Fig.2.8. Choose the constrain: $Kp \geq 0, Ki \geq 0, Kd \geq 0$.

Let Kp=0.4069, Ki=0.7539, Kd=0. The step response and the bode diagram are shown in Fig.2.9.

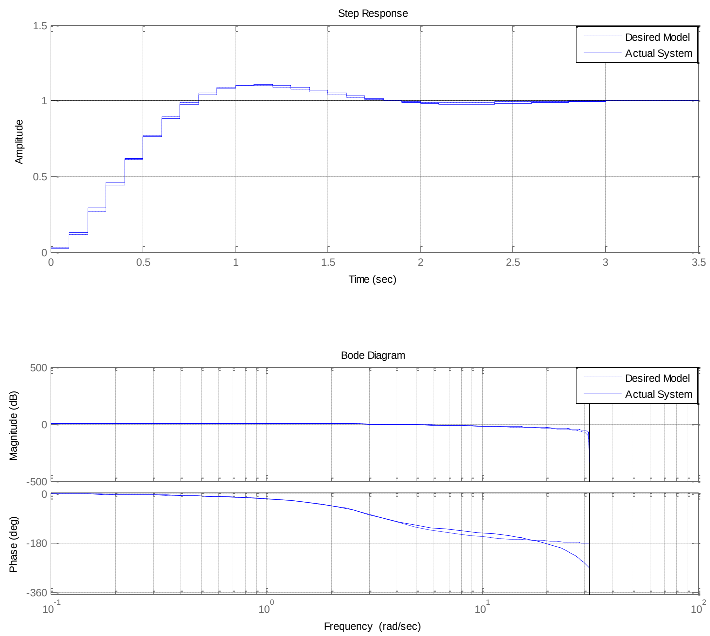


Fig. 2.8 Non-iterative correlation-based tuning method

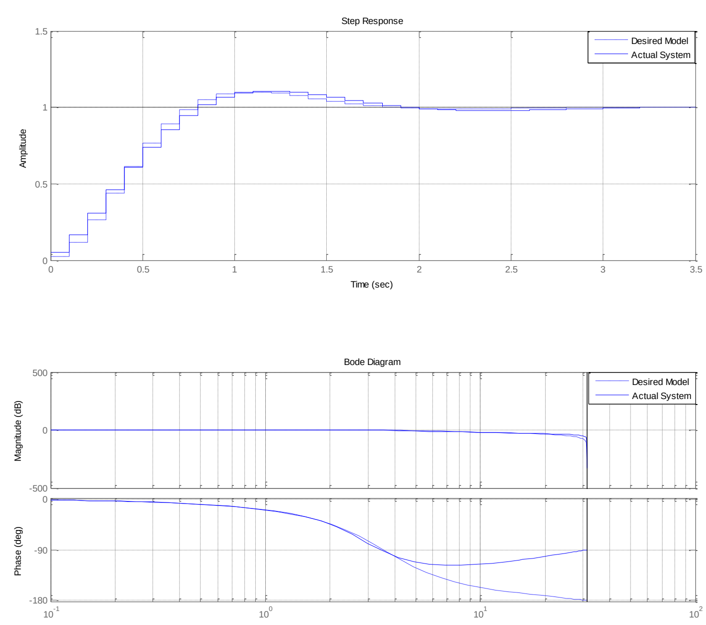


Fig. 2.9 Non-iterative correlation-based tuning method with costrain

2.3 Iterative Correlation-based Tuning Method

The basic idea is to decorrelate the output error between the achieved and designed closed-loop systems by iteratively tuning the controller parameters. At present, there are two different approaches. In the first one, a correlation equation involving a vector of instrumental variables is solved using the stochastic approximation method. It is shown that with an appropriate choice of instrumental variables and a finite number at each iteration, the algorithm converges to the solution of the correlation equation. The second approach is based on the minimization of a correlation criterion. The method is intended for a "fine" tuning of the controller. In other words, it is assumed that the controller obtained in model-based design procedure is in the vicinity of the controller that satisfies all design specifications for the actual system. The controller parameters are calculated as the solution to a cross-correlation equation involving instrumental variables [29-31]. The controller whose parameters are the solution to the cross-correlation equation is called subsequently the decorrelating controller. The diagram of the iterative correlation-based tuning method is shown in Fig.2.10.

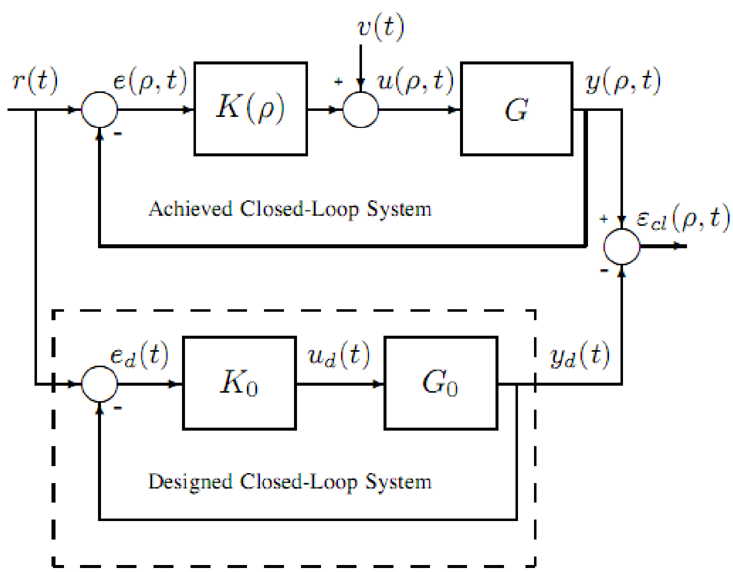


Fig. 2.10 Closed-loop output error resulting from a comparison of the achieved and designed closed-loop systems.

2.3.1 Algorithm of the iterative correlation-based tuning method

The cross-correlation function is as follows:

$$f(\rho) = E\{\overline{f(\rho)}\} \tag{2-36}$$

where E{.} denotes the mathematical expectation and $\overline{f(\rho)}$ is defined as follows:

$$\overline{f(\rho)} = \frac{1}{N}\sum_{t=1}^{N}\zeta(t)\varepsilon_{oe}(\rho,t) \tag{2-37}$$

where N is the number of data and $\zeta(t)$ an n_ζ -dimensional column vector of instrumental variables that are correlated with the reference signal r(t) and independent of the disturbance v(t). The instrumental variable vector $\zeta(t)$ may be a function of the controller parameter vector ρ and will be denoted by $\zeta(\rho,t)$

The controller may be improper or of very high order. Furthermore, it may destabilize the system if the unstable zeros and poles of G are not contained in G_d. On account of this fact, two different situations can be distinguished:

The decorrelating controller K° exists, stabilizes the closed-loop system and belongs to the parameterized set of controllers. Though, this assumption seems to be too restrictive, it allows the parametric convergence and the accuracy of the estimates to be studied. These analyses give important guidelines for the choice of instrumental variables and the improvement of the convergence rate of the iterative algorithm. The parameter vector of this controller ρ°, is evidently a solution of the correlation equation

$$f(\rho) = 0 \tag{2-38}$$

The decorrelating controller K° does not exist or does not belong to the controller set. In this case, the controller parameters can be computed as the minimizing argument of some norm of the correlation function.

A solution of this correlation equation can be found using the following iterative stochastic approximation algorithm:

$$\rho_{i+1} = \rho_i - \gamma_i \overline{f(\rho_i)} \tag{2-39}$$

where γ_i is a positive scalar step size.

2.3.2 Simulation results of the iterative correlation-based tuning method

The simulation is implemented in MATLAB. We choose different models to make a comparison.

(1) Choose the model of the system:

$$G(s) = \frac{1}{s^2 + 3s} \tag{2-40}$$

The initiated parameters we set is:

Kp0=6, Ki0=0.5, Kd0= 0.1,

we choose:

$$\gamma = 0.5, \alpha = 0.9$$

By using the ICBT Method, get the parameters of the controller:

Kp=4.7425, Ki=0.1315, Kd=0.3825

The initiated step response and the step response after ICBT tuning are shown in Fig.2.11.

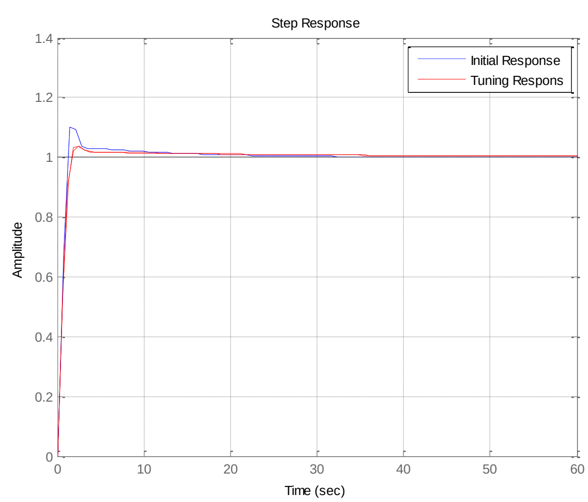


Fig. 2.11 The step response of the system with initiated controller and the tuning controller

(2) Choose the model of the system:

$$G(s) = \frac{1}{s^2 + 4s + 5} \tag{2-41}$$

The initiated parameters we set is:

Kp0=10, Ki0=0.1, Kd0= 0.1

we choose:

$$\gamma = 0.3, \alpha = 0.9$$

By using the ICBT Method, get the parameters of the controller :

Kp=9.6182, Ki=1.9096, Kd=2.8706e-005

The initiated step response and the step response after ICBT tuning are shown in Fig.2.12.

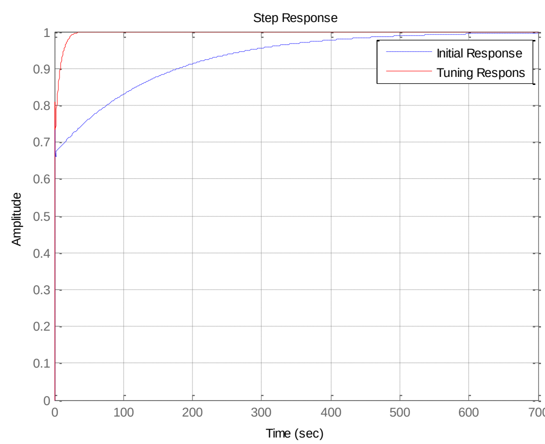


Fig. 2.12 The step response of the system with initiated controller and the tuning controller

(3) Choose the model of the system:

$$G(s) = \frac{s+2}{s^2 + 4s + 5} \tag{2-42}$$

The initiated parameters we set is:

Kp0=20, Ki0=0.1, Kd0= 0.1

we choose:

$$\gamma = 0.1, \alpha = 0.9$$

By using the ICBT Method, get the parameters of the controller:

Kp= 18.6759, Ki= 4.4746, Kd= 1.2876e-004

The initiated step response and the step response after ICBT tuning are shown in Fig.2.13.

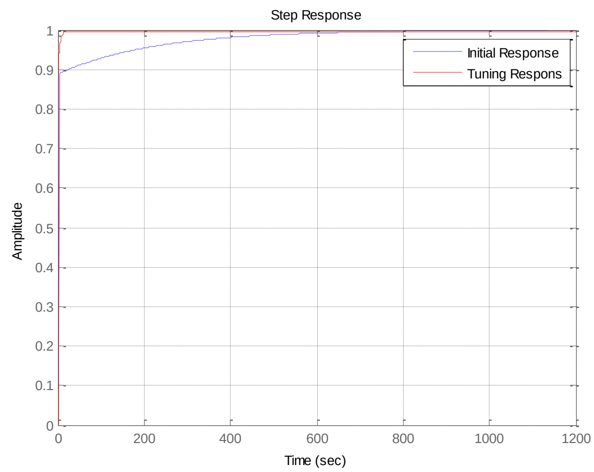


Fig. 2.13 The step response of the system with initiated controller and the tuning controller

2.4 Iterative feedback tuning method

The gradient-based method for the iterative optimization of the performance criterion over the controller parameters has been called Iterative Feedback Tuning (IFT). IFT is a data-based method for the optimal tuning of a low order controller. The tuning of the controller parameters is performed iteratively, using a generalized Robbins-Monro type gradient descent scheme. An update step of the controller parameters is performed at each iteration on the basis of data obtained partly during normal operating conditions and partly from some special experiments. These data come from the closed loop system with the current controller. Iterative Feedback Tuning (IFT) is a model-free data-based method for the optimal tuning of a parameters of a controller of given structure, typically a restricted complexity controller [32-35]. The control performance objective used in IFT is a quadratic performance criterion, which is minimized by a stochastic gradient descent scheme of Robbins-Monro type.

2.4.1 Algorithm of the iterative feedback tuning method

Consider the achieved and designed closed-loop systems shown in figure 25. Here, for simplicity of presentation, the minimization of the following criterion is considered:

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N} E\left\{\varepsilon_{oe}^2(\rho,t)\right\} \tag{2-43}$$

The first derivative of the criterion with respect to the controller parameters is:

$$\frac{\partial J}{\partial \rho} = \frac{1}{N}\sum_{t=1}^{N} E\left\{\varepsilon_{oe}(\rho,t)\frac{\partial y(\rho,t)}{\partial \rho}\right\} \tag{2-44}$$

The minimum of the criterion (1) is attained when the expression (2) is equal to zero. The roots of $\partial J / \partial \rho$ can be found by applying the Robbins Monro stochastic approximation procedure provided that an unbiased estimate of this gradient is available.

Consider now the expression for the output $y(\rho,t)$. The differentiation of $y(\rho,t)$ with respect to the i-th element of ρ gives:

$$\frac{\partial y(\rho,t)}{\partial \rho_i}\Big|_{\rho} = S(\rho,G)G\frac{\partial K(\rho)}{\partial \rho_i}\Big|_{\rho}\left(r(t) - y(\rho,t)\right) \tag{2-45}$$

This expression gives an idea of how to obtain an estimate of $\partial y(\rho,t)/\partial \rho$ by performing two experiments on the actual closed-loop system. That is, perform a first experiment under normal operational conditions and collect measurements of the output $y(\rho,t)$. All signals appearing during this experiment will carry the subscript e1; similarly, the signals from the second experiment will have subscript e2. For example, $y_{e1}(\rho,t)$ denotes the measured output during the first experiment. Then, in the second experiment, form the signal $r(t) - y_{e1}(\rho,t)$ and inject it at the process input. Then collect the output signal $y_{e2}(\rho,t)$ and filter it with the filter $\partial K(\rho)/\partial \rho$ to obtain a realization of $\partial K(\rho)/\partial\rho$. Illustration of this so-called gradient experiment is given in Fig 25. Now, the estimate of $\partial J/\partial\rho$ is calculated as follows:

$$\frac{\partial J}{\partial\rho}(\rho_i)=\frac{1}{N}\sum_{t=1}^{N}\left[\frac{\partial K(\rho)}{\partial\rho}y_{e2}(\rho_i,t)\right]\left(y_{e1}(\rho_i,t)-y_d(t)\right) \tag{2-46}$$

Hjalmarsson and co-workers have shown that this estimate is unbiased. The controller parameters are updated using the following iterative formula:

$$\rho_{i+1}=\rho_i-\gamma_i H_i^{-1}\frac{\partial J}{\partial\rho}(\rho_i) \tag{2-47}$$

Where Hi is some positive-definite matrix. A typical choice for this matrix might be an approximation of Hessian, i.e.

$$H_i=\frac{\partial^2 J}{\partial\rho^2}(\rho_i)=\frac{1}{N}\sum_{t=1}^{N}\left[\frac{\partial K(\rho)}{\partial\rho}y_{e2}(\rho_i,t)\right]\left[\frac{\partial K(\rho)}{\partial\rho}y_{e2}(\rho_i,t)\right]^T \tag{2-48}$$

IFT offers a very nice possibility of obtaining an unbiased estimate of the gradient using only signals collected in closed loop. However, the price to pay is an increased number of experiments. For the case of two-degree-of-freedom SISO controllers, IFT requires three experiments per iteration. In the case of MIMO controllers, the number of experiments per iteration increases to $n_u\times n_y+1$. There have been some attempts to reduce the number of experiments in the case of MIMO systems.

2.4.2 Simulation results for iterative feedback tuning method

(1) Choose the model of the system:

$$G(s)=\frac{1}{s^2+4s+5} \tag{2-49}$$

The discrete transfer function of the desired model:

$$\mathrm{M}(z^{-1})=\frac{0.04223+0.08447z^{-1}+0.04223z^{-2}}{1-1.411z^{-1}+0.5798z^{-2}} \tag{2-50}$$

The overshoot is 10%, the settling time is 15 times the sampling time.

We get the PID parameters of the controller by the non-iterative correlation-based tuning method. Kp= 15.8929, Ki= 14.3852, Kd= 0.9323, the iterative time is 435. The step response and bode diagram of the desired model and the actual system are shown in Fig.2.14.

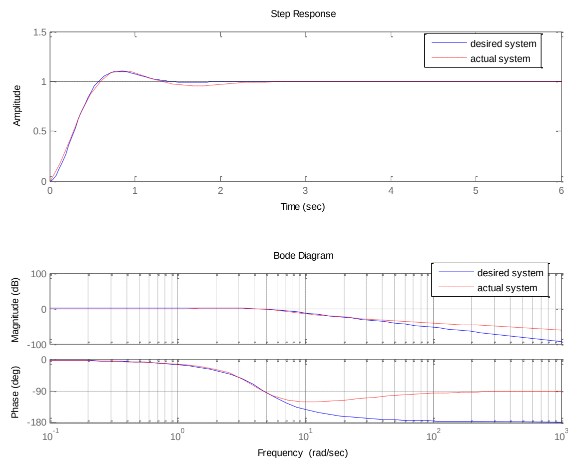


Fig 2.14 Step response and bode diagram of the desired system and the actual system

(2) Choose the model of the system:

$$G(s) = \frac{s+2}{s^2+4s+5} \tag{2-51}$$

The discrete transfer function of the desired model:

$$\mathrm{M}(z^{-1}) = \frac{0.04223 + 0.08447z^{-1} + 0.04223z^{-2}}{1 - 1.411z^{-1} + 0.5798z^{-2}} \tag{2-52}$$

The overshoot is 10%, the settling time is 15 times the sampling time.

We get the PID parameters of the controller by the non-iterative correlation-based tuning method. We can not get a meaningful Kp, Ki, Kd, the algorithm fails. The step response and bode diagram of the desired model and the actual system are shown in Fig.2.15.

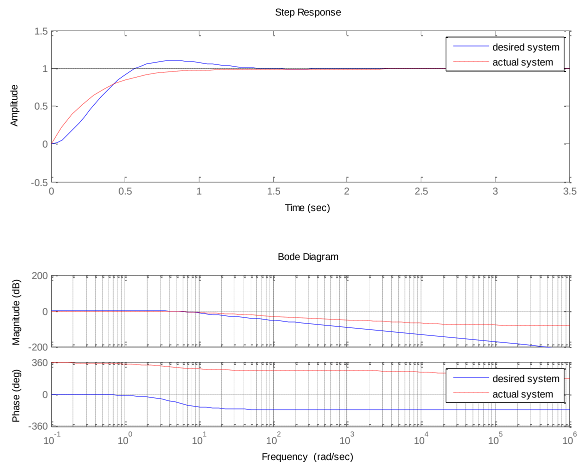


Fig. 2.15 Step response and bode diagram of the desired system and the actual system

2.5 Data-based tuning method based on control criteria

In this section, I will introduce a simple method using the input data and output data getting from the Elmo Drive. Apply the iteration method to tune the controller parameters based on some control criteria which we learn from the classic control system theory.

Elmo drive is a series of very powerful drive in the motion control field and is very widely used in the industrial field. The controller of Elmo drive can be divided into three loops, the current loop, the velocity loop and the position loop. The controller parameters of the current loop are generally tuned by the drive itself, if not necessary, we have no need to tune it by hand. The velocity loop is composed of a PI controller and the position loop is consisted of a P controller, a velocity feedforward and an accelerating feedforward. Though the controller parameters of the velocity loop and the position loop can be auto tuned by the Elmo controller itself, we can not always get a good result with a limit time. While the Elmo drive provides the customers with powerful software called the "Elmo composer" to tune the parameters by hand, it is still a boring work if you have too many drives to tune and the required accuracy is too high for your control system.

By experiments, a 2-order discrete model without dead time is verified to be available for the motion control system and estimated parameters are given, which will be used for the controller design. Besides, other useful information, such as the trace limit in the switched-off mechanism, is also given.

2.5.1 Implementation of the control criteria method

The controller parameters tuning using Elmo Interlude Communication API include the follow processes: communicate with Elmo Drive, set the recorder, get the data from the DSP of Elmo Drive, process the data and auto tune the controller parameters by the controller requirements.

The first step is to make a good communication with Elmo drive using the Elmo Interlude Communication API. In this paper, we just use the RS-232 to communicate with the Elmo drive. The Elmo Interlude is a communication software package designed to enable software developers to program applications for motion control system that integrate Elmo digital servo drives. The software provides an efficient and easy communication channel directly to the connected Elmo drive in order to facilitate integration with the user application. This saves programming time and effort, and ensures compatibility between the use application and the Elmo drive.

The Interlude API is integrated as a DLL directly into the user application, operating as an embedded part of it. It supports the RS232 serial communication and the CiA CANopen communication protocol according to DS301 [36]. By using the Interlude API we can establish (and disconnect) communication with an Elmo drive and send commands to the drive; upload and download applications from/to the drive. Download firmware; create and load a network configuration file; define and activate a callback mechanism for error handling and some other communication with CANopen.

To get the data from the Elmo drive, we should set the recorder with the command of Elmo drive first under a certain sequence. The SimplIQ drive recorder mechanism enables the user to record up eight signals simultaneously. The recorded signals can be uploaded to the host through the communication connection, for presentation and analysis. SimplIQ drives operating with CAN have two communication lines: CAN and RS-232. This arrangement enables the recorder to use one line for performance monitoring while the machine host uses the other line to control the SimplIQ drive in its normal context [37]. In this paper, we just use the RS-232 to communicate with the SimplIQ drive.

The setting process including the following command and they should be set in a certain sequence to make sure we can get the required data. We just introduce the function of them briefly in this paper. To know the details of each command, you can refer to SimplIQ software manual [38]. First, make sure the recorder is killed by setting “RR” command and then set the sampling time. Secondly, mapping the signals we want to record with the “RV” and “RC” commands. Thirdly, define characteristic parameters with the “RP”, “RG” and “RL” commands. Finally, launch the recorder.

We can use the BH command to upload the recorded data by the SimplIQ drive recorder to the host. However, the data is uploaded in hexadecimal form in order to minimize transmission time (relative to ASCII formatted text). In order to analyze the BH record, it is important to understand that the internal representation of quantities in the controller is not in user units. We should transform the hexadecimal form of data into decimal form of data and multiply certain to make the value as what we want. The details can also be known by the SimplIQ software manual. The data we should get from the Elmo Drive is the current command, the current, the speed command, the main speed, the position command, the main position and the position error.

After getting data from Elmo drive, the next thing we should do is to consider how to deal with the data in order to get the information we need and to auto tune the parameters of the controller. In this paper, we just establish some basic functions by VC++ program, and use these functions as a judge to weigh whether a group of parameters of the controller can meet our requirements. These functions are deriving from the control system design textbook which is known by all the people relating to the control field. Although these functions are very easy, they have been tested to be effective and efficient as a judge to our system and is also easy to use and execute. As we all know, before we design a control system, we are confined by some requirements to make sure the control system can have the performance we need or even better. These functions include the calculation of the settling time, overshoot, steady state error, rising time and so forth. We can get the results from the data we get from the recorder. For example, we can get the settling time from the data of position error. We can evaluate the performance of the motion control system driven by Elmo Drive through these functions setting by us.

After the foregoing steps, we have prepared well for the data by the recorder and the evaluation mechanism by the functions we set. Then we can auto tune the parameters as long as we choose a good method to change the parameters in each experiment in the tuning process. In this paper, we tune the velocity loop and the position loop through the method referred by the section 3. There must be many other ways to auto tune the parameters based on the data and the functions we set. This paper only provides other uses an alternative way to tune the parameters of the controller other than using the composer. As to how to make good use of the data and the functions, the condition is quite different. It depends on the complexity of the system and requirements of the user. We just provide a reference in section 3.

2.6 Summary

In this Chapter, several data-driven methods are introduced, including both the non-iterative methods and the iterative methods. The non-iterative methods include the virtual reference feedback method and non-iterative correlation-based tuning method. The iterative methods include the iterative correlation-based tuning method, the iterative feedback tuning method and the data-driven method based on control criteria.

Simulation of these methods is implemented in MATLAB. By the simulation of the result, the validity of these algorithms is proved. By adding constrains to the least square method of the VRFT method, I can get the parameters of the PID controller that can be applied to the actual system.

CHAPTER 3

SYSTEM COMPOSITION

3.1 Golden wire bonding system

The golden wire bonding machine of Parallex Precision co., Ltd is shown in Fig.3.1. It is a very complex system, including the mechanical part, the electronic part, the motion control part and etc. My research focuses on the motion control part. My object is to tune the controller parameters for the all the axises in the golden wire bonding machine, containing the XY axis of the XY table and the Z axis for the wire bonding.

Fig. 3.1 Golden wire bonding machine

3.2 Composition for XYZ axis

X axis and Y axis compose the XY table and the Z axis is the bond head of the golden wire bonding system. My research focus is mainly on tuning the parameters for the controller of the XY table.

3.2.1 XY table

The XY table is shown in Fig 3.2. The XY table is a parallel mechanism, which is the basic component of the golden wire bonding system. The accuracy of the XY table plays key role in the precision of the system. By this means, the parameters tuning for the XY table becomes very important.

Fig 3.2 XY table

3.2.2 Z axis

Fig.3.3 shows the structure of a bond head, which is the Z axis. In the present golden wire bonding system, the Z axis is only controlled by a current loop. The current loop serves as the controller for the open-loop force control.

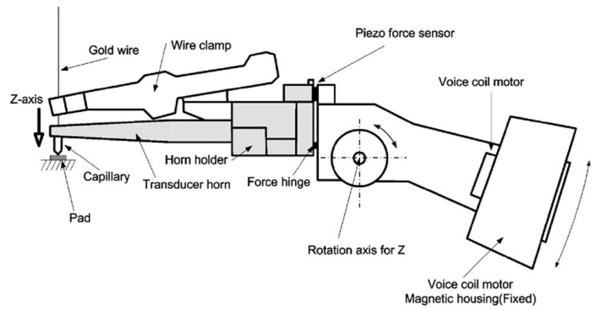


Fig 3.3 Structure of the bond head

3.3 Composition for a single axis

The XY table is composed of two axises and each axis has its own composed elements, including the actuator, the drive, the feedback component and the controller of the motion control card. The control diagram of a single axis is shown in Fig.3.4. The actuator of each single axis is a linear motor and the drive of each single is Elmo Harmonica series drive. The feedback element is the linear optical grating ruler. The motion is controlled by the GTS800 motion control card produced by Googol Technology Co., Ltd. The composition of the actual system is shown in Fig.3.5.

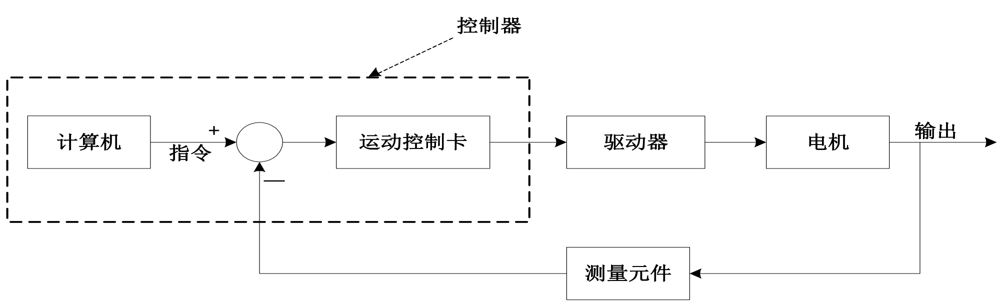


Fig. 3.4 Control diagram of a single axis

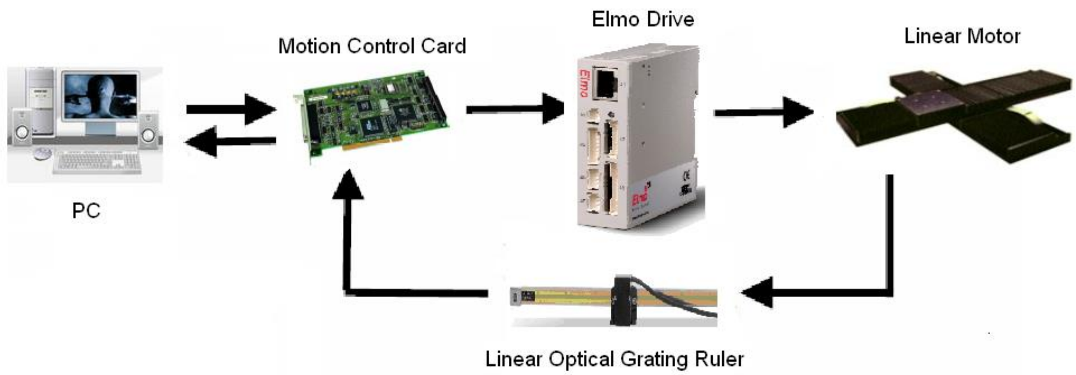


Fig. 3.5 Actual system composition

The system of each axis is composed of three loops as mentioned in chapter 1. They are the current loop, the velocity loop and the position loop. The current loop and the velocity loop are set in the Elmo drive, while the position loop is set in the GTS800 motion control card. The controller parameters of the current loop are tuned by Elmo drive; usually we don't change these parameters. I will only tune the controller parameters of the velocity loop and the position loop by using the methods mentioned in chapter 2.

3.4 Summary

In this chapter, the system composition for my control object is introduced. It is a golden wire bonding system. Although the system is very complex, my research focus is only on the controller parameters tuning for the XY table. Each axis of the XY table is composed of a motor, a drive and the controller of the motion control card. Each axis is controlled by three loops including the current loop, the velocity loop and the position loop. The controller parameters of the velocity loop and the position loop are tuned by the algorithms mentioned in chapter 2.

CHAPTER 4

EXPERIMENT RESULTS

4.1 Input signal

Although we don’t do identification to the motion control system, the selection of the input signal is also very important. Take the VRFT method for an example, though it can tune the parameters for the controller without identify the model of the system. The actual process of it is to identify the parameters of the controller. By this means, it is another way of identification, so an input that contains rich simulated frequency is also needed for the VRFT method. Other data-driven methods mentioned in Chapter 2 also have the same feature, so careful selection for the input signal is also needed.

4.1.1 Step signal

Step signal is the most common signal for the person whose major relates to control. It can be used to evaluate the performance of the system by get the output of a step response. However, we don’t use it often in the real world and we just use it for the simulation. Firstly, the step signal does not have a rich exciting frequency as the PRBS signal, which I will introduce in the following section. Secondly, we can not control the frequency we need by using a step signal as the input as we can do to use the PRBS signal or the chirp signal. In this thesis, I don’t concern the step signal as the input signal.

4.1.2 PRBS signal

PRBS is a periodic, deterministic signal with white-noise-like properties, which is generated by the difference equation

$$u(t) = rem(a_1 u(t-1) + ... + a_n u(t-n), 2) \tag{4-1}$$

Here $rem(x,2)$ is the remainder as x is divided by 2, thus u(t) only assumes the values 0 and 1. And the sequence u(t) is periodic with a period of at most 2^n-1. It's trivial that for each n the actual period will depend on choices of [$a_1 \ldots a_n$] and the order n should be selected specifically for different occasions. In the region as expressed in (4-2), the PRBS behaves like the "periodic white noise" [39-41], where N is the period of u and Δt is the sampling time. Then for systems with different bandwidths, the order can be determined.

$$\omega = \frac{2\pi}{N\Delta t} \sim \frac{2\pi}{3\Delta t} \tag{4-2}$$

The time domain 9 order PRBS signal (positive and negative) is shown in Fig.4.1.

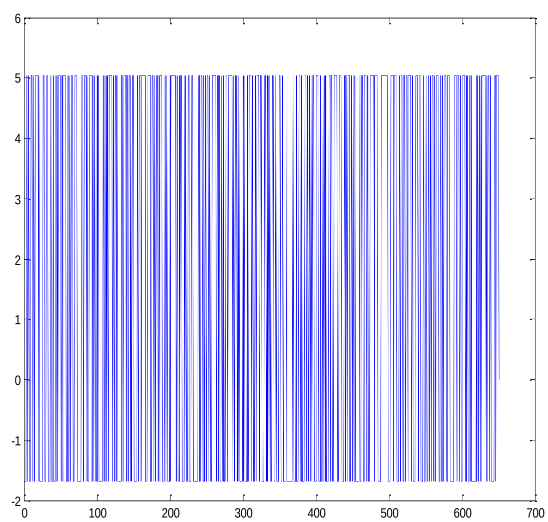


Fig. 4.1 PRBS signal

4.1.3 Chirp signals or Swept Sinusoids

A chirp signal is a sinusoid with a frequency that changes continuously over a certain band Ω: $\omega_1 \le \omega \le \omega_2$ over a certain time period $0 \le t \le M$:

$$u(t) = A\cos(\omega_1 t + (\omega_2 - \omega_1)t^2 / (2M)) \tag{4-3}$$

The "instantaneous frequency" ω_i in this signal is obtained by differentiating the argument w.r.t. time t:

$$\omega_i = \omega_1 + \frac{t}{M}(\omega_2 - \omega_1) \tag{4-4}$$

and we see that it increases from ω_1 to ω_2. This signal has the same crest factor as a pure sinusoid, i.e., $\sqrt{2}$, and it gives good control over the excited frequency band. Due to the sliding frequency, there will however also be power contributions outside the band Ω. The time domain chirp signal is shown in Fig.4.2.

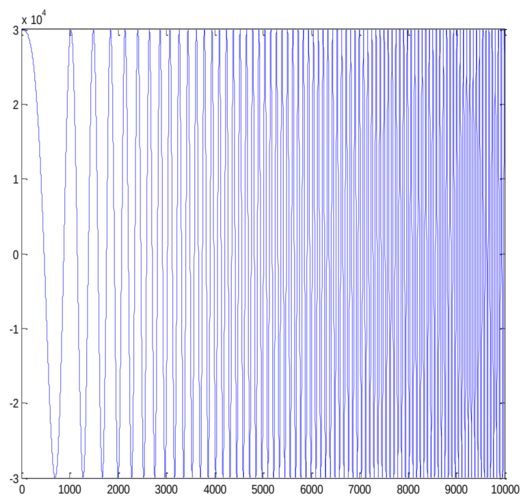


Fig. 4.2 Chirp input signal

4.2 Experiment results for VRFT

In this section, I will use the VRFT method to tune the parameters for both the velocity loop controller and the position loop controller. I will show the results of X axis, Y axis and the synergic motion of XY table. The open loop experiment are implemented in each axis to get the finite set of Input/Output data $\{u(t), y(t)\}_{t=1,\ldots,N}$, as mentioned in chapter 2. The principle of VRFT method is shown in Fig.4.3

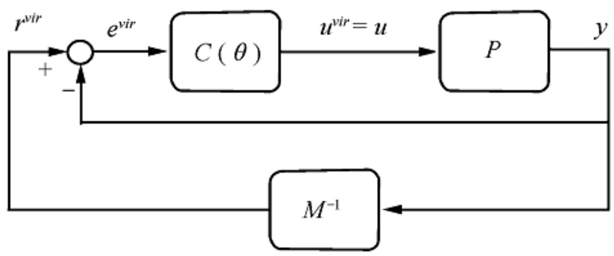


Fig. 4.3 Principle of VRFT method

4.2.1 Velocity loop controller tuning

In this experiment, the system is constructed as in chapter 3. Elmo drive is set to be under current loop. The input signal for the open loop experiment of VRFT method generate from the inner software of Elmo. The input signal is set to be 9 order PRBS signal. The sampling time of the experiment is 280us. I choose the overshoot of the desired model to be 5% and the settling time of the desired model to be 15 times the sampling time. The discrete form of the desired model is shown in equation (4-5). The step response and the bode diagram are shown in Fig.4.4. In the open loop experiment, we get the actual input $\{u(t)\}_{t=1,\ldots,N}$, the actual output $\{y(t)\}_{t=1,\ldots,N}$,. Then by using VRFT method as mentioned in chapter 2, I calculate the virtual input r(t) and the virtual tracking error e(t). Finally, I can get the parameters for PI controller for the velocity loop.

$$M(z^{-1}) = \frac{0.03274 + 0.06549z^{-1} + 0.03274z^{-2}}{1 - 1.436z^{-1} + 0.5672z^{-2}} \quad (4\text{-}5)$$

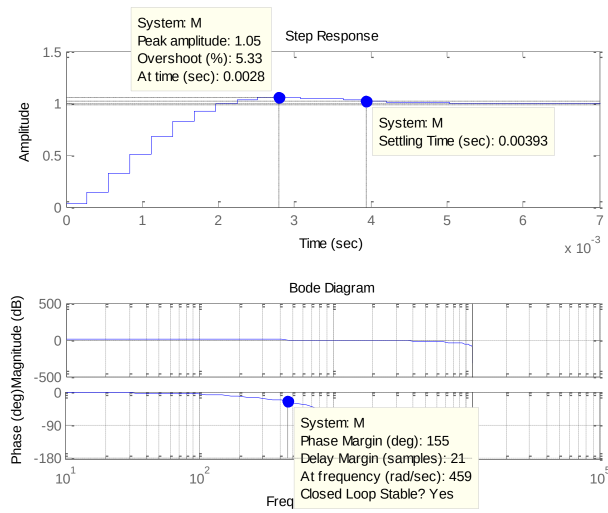


Fig. 4.4 Step response and bode diagram of the desired model of velocity loop

The velocity loop controller parameters are tuned by the VRFT method with filter and constrains. The PI controller for X axis are K_{vp} = 8.5, K_{vi} = 437.6 and for Y axis are K_{vp} = 27.3, K_{vi} = 1215.6. We apply these parameters to the velocity loop of Elmo drive and get the results for X axis as shown in Fig.4.5 and for Y axis as shown in Fig.4.6.

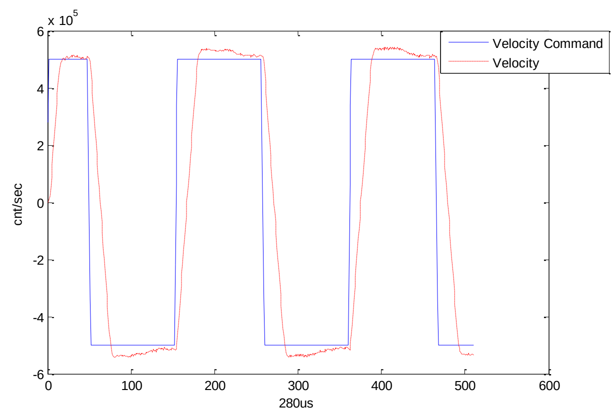


Fig. 4.5 Velocity loop results for X axis (Kp=8.5, Ki=437.6)

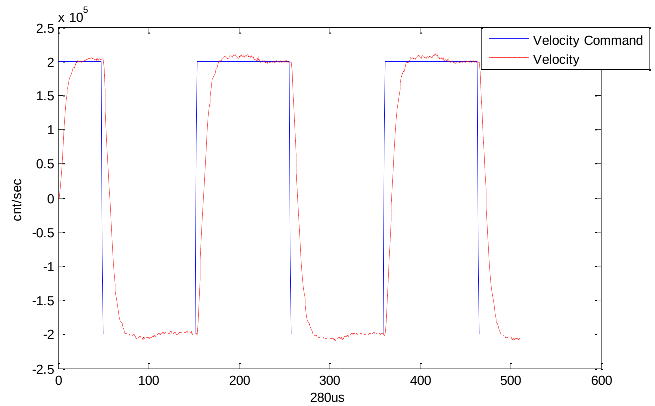


Fig. 4.6 Velocity loop results for Y axis (Kp=27.3, Ki=1215.6)

4.2.2 Position loop controller tuning

The open loop experiment is also implemented to tune the position loop controller parameters. In this experiment, I set the parameters for the velocity loop controller of Elmo drive as the results we get in section 4.2.1. GTS800 motion control card is used as the controller for the position loop of each axis. The input signal is set to be 9 order PRBS signal. The sampling time of the velocity loop is 1ms. I choose the overshoot of the desired model to be 5% and the settling time of the desired model to be 5 times the sampling time. The discrete form of the desired model is shown in equation (4-6). The step response and the bode diagram are shown in Fig.4.7. In the open loop experiment, we get the actual input $\{u(t)\}_{t=1,\ldots,N}$, the actual output $\{y(t)\}_{t=1,\ldots,N}$,. Then by using VRFT method as mentioned in chapter 2, I calculate the virtual input r(t) and the virtual tracking error e(t). Finally, I can get the parameters for PID controller for position loop.

$$M(z^{-1})=\frac{0.1739+0.3478z^{-1}+0.1739z^{-2}}{1-0.5383z^{-1}+0.2338z^{-2}} \tag{4-6}$$

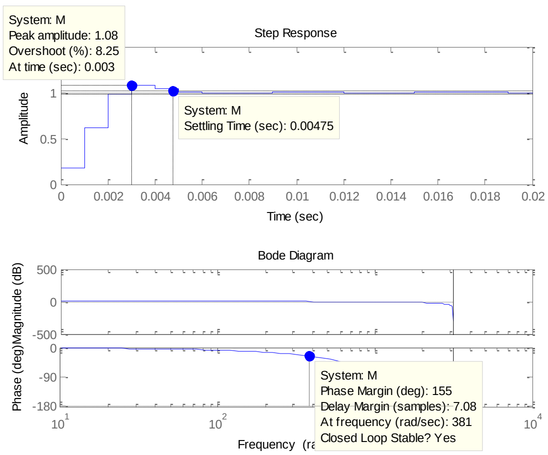


Fig. 4.7 Step response and bode diagram of the desired model for position loop

The position loop controller parameters are tuned by the VRFT method. The results for X axis are K_{PP} =12.8, K_{PI} =0.27, K_{PD}=8.94 and for Y axis are K_{PP} =54.7, K_{PI} =0.37, K_{PD}=153.2. We apply these parameters to the velocity loop of Elmo drive and get position loop results for X axis as shown in Fig.4.8, the position loop results for Y axis as shown in Fig.4.9 and the position loop results for synthesize motion of XY table as shown in Fig.4.10.

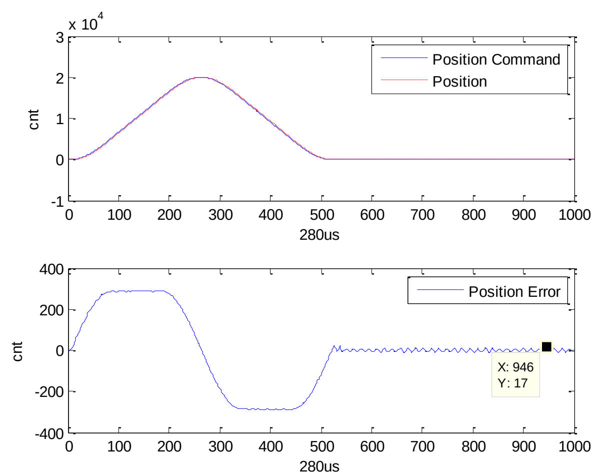


Fig. 4.8 Position loop results of X axis

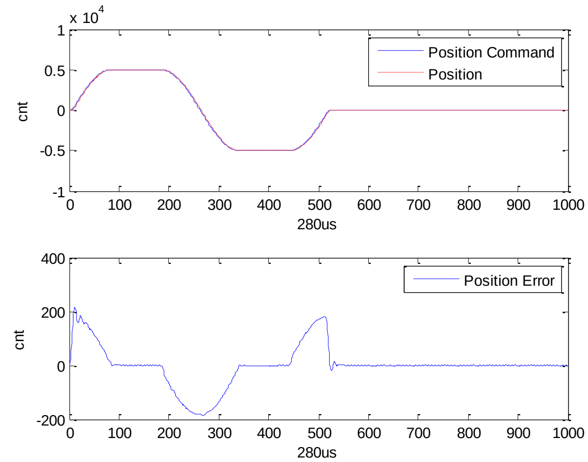


Fig. 4.9 Position loop results of Y axis

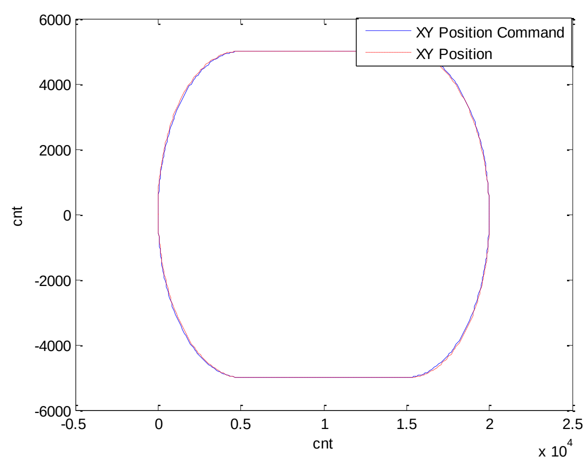


Fig. 4.10 Synthesize motion of XY table

4.3 Experiment results for NICBT method

In this section, I will use the NICBT method to tune the parameters for both the velocity loop controller and the position loop controller. I will show the results of X axis, Y axis and the synergic motion of XY table. The open loop experiment are implemented in each axis to get the finite set of Input/Output data $\{u(t), y(t)\}_{t=1,\ldots,N}$, as mentioned in chapter 2. The experiment diagram is shown in Fig.4.11.

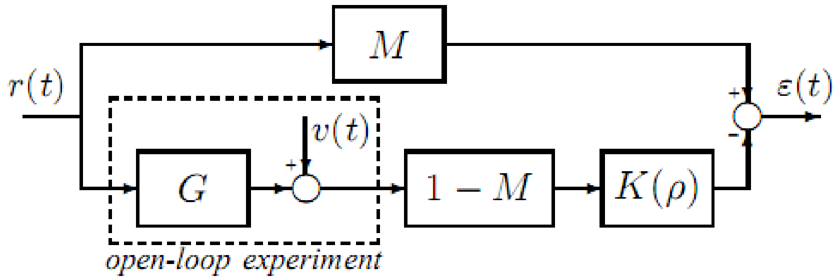


Fig. 4.11 Open loop experiment diagram for NICBT

I can get the data to minimize the criterion through a single experiment on the system. Suppose that the system G is excited in open loop with u(t)=r(t) and the noisy output y(t) is measured. From this experiment, the error signal $\varepsilon(\rho,t)$ can be expressed as a function of the controller parameter, as shown in equation 4-7.

$$\begin{aligned}\varepsilon(\rho,t) &= Mr(t)-(1-M)K(\rho)y(t)\\ &=[M-G(1-M)K(\rho)]r(t)-(1-M)K(\rho)v(t)\end{aligned} \tag{4-7}$$

By using the NICBT as we mentioned in chapter 2, we can get the velocity loop controller parameters and the position loop controller parameters.

4.3.1 Velocity loop controller tuning

In this experiment, the system is constructed as in chapter 3. Elmo drive is set to be under current loop. The input signal for the open loop experiment of NICBT method generates from the inner software of Elmo. The input signal is set to be 9 order PRBS signal. The sampling time of the experiment is 280us. I choose the overshoot of the desired model to be 5% and the settling time of the desired model to be 15 times the sampling time. The desired model is the same as section 4.2.1. In the open loop experiment, we get the actual input $\{u(t)\}_{t=1,\ldots,N}$, the actual output $\{y(t)\}_{t=1,\ldots,N}$,. Then by using NICBT method as mentioned in chapter 2, I can get the parameters for PI controller for the velocity loop. The PI controller parameters for X axis are K_{vp} = 13.2, K_{vi} = 846.5 and for Y axis are K_{vp} = 31.4, K_{vi} = 2765.5. We apply these parameters to the velocity loop of Elmo drive and get the results for X axis as shown in Fig.4.12 and for Y axis as shown in Fig.4.13.

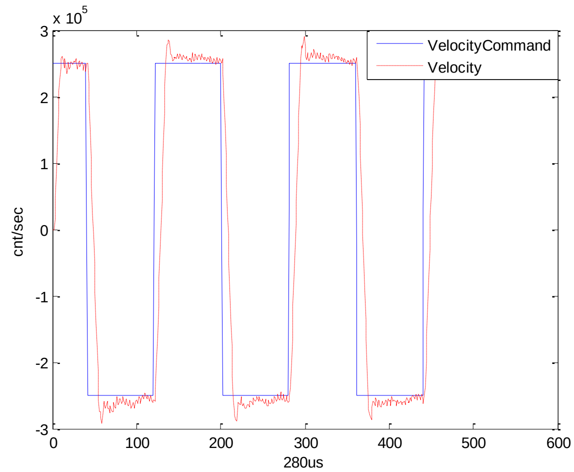


Fig. 4.12 Velocity loop results for X axis (Kp=13.2, Ki=846.5)

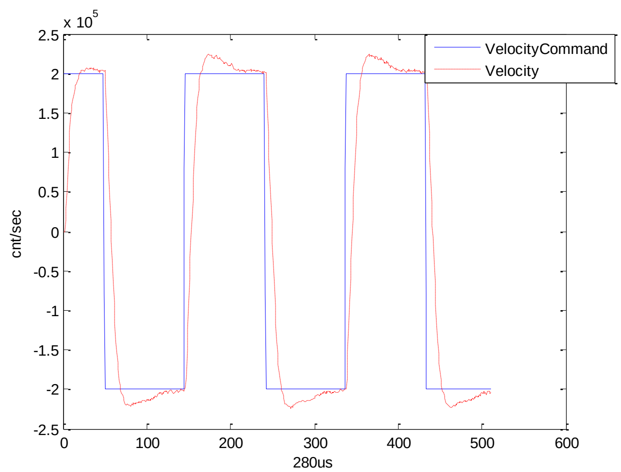


Fig. 4.13 Velocity loop results for Y axis (Kp=31.4, Ki=2765.5)

4.3.2 Position loop controller tuning

The open loop experiment is also implemented to tune the position loop controller parameters. In this experiment, Elmo drive is under velocity loop. I use GTS800 motion control card to set the input signal for the drive. The sampling time of the velocity loop is also 1ms. I choose the overshoot of the desired model to be 5% and the settling time of the desired model to be 5 times the sampling time. The desired model of the system is the same as in section 4.2.2.

The position loop controller parameters are tuned by the NICBT method. The results for X axis are K_{PP} =11.5, K_{PI} =0.13, K_{PD}=211.4 and for Y axis are K_{PP} =21.6, K_{PI} =0.35, K_{PD}=159.3. We apply these parameters to the velocity loop of Elmo drive and get position loop results for X axis as shown in Fig.4.14, the position loop results for Y axis as shown in Fig.4.15 and the position loop results for synthesize motion of XY table as shown in Fig.4.16.

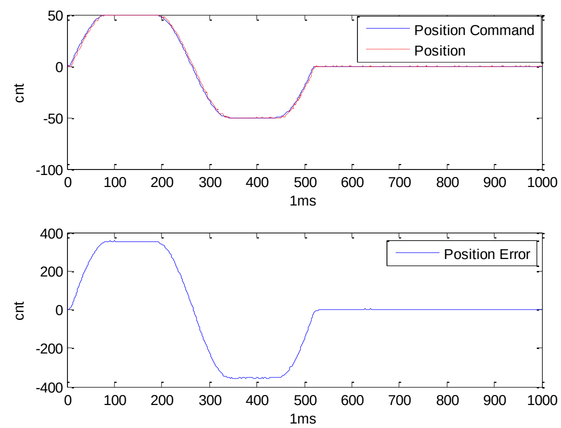


Fig. 4.14 Position loop results of X axis

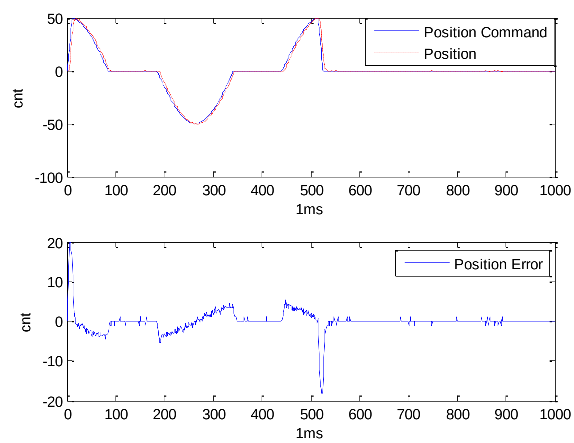


Fig. 4.15 Position loop results of Y axis

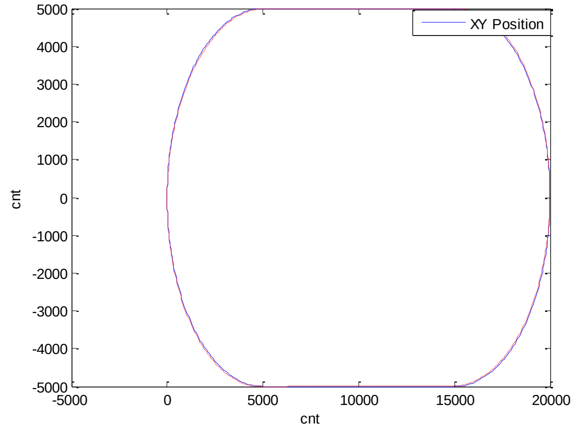
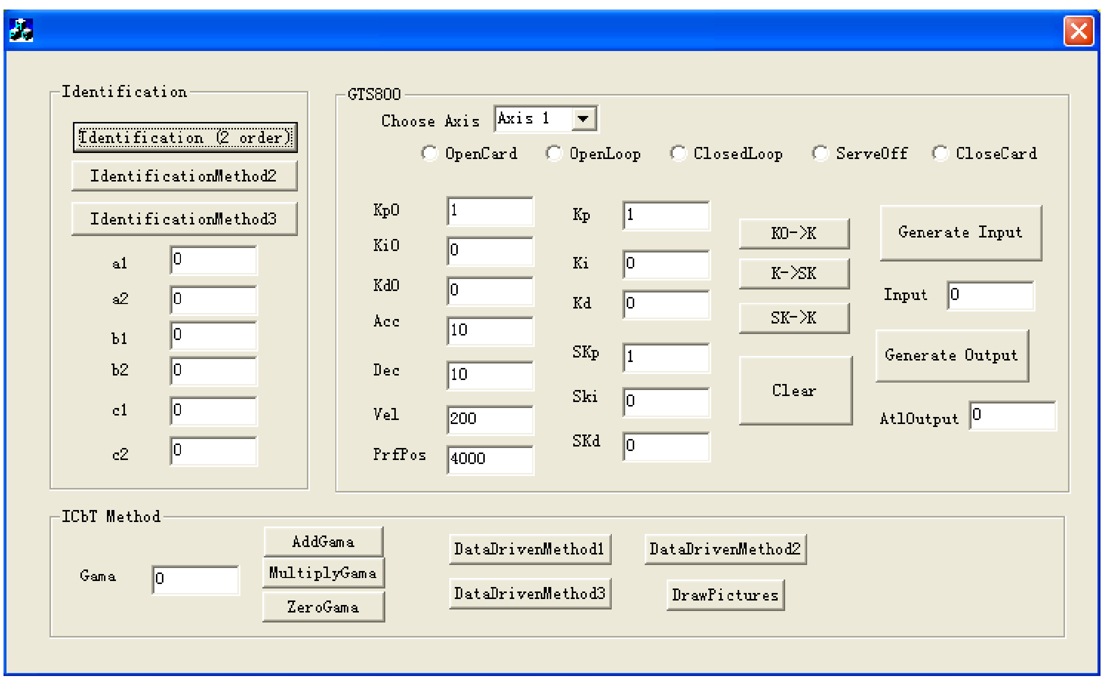


Fig. 4.16 Synthesize motion of XY table

4.4 Experiment results for ICBT method

In industry, the velocity and acceleration feedforward controller are usually used as the feedforward controller. As mentioned above, this kind of feedforward controller is also based on (4-2). It is only a method to approximate the PTC.



4.5 Experiment results for IFT method

Since the perfect cancellation in the presence of uncancellable zeros is not achievable, alternative approaches to optimizing the overall frequency response have

4.6 Experiment results for control criteria based tuning method

In this section, I will apply the foregoing method into our motion control system. Our experimental system is composed of a linear motor as the executing component, Elmo drive, a motion control card and the host. The maximum current of the linear motor is 6.9A, the continuous current of the linear motor is 2.3A. Elmo Drive is working under the position mode, the motion control is only used to plan the position and time but not as a main controller. So the control parameters of Elmo Drive are very important to the performance of the system. As mentioned before, Elmo Drive is composed of three loops, the current loop, the velocity loop and the position loop. We just use the auto tune results by the Drive itself and don’t tune the current loop manually. As to the velocity loop and the position loop, we tune them step by step. The user interface by VC++ program is shown in figure 29.

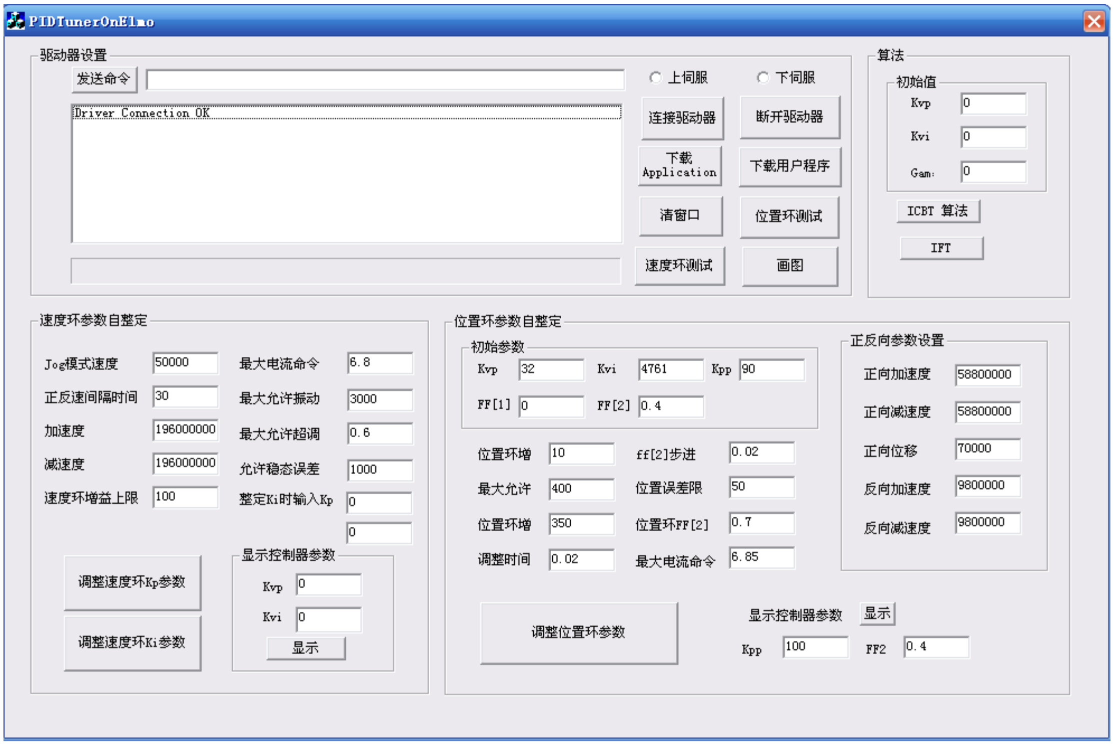


Figure 29 User interface by VC++ program

4.6.1 Velocity loop controller tuning

Because The velocity loop of Elmo controller is composed of a PI controller. There are only two parameters to be tune. As the tuning guide of Elmo manual [5], we set the parameter of P controller Kvp = 10 and set the parameter of I controller Kvi =0 as the initial value. Before tuning the velocity loop, we have to set the parameters of Elmo Drive, including AC, DC, and SD. In this paper, we set AC, DC and SD to be 10g in order to get a quadrate curve of velocity command so as to make the curve clearer. After that, we set JV at a relative large value but take care not to let the current command saturated. In this paper, we set JV to be 40000 with a encoder whose resolution is 2000000.

We do each velocity experiment as the following step in order to get the data we want.

- Step 1: set the recorder as mentioned before.

- Step 2: put the mover of the linear motor at a proper position, ie, at the center of the track.
- Step 3: set JV to be a positive value during a certain period to ensure that the position of the motor is running within a certain range.
- Step 4: change the direction of the linear motor by setting JV to be minus JV and let it running during a same length of period to ensure the position of the linear motor within a certain range.
- Step 5: repeat step 3 to step 4 several times in order to get enough data for tuning

After we have done an experiment and upload the data from the recorder, we can use the functions mentioned above as the criterion to change the parameters of the controller. In our experiment, we first increase Kvp until there is a certain overshoot in the velocity curve which is also evaluate by one of the functions. If the overshoot is bigger than what we want, we reduce Kvp. After that set Kvi to be the value mentioned in [5], that Kvi=1/(Rise time)*Kvp. If there is certain vibration in the velocity curve, we divided it by 2 and then repeat the experiment until all the criteria satisfy the requirements. The initial result when Kvp=10 and Kvi=0 is shown in figure 30, while the final result with Kvp=32, and Kvi= 0 after auto tuning is show in figure 31. Then we set Kvi=4761, as mentioned above, as shown by figure 32. We can see that the initial curve can not satisfy our requirement, after auto tuning the Kvp parameter, there is still a steady state error and after adding the Kvi parameter, the velocity tracking curve can meet our requirement very well.

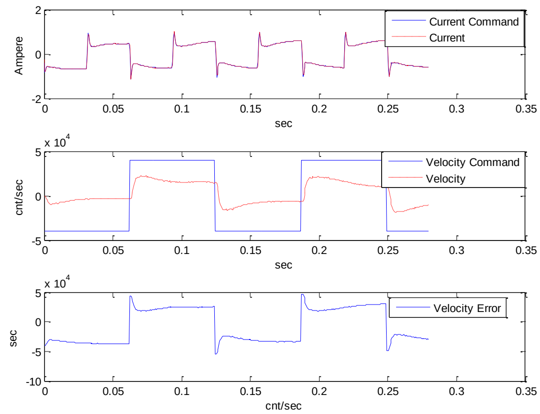


Figure 30 Velocity Loop Results when Kvp=10, Kvi=0

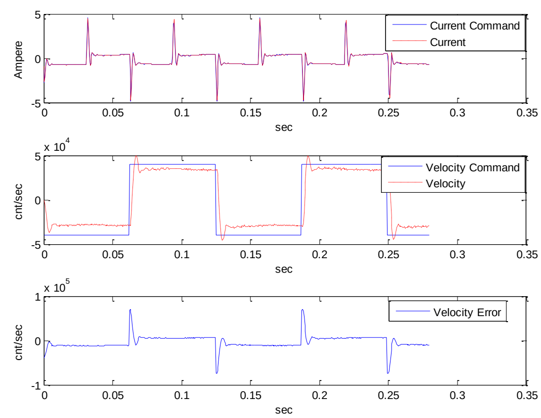


Figure 31 Velocity Loop Results when Kvp=32, Kvi=0

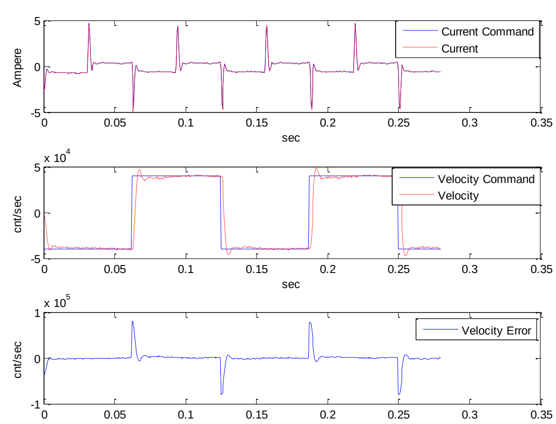


Figure 32 Velocity Loop Results when Kvp=32, Kvi=4076

4.6.2 Position loop controller tuning

The position loop controller of Elmo Drive is composed of a P controller, a velocity feedforward controller and an accelerating feedforward controller. Before we tune the control parameters of the position loop, we should first set the values of AC and DC. In our experiment we set AC and DC both to be 3g so as to simulate the real world condition.

Then we put the linear motor at a proper position, near the beginning of the track, setting PX=0. In each experiment, the running range is best to set to include the whole range of the track in order to get a set of more convincing data. We set the parameter of the accelerating feedforward control equal to zero and set the parameter of the velocity feedforward control equal to a small value by experience, and set the parameter of P control equal to a certain value. In short, the initial value should be setting by tuning experience to maintain the stability of the motion control system.

We do each of the experiment in the position loop as the following step in order to get the data we want, then use the data and the functions we establish to auto tune the parameters controller.

- Step 1: set the recorder as mentioned before.
- Step 2: set PA to be a positive value and run the linear motor.
- Step 3: upload the data and use the function to evaluate the performance of the linear motor.
- Step 4: kill the recorder.
- Step 5: set PA to zero.
- Step 6: repeat step2 to step 1 to step5 until the performance of the linear motor can meet the requirements.

In our experiment, we first increase the parameter of P controller by 10 in each experiment until the position error running into a certain range, but this range must be larger than the required position error. And then we both increase the parameters and increase the accelerating feedforward parameter by combination. We can record some parameters which can meet our requirement, and load the user programs to test them. Also to make sure that the motion control system is stable in all the experiments and if it is not stable the system can stop running. We take some measures using the vibration evaluation, setting the position error limit appropriately and so forth. The experimental result is shown below. The initial result when Kpp=100, FF[2]=0.2 and FF[1]=0 is shown in figure 1. From the evaluation function, we know that the settling time ts=60.48ms. By using the alternative method to auto tune the velocity loop, we get several results that can satisfy our needs. They are shown in figure 33, figure 34, and figure 35. The position error we set is 25um, all of them have the settling time less than 20ms.

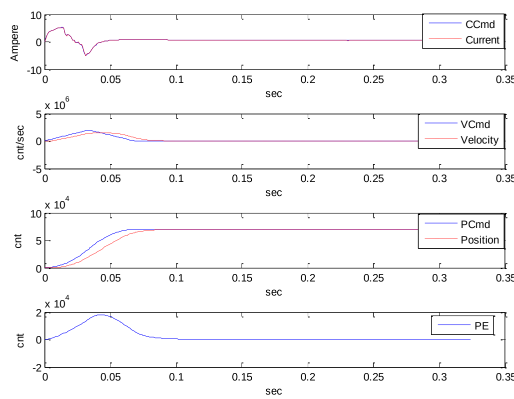


Figure 33 Position Loop Results when Kpp=100, FF[2]=0.2.

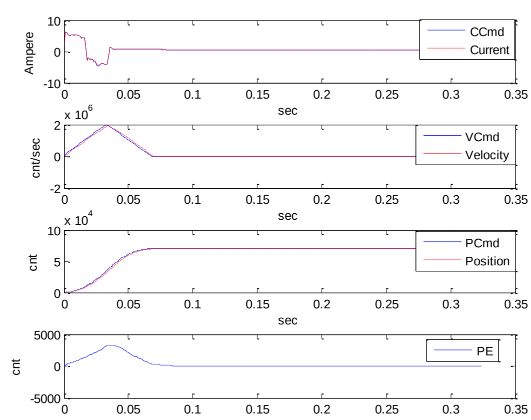


Figure 34 Position Loop Results when Kpp=235, FF[2]=0.62

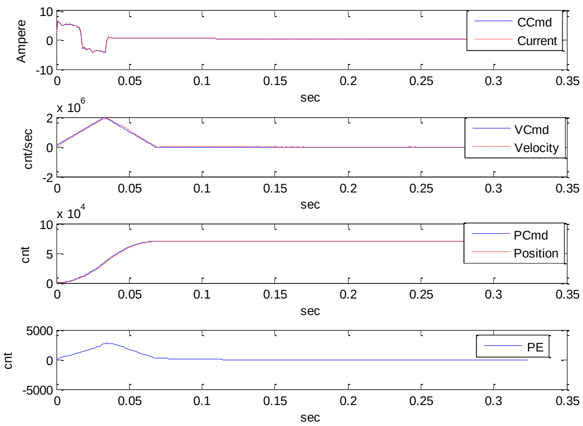


Figure 34 Position Loop Results when Kpp=245, FF[2]=0.66

4.7 Summary

In this section, I have shown the results of the data-driven methods, including the VRFT method, NICBT method, ICBT method, IFT method and the control criteria based method. The experiments are based on the system we mentioned in the chapter 3. As discussed in section 4.1, in the experiment, I choose the chirp signal as the input signal, because it can result in a rich excited frequency and will do little damage to the mechanical part.

From the results of the experiment, we can see that the non-iterative may sometimes corrupt by the noise of the system, because it only uses the data of one experiment. However, the experiment is easy to fulfill and we can get the results as quickly as we can. On the other hand, the iterative method is better in resisting the noise and the disturbance; however, the iterative method need more experiments and cost time to get the results. Furthermore, the control criteria based tuning method is a very practical method for the system driven by the Elmo drive. By using this method, we can find a lot of optimal local solutions and chose the best of them to be the global solution and we can also set constrains for the parameters of the controller.

CHAPTER 5

CONCLUSION

5.1 About the identification

- ELS is available for the closed-loop parameter estimation.
- Model of 2-order is available for a class of motion control systems and the order is lower than the theoretical result.
- For closed-loop identification, the estimation procedure is switched off when the input is not exciting enough. This mechanism is verified to be available to guarantee the accuracy of identification and reliability of self-tuned control parameters.
- The identification is relatively time-consuming so it is important to simplify the identification procedure in the premise of keeping the accurate identification.

5.2 About the controller design

- The off-line identification is necessary and it helps the self-tuning controller work well at start-up.
- The self-tuning PID controller inherits the simplicity of the general PID controller and performs better, especially when controlled plants are varying.
- The whole system becomes a typical 2-order system with the proposed PID controller, of which the physical meaning is clear. Therefore, controller parameters are easy to determine.

- The feedforward controller is good at improving the tracking performance because it can make use of the future information of desired trajectories in motion control systems.
- The tracking error with the proposed self-tuning ZPETC can be guaranteed in the micrometer level and part of the work has been used in the real milling process.
- With the help of the off-line identification, these two controllers are easy to be used in different occasions with little operation and tuning procedure. Thus it can improve the work efficiency a lot.

5.3 Future work

- Both of self-tuning controllers discussed in the dissertation are sensitive to the modeling error so it is important to develop mathematical models of plants and identification algorithms.
- Both of feedback and feedforward controllers mentioned above are based on the estimated linear model without considering the nonlinear disturbances. However, if further performance enhancement is necessary, another control component which is used to deal with nonlinear disturbances maybe enabled [53].

REFERENCES

1 F. L. Lewis. Applied Optimal Control & Estimation: Digital Design & Implementation. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1992.

2 Harman, G.G. Wire bonding in microelectronics: materials, processes, reliability, and yield, McGraw-Hill Professional, 1997.

3 Yu, C.C. Autotuning of PID Controllers, Springer Verlag, 2006.

4 H. Nyquist. Regeneration theory. Bell System Technical Journal, 11:126, 1932.

5 H.W. Bode. Feedback amplifier design. Bell System Technical Journal, 19:42, 1940.

6 R.E. Kalman. Contributions to the theory of optimal control. Boletin de la Sociedad Matematica Mexicana, 5:102–119, 1960.

7 R.E. Kalman. A new approach to linear filtering and prediction problems. Transactions ASME, Series D, J. Basic Eng., 82:34–45, 1960.

8 K.J. ˚Astr ¨ om and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. In Proc. IFAC Symposium on Self-Adaptive Systems, Teddington, UK, 1965.

9 L. Ljung. System Identification: Theory for the User, 2nd Edition. Prentice-Hall, Englewood Cliffs, NJ, 1999.

10 M. Gevers and L. Ljung. Optimal experiment designs with respect to the intended model application. Automatica, 22:543–554, 1986.

11 B. Wahlberg and L. Ljung. Design variables for bias distribution in transfer function estimation. IEEE Trans. Automatic Control, AC-31:134–144, 1986.

12 Hjalmarsson, H. Iterative feedback tuning-an overview. International journal of adaptive control and signal processing. 16. 2002.

13 K. Hamamoto, T. Fukuda, and T. Sugie, "Iterative feedback tuning of controllers for a two-mass-spring system with friction," Control Engineering Practice, vol. 11, pp. 1061-1068, 2003.

14 A. Karimi, L. Miskovic, and D. Bonvin, "Iterative correlation-based tuning:

Application to a magnetic suspension system," Control Engineering Practice, vol. 11, pp. 1069-1087, 2003.

15 L. Miskovic, A. Karimi, and D. Bonvin, "Correlation-based tuning of a restricted-complexity controller for an active suspension system," European Journal of Control, vol. 9, no, 1, pp. 77-83, January 2003.

16 G. H. Er, Y. X. Dou. Motion Control System. Tsinghua University Press, 2002.

17 Steinbuch, M. and Norg, ML, Advanced motion control: An industrial perspective, European Journal of Control: 1998, 4, 278—293.

18 Prentice-Hall, Inc. Upper Saddle River, NJ, USA Modern control engineering. Ogata, K. 1996.

19 T. Hagglund, K. J. Astrom. PID Controllers: Theory, Design, and Tuning. Research Triangle Park, NC: Instrument Soc. Amer., 1995.

20 MISKOVIC, L. Data-driven controller tuning using the correlation approach. M.Sc. in Electrical Engineering, University of Belgrade, Yougoslavie de nationalité serbe et monténégrine. 2006.

21 M. C. Campi, A. Lecchini, and S.M. Savaresi. Virtual reference feedback tuning: a direct method for the design of feedback controllers. Automatica, vol. 38, no. 8, pp. 1337-1346, 2002.

22 Campi M. C, Lechini A, Savaresi SM. Virtual reference feedback tuning (VRFT): a new direct approach to the design of feedback controllers. In Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, December, 623-628, 2000.

23 Lecchini A. Virtual reference feedback tuning: a new direct data-based method for the design of feedback controllers. Ph. D Thesis-Universita di Brescia, 2001.

24 G. O. Guardabassi and S. M. Savaresi. Virtual reference direct design method: an on-line approach to data-based control system design. IEEE Trans. on Automatic Control, 45(5):954–959, 2000.

25 Antonio Sala, Alicia Esparza. Extensions to “virtual reference feedback tuning: A direct method for the design of feedback controllers”. Automatica, 41: 1473-1476, 2001.

26 A. Lecchini, M.C. Campi, and S.M. Savaresi. Virtual reference feedback tuning for two degree of freedom controllers. International journal of adaptive control and signal processing. 16: 355-371, 2002.

27 Campi, M. C., Lecchini, A., & Savaresi, S. M. An application of the virtual reference feedback tuning method to a benchmark problem. European Journal of Control, 9(1): 66-76, 2003.

28 Alireza Karimi, Klaske van Heusden and Dominique Bonvin. “Noniterative Data-driven Controller Tuning Using the Correlation Approach”. European Control Conference, 2002.

29 Karimi, A., L. Miskovic and D. Bonvin. Convergence analysis of an iterative correlation based controller tuning method. 15th IFAC World Congress, Barcelona, Spain, 2002.

30 Karimi, A., L. Miskovic and D. Bonvin. Iterative correlation-based controller tuning: Frequency-domain analysis. In: 41st IEEE-CDC, Las Vegas, USA, 2002.

31 A. Karimi, L. Miskovic, and D. Bonvin. Iterative correlation-based controller tuning”, International Journal of Adaptive Control and Signal Processing, 18: 645-664, 2004.

32 Hjalmarsson H., Gevers M., Gunnarson S., and Lequin O. Iterative Feedback Tuning: Theory and Applications. IEEE Control Systems Magazine, 18(4): 26-41, 1998.

33 Hjalmarsson, H., Gunnarsson, S., and Gevers, M. Aconvergent iterative restricted complexity control design scheme. In Proceedings of the 33rd IEEE conference on decision and control, Orlando, FL. pp: 1735-1740, 1994.

34 Olivier Lequin, Michel Gevers, Magnus Mossberg, Emmanuel Bosmans, and Lionel Triest. Control Engineering Practice. 11: 1023-1033, 2003.

35 Fotios N. Koumboulis, Member, IEEE, Maria P. Tzamtzi, Member, IEEE, George E. Chamilothoris. Iterative Feedback Tuning Safe Switching Controllers. Proceedings of the 13th Mediterranean Conference on Control and Automation Limassol, Cyprus, June 27-29, 2005.

36 Elmo Interlude Communication API for CANopen and RS-232.

37 SimplIQ Software Manual

38 Elmo HARmonica Hands-on Tuning Guide

39 潘立登，潘仰东。系统辨识与建模。化学工业出版社。2004，1

40 李言俊，张科。系统辨识理论及应用。国防工业出版社，2003，5

41 杨承志，孙棣华，张长胜。系统辨识与自适应控制。重庆大学出版社。2003，7

42

哈尔滨工业大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《运动控制系统中控制算法的研究》，是本人在导师指导下，在哈尔滨工业大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字：　　　　　　日期：　　年　月　日

哈尔滨工业大学硕士学位论文使用授权书

《运动控制系统中控制算法的研究》系本人在哈尔滨工业大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归哈尔滨工业大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅，同意学校将论文加入《中国优秀博硕士学位论文全文数据库》和编入《中国知识资源总库》。本人授权哈尔滨工业大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

作者签名：　　　　　　日期：　　年　月　日

导师签名：　　　　　　日期：　　年　月　日

哈尔滨工业大学硕士学位涉密论文管理

根据《哈尔滨工业大学关于国家秘密载体保密管理的规定》，毕业论文答辩必须由导师进行保密初审，外寄论文由科研处复审。涉密毕业论文，由学生按学校规定的统一程序在导师指导下填报密级和保密期限。

本学位论文属于　　保密□，在　　年解密后适用本授权书

不保密□

作者签名：　　　　日期：　年　月　日

导师签名：　　　　日期：　年　月　日

ACKNOWLEDGEMENT

I would like to make use of this opportunity to express my acknowledgement to all the people who gave me help before.

First of all, I would like to say thanks to my supervisor, Prof. Zexiang Li. As for me, he is more than an academic master with passion and foresight. His distinctive attitude to research and engineering will impact me for a long time in the future.

Besides, I want to say thanks to my senior fellow apprentice, Mr. Xinlu Yu, Mr. Wei Cheng and Jianjun Min. They give me endless care and help. Their inspiration gives me confidence and motivation to keep going forward.

Also, my friends and roommates, Ms. Zhihong Tan and Ms. Xin Guan who are always with me deserve my gratitude. We have been enjoying two fruitful years and sharing much together. I learnt much from them.

At last, I would like to take some time for my great family, especially my parents, Ms. Jingwei Xu and Mr. Guoping Jiang. They give me life, confidence, knowledge, freedom and even more. My achievements will be nothing without my

family’s happiness and appreciation. The responsibility to never let them down will always encourage me to pursuit my dream.