Proceedings of the 30th Chinese Control Conference
July 22-24, 2011, Yantai, China

Tuning Method for Second-order Active Disturbance Rejection Control

CHEN Xing[1], LI Donghai[1], GAO Zhiqiang[2], WANG Chuanfeng[1]

1 State Key Lab of Power Systems, Dept of Thermal Engineering, Tsinghua University, Beijing 100084, P. R. China
E-mail: lidongh@mail.tsinghua.edu.cn

2 The Center for Advanced Control Technologies, Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH 44115, USA.
E-mail: z.gao@ieee.org

Abstract: A simple tuning method for second-order active disturbance rejection control (ADRC) that achieves high performance and good robustness for a wide range of processes is presented. ADRC is a novel control strategy whose exciting performance has been shown by literatures. The proposed method makes ADRC become easy to tune and more practical. Once the desired settling time is given, only one parameter need to be tuned during the design procedure, which can be found by monotonously increasing. Examples are given to show the effectiveness and flexibility of the method. Simulations demonstrate that second-order ADRC can handles processes with various characteristics, including low- and high-order, large dead time, non-minimum phase, unstable and distributed parameter systems.

Key Words: Second-order ADRC, One Parameter Tuning Method, Monte-Carlo Experiment

1 Introduction

Proportional-integral-derivative (PID) controllers are widely used in the process control industry [1]. The main reason is its relatively simple structure, which can be easily understood and implemented in practice. Another reason is that users do not have to change the structure of PID, only re-tuning of its parameters is necessary, while handling different kinds of problem. However, many PID controllers are poorly tuned and limited in performance, especially while dealing with dynamic uncertainties.

To overcome the limitations of PID, the active disturbance rejection controller (ADRC), a novel controller, was developed by Han [2, 3]. The basic idea of ADRC is to use an extended state observer (ESO) to estimate the internal and external disturbances in real time. Then, through disturbance rejection, the originally complex and uncertain plant dynamics is reduced to a simple cascade integral plant, which can be easily controlled. Due to its strong robustness and disturbance rejection, ADRC has been successfully applied in many fields [4-6].

However, the tuning procedure of ADRC is very complicated due to its large number of parameters. The tuning is usually relied on the human experiences. Some artificial intelligence approaches were adopted to regularize the ADRC parameters (see, for example, Tong[7] and Shi[8]), but the algorithm was complex and the computational cost was large. To simplify the problem, Gao[9] used linear gains in place of the original nonlinear gain in ADRC. Thus, the number of parameters was reduced obviously. This made the tuning more realistic. Researches showed that the linear ADRC still achieved high performance and good robustness (see, for example, Sun[10] and Miklosovic[11]). The discussions in this paper are limited to the linear case.

This work is supported by National Nature Science Foundation under Grant : 51076071

Besides the difficulty in tuning, there is another obstacle in the implementation of ADRC. That is, the order of ADRC is generally varied with the controlled plant. Much work has been done in the cases of n-order ADRC applying to n-order plants. However, the order of practical plants exists from one to infinite, even unknown or time-varying. Therefore, it's necessary to study how a fixed-order ADRC can deal with the real processes, and how to tune its parameters. To the best of our knowledge, however, little has been done about this. Yao[12] discussed the case of second-order ADRC, but the controlled objects were limits to third-order, and that no specific tuning rule was provided therein.

In this paper, we present a simple tuning method for second-order ADRC that demonstrates promising results in controlling an extensive class of linear processes, including low- and high-order, large dead time, non-minimum phase, unstable and distributed parameter systems. Only one parameter needs to be tuned in this method under the condition that the desired settling time is given. To verify the inherent robustness of ADRC, the Monte-Carlo method[13] is used. Simulations show that both high performance and good robustness are achieved.

The remainder of this paper is organized as follows. Section 2 formulates the design problem, in which the second-order ADRC is presented. A result about stability of ADRC system is discussed in section 3, which is an important hint for the tuning. Then, the specific tuning method is given in section 4. In section 5, the method is applied to several examples in simulation. Finally, section 6 offers concluding remarks.

2 Problem Formulation

2.1 Process Description

The active disturbance rejection concept has been applied to problems of different kinds, including single-input-single-output (SISO), as well as multi-input-multi-output. (MIMO), plants that are nonlinear, time-varying, and most of all, uncertain. In this paper, however, the process is assumed to be linear, time invariant, and specified by a transfer function:

$$\frac{y(s)}{u(s)} = G_p(s,\mathbf{p}) \tag{1}$$

where $\mathbf{p}$ denotes the parameters of system. The description covers finite dimensional systems with time delays and infinite dimensional systems described by linear partial differential equations. To verify the robustness of ADRC, we assume that $\mathbf{p}$ may have a 10% change, that is, $\mathbf{p} \in [0.9\mathbf{p}_0, 1.1\mathbf{p}_0]$, where $\mathbf{p}_0$ is the nominal value of $\mathbf{p}$.

2.2 Second-order ADRC

The structure of second-order ADRC is illustrated in Fig. 1.

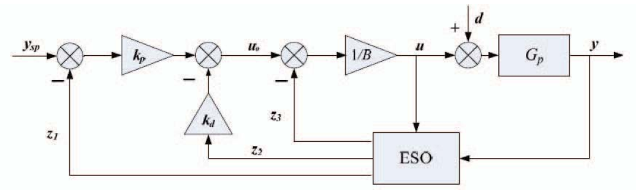

Fig.1 Structure of second-order ADRC

$G_\mathbf{p}$ is the process to be controlled. Two external signals act on the control loop, namely set point y_{sp} and load disturbances d . An extended state observer (ESO) is used to estimate the external disturbances d and internal uncertainty (such as parameters perturbation in $G_\mathbf{p}$) in real time. The inputs of ESO are control signal u and process output y . The outputs of ESO are z_1, z_2 and z_3 . Then, a simple control law is applied, in which k_p, k_d and B are the control parameters.

The ESO in second-order ADRC is

$$\begin{cases} \dot{z}_1 = z_2 + \beta_1(y - z_1) \\ \dot{z}_2 = z_3 + \beta_2(y - z_1) + Bu \\ \dot{z}_3 = \beta_3(y - z_1) \end{cases} \tag{2}$$

where β_1, β_2 and β_3 are observer parameters to be determined. If $G_\mathbf{p}$, somehow, can be approximated by a second order system model structure:

$$\ddot{y} = f(t, y, \dot{y}, \omega) + Bu \tag{3}$$

and the observer (2) is well tuned, z_1, z_2, z_3 will track $y, \dot{y}$ and f respectively. The term f represents the combined effect of the internal dynamics and external disturbances ω . Note that, there is no need to know the accurate mathematic description of f in model (3).

The control law in second-order ADRC is

$$u_0 = k_p(y_{sp} - z_1) - k_d z_2 \tag{4}$$

$$u = (u_0 - z_3)/B \tag{5}$$

With $z_3 \approx f$ obtained from the ESO, (5) reduces (3) to an approximate double integral plant:

$$\ddot{y} = f + u_0 - z_3 \approx u_0 \tag{6}$$

Then, substituting (4) into (6) yields the closed-loop dynamic characteristic:

$$\ddot{y} + k_d\dot{y} + k_p y = k_p y_{sp} \tag{7}$$

Taking the Laplace transform yields the close-loop transfer function:

$$G_d(s) = \frac{y(s)}{y_{sp}(s)} = \frac{k_p}{s^2 + k_d s + k_p} \tag{8}$$

Remarks:

a. There are six tuning parameters in second-order ADRC, namely control parameters B, k_p and k_d , and observer parameters β_1, β_2 and β_3 .

b. In the frame of ADRC, the external disturbances and internal uncertainty can be estimated and canceled in real time. This is the reason that ADRC is model-independent and inherent robust.

2.3 Design Objective

The design objective of this paper is to determine the six parameters in ADRC so that the system behaves well with respect to changes in the two signals y_{sp} and d , as well as in the process model G_p .Hence, the specification will express requirements on

- Set point response. The system asymptotically tracks stepwise set point changes, and the settling time and overshoot $\sigma\%$ are smaller than desired value.
- Load disturbance response. The process output y comes back to set point y_{sp} quickly and the impact of disturbance is low.
- Robustness with respect to model uncertainties. That is, for a 10% change in parameter $\mathbf{p}$ of the process, the system is still stable and the control performance is still good.

3 Stability and the Parameter B

Converting the ADRC equations to frequency domain, the closed-loop transfer function $G_{cl}(s)$ of the control system is readily available:

$$G_{cl}(s) = \frac{G_p(s)k_p(s^3 + \beta_1 s^2 + \beta_2 s + \beta_3)}{BA_1(s) + G_p(s)A_2(s)} \tag{9}$$

where

$$A_1(s) = s^3 + (\beta_1 + k_d)s^2 + (\beta_1 k_d + \beta_2 + k_p)s$$
$$A_2(s) = (\beta_3 + \beta_2 k_d + \beta_1 k_p)s^2 + (\beta_3 k_d + \beta_2 k_p)s + \beta_3 k_p$$

From this transfer function, the stability analysis will proceed.

We learned from experiments that, in the case of a second-order ADRC dealing with a plant whose order is higher than two, the parameter B is important concerning to the stability of system. To study how the parameter B influence the stability of system, we simplify the problem by making $\beta_1, \beta_2, \beta_3$ a function of ω_o and k_p, k_d a function of ω_c, as proposed in Gao[9]:

$$\begin{aligned} s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 &= (s+\omega_o)^3 \\ s^2 + k_d s + k_p &= (s+\omega_c)^2 \end{aligned} \tag{10}$$

such that there are only three parameters left in (9),that is, B, ω_c and ω_o .Then, a search program is used to determine the region in $\omega_o - \omega_c$ plane where the closed-loop system is stable. And this is repeated for different B. The results for an example of $G_p(s) = 1/(s+1)^3$ are shown in Fig. 2, where the area to the lower-left side of the curve is the stable region. (See appendix 1 for more examples).

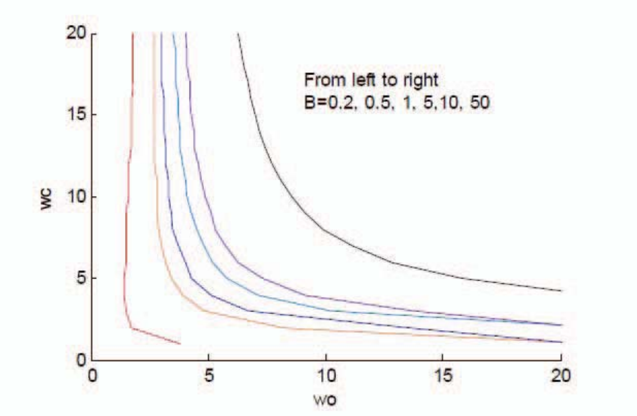

Fig.2 Stability in $\omega_o - \omega_c$ plane of $G_p(s) = 1/(s+1)^3$

Results show that as B increases within certain bound, the stable area in $\omega_o - \omega_c$ plane is expanded. This is an important hint for the tuning method discussing subsequently.

Remark:

c. In the tuning procedure, B is chosen to make sure the closed-loop system is stable. It must be large enough so the right control parameters, which are determined to achieve the desired performance, lie in the stable region. On the other hand, B can't be too large, otherwise the control signal u is weak and the system becomes slow. Therefore, a trade-off has to be made between stability (robustness) and response speed.

4 Tuning Method

4.1 Tuning of Control Parameters k_p and k_d

Note from (8) that k_p and k_d decide the closed-loop transfer function. By setting

$$k_p = \omega_c^2, k_d = 2\omega_c \tag{11}$$

where ω_c is denoted as control bandwidth by Gao[9].Thus, the desired transfer function becomes:

$$G_d(s) = \frac{\omega_c^2}{(s+\omega_c)^2} \tag{12}$$

As the settling time t_s and the overshoot $\sigma\%$ is the main dynamic performance index, and the overshoot of (12) is constantly zero, t_s becomes the only factor taken into account while choosing ω_c. Therefore, the relation between t_s and ω_c is derived subsequently.

Under a unit step change in input signal, the output of (12) with Laplace transform is

$$Y(s) = \frac{\omega_c^2}{s(s+\omega_c)^2} = \frac{1}{s} - \frac{\omega_c}{(s+\omega_c)^2} - \frac{1}{s+\omega_c} \tag{13}$$

Thus, the unit step response of system is

$$y(t) = 1 - \omega_c t e^{-\omega_c t} - e^{-\omega_c t} = 1 - (1+\omega_c t)e^{-\omega_c t} \tag{14}$$

According to the definition of setting time

$$|y(t_s) - y(\infty)| = \Delta \tag{15}$$

where $y(\infty) = 1$ and $\Delta \triangleq 2\%$,we have

$$(1+\omega_c t_s)e^{-\omega_c t_s} = 0.02 \tag{16}$$

which yields to

$$t_s = 5.85/\omega_c \tag{17}$$

Once the setting time is given, the parameter ω_c can be determined by (17). Thus, the desired dynamic characteristic of systems is decided. However, the actual output, somehow, can't act the desired dynamic characteristic exactly. Therefore, some margins are considered to make sure the design is dependable. In this paper, ω_c is determined as following

$$\omega_c \approx 10/t_s \tag{18}$$

Then, k_p and k_d can be computed from Eq. (11).

4.2 Tuning of Observer Parameters β_1, β_2 and β_3

For tuning simplicity, Gao[9] suggested the observer gains, β_1, β_2 and β_3 ,can be chosen as following

$$\beta_1 = 3\omega_o, \beta_2 = 3\omega_o^2, \beta_3 = \omega_o^3 \tag{19}$$

where ω_o is denoted as observer bandwidth. This make all three of the observer poles be placed at ω_o. The larger the ω_o, the sooner the disturbance is observed by ESO and cancelled by the controller. But, according to the analysis in section 3, ω_o cannot be too large or it will lie out of the stable region, especially when the order of the controlled plan is higher than two. Therefore, to make the ESO work well in the case of small ω_o, it is necessary to find a new way to tune the observer parameters.

To this end, the transfer function between $z_3(s)$ and $f(s)$ is derived (See appendix 2 for details):

$$\frac{z_3(s)}{f(s)} = \frac{\beta_3}{\beta_3 + \beta_2 s + \beta_1 s^2 + s^3} \tag{20}$$

Due to the following reasons:

(1) in the actual control situation, low and middle frequencies are much more important than high frequencies;

(2) in general, the coefficients of low and middle frequencies (β_3 and β_2) are much more larger than the coefficients of high frequencies (β_1 and 1);

only the first two terms in the denominator of Eq. (20) are often sufficient to describe the character that z_3 tracks f. Thus, we have

$$\frac{z_3(s)}{f(s)} \approx \frac{k}{s+k} \tag{21}$$

where $k \equiv \beta_3 / \beta_2$.

With the knowledge of first-order systems, we can get from Eq. (21) that the larger the $k \equiv \beta_3 / \beta_2$, the sooner the ESO. Furthermore, in accordance with the definition of 2% setting time, we can get

$$T_t \approx 4 / k \tag{22}$$

where T_t is defined as the time for z_3 tracks f. In general, the tracking time T_t of ESO should be smaller than the desired setting time t_s of system.

Now, we go back to the Eq. (19), find that it results in $k = \omega_o / 3$, thus the value of k is limited by ω_o (note that, in many cases, ω_o must be in a small value to make the system stable). To free k from ω_o, we improve Gao's method as following:

$$\beta_1 = 3\omega_o, \beta_2 = 3\omega_o^2, \beta_3 = k\beta_2 \tag{23}$$

where k is a constant which can be conveniently determined by the characteristic of the controlled process. For example, if the desired setting time t_s is larger than 1 second, we can choose $k = 4$, which make $T_t = 1s$ according to Eq. (22).

Once k, the key parameter of ESO, is determined, we choose ω_o using a rule of thumb:

$$\omega_o = 4\omega_c \tag{24}$$

Then β_1, β_2 and β_3 can be computed from Eq. (23).

4.3 Tuning Procedure

The above development can be summarized into the following procedure:

(1) Get the desired setting time t_s^*. (we assume that t_s^* is known in a practical problem)

(2) Let $\omega_c \approx 10 / t_s^*$, compute k_p and k_d from Eq. (11).

(3) Let $\omega_o = 4\omega_c$, and $k = 4$, compute β_1, β_2 and β_3 from Eq. (23).

(4) Monotonously increase the value of parameter B from a small value, until the dynamic performance is satisfactory.

(5) Verify the robustness of controller by Monte-Carlo simulation. (As all the control parameters were decided from the former steps, we let the plant coefficients $\mathbf{p}$ have a $\pm 10\%$ stochastic change, simulate the close-loop system and record the result. By repeating it, 200 times in this paper, we believe the results cover most of the case under parameters perturbation. Then, we can see whether the controller is robust or not).

Note that, since the desired setting time is given, only one parameter B, need to be tuned.

Remark

d. For a wild range of process with the desired setting time is larger than 1 second, we make $k \equiv 4$ in step (3). If the desired setting time is smaller than 1 second, we can adjust the value of k in step (3) from Eq. (22) to enhance the speed that z_3 tracks f. There is an example in section 5 for illustration.

e. It is no doubt that there are always a conflict between performance and robustness. The conflict in ADRC is not as strong as in many other controllers, which make ADRC advanced. Therefore, the result in step (5) is generally satisfactory. However, if a better robustness is expected, we can go on increasing B in step (4), making a trade-off between robustness and performance.

5 Simulation Examples

We shall now look at some examples and demonstrate the use of the method. Comparisons of both performance and robustness will be made with Panagopoulos and Åström's two degrees of freedom PID (2DOF-PID)[14, 15]. The following transfer functions have been considered

$$G_1(s) = \frac{e^{-ps}}{(s+1)^3}, p = 5 \quad G_2(s) = \frac{1-ps}{(s+1)^3}, p = 2$$

$$G_3(s) = \frac{1}{s(s+1)^3}, p = 1$$

$$G_4(s) = \frac{1}{(s+p_1)(1+p_2 s)(1+p_3 s)(1+p_4 s)},$$

$$p_1 = 1, p_2 = 0.2, p_3 = 0.04, p_4 = 0.008$$

$$G_5(s) = \frac{1}{(s+p)^5}, p = 1 \quad G_6(s) = \frac{1}{(s+1)^6}, p = 1$$

$$G_7(s) = \frac{p}{(s+p)(s-1)}, p = 4 \quad G_8(s) = e^{-p\sqrt{s}}, p = 1$$

The first two models describe process of 3-order. G_1 models a process with long dead time and G_2 is a non-minimum phase process. Model G_3 and G_4 are 4-order process, in which G_3 is an integrating process and G_4 is a process with four different poles. Model G_5 and G_6 represent processes of 5-order and 6-order respectively. Model G_7 is considered to show that the proposed method can also be used for unstable systems. G_8 is a distributed parameter system, whose dynamic is described by partial differential equations in nature. Models $G_1 - G_8$ which represent processes with large variations in process dynamic, are included to demonstrate the wide applicability of the design procedure.

Firstly, we make model G_1 as an example to illustrate the design procedure specifically. As the desired setting time $t_s^* \approx 30s > 1s$,we have $\omega_c = 10/t_s^* = 0.33 \approx 0.4$ .Then $\omega_o = 4\omega_c = 1.6$, $k = 4$. Thus, $\beta_1, \beta_2, \beta_3$ and k_p, k_d can be obtained from Eq. (11) and Eq. (23). The last tuning parameter B is monotonously increased by a step of 1 from $B = 1$. We can find that the dynamic performance is satisfactory at $B = 3$. The tuning parameters we get can be summarized as $[B, \omega_c, \omega_o, k] = [3, 0.4, 1.6, 4]$. The tuning procedures of the other processes are similar. Note that, model G_8 is different from others for it's desired setting time is smaller that 1s. Therefore, according to Eq. (22), we increase the value of k to 20.

Fig. 3 shows the responses to changes in set point and load. The details of the design calculations and simulations are summarized in Table 1. We can see that the proposed method yields a faster and smoother respond when compared to Panagopoulos and Åström's 2DOF-PID (There were two sets of controller given in Panagopoulos and Åström's paper, and we choose the one which yields a faster respond to compare). Especially, the overshoot of ADRC is quite smaller. Furthermore, the control signal of ADRC is smaller. Note that, the control problem of model G_7 and G_8 were not presented in Panagopoulos[14]. Therefore, the ADRC controller obtained of them is compared to the corresponding 2DOF-PI controller in Åström [15]. Then, the resulting performance of ADRC is great superior.

The Monte-Carlo method was adopted to verify the robustness of system. For a ±10% stochastic change in $\mathbf{p}$ of models $G_1 - G_8$, we do simulations applying the same controller designed for nominal system, and record the values of $t_s - \sigma\%$. This is repeated 200 times, and the results are shown in Fig. 4. The more concentrated the distribution of $t_s - \sigma\%$, the better the robustness. The bounds of the distribution, $[t_s^-, t_s^+]$ and $[\sigma^-\%, \sigma^+\%]$, are listed in Table 1. We can see from Fig. 4 and Table 1 that, the robustness of ADRC is superior.

6 Conclusions

A simple tuning method for second-order ADRC is presented. Once the desired settling time is given, only one parameter need to be tuned. The method has been applied to a variety of systems, low- and high-order, with long dead times and with right half-plane zeros, unstable and of distributed parameters. It was shown that both satisfactory performance and robustness can be obtained.

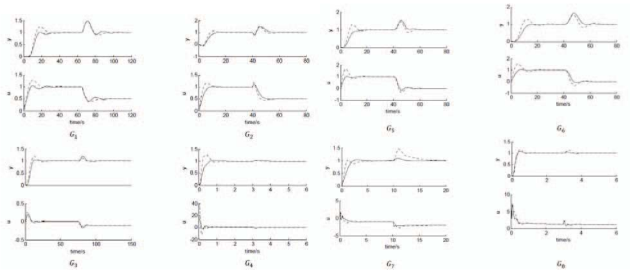

Fig.3 Comparison between ADRC (solid line) and 2DOF-PID (dashed line), showing step response followed by load disturbance of lose loop system

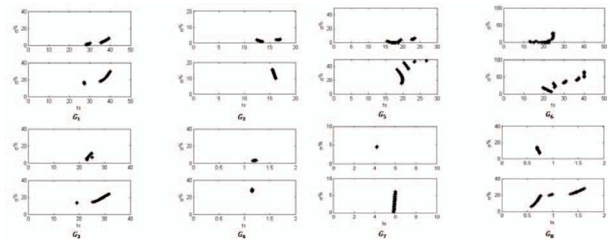

Fig.4 Comparison between ADRC (upside) and 2DOF-PID (downside), showing the distribution of $t_s - \sigma\%$ under parameters perturbation

Table1 Properties of ADRC and 2DOF-PID for system $G_1 - G_8$

Process	Controller	Design parameters*	t_s	$[t_s^-, t_s^+]$	$\sigma\%$	$[\sigma^-\%, \sigma^+\%]$
$G_1(s)$	ADRC	[3, 0.4, 1.6, 4]	29.8	[28.0, 39.6]	1.8	[0.79, 8.50]
	2DOF-PID	[0, 0.56, 0.17, 0.97]	38.3	[27.3, 40.1]	24.4	[14.6, 30.4]
$G_2(s)$	ADRC	[15, 1, 4, 4]	12.9	[12.4, 17.1]	1.4	[0.76, 2.30]
	2DOF-PID	[0, 0.54, 0.26, 0.43]	15.8	[15.5, 16.1]	12.3	[9.58, 15.8]
$G_3(s)$	ADRC	[3, 0.3, 1.2, 4]	23.0	[22.8, 25.0]	4.0	[3.26, 11.6]
	2DOF-PID	[0, 0.68, 0.15, 1.54]	28.7	[19.0, 31.8]	19.0	[13.2, 24.2]
$G_4(s)$	ADRC	[5, 10, 40, 4]	1.2	[1.15, 1.24]	3.1	[2.83, 3.49]
	2DOF-PID	[0.81, 43.1, 228, 5.6]	1.2	[1.14, 1.16]	27.6	[26.6, 30.1]
$G_5(s)$	ADRC	[2, 0.5,2, 4]	16.6	[15.5, 24.8]	0.0	[0, 6.62]
	2DOF-PID	[0, 1.47, 0.63, 1.84]	19.2	[18.3, 26.9]	30.1	[15.1, 48. 3]
$G_6(s)$	ADRC	[3, 0.5, 2, 4]	14.4	[12.8, 25.0]	0.2	[0, 27.4]
	2DOF-PID	[0, 1.15, 0.42, 1.71]	24.8	[19.5, 40.0]	30.3	[6.37, 65.9]
$G_7(s)$	ADRC	[1.5, 2, 8, 4]	4.2	[4.18, 4.26]	4.5	[4.41, 4.60]
	2DOF-PI	[0.5, 3.31, 0.82, 0]	5.9	[5.84, 5.99]	2.8	[0.03, 6.23]
$G_8(s)$	ADRC	[35, 15, 60, 20]	0.7	[0.69, 0.75]	9.4	[6.74, 14.8]
	2DOF-PI	[0.48, 5.31, 27, 0]	0.7	[0.57, 1.63]	16.8	[5.72, 28.3]

*The design parameters of ADRC and 2DOF-PID are $[B, \omega_c, \omega_o, k]$ and $[b, k_p, k_i, k_d]$ respectively.

References

[1] K. J. Åström, T. Hagglund. The future of PID control. *Control Engineering Practice*, 2001, 9(11):1163-1175.

[2] J. Han. Auto-disturbance rejection control and its applications. *Control and Decision*, 1998, 13(1): 19-23.

[3] Z. Gao, Y. Huang, J. Han. An alternative paradigm for control system design. *Proceedings of the IEEE Conference on Decision and Control*, Orlando, FL, 2001: 4578-4585.

[4] J. F. Pan, N. C. Cheung, J. M. Yang. Auto-disturbance rejection controller for novel planar switched reluctance motor. *IEE Proceedings - Electric Power Applications*, 2006, 153(2):307-316.

[5] H. Huanpao, L. Wang, J. Han, F. Gao, Y. Lin. A new synthesis method for unit coordinated control system in thermal power plant - ADRC control scheme. *2004 International Conference on Power System Technology - POWERCON (IEEE Cat. No.04EX902),* Singapore, 2004: 133-138.

[6] J. B. Su, W. B. Qiu, H. Y. Ma, P. Y. Woo. Calibration-free robotic eye-hand coordination based on an auto disturbance-rejection controller. *IEEE Transactions on Robotics and Automation*, 2004, 20(5): 899-907.

[7] C. Tong, L. Wang, X. Yin, J. Dong. Finishing width control system based on active disturbance rejection control and chaos optimization. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, Dalian, China, 2006: 7751-7755.

[8] Y. L. Shi, C. Z. Hou. Auto-disturbance-rejection controller design based on RBF neural networks. *Intelligent Control and Automation*, 2006, 344: 500-505.

[9] Z. Gao. Scaling and Bandwidth-Parameterization based Controller Tuning. *Proceedings of the American Control Conference, Denver, CO,* United States, 2003: 4989-4996.

[10] B. Sun, Z. Gao. A DSP-based active disturbance rejection control design for a 1-kW H-bridge DC-DC power converter. *IEEE Transactions on Industrial Electronics*, 2005, 52(5): 1271-1277.

[11] R. Miklosovic, Z. Gao. A robust two-degree-of-freedom control design technique and its practical application. *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)*, Seattle, WA, United States, 2004: 1495-1502.

[12] X. Yao, Q. Wang, W. Liu, R. Liu. Two-order ADRC control for general industrial plants. *Control Engineering China*, 2002, 9(5): 59-62.

[13] I. Beichl, F. Sullivan. Monte Carlo methods. *Computing in Science and Engineering*, 2006, 8(2): 7-8.

[14] H. Panagopoulos, K. J. Åström, T. Hagglund. Design of PID controllers based on constrained optimization. *IEE Proceedings - Control Theory and Applications*, 2002, 149 (1): 32-40.

[15] K. J. Åström, H. Panagopoulos, T. Hagglund. Design of PI controllers based on non-convex optimization. *Automatica*, 1998, 34(5): 585-601.

Appendices

A.1 Examples for Stable Region

$$(a)\ \ G_a(s)=\frac{-2s+1}{(s+1)^3}$$

$$(b)\ \ G_b(s)=\frac{1}{(s+1)(1+0.2s)(1+0.04s)(1+0.008s)}$$

$$(c)\ \ G_c(s)=\frac{1}{(s+1)^5}$$

$$(d)\ \ G_d(s)=\frac{100}{(s+1)^2}\left(\frac{1}{s+1}+\frac{0.5}{s+0.05}\right)$$

All these results show that, as *B* increases within certain bound, the stable area in $\omega_o-\omega_c$ plane is expanded. In fact, for some special processes, there exists a constant value B_0, and the above law is just correct when *B* increases within the range of [0, B_0]. For example, in the case of G_d, the stable area is no longer expanded, but contracted, while *B*>5. However, in our tuning procedure, *B* shall not be too large, and we consider that the law is always correct.

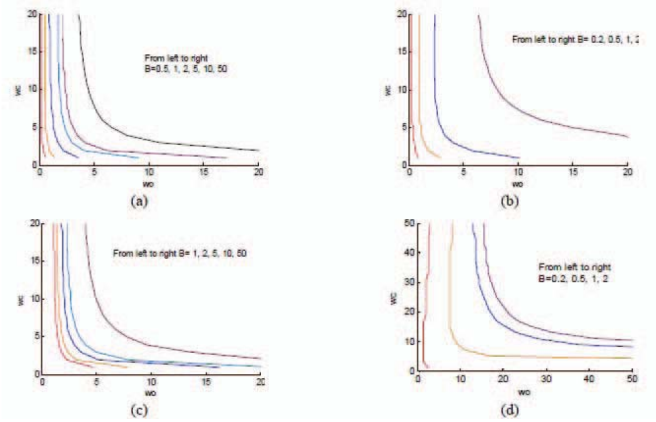

Fig.A1 Stable region of models G_a-G_d

A.2 Transfer Function Between $z_3(s)$ and $f(s)$!

Rewrite the Eq. (2) and Eq. (3) as following

$$\dot z_1 = z_2+\beta_1(y-z_1) \qquad (A1.1)$$

$$\dot z_2 = z_3+\beta_2(y-z_1)+Bu \qquad (A1.2)$$

$$\dot z_3 = \beta_3(y-z_1) \qquad (A1.3)$$

$$\ddot y = f+Bu \qquad (A2)$$

Combine Eq. (A1.2) and Eq.(A2),and cancel the term Bu yielding

$$f = z_3+\beta_2(y-z_1)+(\ddot y-\dot z_2) \qquad (A3)$$

Taking the Laplace transform of Eq.(A1.1) and Eq.(A3),we have

$$sz_1(s)=z_2(s)+\beta_1[y(s)-z_1(s)] \qquad (A4)$$

$$f(s)=z_3(s)+\beta_2[y(s)-z_1(s)]+[s^2y(s)-sz_2(s)] \qquad (A5)$$

Combine Eq.(A4) and Eq.(A5), and cancel the term $z_2(s)$, yielding

$$f(s)=z_3(s)+\beta_2[y(s)-z_1(s)]+\beta_1 s[y(s)-z_1(s)]+ s^2[y(s)-z_1(s)] \qquad (A6)$$

Taking the Laplace transform of Eq.(A1.3),we have

$$z_3(s)s=\beta_3[y(s)-z_1(s)] \qquad (A7)$$

Times s to both sides of Eq. (A6), then substitute Eq. (A7) into it, yielding

$$f(s)s=(\beta_3+\beta_2 s+\beta_1 s^2+s^3)[y(s)-z_1(s)] \qquad (A8)$$

Divide Eq. (A7) by Eq. (A8), yielding

$$\frac{z_1(s)}{f(s)}=\frac{\beta_3}{\beta_3+\beta_2 s+\beta_1 s^2+s^3} \qquad (A9)$$