Scaling and Bandwidth-Parameterization Based Controller Tuning

Zhiqiang Gao
Dept. of Electrical and Computer Engineering
Cleveland State University, Cleveland, Ohio 44115

Abstract: A new set of tools, including controller scaling, controller parameterization and practical optimization, is presented to standardize controller tuning. Controller scaling is used to frequency-scale an existing controller for a large class of plants, eliminating the repetitive controller tuning process for plants that differ mainly in gain and bandwidth. Controller parameterization makes the controller parameters a function of a single variable, the loop-gain bandwidth, and greatly simplifies the tuning process. Practical optimization is defined by maximizing the bandwidth subject to the physical constraints, which determine the limiting factors in performance. Collectively, these new tools move controller tuning in the direction of science.

Keywords: Tuning, PID, Scaling, Auto-Scaling, Auto-Tuning, Adaptive Self-Tuning, Gain-Scheduling, Disturbance Observer, Computer Aided Controller Design

I. Introduction

The proportional-integral-derivative (PID) controller, first proposed by N. Minorsky in 1922 [1], is used in over 90% of current industrial control applications [2]. In addition, the controller parameters are still determined by rules of thumb, such as look-up tables [3].

Classical control theory has successfully provided the analysis and design tools for single-input single-output (SISO), linear, time-invariant systems, since the 1940s. The PID design approach moved from empirical (i.e., ad hoc tuning methods such as Ziegler and Nichols tuning tables [3]) to analytical (i.e., pole placement, frequency response). In particular, the frequency response-based methods (Bode and Nyquist plots, stability margins, lead-lag compensators) have proved to be especially useful in solving control problems.

Historically, determining controller parameters to meet design specifications (tuning), rather than the design of the controller itself, has been the main concern in industry. Most industrial plants are inherently stable and consist of SISO subsystems. Simple PID controllers implemented in a digital form can usually meet the performance needs. But the problem of tuning has hardly received much attention in the existing control theory. The variety of ad-hoc tuning algorithms in industrial control products shows the lack of in-depth understanding of the problem and the need for further research.

The PID gains are commonly "tuned" on a trial-and-error basis in practice. A general lack of knowledge regarding the relationship between "design objectives" and "practical performance measures" makes the use of well-known design techniques such as Root Locus (pole-placement) and linear optimal control difficult. For example, in pole-placement design, the objective is to place the closed-loop poles at given locations, based on the understanding of how the location of poles affects the transient response of a system. Although the transient response is usually an important design consideration, it is not the only issue in pole-placement methods with which to contend. The pole-placement method is ill-equipped to handle other common design specifications including disturbance rejection, noise sensitivity, stability margins, and smoothness of the control signal.

This lack of design insight leads to the heuristic nature of the tuning methods implemented in industry. Furthermore, the practice of control design and tuning tended in the direction of art rather than science. This paper presents a comprehensive approach that moves control design and tuning in the direction of science.

The paper is organized as follows. Controller scaling is introduced in Section II. Parameterization and optimization of model-based controllers are discussed in Section III. Design, parameterization, and optimization of a model-independent controller design method are discussed in Section IV. Finally, some concluding remarks are given in Section V.

II. Controller Scaling

A controller is generally not "portable", i.e., a controller designed for one plant is usually not applicable to another plant. The objective of controller scaling is to make a good controller "portable", much like the filter design. With the bandwidth, pass band, and stop band requirements given, the filter design is straightforward. First, a unit bandwidth filter, such as an *nth* order Chebeshev filter H(s), is found that meets the pass band and stop band specifications; then it is frequency scaled by ω_0 to achieve the desired bandwidth of ω_0. It is shown in this section that the controller design can be performed similarly.

2.1 Frequency Scale and Time Scale

Consider a unit feedback control system with the plant $G_p(s)$ and the controller $G_c(s)$, as shown in Figure 2.1. Assume that $G_c(s)$ was designed for desired command following, disturbance and noise rejection, and stability robustness. Now, consider a similar class of plants $G_p(s/\omega_p)$, for any given ω_p. Can a controller be found without a repetition of the tedious loop shaping design process?

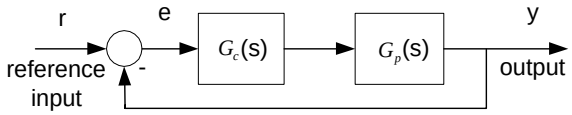

Figure 2.1 Feedback Control Configuration

Definition2.1: Denote ω_p as the frequency scale of the plant $G_p(s/\omega_p)$ with respect to $G_p(s)$, and $\tau_p=1/\omega_p$, the corresponding time scale.
Definition2.2: Denote k as the gain scale of the plant $kG_p(s)$ with respect to $G_p(s)$.

The differences in many industrial control problems can be described in terms of the frequency and gain scales defined here, such as the temperature processes with different time constants (in first-order transfer functions), motion control problems with different inertias, motor sizes and frictions.

The use of the scales allows the development of a generic solution for a class of problems. Any linear time-invariant plant, strictly proper and without a finite zero, can be reduced to one of the following forms

$$\frac{1}{s+1}, \frac{1}{s}, \frac{1}{s^2+2\xi s+1}, \frac{1}{s(s+1)}, \frac{1}{s^2}, \frac{1}{s^3+\xi_1 s^2+\xi_2 s+1}, \ \ldots \quad (2.1)$$

through gain and frequency scaling. For example, the motion control plant of

$$G_p(s)=\frac{23.2}{s(s+1.41)}$$

is simply a variation of a generic motion control plant

$$G_p(s)=\frac{1}{s(s+1)}$$

with gain and frequency scales of k = 11.67 and ω_p = 1.41, respectively. That is

$$\frac{23.2}{s(s+1.41)}=\frac{11.67}{\frac{s}{1.41}(\frac{s}{1.41}+1)} \tag{2.2}$$

Equation (2.1) covers the majority of industrial control plants, which are usually approximated by a pure first order or a second order transfer function response. For completeness, (2.1) may be appended by terms such as

$$\frac{s+1}{s^2+2\xi s+1}, \frac{s^2+2\xi_z s+1}{s^3+\xi_1 s^2+\xi_2 s+1}, \ \ldots \tag{2.3}$$

to include systems with finite zeros. Furthermore, for a particular class of plants, the scaling concept can be applied accordingly to reflect the unique characteristics of the class. For example, plants commonly seen in motion control with significant resonant problems can be modeled and scaled as:

$$\frac{k}{\frac{s}{\omega_p}(\frac{s}{\omega_p}+1)}\frac{\left(\frac{s}{\omega_{rz}}\right)^2+2\xi_z\frac{s}{\omega_{rz}}+1}{\left(\frac{s}{\omega_{rp}}\right)^2+2\xi_p\frac{s}{\omega_{rp}}+1}\Rightarrow\frac{1}{s(s+1)}\frac{\left(\frac{s}{m}\right)^2+2\xi_z\frac{s}{m}+1}{\left(\frac{s}{n}\right)^2+2\xi_p\frac{s}{n}+1} \tag{2.4}$$

where the resonant frequencies satisfy $\omega_{rp}=n\omega_p$, $\omega_{rz}=m\omega_p$. Such problems with multiple frequency scales, ω_p, $n\omega_p$, and $m\omega_p$, are referred to as multi-scale problems.

2.2 Controller Scaling

Theorem 2.1: Assuming $G_c(s)$ is a stabilizing controller for plant $G_p(s)$, and the loop gain crossover frequency is ω_c, then the controller

$$\bar{G}_c(s)=G_c(s/\omega_p)/k \tag{2.5}$$

will stabilize the plant $\bar{G}_p(s)=kG_{p1}(s/\omega_p)$, and new loop gain $\bar{L}(s)=\bar{G}_p(s)\bar{G}_c(s)$ will have a bandwidth of $\omega_c\omega_p$, and the same stability margins of L(s)= $G_p(s)G_c(s)$.

Proof: The proof is obvious since $\bar{L}(s)=L(s/\omega_p)$ Q.E.D.

Note that the new closed-loop system has the same frequency response shape as the original system, except that it is shifted by ω_p. That is, all feedback control properties, such as disturbance and noise rejection, as well as stability robustness, are retained from the previous design, except that their frequency ranges are all shifted by ω_p.

The use of controller scaling eliminates the repetitiveness of control design and tuning in industry today. Applying controller scaling in (2.5) to a PID controller,

$$G_c(s)=k_p+\frac{k_i}{s}+k_d s \tag{2.6}$$

results in

$$G_c(s)=(k_p+k_i\frac{\omega_p}{s}+k_d\frac{s}{\omega_p})/k \tag{2.7}$$

That is, the new PID gains, $\bar{k}_p$, $\bar{k}_i$, *and* $\bar{k}_d$ are obtained from the original ones as

$$\bar{k}_p=\frac{k_p}{k},\bar{k}_i=\frac{k_i\omega_p}{k},\bar{k}_d=\frac{k_d}{k\omega_p} \tag{2.8}$$

Example 2.1 Consider one of the plants in (2.1), which has a transfer function of

$$G_p(s)=\frac{1}{s^2+s+1}$$

and the PID controller gains of $k_p=3$, $k_i=1$, and $k_d=2$. Now, assume that the plant has changed to

$$G_p(s)=\frac{1}{(\frac{s}{10})^2+\frac{s}{10}+1}$$

The new PID gains determined from (2.8) are $\bar{k}_p=3,\bar{k}_i=10,\bar{k}_d=.2$. Applying a unit step function as the set point, the responses of the original controller and the scaled controller are illustrated in Figure 2.2, demonstrating that the new response is exactly the same as the original scaled by $\tau=1/\omega_p$, $\omega_b=10$ rad/sec.

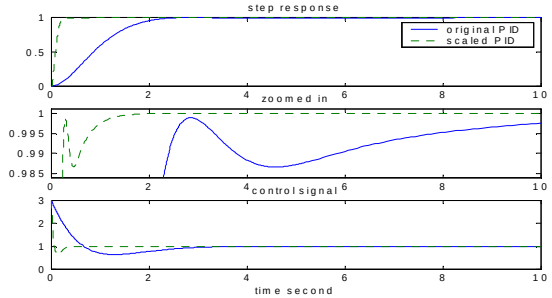

Figure 2.2 Auto-Scaling of PID

From the frequency response of loop gain transfer function (i.e., the product of the controller and the plant in this case), it is determined that the gain margins of both systems are infinite, and the phase margins are both 82.372 degrees; and the 0 dB crossover frequency for both systems are 2.3935 and 23.935 r/s, respectively.

III. Bandwidth-Parameterization and Optimization

The controller scaling method that is demonstrated above resolved the long-standing issue of portable controller design. Once a good controller is obtained for one plant, it is easily scaled to control similar plants that are different only in gain and frequency scales, thus avoiding tedious control redesign.

However, there are always differences in design specifications and constraints for different control problems. The question we want to address now is how do we realistically "optimize" controller design for each application, i.e., how does an engineer get "the most performance" out of a given set of hardware/software?

For the sake of simplicity and practicality, bandwidth, denoted as ω_c, is selected here as the measure of performance. It is well-known that higher bandwidth corresponds to better command

following, disturbance rejection and sensitivity to parameter variations. On the other hand, achievable bandwidth is limited by the presence of sensor noise and dynamic uncertainties. Most design trade-offs are reflected in the selection of the ω_c. For this reason, a unique controller parameterization is proposed that makes ω_c the only design parameter to be determined (tuned).

Definition 3.1: Bandwidth-Parameterization, also known as ω_c-Parameterization, refers to assigning all closed-loop poles at $-\omega_c$ and making all parameters of the controller a function of ω_c. Here, ω_c is denoted as the bandwidth of the feedback control system and $G_c(s,\omega_c)$ as the controller.

The concept of ω_c-Parameterization applies to all controller design methods. The key is to make all controller parameters as functions of ω_c. Two particular design techniques are used in the following subsections to show how this is accomplished.

3.1 Parameterization of the Pole-Placement Design

Consider the normalized plants in (2.1) and assume that the desired closed-loop transfer functions are

$$\frac{\omega_c}{s+\omega_c}\ ,\ \frac{\omega_c^2}{(s+\omega_c)^2}\ ,\ \frac{\omega_c^3}{(s+\omega_c)^3},\ \ldots \tag{3.1}$$

Applying the simple pole-placement design to the first and second order plants in (2.1), a set of ω_c-parameterized controllers is obtained as shown in Table 2.1. Similar solutions for higher order plants in (2.1) can be easily obtained. Note that with ω_c as the only parameter, the tuning of the controller is greatly simplified.

Table 3.1 Examples of ω_c-parameterized controllers

Plant $G_p(s)$	$\frac{1}{s+1}$	$\frac{1}{s}$	$\frac{1}{s^2+2\xi s+1}$	$\frac{1}{s(s+1)}$	$\frac{1}{s^2}$
Controller $G_c(s,\omega_c)$	$\frac{\omega_c(s+1)}{s}$	ω_c	$\omega_c^2\frac{s^2+2\xi s+1}{s(s+2\omega_c)}$	$\frac{\omega_c^2(s+1)}{s+2\omega_c}$	$\frac{\omega_c^2 s}{s+2\omega_c}$
Closed-loop Tranferfunction	$\frac{\omega_c}{s+\omega_c}$	$\frac{\omega_c}{s+\omega_c}$	$\frac{\omega_c^2}{(s+\omega_c)^2}$	$\frac{\omega_c^2}{(s+\omega_c)^2}$	$\frac{\omega_c^2}{(s+\omega_c)^2}$

3.2 Parameterization of the Loop-Shaping Design

The loop-shaping design method, also known as the frequency response-based controller design method, is an insightful method used by many engineers in practice. It is the only design method in classical control theory that comprehensively addresses multiple design concerns, such as transient response, disturbance rejection, stability margins, and noises. Loop-shaping as a concept, and as a design tool, helps practicing engineers greatly in improving the PID loop performance and stability margins. For example, a PID with a lead-lag compensator is commonly seen in industry today. Unfortunately, manipulating loop gain frequency response and managing competing performance measures can be tedious and, in some cases, frustrating. A computer algorithm to automate this process was proposed in [4-7], which not only replaces the manual design but also leads to a self-tuning controller design and implementation.

The loop-shaping process consists of two steps: 1) convert all design specifications to loop gain constraints, as shown in Figure 3.1; and 2) find a controller $G_c(j\omega)$ to meet the specifications.

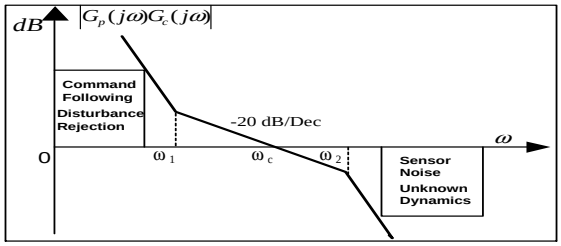

Figure 3.1 Loop-shaping

Consider Figure 3.1, the desired loop-gain can be characterized as

$$L(s)=G_p(s)G_c(s)=\left(\frac{s+\omega_1}{s}\right)^m\frac{1}{\frac{s}{\omega_c}+1}\frac{1}{\left(\frac{s}{\omega_2}+1\right)^n} \tag{3.2}$$

where ω_c is the bandwidth, and

$$\omega_1<\omega_c,\ \omega_2>\omega_c,\ m\geq 0,\ \text{and}\ n\geq 0 \tag{3.3}$$

are selected to meet constraints demonstrated in Figure 3.1. Both *m* and *n* are integers. The default values for ω_1 and ω_2 are

$$\omega_1=\omega_c/10\ \text{and}\ \omega_2=10\omega_c \tag{3.4}$$

which yields a phase margin of approximately 90 degrees.

Once the appropriate loop gain constraints are derived and the corresponding lowest order L(s) in (3.2) is selected, the controller can be determine from

$$G_c(s)=\left(\frac{s+\omega_1}{s}\right)^m\frac{1}{\frac{s}{\omega_c}+1}\frac{1}{\left(\frac{s}{\omega_2}+1\right)^n}G_p^{-1}(s) \tag{3.4}$$

An additional constraint on *n* is that

$$\frac{1}{\frac{s}{\omega_c}+1}\frac{1}{\left(\frac{s}{\omega_2}+1\right)^n}G_p^{-1}(s) \tag{3.5}$$

is proper. This design is valid only if the plant is minimum phase, as assumed. For a non-minimum phase plant, a minimum phase approximation of $G_p^{-1}(s)$ should be used instead.

A compromise between ω_1 and the phase margin can be made by adjusting ω_1 upwards, which will increase the low frequency gains at the cost of reducing the phase margin. Similar compromise can be made between phase margin and ω_2.

3.3 Parameterization of State Feedback Controller and State Observers

The state feedback controller can be represented by

$$u=r+K\hat{x} \tag{3.6}$$

and is based on the state space model of the plant:

$$\begin{aligned}\dot{x}(t)&=Ax(t)+Bu(t)\\ y(t)&=Cx(t)+Du(t)\end{aligned} \tag{3.7}$$

Here, u is the control signal, r is the setpoint for the output to follow, x is the state vector, and {A,B,C,D} matrices are given. When the state x is not accessible, a state observer (SO):

$$\dot{\hat{x}}=A\hat{x}+Bu+L(y-\hat{y}) \tag{3.8}$$

is often used to find its estimate, $\hat{x}$. The state feedback gain K and the observer gain L are determined using the standard eigenvalue assignment technique based on the equations:

$$eig(A+BK)=\lambda_c(s)\ \text{and}\ eig(A+LC)=\lambda_o(s)$$

where $\lambda_c(s)$ and $\lambda_o(s)$ are polynomials of s that are chosen by the designer. Usually K and L have many parameters and are

difficult to tune. The parameterization of state feedback and state observer gains are achieved by making

$\lambda_c(s)=(s+\omega_c)^n = eig(A+BK)$ and $\lambda_o(s)=(s+\omega_o)^n = eig(A+LC)$ (3.9)

where ω_c and ω_o are the bandwidth of the state feedback system and the state observer, respectively, and n is the order of the system. Now the tuning of K and L become much simpler and intuitive since ω_c and ω_o have explicit physical meanings. An example will be given in a later section of this paper.

3.4 ω_c-Optimization

Consider a common controller design scenario in Figure 3.2. The goal of controller design is to achieve maximum closed-loop bandwidth, subject to design constraints. This is denoted as the ω_c-Optimization.

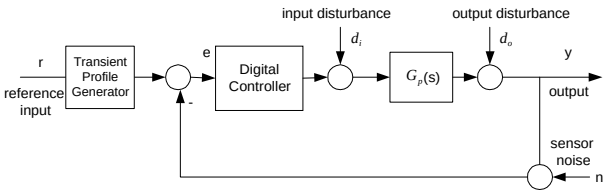

Figure 3.2 A Common Controller Design Scenario

Design Procedure:

1. From the given plant transfer function, determine its frequency and gain scales, ω_p and k;
2. Determine the type of controller required (either a simple pole placement design in Table 3.1, or a more comprehensive design in equation (3.4)) based on the design specifications;
3. Select the $G_c(s, \omega_c)$ corresponding to the scaled plant in the form of (2.1);
4. Scale the controller to $\frac{1}{k}G_c(\frac{s}{\omega_p},\omega_c)$
5. Digitize and implement $G_c(s/\omega_p, \omega_c)/k$;
6. Set an initial value of ω_c based on the bandwidth requirement from the transient response;
7. Gradually increase ω_c while performing tests on the simulator, until either one of the followings is observed:
   a. Control signal becomes excessively noisy;
   b. Control signal exceeds physical limits in the magnitude and/or the rate of change;
   c. Indication of instability (oscillatory behavior)

Example 3.1

Consider a motion control test bed as pictured in Figure 3.3. The mathematical model of the motion system was derived and verified in hardware test, as

$$\ddot{y} = (-1.41\dot{y} + 23.2T_d) + 23.2u \qquad (3.10)$$

where y is the output position, u is the control voltage sent to the power amplifier that drives the motor, and T_d is the torque disturbance.

Figure 3.3 The Motion Control Test Plant

The design objective is to rotate the load one revolution in one second with no overshoot. The physical characteristics of this control problem are: 1) |u|<3.5 volt, 2) the sampling rate is 1 kHz, 3) the sensor noise is 0.1% white noise, 4) there could be a torque disturbance up to 10% of the maximum torque, and 5) the control signal should be smooth with noise level limited to ±100 mV.

The plant transfer function is

$$G_p(s) = \frac{k}{\frac{s}{\omega_p}(\frac{s}{\omega_p}+1)}, \; k = 11.67 \text{ and } \omega_p = 1.41$$

A simple PD design of

$$u = k_p(r-y) + k_d(-\dot{y})$$

with

$$k_p = \omega_c^2 \text{ and } k_d = 2\omega_c - 1$$

results in a closed-loop transfer function of

$$G_{cl}(s) = \frac{\omega_c^2}{(s+\omega_c)^2}$$

Considering the plant gain scale of k and the frequency scale of ω_p,, the PD gains are then scaled as

$$k_p = \frac{\omega_c^2}{k} = .086\omega_c^2 \text{ and } k_d = \frac{2\omega_c - 1}{k\omega_p} = .061(2\omega_c - 1)$$

To avoid noise corruption of the control signal, an approximate differentiator

$$\frac{s}{(\frac{s}{10\omega_c}+1)^2}$$

is applied with a corner frequency of $10\omega_c$ selected so that the approximation of the differentiator does not introduce problematic phase delays at the crossover frequency.

Using the conventional root locus method, the one second settling time would require a closed-loop bandwidth of 4 rad/sec. By applying the proposed single parameter design and tuning method, it was determined that the maximum ω_c is 20 rad/sec, beyond which the noise level in the control signal will exceed the design limit. Simulation results are illustrated in Figure 3.4. Note that a step disturbance of 1 volt, which corresponds to 10% of the maximum torque, is added at t=3 sec. to test the disturbance rejection property of the controller. A trapezoidal transient profile is used as the desired output trajectory.

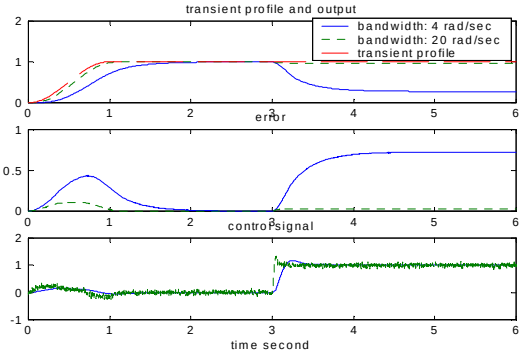

Figure 3.4 Effects of Transient Profile

IV. Parameterization of Model-Independent Controllers

The design techniques presented so far are based on a linear time-invariant mathematical model of the plant. In practice, however, the physical systems are usually nonlinear and time varying. The

lack of a descriptive mathematical model is often a limitation to the systematic controller design. In this section, the parameterization and optimization methods introduced above are applied to a model-independent design approach proposed in [8-11].

4.1 A Model Independent Controller Design

For the sake of simplicity, consider a second order plant

$$\ddot{y} = -a\dot{y} - by + w + bu \tag{4.1}$$

where *y* and *u* are output and input, respectively, and *w* is the external disturbance. Here, the parameters, *a* and *b*, are both unknown, although we have some knowledge of *b*, i.e., $b_0 \approx b$. Rewrite (4.1) as

$$\ddot{y} = -a\dot{y} - by + w + (b - b_0)u + b_0 u = f + b_0 u \tag{4.2}$$

where $f = -a\dot{y} - by + w + (b - b_0)u$. Here, *f* is referred to as the *generalized disturbance*, or *disturbance*, because it represents both the unknown internal dynamics, $-a\dot{y} - by + (b - b_0)u$ and the external disturbance *w*.

The basic idea is to obtain $\hat{f}$, an estimate of *f*, and use it in the control law, $u = (-\hat{f} + u_0)/b_0$, to reduce the plant to a unit-gain double integrator control problem, $\ddot{y} = (f - \hat{f}) + u_0$, with a disturbance $(f - \hat{f})$.

The plant in (4.2) is written in state equation form

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + b_0 u \\ \dot{x}_3 = h \\ y = x_1 \end{cases} \tag{4.3}$$

with $x_3 = f$ added as an augmented state, and $h = \dot{f}$ as unknown disturbance. Now *f* can be estimated using a state observer based on the state space model

$$\begin{aligned} \dot{x} &= Ax + Bu + Eh \\ y &= Cz \end{aligned} \tag{4.4}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ b_0 \\ 0 \end{bmatrix}, C = [1 \quad 0 \quad 0], E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The state space observer, denoted as the linear extended state observer (LESO), of (4.4) is constructed as

$$\begin{aligned} \dot{z} &= Az + Bu + L(y - \hat{y}) \\ \hat{y} &= Cz \end{aligned} \tag{4.5}$$

and L is the observer gain vector, which can be obtained using any known method such as the pole placement technique,

$$L = [\beta_1 \; \beta_2 \; \beta_3]^T \tag{4.6}$$

where $[]^T$ denotes transpose. With the state observer properly designed, the controller is given by

$$u = \frac{-z_3 + u_0}{b_0} \tag{4.7}$$

Ignoring the estimation error in z_3, the plant is reduced to a unit gain double integrator,

$$\ddot{y} = (f - z_3) + u_0 \approx u_0 \tag{4.8}$$

which is easily controlled with a PD controller

$$u_0 = k_p(r - z_1) - k_d z_2 \tag{4.9}$$

where *r* is the setpoint. Note that $-k_d z_2$, instead of $k_d(\dot{r} - z_2)$, is used to avoid differentiation of the setpoint and to make the closed-loop transfer function pure second order without a zero:

$$G_{cl} = \frac{k_p}{s^2 + k_d s + k_p} \tag{4.10}$$

Here, the gains can be selected as

$$k_d = 2\xi\omega_c \;\text{ and }\; k_p = \omega_c^2 \tag{4.11}$$

where ω_c and ζ are the desired closed loop natural frequency and damping ratio. ξ is selected to avoid any oscillations.

Remarks:

1. The disturbance observer-based PD controller achieves zero steady state error without using an integrator;
2. The design is model independent. The only parameter needed is the approximate value of b in (4.1).
3. The combined effects of the unknown disturbance and the internal dynamics are treated as a generalized disturbance. By augmenting the observer to include an extra state, it is actively estimated and canceled out, thereby achieving active disturbance rejection.
4. The PD controller in (4.9) can be replaced with a more elaborate loop-shaping design, if necessary.

The above controller, (4.7) and (4.9), combines with the LESO to actively compensate for the disturbances and is, therefore, denoted as Linear Active Disturbance Rejection Controller (LADRC). This is a special case of the ADRC controller originally proposed by Han in [8, 9], which uses nonlinear gains in place of the linear ones in (4.6) and (4.9). While the nonlinear gains may be more effective, they also produce extra complexity in the control algorithm implementation and tuning. In addition, stability of the system becomes more difficult to prove. The discussions in this paper are limited to the linear case.

4.2 Stability

Let $e_i = x_i - z_i$, *i=1, 2, 3*, and combine equation (4.5) and (4.6) and subtract it from (4.4), the error equation can be written as:

$$\dot{e} = A_e e + Eh \tag{4.12}$$

where

$$A_e = A - LC = \begin{bmatrix} -\beta_1 & 1 & 0 \\ -\beta_2 & 0 & 1 \\ -\beta_3 & 0 & 0 \end{bmatrix}$$

and *E* is defined in (4.4). Obviously, the LESO is bounded-input bounded-output (BIBO) stable if the roots of the characteristic polynomial of A_e

$$\lambda(s) = s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 \tag{4.13}$$

are all in the left half plane and *h* is bounded. The separation principle also applies to LADRC.

Theorem 4.1: The LADRC design from (4.5) to (4.8) yields a BIBO stable closed-loop system if the observer in (4.5) and (4.6) and the control law (4.8) for the double integrator are stable.

Proof: Equation (4.7) and (4.9) can be combined into a state feedback form of $u=(1/b_0)[-k_p \; -k_d \; -1]z=Fz$, where $F = (1/b_0)[-k_p \; -k_d \; -1]$. The closed-loop system is then represented by the state-space equation of

$$\begin{bmatrix}\dot{x}\\ \dot{z}\end{bmatrix}=\begin{bmatrix}A & \bar{B}F\\ LC & A-LC+\bar{B}F\end{bmatrix}\begin{bmatrix}x\\ z\end{bmatrix}+\begin{bmatrix}\begin{bmatrix}\bar{B} & E\end{bmatrix}\\ \begin{bmatrix}\bar{B} & 0\end{bmatrix}\end{bmatrix}\begin{bmatrix}r\\ h\end{bmatrix} \quad (4.14)$$

where $\bar{B}=B/b_0$. (4.14) is BIBO stable if its eigenvalues are in the left half plane. By applying elementary row and column operations, it is obvious that the closed-loop eigenvalues satisfy

$$eig\left(\begin{bmatrix}A & \bar{B}F\\ LC & A-LC+\bar{B}F\end{bmatrix}\right)=eig\left(\begin{bmatrix}A+\bar{B}F & \bar{B}F\\ 0 & A-LC\end{bmatrix}\right)$$

$$=eig\left(A+\bar{B}F\right)\cup eig\left(A-LC\right)$$

$$=\{\text{roots of } s^2+k_d s+k_p\}\cup\{\text{roots of } s^3+\beta_1 s^2+\beta_2 s+\beta_3\} \quad \text{Q.E.D.}$$

Since *r*, as the reference signal, is always bounded, the only nontrivial condition on the plant is that $h=\dot{f}$ is bounded. In other words, the disturbance *f* must be differentiable, which is a reasonable assumption.

4.3 Parameterization and ω_o-Optimization of LESO

State observers provide information on the internal states of the plants that are otherwise unavailable. They also function as noise filters. The primary concern in observer design is the selection of the bandwidth. The closed-loop observer, or the correction term, $L(y-\hat{y})$ in particular, is used to accommodate the unknown initial states, the uncertainties in parameters, and the disturbances. The ability to meet the control requirements is largely dependent on the speed at which the observer can track the states and, in case of LESO, the disturbance $f(t,x_1,x_2,w)$. In general, observers should be made to work as fast as the measurement noise allows.

4.3.1 Limiting Factors In Observer Design

There are three common limiting factors in design: 1) sensor noise; 2) sampling rate; 3) dependency on the state space model of the plant.

The level of sensor noise is hardware-dependent but it is reasonable to assume that it is a white noise with the peak value at 0.1% to 1% of that of the output. The observer bandwidth should be selected so that there is no significant oscillation in its states due to noises. This bandwidth will also be limited by the given sampling rate.

The dependency of the design on the state space model limits its application to problems where this type of model is available. The observer also becomes sensitive to the inaccuracies of the model, which cause the plant dynamic to change. These limitations make the LESO, which is largely model-independent as shown above, unique and appealing.

4.3.2 Parameterization of LESO

Although the pole placement technique is used widely for the state observer design, the location of observer poles has never been systematically addressed. In practice, a compromise is made between the speed at which the observer tracks the states and its sensitivity to the sensor noises. Sampling rate also limits how fast the observer can operate. These design issues are addressed here using the same parameterization and optimization method in the controller design, introduced in Section III.

Definition 4.1: ω_o-Parameterization refers to assigning all observer eigenvalues at $-\omega_o$ and making all parameters of an observer a function of ω_o. Here, ω_o is denoted as the bandwidth of the observer.

The plant (4.4) for which LESO is designed has all three poles at the origin. Intuitively, the observer will be the least sensitive to noises if the observer gains in (4.6) are the smallest for a given bandwidth of ω_o (a proof of this would be interesting). But the observer gains are proportional to the distance from the poles of the plant to those of the observer. Both this and simplicity suggest that all three of the observer poles should be placed at $-\omega_o$, or equivalently,

$$\lambda(s)=s^3+\beta_1 s^2+\beta_2 s+\beta_3=(s+\omega_o)^3 \quad (4.15)$$

That is,

$$\beta_1=3\omega_o,\ \beta_2=3\omega_o^2,\ \beta_3=\omega_o^3 \quad (4.16)$$

Remarks:

1) Equations (4.15) and (4.16) are easily extended to an *nth* order LESO;
2) This parameterization method can also be readily extended to the Luenberger Observer for arbitrary A, B, and C matrices, using the following steps:
   a. Obtain $\{\bar{A},\bar{B},\bar{C}\}$ as observable canonical form of $\{A,B,C\}$;
   b. Determine the observer gain, $\bar{L}$, so that all the poles of the observer are at $-\omega_o$;
   c. Use the inverse state transformation to obtain the observer gain, L, for $\{A,B,C\}$.

   The parameters in L are all functions of ω_o which can be easily adjusted.

4.3.3 ω_o-Optimization

The practical optimality of the observer is similarly defined to that of the controller as shown in Section III. That is, increase the bandwidth as much as allowed by the hardware and software limitations, which are mainly noises and the fixed sampling rate.

Definition 4.2: ω_o-Optimization refers to maximizing the observer bandwidth ω_o, subject to the condition that the sensitivity to sensor noises and the delay in sampling are acceptable.

In general, the faster the LESO, the sooner the disturbance is observed and cancelled by the controller. Therefore, the ω_o-optimization should be applied for each application. More importantly, the repetitive observer design and tuning is reduced to the adjustment of one parameter: ω_o.

Relationship between ω_o and ω_c

A common rule of thumb is to choose

$$\omega_o\approx 3\sim 5\omega_c \quad (4.17)$$

This applies to the state feedback control system where ω_c is determined based on the transient response requirements, particularly the settling time specification. The controller design can be more aggressive by using a smooth transient profile, instead of a step command, as the desired trajectory to allow the output to follow more closely. In this case, there are two bandwidths to consider: the actual control loop bandwidth, ω_c and the equivalent bandwidth of the transient profile, $\bar{\omega}_c$. Since the observer is evaluated on how closely it tracks the states and $\bar{\omega}_c$ is more indicative than ω_c of the speed at which the plant states move, $\bar{\omega}_c$ is used in place of ω_c in (4.17). Furthermore, taking other design issues, such as the sampling delay, into

consideration, a more appropriate minimum ω_o is found through simulation and experimentation to be

$$\omega_o \approx 5\sim 10\,\bar{\omega}_c \tag{4.18}$$

4.4 Optimization of LADRC

To summarize the observer and controller design methods discussed in section 4.1 and 4.3, a cohesive LADRC design and optimization procedure is given as follows:

Step 1: Design parameterized LESO and controller where ω_o and ω_c are design parameters;

Step 2: Design a transient profile with the equivalent bandwidth of $\bar{\omega}_c$;

Step 3: Select an ω_o from (4.18);

Step 4: Set $\omega_c=\omega_o$ and simulate/test the LADRC in a realistic software simulation or a hardware set-up;

Step 5: Incrementally increase both by the same amount until the noise levels and/or oscillations in the control signal and output exceed the tolerance;

Step 6: Incrementally increase or decrease ω_c and ω_o individually, if necessary, to make trade-offs among different design considerations, such as the maximum error during the transient period, the disturbance attenuation, and the magnitude and smoothness of the controller.

Remarks:

1. If the simulation/test results in Step 4 are unsatisfactory, it is likely that the transient design specification described by $\bar{\omega}_c$ is untenable due to the noise and/or sampling limitations. In this case, the control "goals" may have to be lowered by reducing $\bar{\omega}_c$ and therefore ω_c and ω_o.
2. Note that this approach can be easily extended to the Luenberg state observer- based state feedback design.

Example 4.1: Consider the same control problem in Example 3.1, but apply the LADRC in (4.5) to (4.10). Note that b=23.2 for this problem, but to make the design realistic, assume the best estimate of b is b_0=40. Rewrite the plant differential equation (3.10) as

$$\ddot{y} = (-1.41\dot{y} + 23.2T_d) + (23.2-40)u + 40u = f + 40u$$

The LESO is

$$\dot{z} = \begin{bmatrix} -3\omega_o & 1 & 0 \\ -3\omega_o^2 & 0 & 1 \\ -\omega_o^3 & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & 3\omega_o \\ 40 & 3\omega_o^2 \\ 0 & \omega_o^3 \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix}$$

and

$$z_1 \to y, z_2 \to \dot{y}, and$$

$$z_3 \to f = -1.41\dot{y} + 23.2T_d + (23.2\text{-}40)\text{u},\ as\ t \to \infty$$

The controller is defined as

$$u = \frac{u_0 - z_3}{40} \text{ and } u_0 = k_p(r - z_1) - k_d z_2$$

$$k_d = 2\xi\omega_c,\ \ \xi = 1,\ \text{and}\ \ k_p = \omega_c^2$$

where ω_c is the only design parameter to be tuned. A trapezoidal transient profile is used with a settling time of one second, or $\bar{\omega}_c$=4. From (4.18), ω_o is selected to be 40 rad/sec.

Simulation Setup:

The LADRC is simulated using Simulink (using ode1 with a fixed step of 1 ms) with a white sensor noise of the peak value 0.1% of that of the output, a 1 ms sampling period. As in Example 3.1, a step torque disturbance of 10% of the maximum torque is added at t=3 sec. The threshold for the ripples in the control signal is ±100mV peak to peak, which is in line with that of the PD controller in Example 3.1.

Tuning/Optimization

Following the design procedure describe above, ω_c and ω_o are increased together, with an initial value of 40 rad/sec. When they reach 95 rad/sec, the ripple threshold for control signal is reached, indicating that they cannot be increased further. The corresponding error of r-y has a maximum value of 3.4% and 0.5% for the transient period (t from 0 to 1 sec) and the disturbance rejection period (t from 3 to 3.2 sec), respectively. The subsequent changes in ω_c and ω_o and simulations show interesting facts:

1. The noise ripples in the control signals are virtually unchanged for $\omega_c + \omega_o$=190 and min{ω_c and ω_o}≥ 40;
2. Further increase of ω_o beyond 40 only affects the disturbance rejection error. The transient error in this case appears to be solely dependent on ω_c;
3. Bringing the transient error and the disturbance error to the same level while maintaining the same noise level in the control signal leads to ω_o=40 rad/sec and ω_c=150 rad/sec;
4. The maximum bandwidth is quickly determined using the proposed design procedure.

Figure 4.1 shows comparisons between the LADRC and the regular PD controller in Example 3.1 (ω_c=20 rad/sec). The fairness of the comparison is based on the fact that both control signals have the same level of noises.

The LADRC shows distinct advantages over the regular PD controller because:

1) No detailed mathematical model is required;
2) Zero steady state error is achieved without using the integrator term in PID;
3) Much better command following is demonstrated during the transient stage;
4) The controller is extremely robust as the plant's damping coefficient deviates from its original value of 1.41. In fact, the output response is essentially unchanged when this value changes from –30 to 30!

High performance is achieved through the use of a disturbance observer and is examined in Figure 4.2. The further increase in ω_o from 40 to 80 rad/sec does not greatly improve tracking performance, but it allows more noise into the observer states.

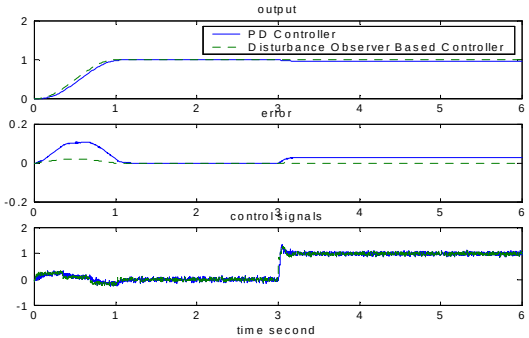

Figure 4.1 Comparison of PD and LADRC Controllers

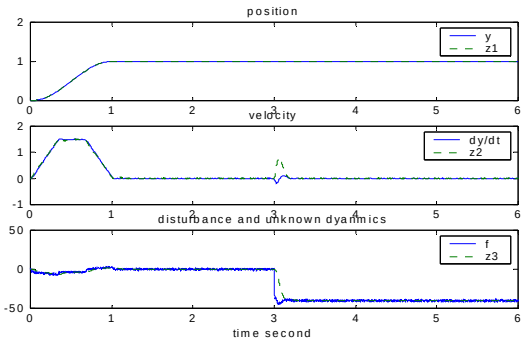

Figure 4.2 Performance of the LESO

4.5 Extensions to a Plant of an Arbitrary Order

For a general *nth* order plant with unknown dynamics and external disturbances,

$$y^{(n)} = f(t, y, \dot{y}, \cdots, y^{(n-1)}, u, \dot{u}, \ldots u^{(n-1)}, w) + bu \quad (4.19)$$

Han's observer can be similarly derived, starting from the extended state equation

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \ldots \\ \dot{x}_n = x_{n+1} + b_0 u \\ \dot{x}_{n+1} = h \\ y = x_1 \end{cases} \quad (4.20)$$

with $x_{n+1} = f$ added as an augmented state, and $h = \dot{f}$ unknown.

The LESO of (4.20) with the observer gain

$$L = [\beta_1\ \beta_2 \ldots \beta_{n+1}] \quad (4.21)$$

has the form of

$$\begin{cases} \dot{z}_1 = z_2 - \beta_1 (z_1 - y(t)) \\ \dot{z}_2 = z_3 - \beta_2 (z_1 - y(t)) \\ \ldots \\ \dot{z}_n = z_{n+1} - \beta_n (z_1 - y(t)) + b_0 u \\ \dot{z}_{n+1} = -\beta_{n+1} (z_1 - y(t)) \end{cases} \quad (4.22)$$

With the gains properly selected, the observer will track the states and yield

$$\begin{aligned} & z_1(t) \to y(t), z_2(t) \to \dot{y}(t), \cdots, z_n(t) \to y^{(n-1)}(t) \\ & z_{n+1}(t) \to f(t, y, \dot{y}, \cdots, y^{(n-1)}, u, \dot{u}, \ldots u^{(n-1)}, w) \end{aligned} \quad (4.23)$$

The controller can also be similarly designed as in (4.7) and (4.9):

$$u = -\frac{z_{n+1} + u_0}{b_0} \quad (4.24)$$

which reduces the plant to approximately a unit gain cascaded integrator plant

$$y^{(n)} = (f - z_{n+1}) + u_0 \approx u_0 \quad (4.25)$$

and

$$u_0 = k_p (r - z_1) - k_{d_1} z_2 - \ldots - k_{d_{n-1}} z_n \quad (4.26)$$

where the gains are selected so that the closed-loop characteristic polynomial has n poles at $-\omega_c$, i.e.,

$$s^n + k_{d_{n-1}} s^{n-1} + \ldots + k_{d_1} s + k_p = (s + \omega_c)^n \quad (4.27)$$

Here, ω_c is the closed-loop bandwidth to be optimized by ω_c-optimization, introduced in Section 3.3. The ω_o-optimization can be similarly applied using

$$s^n + \beta_1 s^{n-1} + \ldots + \beta_{n-1} s + \beta_n = (s + \omega_o)^n \quad (4.28)$$

V. Concluding Remarks

A new set of controller design, tuning, and optimization tools are presented, based on innovative scaling, parameterization, and optimization concepts. These concepts prove to be applicable not only to the existing model-based controller design methods such as pole placement and loop shaping, but also to model-independent design methods. These new tools make controller design and tuning easier and more effective. For example, the tuning of a PID controller is reduced to adjusting a single parameter, instead of three. Practical optimization is successfully incorporated into the one-parameter tuning. The new concepts and methods were tested successfully in software simulations, which incorporated real world scenarios such as sampling, sensor noise, disturbance, and the lack of mathematical model.

References

[1] N. Minorsky, "Directional Stability and Automatically Steered Bodies," *J. Am. Soc. Nav. Eng.*, vol. 34, p. 280, 1922.

[2] K. J. Astrom and T. Hagglund, "PID Control," *The Control Handbook*, W.S. Levine, Ed. CRC Press and IEEE Press, 1996, p. 198.

[3] J.G. Ziegler and N.B. Nichols, "Optimal Settings for Automatic Controllers," *Trans. ASME*, vol. 64, pp. 759-768, 1942.

[4] R. Eucker and Z. Gao, "A Novel Self-Tuning Control Design Approach for Industrial Applications," *Proc. of the IEEE Industrial Application Society 2000 Annual Meeting and World Conference on Industrial Applications of Electrical Energy*, Oct. 8-12, 2000.

[5] B. Boulter and Z. Gao, "A Novel Approach for Efficient Self-Tuning Web Tension Regulation," *Proc. of the 4th IEEE Conference on Control Applications*; *Real-Time Systems*, *The International Journal of Time-Critical Computing Systems*, vol. 11, no. 3, Nov. 1996.

[6] Z. Gao, "An Algorithmic Approach to Loop Shaping with Applications to Self-Tuning Control Systems," *Journal of the Franklin Institute*, vol. 332B, no.6, pp. 643-656, 1995.

[7] Z. Gao and P. Antsaklis, "New Methods for Control System Design Using Matrix Interpolation", *Proc. of the 33rd IEEE Conference on Decision and Control*, pp. 2506-2511, December 1996.

[8] J. Han, "A Class of Extended State Observers for Uncertain Systems," *Control and Decision*, vol. 10, no.1, pp. 85-88, 1995.(in Chinese)

[9] J. Han, "Nonlinear Design Methods for Control Systems," *Proc. Of The 14th IFAC World Congress*, Beijing, 1999.

[10] Z. Gao, Y. Huang, and J. Han, "An Alternative Paradigm for Control System Design," *Proc. of the 2001 IEEE Conference on Decision and Control*, Dec. 2001.

[11] Z. Gao, "From Linear to Nonlinear Control Means: A Practical Progression," *ISA Transactions*, vol.41, no.2, pp. 177-89, April 2002.