EE3304 Digital Control Systems (Part II)

Xiang Cheng
Dept. of Electrical and Computer Engineering
National University of Singapore

Contents

Chapter 1

Systems and Control —Background Overview

Questions of control have been of great interest since ancient times, and are assuming major importance in modern society. By the “control of a system” we mean the ability to influence its behavior so as to achieve a desired goal. Control mechanisms are ubiquitous in living organisms, and the design of self-regulating systems by human beings also has a long history. These systems range from water clocks in antiquity to aqueducts in early Rome, from Watt’s steam engine governor in 1769 which ushered in the Industrial Revolution in England, to the sophisticated automobiles, robots, and unmanned aircraft in the modern times.

The main purpose of this module is to teach you the most fundamental concepts and tools in designing a control system for real world problems. Both computer control theory and applications will be discussed. The first question I would like to raise is: what is control theory all about?

1.1 What is Control Theory?

If the answer is given in one short sentence, then control theory centers about the study of systems. Indeed, one might describe control theory as the care and feeding of systems. But, what is a system?

1.1.1 Systems

Intuitively, we can consider a system to be a set of interacting components subject to various inputs and producing various outputs. In fact system is such a general concept that there is no universally accepted definition. Everyone has his/her own view of what a system should look like. It is like the concept of set in mathematics, everyone knows what a set is, but we cannot define a set! Likewise, we cannot define a system precisely!

The student may object, "I do not like this imprecision, I should like to have everything defined exactly; in fact, it says in some books that any science is an exact subject, in which everything is defined precisely." But the truth is, if you insist upon a precise definition of system, you will never get it! For example, philosophers are always saying, "Well, just take a chair for example." The moment they say that, you know that they do not know what they are talking about any more. What *is* a chair? Well, a chair is a certain thing over there ... certain? how certain? The atoms are evaporating from it from time to time – not many atoms, but a few — dirt falls on it and gets dissolved in the paint; so to define a chair precisely, to say exactly which atoms are paint that belongs to the chair is impossible.

A mathematical definition will be good for mathematics, in which all the logic can be followed out completely, but the physical world is too complex. In mathematics, almost everything can be defined precisely, and then we do not know what we are talking about. In fact, the glory of mathematics is that *we do not have to say what we are talking about.* The glory is that the laws, the arguments, and the logic are independent of what "it" is. If we have any other set of objects that obey the same system of axioms as Euclid's geometry, then if we make new definitions and follow them out with correct logic, all the consequences will be correct, and it makes no difference what the subject was. In nature, however, when we draw a line or establish a line by using a light beam, as we do in surveying, are we measuring a line in the sense of Euclid? No, we are making an approximation; the light beam has some width, but a geometrical line has no width, and so, whether Euclidean geometry can be used for surveying or not is a physical question, not a mathematical question. However, from an experimental standpoint, not a mathematical standpoint, we need to know whether the laws of Euclid apply to the kind of geometry that we use in measuring land; so we make a hypothesis that it does, and it works pretty well; but it is not precise, because our surveying lines are not really geometrical lines.

Let's return to the concept of "system". Some people would insist that system must contain certain internal components, while others may argue that any object in the world can be considered as a system. Can we say that a single electron is a system? That is definitely debatable! How about your mind? do you consider it as a system? Well, we don't have to quarrel over the definition of the system, which we will happily leave to the philosophers. We can attack a problem without knowing the precise definition! Let's just look at the various types of systems around us:

- Mechanical systems: Clocks, Pistons
- Electrical systems: Radios, TVs
- Electrical-mechanical-chemical systems: Automobiles, Hard-disk-drives
- Industrial systems: Factories
- Medical systems: Hospitals
- Educational systems: Universities
- Biological systems: Human beings
- Information processing systems: Digital computers

This brief list, however, is sufficient to emphasize the fact that one of the most profound concepts in current culture is that of a "system".

1.1.2 Block Diagrams

From the standpoint of control engineering, there is always something going on in the system. If you are particularly interested in one of the variables, then you may want to measure it with some sensors. And we call these variables, outputs. Sometimes, you can even affect the output of the system by tuning some other signals, which we call inputs. And it is often a convenient guide to conceive of a system in terms of a block diagram.

The first attempt might be the one shown in Figure 1.1.

Frequently, Fig1.1 is much too crude and one prefers the following diagram to indicate the fact that a system usually has a variety of inputs and outputs:

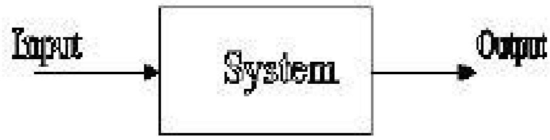


Figure 1.1: The simplest block diagram of a system

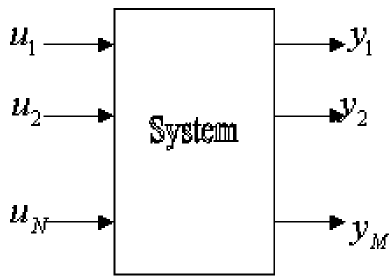


Figure 1.2: A block diagram of a system by variety of inputs and outputs

But wait a minute, we have not discussed the meaning of input. What is input?
Input: the attributes which can be varied by the controller. (Car driving example: brake, gas paddle, steering wheel)

Hey, you may wonder: can we consider noise, or disturbance (which cannot be varied by the man-made controller easily) as input? Yes and No. It all depends upon your point of view. If you adopt a very broad view of the input as any factors that can affect the behavior of the system, then the answer is yes. But if you limit it to the things that can be manipulated, then all those uncontrollable factors such as noise have to be distinguished from a normal input. Once again, don't try to find a precise definition of "input", as you may get into trouble again. But you never bother about it, as in reality, the meaning of the input is usually self evident for any specific control system.

What is output?
Output: the attributes of a system which are of interest to you, and are

observed carefully, but not varied directly. (Car driving example: position and velocity of the car)

But there is no clear cut between input and output, it all depends upon which aspect of the system you are looking at! In multi-stage production process, or chemical separation process, we may employ a block diagram as shown in Fig.1.3. Fig 1.3 indicates that the inputs u_1, u_2 to subsystems S_1 produce outputs y_1, y_2 , that are themselves inputs to subsystem S_2, and so on until the final output.

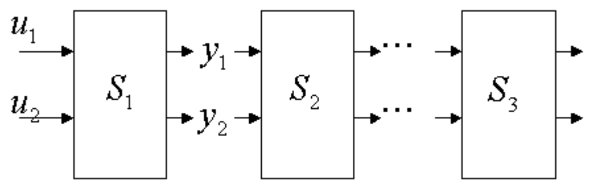


Figure 1.3: A block diagram of a multi-stage system

It is better to think of inputs and outputs in terms of cause and effect. For a certain system, inputs are the signals to drive the dynamics of the system and the outputs are the effects of the inputs. However, we cannot define precisely what inputs and outputs are!

In the study of atmospheric physics, geological exploration, and cardiology, we frequently think in terms of diagrams such as:

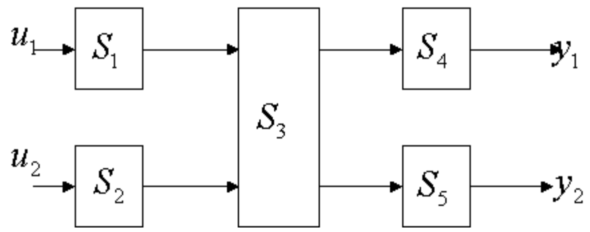


Figure 1.4: A block diagram of systems in atmospheric physics

Fig. 1.4 brings clearly to the fore that there are internal subsystems which we can neither directly examine nor influence. The associated question of identification and control of S_3 are quite challenging. And it depends upon how much you know about the internal system. In particular, it depends upon whether any mathematical models are available for the various subsystems.

1.1.3 Mathematical Systems

It cannot be too strongly emphasized that real systems possess many different conceptual and mathematical realizations. Each mathematical description serves its own purpose and possesses certain advantages and disadvantages. There is no such universal mathematical model that can serve every purpose.

For example, consider an airplane which can be analyzed by many mathematical models from different viewpoints:

- Electrical engineers: autopilot design; sensor systems (signal processing models);
- Mechanical engineers: air dynamic models; structure optimization; material analysis.

Different models serve different purposes. Once the mathematical die has been cast, equations assume a life of their own and can easily end by becoming master rather than servant. But as an engineer, you must always check the physical meanings of the various variables you are studying.

In any case, it should be constantly kept in mind that the mathematical system is never more than a projection of the real system on a conceptual axis. Mathematical model is only an approximation of the real world. It does not make sense to talk about the "true" model, you can only argue that a model is good or bad based upon the experimental evidence. But you will never be 100% sure that a model is "true". This is one of the lessons we have learned from the history of science. Take an example of Newton's law, almost everyone believed it was a "true" law governing the nature, until Einstein showed the world his Relativity. If Newton's law is not accurate, or is not "true", every model can be "wrong"!

1.1.4 The Behavior of Systems

In all parts of contemporary society, we observe systems that are not operating in a completely desired fashion.

- Economical systems: inflation, depression and recession;

- Human systems: cancer, heart diseases;
- Industrial systems: unproductivity and non-profitability;
- Ecological systems: pests and drought; green house effect.

What we can conclude from this sorry recital is that systems do indeed require care. They do not operate in a completely satisfactory manner by themselves. That's why control theory can play very important role in the real world. On the other hand, as we shall see, the cost of supervision must be kept clearly in mind. This cost may be spelled out in terms of resources, money, manpower, time, or complexity.

To control or not to control, that is a difficult question under certain circumstances. For instance, there is this huge debate on what is the role of government in the economy. The advocates of the free economy theory strongly oppose any interference of the government into the market. They believe that the "invisible hand" can handle everything. The opposite view is that the government should play active role in regulating the market. Consider the extreme example of the Hong Kong's Financial Crisis 1997-1998, would it be better that the Hong Kong government did not react to the attack of the international hedge funds? You can cast your own vote.

Fortunately, for engineers, we don't have to make this hard decision. Your boss makes the call! (but what if you become the boss in the future?) It is our primary concern to improve the behavior of any specific engineering system.

So in the next section, we are going to discuss how to improve the performance of a system.

1.1.5 Improvement of the Behavior of Systems

There are several immediate ways of improving the performance of a system.

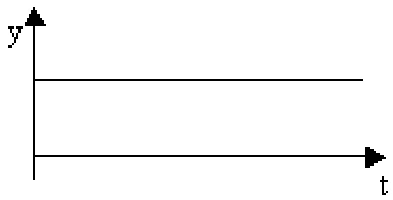
1. Build a new system;
2. Replace or change the subsystems;
3. Change the inputs.

In some cases, such as, say fixing an old car, a new system may be the ideal solution. In other cases, say the human body or an economical system, replacement is not a feasible procedure. We shall think in terms of the more feasible program of altering the design of the system or in terms of modifying the inputs, or both. In reality, there are always multiple solutions to fix the problems. The choice of the solutions would depend upon your requirement and your resource. For instance, if an old car keeps giving you headaches, how are you going to fix that? If you are a rich man, then the solution is simple, just buy a new one. If you are a poor student, probably you will send it to a garage and have the engine or other parts replaced. But before we go into further details, let's try to give a rough definition of control.

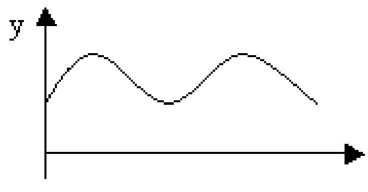
Control: influence the behavior of the system to achieve a desired goal.

Control theory mostly concerns about the third way: how to change the inputs such that the outputs behave in a desired fashion.

Usually the goal of automatic control for industrial systems is maintaining one or more outputs around some desired constant or time-varying values.



(a) Constant—Regulation, or Set point control: Air-conditioning system; Cruise system.



(b) Time-varying—Tracking: Autopilot; Guided missile.

Figure 1.5: The goal of automatic control

How do we achieve such goals? Is there any fundamental principle to guide us to design self-regulating systems?

Let's try to learn from our own experiences. There are many things for us to control in our daily life. How do we do that? For instance, consider a driving example – steering the car within one lane. There are three processes involved:
Observe the heading—compare with the lane lines —adjust the wheel

Trio: Measurement — comparison — adjustment

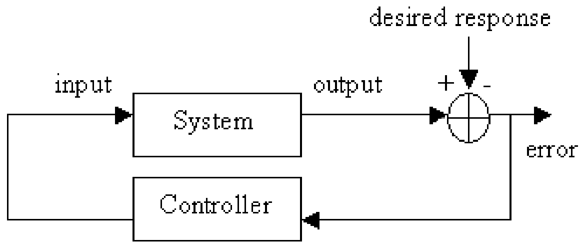


Figure 1.6: A closed-loop system

By measuring the quantities of interest, comparing it with the desired value, and using the error to correct the process, the familiar chain of cause and effect in the process is converted into a closed loop of interdependent events as shown in Fig. 1.6. That is the

Fundamental Concept of Control: ***Feedback***

How to apply this idea of feedback to build self-regulating system is the central problem of automatic control.

Once again we have to emphasize that control theory is about applying the idea of feedback to build a self-regulating system. Since the systems can be represented in terms of differential or difference equations, it is sometimes misleading to think that the control theory is about solving or analyzing the differential equations, which is totally wrong.

In the next, we are going to study a simple example to show why feedback is playing the central role in controller design.

1.2 A Simple Example of Feedback Controller

Following we will use a very simple physical example to give an intuitive presentation of some of the goals, terminology and methodology of automatic control. The emphasis of the discussion will be put on motivation and understanding of the basic concepts rather than mathematical rigidity.

1.2.1 Model of the Robotic Single-link Rotational Joint

One of the simplest problems in robotics is that of controlling the position of a single-link rotational joint using a motor placed at the pivot. Mathematically, this is just a pendulum to which one can apply a torque as an external force.

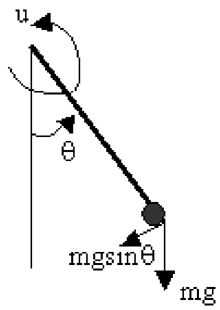


Figure 1.7: A single-link rotational joint

m — mass
l — length of rod
g — acceleration due to gravity
$u(t)$ — external torque at time t — input
$\theta(t)$ — counterclockwise angle of the pendulum at time t — output

Once the system is chosen, the first step is building a mathematical model. But how to build a model? It depends upon how much knowledge you have. There are basically three ways:

- White Box —First principles.
- Black Box —- Assume a function or equation form, then estimate the parameters by data fitting.
- Grey Box

White-box is always the first option you should consider for modeling if possible. We assume the friction is negligible, and all of the mass is concentrated at the end. From Newton's second law, the change rate of the angular momentum is proportional to the external torque, we have

$$ml^2\ddot{\theta}(t) + mgl\sin\theta(t) = u(t) \tag{1.1}$$

To avoid keeping track of constants, let's assume that the units of time and distance have been chosen such that

$$ml^2 = mgl = 1$$

So we are looking at a system describing by a second order nonlinear differential equation,

$$\ddot{\theta}(t) + \sin\theta(t) = u(t) \tag{1.2}$$

1.2.2 Mathematical Analysis of the Model

What are the equilibrium positions when there is no torque?

The equilibrium positions are the positions where $\ddot{\theta} = 0$, and $\dot{\theta} = 0$.

Equilibrium positions:

1. $\dot{\theta} = 0, \theta = 0$: Stable position
2. $\dot{\theta} = 0, \theta = \pi$: Unstable position

Stable:
If you change the initial condition a little bit, the dynamics of the system will change a little bit, but not too much.

Unstable:
If the initial conditions are just a bit different, the dynamic behaviors of the system are dramatically different.

Asymptotic stable:
Not only stable but also the response of the system will converge to the equilibrium point.

Before we design the controller, we have to be clear about the objective we try to achieve.

Control objective:
Apply torque $u(t)$ to stabilize the inverted pendulum such that the pendulum will maintain the upright position ($\theta = \pi$).

Now please try to put a pen or pencil in your palm, and keep it upright. Can you do that? Is it simple? Then try it again, but this time with your eyes closed! Can you do that? Why can't you do it? Because there is little feedback information when your eyes are closed! By now I hope you would appreciate the importance of the feedback principle. Now let's see if we can make a machine to imitate what we can do.

Stabilization or Set-point Control:
In most of the control problem in industry, the objective is to make the

overall system asymptotic stable by proper design of the controller, such that any deviation from the desired goal will be corrected. This will also be the central topic of this module.

Since equation (1.2) is just a second-order differential equation, why don't we try to solve it directly? Would it be wonderful that we can work out the solution to this ODE? If the solution is available to us, perhaps it can help us in designing the control system.

Is there analytical solution to this problem? Unfortunately, the answer is No. This is a nonlinear system, and normally, it is very hard to find analytical solution to nonlinear differential equations.

Although we don't know how to solve nonlinear equations in general, we do know how to solve linear equations. And if we can turn the nonlinear system into linear one by some magic, then the problem is finished! Now here is the trick. Remember, you are free to choose any control signal in any way you like (as long as feasible), right? Is it possible to design a feedback controller such that the nonlinear term $\sin(\theta(t))$ disappears? From the equation (1.2), we may simply let

$$u(t) = \sin(\theta(t))$$

and the closed loop system becomes

$$\ddot{\theta}(t) = 0$$

Although it becomes a linear system now, it is not stable! To further stabilize it, we need to add extra terms like

$$u(t) = \sin(\theta(t)) - 1.4\dot{\theta}(t) - \theta(t)$$

And we have

$$\ddot{\theta}(t) + 1.4\dot{\theta}(t) + \theta(t) = 0$$

But still it is not exactly what we want. We can easily find out that the angle $\theta(t)$ would eventually converge to 0! But we want it to converge to π (upright position)! Well, that is simple once you are at this stage. You just add a feed-forward term, $u_{ff}(t)$ in your controller as

$$u(t) = \sin(\theta(t)) - 1.4\dot{\theta}(t) - \theta(t) + u_{ff}(t)$$

Plug it into equation (1.2), we have

$$\ddot{\theta}(t) + 1.4\dot{\theta}(t) + \theta(t) = u_{ff}(t)$$

Now the nonlinear system becomes a well-behaved second order linear system. Remember you still have freedom to use $u_{ff}(t)$. Simply choose a constant $u_{ff}(t) = \pi$. This trick used the idea of converting a nonlinear system into linear ones, which actually is the most widely used technique to study nonlinear systems. Of course, we don't have time to study all those tricks for nonlinear systems, we will limit our interests to linear system in our module.

How to implement a nonlinear controller is not an easy issue, especially for analog controller. In many cases, we may have to use simple linear controller rather than a nonlinear one to solve this problem. There is another approach, which is also very commonly used —- Linearization! Through linearization, we are still trying to turn the nonlinear systems into linear systems. But this time, we are going to do it by approximation, which does not involve manipulating the control input u(t) with the nonlinear terms at all!

If only small deviations are of interest, we can linearize the system around the equilibrium point.
For small $\theta - \pi$,

$$\sin\theta = -\sin(\theta - \pi) = -(\theta - \pi) + o(\theta - \pi)$$

Define $y = \theta - \pi$, we replace the nonlinear equation by the linear equation as our object of study.

$$\ddot{y}(t) - y(t) = u(t) \tag{1.3}$$

Why linearization?

1. There is no general method to solve nonlinear equations, and linear system is much easier to analyze.

2. The local behavior of the nonlinear system can be well approximated by linear system. Just like any curve looks like a line locally. But does it work? The good news is that we have the following linearization principal.

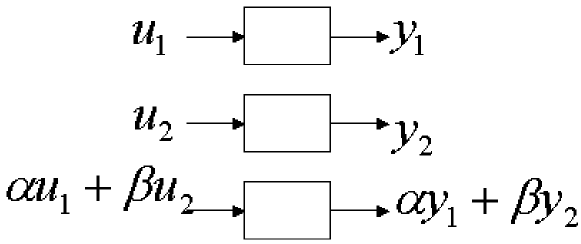


Figure 1.8: Superposition Principle

Linearization principal:

Design based on linearizations works locally for the original nonlinear system. Local means that satisfactory behavior only can be expected for those initial conditions that are close to the point about which linearization was made. The bad news is that the design may not work if the deviations from the operating points are large.

Why is linear system easier to analyze? What's the fundamental difference between linear and nonlinear systems?

Fundamental property of Linear System —-Superposition Principle

More specifically, suppose that the input u is resolved into a set of component time functions ϕ_1, ϕ_2 ... by expressing u as an infinite series

$$u = a_1\phi_1 + a_2\phi_2 + ...$$

where the coefficients $a_\lambda, \lambda = 1, 2, \ \ldots$, are constants representing the "weights" of the component time functions ϕ_1, ϕ_2 ... in u. Now, if the system is linear, then the response of the system to input u, $A(u)$, can be expressed as an infinite series

$$A(u) = a_1A(\phi_1) + a_2A(\phi_2) + ...$$

Where $A(\phi_\lambda)$ represents the response of the system to the component $\phi_\lambda, \lambda = 1, 2, \ \ldots$. Thus, if ϕ_λ are chosen in such a way that the determination of the response of system to input ϕ_λ is a significantly simpler problem than the direct calculation of $A(u)$, then it may be advantageous to determine $A(u)$ indirectly by

1. resolving u into the component functions ϕ_λ

2. calculating $A(\phi_\lambda), \lambda = 1, 2, .$

3. obtaining $A(u)$ by summing the series

This basic procedure appears in various guises in many of the methods used to analyze the behavior of linear systems. In general, the set of component time functions ϕ_λ is a continuum rather than a countable set, and the summations are expressed as integrals in which λ plays the role of variable of integration.

Remark: It is important to think of the integration as approximation by summation for understanding many of the concepts and properties of the system.

Example of application of superposition principle:
Choose component functions as e^{st}, where $s = \sigma + j\omega$ is the complex variable.

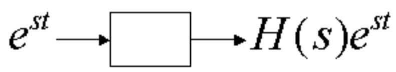


Figure 1.9: The response function for exponential input e^{st}

Try the response function as $H(s)e^{st}$, and plug it into the pendulum Eq 1.3, we have

$$s^2 H(s)e^{st} - H(s)e^{st} = e^{st} \tag{1.4}$$

we obtain

$$H(s) = \frac{1}{s^2 - 1} \tag{1.5}$$

Hence $H(s)e^{st}$ is indeed the response to e^{st}, which can be easily determined. Consider the special case of unit step input, where $s = 0$, the response is $H(0)$. Therefore, the static (or steady state) gain is simply $H(0)$.

Question: Can we decompose $u(t)$ by e^{st}?
Yes! The input $u(t)$ can be decomposed by e^{st}

$$u(t) = \frac{1}{2\pi j} \int_{a-j\infty}^{a+j\infty} F(s)e^{st} ds \tag{1.6}$$

But what is $F(s)$?
$F(s)$ is the Laplace transform of $u(t)$!

$$L[u(t)] = F(s) = \int_0^\infty u(t)e^{-st} dt \tag{1.7}$$

Did you ever wonder why Laplace transform is such a wonderful tool for analyzing linear system? It is rooted in the superposition principle!

There are many nice properties of Laplace transform, among which two of them are the most often used.

1. $L[f^{(n)}(t)] = s^n L[f(t)] - s^{n-1}f(0) - s^{n-2}f'(0) - ... - f^{(n-1)}(0)$

   Remark: if the initial conditions are zero, we simply replace the differential operator d/dt with s:

$$\frac{d}{dt} \Leftrightarrow s$$

2. $L[f_1(t) * f_2(t)] = L[f_1(t)] \cdot L[f_2(t)]$

   where

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau) d\tau \qquad \text{——} \qquad \textit{convolution}$$

   If $f1$ and $f2$ are one sided, ie. vanish for all $t < 0$, then

$$f_1(t) * f_2(t) = \int_{0}^{t} f_1(\tau) f_2(t-\tau) d\tau$$

Now let's apply the Laplace transform to our example, denote

$$L[y(t)] = Y(s)$$
$$L[u(t)] = U(s)$$

Then

$$L[\ddot{y}(t)] = s^2 Y(s) - sy(0) - y'(0) \tag{1.8}$$

Substitute them into Eq 1.3, we have

$$\begin{aligned}
&s^2 Y(s) - sy(0) - y'(0) - Y(s) = U(s) \\
&(s^2 - 1)Y(s) = sy(0) + y'(0) + U(s) \\
&Y(s) = \frac{1}{s^2 - 1}(sy(0) + y'(0) + U(s))
\end{aligned}$$

Remark: Note that Laplace transform magically turns the original differential Eq (1.3) into an algebraic equation that can be easily solved even by middle school students!

$$Y(s) = H(s)(sy(0) + y'(0)) + H(s)U(s) \tag{1.9}$$

But hold on, you may argue that this solution $Y(s)$ does not look like any other familiar solutions in time domain. In order to get the time-domain solution, you may just transform it back by partial fraction expansion and look them up in the Laplace transform table.

$$L^{-1}[H(s)] = L^{-1}[\frac{1}{s^2-1}] = L^{-1}[\frac{1}{2}(\frac{1}{s-1} - \frac{1}{s+1})] = \frac{1}{2}(e^t - e^{-t})$$
$$L^{-1}[sH(s)] = L^{-1}[\frac{s}{s^2-1}] = L^{-1}[\frac{1}{2}(\frac{1}{s-1} + \frac{1}{s+1})] = \frac{1}{2}(e^t + e^{-t})$$
$$L^{-1}[H(s)U(s)] = h(t) * u(t) = \int_0^t \frac{1}{2}(e^{t-\tau} - e^{-(t-\tau)})u(\tau)d\tau$$

Overall, we obtain the solution of $y(t)$ as

$$\begin{aligned} y(t) &= \frac{1}{2}y'(0)(e^t - e^{-t}) + \frac{1}{2}y(0)(e^t + e^{-t}) + \int_0^t \frac{1}{2}(e^{t-\tau} - e^{-(t-\tau)})u(\tau)d\tau \\ &= \frac{1}{2}(y'(0) + y(0))e^t + \frac{1}{2}(y(0) - y'(0))e^{-t} + \int_0^t \frac{1}{2}(e^{t-\tau} - e^{-(t-\tau)})u(\tau)d\tau \end{aligned} \tag{1.10}$$

If the initial conditions are zero, we have a very simple relationship between the output and input,

$$Y(s) = H(s)U(s) \tag{1.11}$$

It seems that the output Y(s) is simply transformation of the input U(s), and we have a special name for this transformation H(s).

Transfer function — $H(s)$

The transfer function is an alternative and convenient mathematical description of the system. But it is only applicable to linear systems since it is based upon superposition principle. Given any differential equation, the transfer function can be easily obtained by replacing the differential operator "d/dt" with "s". And given any transfer function, the corresponding differential

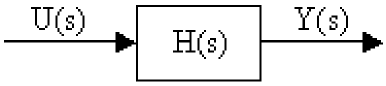


Figure 1.10: Transfer Function

equation can be found by simply replacing "s" with "d/dt". In addition to this, transfer function also has other meanings.

What's the relation between impulse response and transfer function? Impulse function is a mathematical description of a signal which takes a huge value for a very short period and then vanish elsewhere. For instance, impulse function can be used to describe the force resulting from a hammer hitting the table.

Now let the input be the impulse, $u(t) = \delta(t)$, then $U(s) = 1$, and $Y(s) = U(s)H(s) = H(s)$

Hence the transfer function is the Laplace transform of the impulse response. In other words, transfer function is the impulse response in the frequency domain.

In general, consider

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + ... + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + ... + b_0 u$$
$$(a_n s^n + a_{n-1} s^{n-1} + ... + a_0) Y(s) = (b_m s^m + b_{m-1} s^{m-1} + ... + b_0) U(s)$$
$$Y(s) = H(s)U(s)$$

$$H(s) = \frac{P(s)}{Q(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + ... + b_0}{a_n s^n + a_{n-1} s^{n-1} + ... + a_0} \tag{1.12}$$

The roots of the denominator and the numerator of the transfer function H(s) turn out to be very important for mathematical analysis of the system.

Poles: roots of $Q(s)$

The poles are important because the stability of the system can be determined by the following well known criterion (assuming you still remember what you learned in EE2010, systems and control).

Stability criterion:

If all poles have negative real parts then the system is stable!

why?

If λ is the pole, then $e^{\lambda t}$ is one component of the impulse response (verified by partial fraction expansion of the transfer function). So if all the poles have the negative real parts, the impulse response will decrease to zero! But what does impulse response have to do with stability?

Stability:
concerns the behavior of the system without external input when the initial condition is perturbed from the equilibrium point, say zero. What is the catch here?
If an impulse input is applied to a system in equilibrium position, this impulse will certainly excite the states of the system from zero conditions (equilibrium positions) into some non-zero values (non-equilibrium positions), and then all the inputs are zero immediately after $t = 0$. Therefore, impulse response can describe the behavior of the system with deviations from the equilibrium points. Then the stability will depend upon whether the deviations will be magnified or attenuated. If all the poles contain negative real parts, then the impulse response will converge to zero since each component $e^{\lambda t}$ goes to zero. Hence the system is stable.

Zeros: roots of $P(s)$

The concept of zeros is useful for stability analysis for the inverse system described by transfer function

$$\frac{1}{H(s)} = \frac{Q(s)}{P(s)} \tag{1.13}$$

There is also another reason for why they are called "zero". Remember that $H(\lambda)e^{\lambda t}$ is the output of the system for input signal $e^{\lambda t}$. If λ is the zero, input $e^{\lambda t}$ will have no effect on the system, just like zero input. So "zero" implies zero output for certain inputs. When do we want to use this property? It is useful for disturbance rejection such that the output of the system is not affected by the disturbance!

1.2.3 Open Loop and Feedback Controllers

Let's return to our control problem of the inverted pendulum. Since we already obtained the solution of the approximated linear system, it makes it easier for us to design controller.
What would happen without any control action? Would the pendulum maintain its upright position without any control?

Control Goal

Our objective is to bring y and y' to zero, for any small nonzero initial conditions, and preferably to do so as fast as possible, with few oscillations, and without ever letting the angle and velocity become too large. Although this is a highly simplified system, this kind of "servo" problem illustrates what is done in engineering practice. One typically wants to achieve a desired value for certain variables, such as the correct idling speed in an automobile's electronic ignition system or the position of the read/write head in a disk drive controller.
But why do we require that the initial values are small? Because the linearized model only works locally for nonlinear systems.

First attempt: open loop control – control without feedback

Since we have already obtained the solution, we can now try to solve the control problem. How should we choose the control function $u(t)$? There is no difficulty in choosing $u(t)$ to dispose of any disturbance corresponding to a particular initial conditions, $y(0)$ and $y'(0)$.

For example, assume we are only interested in the problem of controlling the pendulum when starting from the initial position $y(0) = 1$ and velocity $y'(0) = -2$. In this case, from solution (1.9), we have

$$Y(s) = \frac{s-2}{s^2-1} + \frac{U(s)}{s^2-1} = \frac{s-2+U(s)}{(s+1)(s-1)}$$

If we choose $U(s) = 1$, then y(t) will go to zero. But the corresponding signal is impulse function, hard to implement precisely. Let's instead try

$$u(t) = 3e^{-2t}$$

or

$$U(s) = \frac{3}{s+2}$$

Then

$$\begin{aligned}
Y(s) &= \frac{1}{s^2-1}(s-2) + \frac{3}{(s^2-1)(s+2)} \\
&= \frac{(s-2)(s+2)+3}{(s^2-1)(s+2)} \qquad (1.14)\\
&= \frac{1}{s+2} \Rightarrow e^{-2t}
\end{aligned}$$

It is certainly true that $y(t)$ and its derivative approach zero, actually rather quickly. It looks like the control problem is solved satisfactorily. However, there is one serious problem with this control method. If we make any mistakes in estimating the initial velocity, the control result will be disastrous.

For instance, if the differential equation is again solved with the same control input, but now using instead the initial conditions:

$$\begin{aligned}
&y(0) = 1, y'(0) = -2+\varepsilon \\
Y(s) &= H(s)(sy(0)+y'(0)) + H(s)U(s) \\
&= \frac{1}{s^2-1}(s-2+\varepsilon) + \frac{3}{(s^2-1)(s+2)} \\
&= \frac{\varepsilon}{s^2-1} + \frac{1}{s+2} \qquad (1.15)\\
&= \frac{\varepsilon}{2}(\frac{1}{s-1} - \frac{1}{s+1}) + \frac{1}{s+2} \Rightarrow \frac{\varepsilon}{2}e^{t} - \frac{\varepsilon}{2}e^{-t} + e^{-2t}
\end{aligned}$$

It is obvious that no matter how small ε is, the solution will diverge to infinity, namely,

$$\lim_{t\to\infty} y(t) = \infty$$

What about the other types of open-loop control?

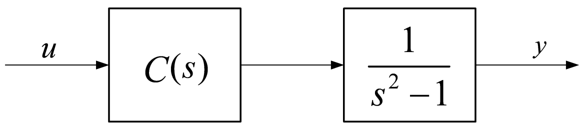


Figure 1.11: Open Loop Control

No matter what type of $C(s)$ is used, the unstable pole at $s = 1$ cannot be eliminated. Then you may argue how about choosing $C(s) = (s-1)/(s+2)$? That is an interesting solution! This raises the following question: *Can unstable poles be canceled out by the zeros?*

Let's just examine a simple example with transfer functions $H_1(s) = \frac{1}{s+1}$ and $H_2(s) = \frac{s-1}{(s+1)(s-1)}$. Are these two transfer functions equivalent?

The simplest way to test it out is to write down the corresponding differential equations and set all the inputs to be zero, and examine whether the solutions to the two differential equations are the same or not.
Then from $H_1(s) = \frac{1}{s+1}$, we have

$$\frac{dy}{dt} + y = 0$$

and from $H_2(s) = \frac{s-1}{(s+1)(s-1)}$, we have

$$\frac{d^2y}{dt^2} - y = 0$$

Obviously one solution will converge to zero, and the other solution will blow up to infinity!

I hope at this moment you are certain that the unstable poles cannot be simply canceled out by zeros. And we can conclude that the open loop control can not achieve the goal very well. What we need is a controller such that the system returns to equilibrium position rapidly regardless of the nature of the small perturbations, or the time at which it occurs.

One way to accomplish this is to take $u(t)$ not to depend upon the time t, but rather upon the state of the system, y and y'. Thus, we write

$$u = g(y, y') \tag{1.16}$$

This is exactly a feedback controller: the control inputs depend upon the output, not upon time! This also how you keep your pen upright in your palm. You watch the pen (measure and compare) and move your hand accordingly (control). Again, we can see control theory is not about how to solve the differential equations, but on how to apply the idea of feedback.

Second attempt: Proportional control – the simplest feedback control

A naive first attempt using the idea of feedback would be as follows: If we to the left of the vertical, that is, if $y > 0$, then we wish to move to the right, and therefore we apply a negative torque. If instead we are to the right, we apply a positive torque. In other words, we apply proportional feedback

$$u(t) = -K_p y(t) \tag{1.17}$$

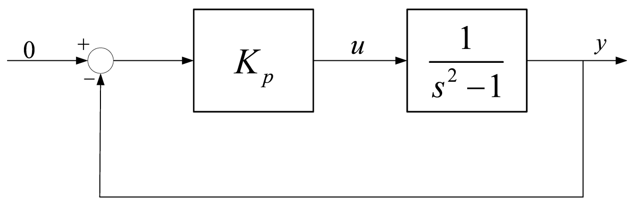


Figure 1.12: Proportional Control

where K_p is some positive real number, the feedback gain.
What is the transfer function of the closed loop system?

$$H_{cl}(s) = \frac{K_p \frac{1}{s^2-1}}{1 + \frac{K_p}{s^2-1}} = \frac{K_p}{s^2 - 1 + K_p}$$

Where are the poles?

$$s = \pm\sqrt{1 - K_p} \tag{1.18}$$

1. If $K_p < 1$, then all of the solutions diverge to infinity.
2. If $K_p = 1$, then each set of initial values with $y'(0) = 0$ is an equilibrium point.
3. If $K_p > 1$, then the solutions are all oscillatory.

Therefore, in none of the cases is the system guaranteed to approach the desired position. We needs to add damping to the system. We arrive then at a PD, or

Third attempt: Proportional and Derivative Control

Consider the resulting transfer function of the closed-loop system,

$$H_{cl}(s) = \frac{(K_p + K_d s)\frac{1}{s^2-1}}{1 + \frac{(K_p + K_d s)}{s^2-1}} = \frac{K_p + K_d s}{s^2 + K_d s + K_p - 1} \tag{1.19}$$

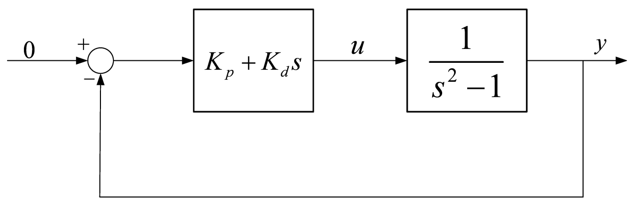


Figure 1.13: Proportional and Derivative Control

Then is it possible to choose the gains K_p and K_d such that the overall system is asymptotically stable?

It is evident that as long as $K_p > 1$ and $K_d > 0$, the closed loop will be stable. Further more, the poles of the closed loop can be placed at any desired position by choosing proper gains.

We conclude from the above discussion that through a suitable choice of the gains K_p and K_d it is possible to attain the desired behavior, at least for the linearized model. That this same design will still work for the original nonlinear model is due to what is perhaps the most important fact in control theory – and for the matter in much of mathematics – namely that first order approximations are sufficient to characterize local behavior. Informally, we have the following linearization principle:

Design based on linearizations works locally for the original nonlinear system.

1.3 Feedback Control System

Many advantages may be obtained by using feedback in control system design. Feedback, for instance, can do the following:

- Improve the transient behavior of the system

- Decrease the sensitivity to parameter changes in the open-loop system
- Eliminate steady-state errors if there are enough integrators in the open-loop system
- Decrease the influence of load disturbances and measurement errors

In general, a feedback control system can be represented by the block diagram below:

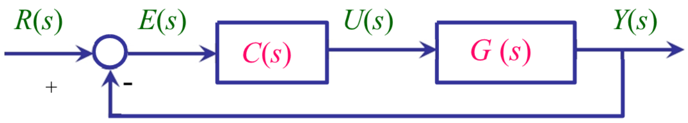


Figure 1.14: Feedback Control System

Given a system represented by $G(s)$ and a reference $R(s)$, the objective of control system design is to find a control law (or controller) $C(s)$ such that the resulting output $Y(s)$ is as close to reference $R(s)$ as possible, or error $E(s) = R(s) - Y(s)$ is as small as possible.

However, many other factors have to be carefully considered when dealing with real-life problems. These factors include:

- The effect of the disturbances. The process is always subject to different types of disturbances, such as load disturbances, measurement errors, and parameter variations. How to eliminate or minimize the effect of the disturbances is called disturbance rejection problem in control system.
- Uncertainties. As discussed before, all the mathematical models are only approximation of the real system, and there are always some uncertainties regarding how close the model is to the reality.
- Nonlinearities. Most of the real systems are not linear in general. How to make sure that the design based upon the linear systems would still work for the nonlinear system just as what we did for the robotic manipulator example?
- Noise. Measurement noise is inevitable in real world, and how to cope with that?

For simplicity, let's ignore all those complicated factors and take a closer look at the block diagram of the simple feedback control system as shown in Fig. 1.14.
Recall that the objective of control system design is trying to match the output Y(s) to the reference R(s). Thus, it is important to find the relationship between them. Recall that

$$Y(s) = G(s)U(s) = G(s)C(s)E(s)$$

Similarly we have $E(s) = R(s) - Y(s)$. Thus,

$$Y(s) = G(s)C(s)(R(s) - Y(s))$$

which leads to the closed loop transfer function from the reference signal $R(s)$ to output $Y(s)$:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{1.20}$$

Thus, the block diagram of the control system can be simplified as,

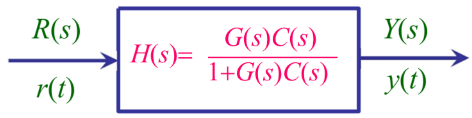


Figure 1.15: Closed Loop Transfer Function

The problem becomes how to choose an appropriate $C(s)$ such that $H(s)$ will have desired properties to meet certain specifications.

On the other hand, we have

$$E(s) = R(s) - Y(s) = (1 - \frac{G(s)C(s)}{1 + G(s)C(s)})R(s) = \frac{R(s)}{1 + G(s)C(s)} \tag{1.21}$$

The problem is equivalent to finding an appropriate control law $C(s)$ such that the resulting error signal $e(t)$ goes to zero as quickly and as smoothly as possible, which is the same as saying that the output $y(t)$ is tracking the given reference $r(t)$ as quickly and as smoothly as possible. In this course, we will mainly focus on the cases when $r(t)$ is either a step function or a ramp function.
It is obvious that the realizability of a control task is not only depending on the selection of an appropriate controller $C(s)$, but also related to the

nature of the task $R(s)$, as well as the plant $G(s)$. Clearly, an appropriate controller is the simplest one which can achieve the control task $R(s)$ based on the plant $G(s)$. In general, to accomplish a more complex control task, a more complex controller will be needed, as we will further demonstrate subsequently.

In practice, the following control system configuration is often encountered where $C_2(s)$ is either a part of controller, or a dynamics of sensory device.

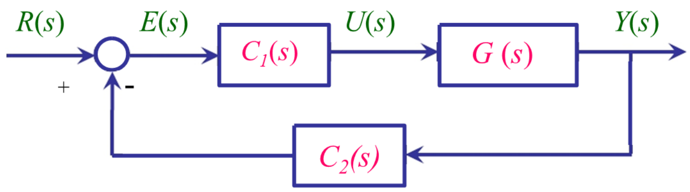


Figure 1.16: A More Complex Configuration of Feedback Control System

The closed loop transfer function can be easily obtained as

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G(s)C_1(s)}{1 + G(s)C_1(s)C_2(s)}$$

Then the question becomes how to choose the controller $C_1(s)$ such that the closed loop transfer function $H(s)$ satisfies some desired properties.

1.4 Notes and References

Most of the materials covered in the lecture notes can be found in following references:

1. G. F. Franklin, J. D. Powell and M. L. Workman, *Digital Control of Dynamic Systems, 2nd ed.*, Addison-Wesley Publishing Company, 1990.

2. K. J. Astrom and B. Wittenmark, *Computer-Controlled Systems*, Prentice Hall , 1997.

3. K. Ogata, *Modern Control Engineering, 4th ed.*, Prentice Hall , 2002.

Chapter 2

Digital Control System and Performance Specifications

2.1 Why Digital Control?

What is Digital Control System? Literally, the answer is simple. The controller is implemented by a digital computer and the input is calculated by the controller!

There are many advantages gained by switching from analog control to digital control:

- Cost is the major argument for changing the technology of analog control to digital control. The cost of an analog system increases linearly with the number of control loops; the initial cost of a digital system is large, but the cost of adding an additional loop is small. The digital system is thus cheaper for large installations.
- Flexibility is another advantage of the digital control systems. Analog systems are changed by rewiring; computer-controlled systems are changed by reprogramming.
- Efficiency is also a factor. For instance, nonlinear control cannot be easily wired, but it can be easily programmed!

Most of control systems nowadays are implemented using either computers such as PCs or Digital Signal Processors (DSP), which are specially designed to carry out computations related to control algorithm realizations. The advantages of digital controllers using PC or DSP are obvious: it is

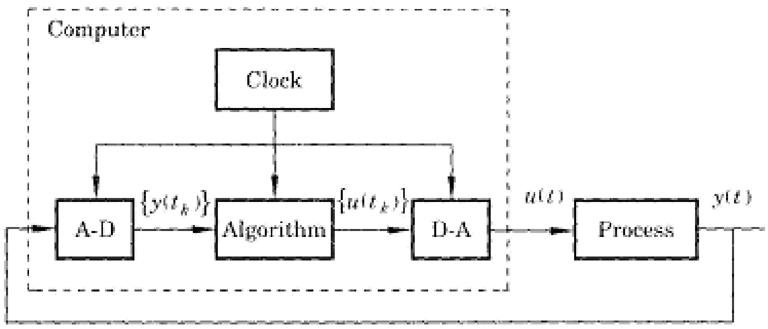


Figure 2.1: Schematic diagram of a computer-controlled system.

fast, reliable, reusable and can be modified through simple coding whenever needed.

Practically all control systems that are implemented today are based on computer control. It is therefore important to understand computer-controlled systems well. Such systems can be viewed as approximations of analog-control systems, but this is a poor approach because the full potential of computer control is not used. At best the results are only as good as those obtained with analog control. It is much better to master computer-controlled systems, so that the full potential of computer control can be used. There are also phenomena that occur in computer-controlled systems that have no correspondence in analog systems. It is important for an engineer to understand this. The main goal of this module is to provide a solid background for understanding, analyzing, and designing computer-controlled systems.

A computer-controlled system can be described schematically as in Fig. 2.1. The output from the process $y(t)$ is a continuous-time signal. Can you directly feed a continuous-time signal into a computer? Of course not! The computer can only deal with digits. That's why you need the A-D converter. The output is converted into digital form by the analog-to-digital $(A - D)$ converter. The $A-D$ converter can be included in the computer or regarded as a separate unit, according to one's preference. The conversion is done at the sampling instants, t_k. The time between successive samplings is called the sampling period and is denoted by T. Periodic sampling is normally used, but there are, of course, many other possibilities. For example, it is possible to sample when the output signals have changed by a certain amount. It is also possible to use different sampling periods for different loops in a system. This is called multirate sampling.

The computer interprets the converted signal, $y(t_k)$, as a sequence of numbers, processes the measurements using an algorithm, and gives a new sequence of numbers, $u(t_k)$. This sequence is converted to an analog signal by a digital-to-analog $(D - A)$ converter. The events are synchronized by the realtime clock in the computer. The digital computer operates sequentially in time and each operation takes some time. The D-A converter must, however, produce a continuous-time signal. This is normally done by keeping the control signal constant between the conversions, which is called Zero-order Hold.

Between the sampling instants, the input would be a constant due to zero-order hold. Now an interesting question is: is the controller open-loop, or feedback control between the sampling instants? Since the input is a constant regardless of what happens in the output during this period, the system runs open loop in the time interval between the sampling instants.

The computer-controlled system contains both continuous-time signals and sampled, or discrete-time, signals. The mixture of different types of signals sometimes causes difficulties. In most cases it is, however, sufficient to describe the behavior of the system at the sampling instants. The signals are then of interest only at discrete times. Such systems will be called discrete-time systems. Discrete-time systems deal with sequences of numbers, so a natural way to represent these systems is to use difference equations.

Using computers to implement controllers has substantial advantages. Many of the difficulties with analog implementation can be avoided. For example, there are no problems with accuracy or drift of the components. It is very easy to have sophisticated calculations in the control law, and it is easy to include logic and nonlinear functions. Tables can be used to store data in order to accumulate knowledge about the properties of the system. It is also possible to have effective user interfaces.

2.2 Design a Digital Controller

There are two ways to design a digital controller or a discrete-time control system.

The first approach is based upon the analog control design. The idea is simple. Because analog (continuous-time) control system designs are well established, we should like to know how to take advantage of a good continuous-time design and try to make a digital computer to produce a discrete equivalent to the continuous compensator. This method of design is called emulation.

The first step is to follow whatever we have learnt in EE2010 (systems and control) to design a continuous-time controller. In other words, the first design is done in the s-plane, using root-locus or frequency-response techniques to derive a satisfactory analog controller $C(s)$. This step totally ignores the fact that a sampler and digital computer will eventually be used.

Having the analog controller $C(s)$, we then convert the design to a digital control by applying one of the discretization techniques you have learned in part I of this module to obtain an equivalent $C(z)$.

Emulation design works very well if sampling period T is sufficiently small. Carrying out the initial design using continuous methods is a good idea independent of whether it will be used in a subsequent emulation step or merely as a guide for a direct discrete design. Knowing how the system could perform if implemented with analog hardware provides a target for how well the digital system should perform and aids in selecting the sample rate.

The second approach is called the direct discrete design. Basically, we just discretize the plant first to obtain a sampled-data system or discrete-time system $G(z)$, and then apply digital control system design techniques to design a digital controller $C(z)$. In this approach, we totally ignore the analog controller design.

We will discuss both design methods in this course.

2.3 Feedback Control in Discrete Setting

Let us examine the following block diagram of digital control system:
The closed loop transfer function can be easily obtained as

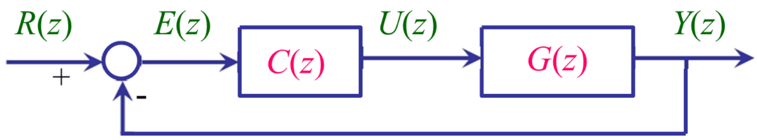


Figure 2.2: Digital Feedback Control System

$$H(z) = \frac{Y(z)}{R(z)} = \frac{G(z)C(z)}{1 + G(z)C(z)} \tag{2.1}$$

Thus, the block diagram of the control system can be simplified as,

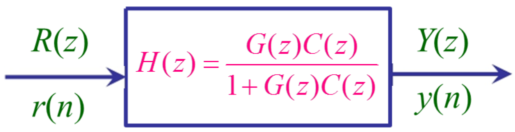


Figure 2.3: Closed Loop Transfer Function

The problem becomes how to choose an appropriate controller $C(z)$ such that the closed loop system (described by transfer function $H(z)$) will have desired properties.

Now it is time to discuss in details about what are the desired properties that we want the closed-loop system to have.

2.4 Control System Performance Specifications

Before we design the controller, we must first understand the control performance specifications clearly. Fig. 2.4 shows a generic block diagram of a unity feedback closed loop system that applies to many systems. In this figure, the load disturbance d affects the system, the block containing C is the digital controller that we will be designing.

Performance characteristics are normally specified as follows:

- Stability
- Steady-state accuracy
- Settling time

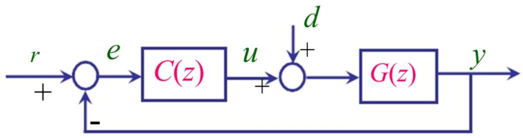


Figure 2.4: A unity feedback system.

- Overshoot
- Rise time
- Disturbance rejection
- Control effort required – maximum magnitude of u and energy $K\int u^2dt$
- Robustness – Sensitivity to parameter changes
- others

2.4.1 Stability

The stability of the systems concerns about the fundamental question: what would happen if there is a small disturbance to the system?
The concept of stability is of primary importance for control system design. The reason is very simple. For any control system design, stability has to be guaranteed first before its practical implementation. Disaster would follow if the system is unstable as any small deviations from the set point or equilibrium position will lead to divergence.
Stabilization: In most of the control problem in industry, the objective is to make the system stable by proper design of the controller, such that any small deviation from the desired point will be corrected.
How to check whether the system is stable or not? It depends upon where the poles are! Let's review what you have already learned in the past.

Stability criterion: Continuous-time systems

A continuous-time system is said to be stable if its denominator of the system has no roots or poles with a positive real part. It is unstable if it has poles or roots with a positive real part.
What would happen if the poles lie on the stability boundary, the imaginary axis? Then it depends upon whether the pole is a single root, or a repeated root.

- If the pole lies on the imaginary axis (with zero real part), and is a single root, then the system is marginally stable, which implies that the impulse response will neither decrease to zero, nor blow up to infinity. For example, the integrator, $\frac{1}{s}$ is marginally stable.

- If the pole lies on the imaginary axis and is a repeated root, then the system is unstable. For example, the double integrator, $\frac{1}{s^2}$ is unstable.

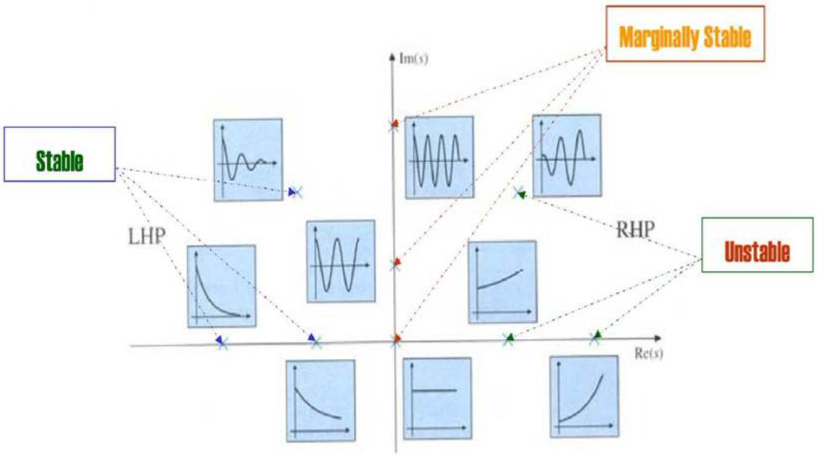


Figure 2.5: The stability criterion: Continuous-time Systems

Stability criterion: Discrete-time systems

A discrete-time system is said to be stable if its denominator of the system has no roots or poles outside unit circle. It is unstable if it has poles or roots outside unit circle.
What would happen if the poles lie on the stability boundary, i.e. the unit circle? Then it depends upon whether the pole is a single root, or a repeated root.

- If the pole lies on the unit circle, and is a single root, then the system is marginally stable, which implies that the impulse response will neither decrease to zero, nor blow up to infinity. For example, the discrete-time integrator, $\frac{1}{z-1}$ is marginally stable.

- If the pole lies on the unit circel and is a repeated root, then the system is unstable. For example, the discrete-time double integrator, $\frac{1}{(z-1)^2}$ is unstable.

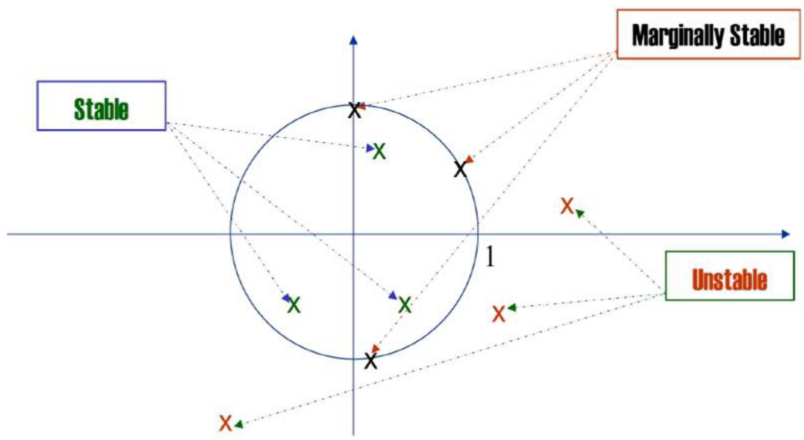


Figure 2.6: The stability criterion: Discrete-time Systems

2.4.2 Steady-state Accuracy

Only after the stability is guaranteed, we may move on to consider other requirements such as steady-state accuracy. Steady-state accuracy refers to the requirement that after all transients are negligible, the output error, r-y, must be acceptably small. The two causes of nonzero error are the reference r and the disturbance d.

Consider first the reference. Some control systems have a finite nonzero steady-state error when the reference is a constant. Such systems are labeled "type 0," because there is finite error with a zero-order polynomial input (i.e. constant). The reason for the error can be seen from Fig. 2.4: The finite dc gain between e and y requires that e be nonzero in order for y to be nonzero. Similarly, a control system that has finite nonzero steady-state error to a first-order polynomial input (a ramp) is called a "Type I" system. In each case, the disturbance d is taken to be zero. However, in evaluating the system we must include the effects of d in the final calculations. In general, we add the errors due to reference and disturbance to find a total system error, which must be within acceptable limits.

Steady state accuracy–Continuous-time systems

Suppose the unity feedback system shown in Fig. 2.4 has a reference input that is a step function and that the disturbance is zero. It can be easily seen that

$$E(s) = \frac{R(s)}{1+G(s)C(s)} = \frac{1}{s}\frac{1}{1+G(s)C(s)} \tag{2.2}$$

To compute the steady-state error due to r, we assume that the closed-loop system is stable and use the final-value theorem. The final-value theorem states that for any stable continuous-time signal $f(t)$ which has a steady-state value, its final value is

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) \tag{2.3}$$

Applying the final value theorem, the final value of $e(t)$ is

$$\begin{aligned} e(\infty) &= \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\frac{1}{s}\frac{1}{1+G(s)C(s)} \\ &= \frac{1}{1+G(0)C(0)} = \frac{1}{1+\beta_p} \end{aligned}$$

where $\beta_p = G(0)C(0)$ is called position error constant. Now the question is: is it possible to make the steady-state error go to zero?

The answer is of course YES as long as the position error constant goes to infinity. But when will the position error constant become infinity? If either the plant $G(s)$ or the controller $C(s)$ contains one integrator $1/s$, the position error constant is infinity, and such system will be "Type I" system. If there is no integrator inside $G(s)C(s)$, then the position error constant is a finite constant, and the steady-state error is finite. Such system is called "Type 0" system as discussed before.

For a unit ramp input, $r(t) = t$, and $R(s) = \frac{1}{s^2}$, applying the final value theorem, the final value of $e(t)$ is

$$\begin{aligned} e(\infty) &= \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\frac{1}{s^2}\frac{1}{1+G(s)C(s)} \\ &= \lim_{s\to 0}\frac{1}{sG(s)C(s)} = \frac{1}{\lim_{s\to 0} sG(s)C(s)} = \frac{1}{\beta_v} \end{aligned}$$

where $\beta_v = \lim_{s\to 0} sG(s)C(s)$ is called velocity error constant. Now the question is: is it possible to make the steady-state error for a ramp input go to zero?

Similarly, we need the velocity error constant to be infinity, which implies that $G(s)C(s)$ has at least two integrators $1/s^2$ in order to make the steady-state error go to zero, and such system is called "Type II" system. The error

will be finite if the velocity error constant is finite, which also implies that the system contains one integrator, which makes it a "Type I" system.
All these ideas can be extended to the discrete-time system.

Steady State Accuracy–Discrete-time Systems

Still consider the unity feedback system shown in Fig. 2.4 where the transfer functions of the plant and the controller are discrete-time version, $G(z)$ and $C(z)$. The reference r and the disturbance d are the sampled versions of their continuous-time counterparts.
Proceeding as we did for the continuous-time system, suppose the input r is a step, and the disturbance d is zero. It can be easily seen that

$$E(z) = \frac{R(z)}{1+G(z)C(z)} = \frac{z}{z-1}\frac{1}{1+G(z)C(z)} \tag{2.4}$$

To compute the steady-state error due to r, we assume that the closed-loop system is stable and use the discrete-time final-value theorem. The discrete-time final-value theorem states that for any stable discrete-time signal $f(k)$ which has a steady-state value, its final value is

$$\lim_{k\to\infty} f(k) = \lim_{z\to 1}(z-1)F(z) \tag{2.5}$$

Applying the final value theorem, the final value of $e(k)$ is

$$\begin{aligned} e(\infty) &= \lim_{z\to 1}(z-1)E(z) = \lim_{z\to 1}(z-1)\frac{z}{z-1}\frac{1}{1+G(z)C(z)} \\ &= \frac{1}{1+G(1)C(1)} = \frac{1}{1+\beta_p} \end{aligned}$$

where $\beta_p = G(1)C(1)$ is called position error constant for discrete-time system. Now the question is: is it possible to make the steady-state error go to zero?

Similarly, as in the continuous-time case, we need the position error constant to be infinity, which implies that either the plant $G(z)$ or the controller $C(z)$ contains one integrator $1/(z-1)$, and such system will be "Type I" system. If there is no integrator inside $G(z)C(z)$, then the position error constant is a finite constant, and the steady-state error is finite. Such system is called "Type 0" system as discussed before.

For a unit ramp input, $r(k) = k$, and $R(z) = \frac{Tz}{(z-1)^2}$, applying the final value theorem, the final value of $e(k)$ is

$$\begin{aligned} e(\infty) &= \lim_{z\to 1}(z-1)E(s) = \lim_{z\to 1}(z-1)\frac{Tz}{(z-1)^2}\frac{1}{1+G(z)C(z)} \\ &= \lim_{z\to 1}\frac{T}{(z-1)G(z)C(z)} = \frac{1}{\lim_{z\to 1}(z-1)G(z)C(z)/T} = \frac{1}{\beta_v} \end{aligned}$$

where $\beta_v = \lim_{z\to 1}(z-1)G(z)C(z)/T$ is called velocity error constant. Now the question is: is it possible to make the steady-state error for a ramp input go to zero?
Similarly, we need the velocity error constant to be infinity, which implies that $G(z)C(z)$ has at least two integrators $1/(z-1)^2$ in order to make the steady-state error go to zero, and such system is called "Type II" system. The error will be finite if the velocity error constant is finite, which also implies that the system contains one integrator, which makes it a "Type I" system.

From above discussion, we can see that if we want the system to track a constant reference perfectly, at least one integrator is needed. The control system for perfectly tracking a ramp signal will be more complex as 2 integrators are needed. Adding integrators may increase the order of the whole system, and affect the stability of the closed-loop, which makes the design more complicated. For many practical system, tracking a constant is more common requirement.

2.4.3 Transient Behavior of Second-order Systems

Transient accuracy refers to the ability of the system to keep the error small as r(t) changes. Specifications of transient performance can be made in the time domain and then translated to the frequency domain either in terms of characteristic pole locations in s or z, or in terms of frequency-response features such as bandwidth and phase margin. We will aim to consider specifications in terms of characteristic root locations in the z-plane by transforming the desired s-plane specifications to equivalent locations in the z-plane. This is accomplished by using the relation $z = e^{sT}$ to map the poles in the s-plane to the z-plane.
First we need to transfer the transient specifications from a time description to an s-plane pole-location requirements. Consider the following block diagram with a standard 2nd order system under unit step.
The transfer function is fully specified by two factors: the damping ratio

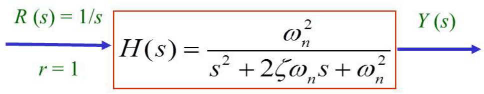


Figure 2.7: The standard second order system.

ζ and the natural frequency ω_n. The transient behavior of the standard second-order system is shown in the figure below.

The transient behavior is characterized by three factors: percent overshoot

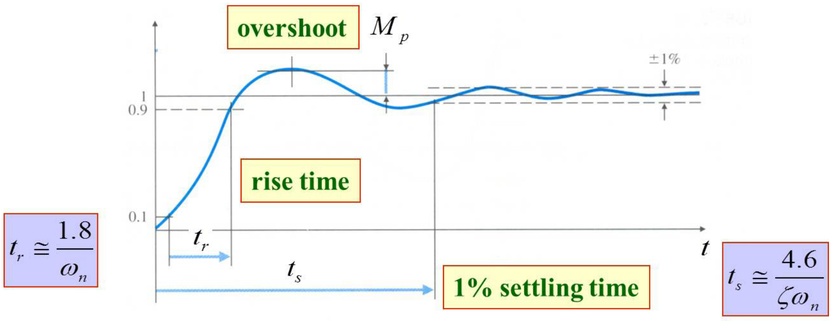


Figure 2.8: Typical step response of the standard second order system.

M_p, rise time t_r and settling time t_s. The percent overshoot, M_p, depends only upon the damping ratio, ζ, as shown in Fig. 2.9. Thus, given a requirement on percent overshoot, we can easily find out the requirement on the damping ratio by this formula.

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{2.6}$$

Another feature of the interest is the rise time of the response towards its final value, t_r. The rise time (for the response to rise from 0.1 to 0.9) is related to the natural frequency only, and can be approximated by

$$t_r \cong \frac{1.8}{\omega_n} \tag{2.7}$$

A requirement on t_r thus becomes a requirement that ω_n satisfies

$$\omega_n \geq \frac{1.8}{t_r} \tag{2.8}$$

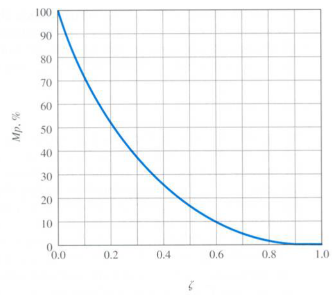


Figure 2.9: Dependence of overshoot on damping ratio for second-order system.

The final time-domain feature of importance to us is the settling time. This is the time required for the response to settle to within some small fraction of its steady-state value and stay there. A typical value of the error-tolerance is 1%, for which we can compute the settling time as

$$t_s \cong \frac{4.6}{\zeta\omega_n} \tag{2.9}$$

In order to settle in t_s sec or less, we require that

$$\zeta\omega_n \geq \frac{4.6}{t_s}. \tag{2.10}$$

Once the three time-domain features are specified, we can use the three formulae, (2.6),(2.7) and (2.9) to find out the requirements on damping ratio, ζ and natural frequency ω_n, from which we can easily translate them to the specifications on the pole positions in the s-plane,

$$s = -\zeta\omega \pm j\omega_n\sqrt{1-\zeta^2}. \tag{2.11}$$

whose amplitude and phase are ω_n and $\cos\varphi = \zeta$, which are shown in the Fig. 2.10.

Example 2.1 How to specify the poles using the settling time, rise time and overshoot?

Consider the following requirement on the transient response:

- The percent overshoot is not greater than 10%.
- The rise time is not more than 0.9 sec.

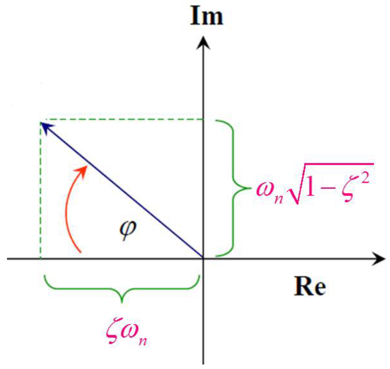


Figure 2.10: Poles of the second order system.

- The settling time is not more than 3.45 sec.

From the requirement on rise time, using (2.7), we can obtain

$$\frac{1.8}{\omega_n} \leq 0.9 \Longrightarrow \omega_n \geq 2$$

From the requirement on overshoot, using (2.6), we have

$$e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 0.1 \Longrightarrow \zeta \geq 0.59$$

Finally, from the requirement on settling time, using (2.9), we have

$$\frac{4.6}{\zeta\omega_n} \leq 3.45 \Longrightarrow \ \zeta\omega_n \geq \frac{4}{3}$$

Overall, all the poles have to be in the shaded area as shown in Fig. 2.11. After the poles of the systems are specified in the s-plane, they can be easily converted into the requirements in the z-plane using the relation between the poles of the s-plane and the z-plane for the sampled-data system,

$$s \Longrightarrow z = e^{Ts} \tag{2.12}$$

$$s = \sigma + j\omega \Longrightarrow z = e^{Ts} = e^{T(\sigma+j\omega)} = e^{\sigma T}e^{j\omega T}$$

Example 2.2 How to specify the poles in the discrete-time system?

Suppose that the percent overshoot is required to be 18%, and the settling is 7.3 sec. For a digital control system with sampling time $T = 1$, please

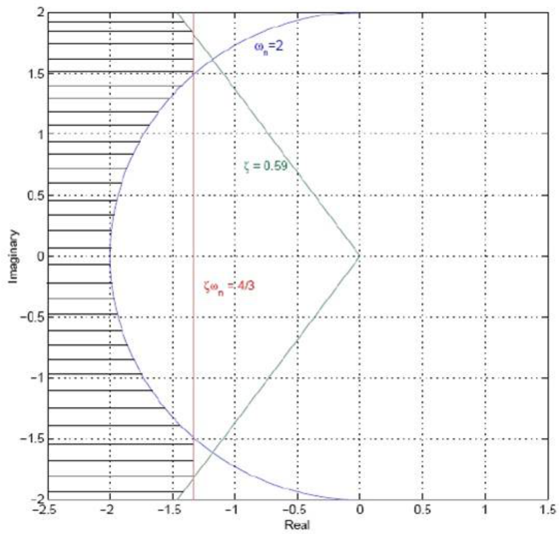


Figure 2.11: Position of the poles to satisfy the time domain performance requirement.

find out the desired positions of the poles in the z-plane.

From the requirement on overshoot, using (2.6), we have

$$e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} = 0.18 \Longrightarrow \zeta = 0.5$$

From the requirement on settling time, using (2.9), we have

$$\frac{4.6}{\zeta\omega_n} = 7.3 \Longrightarrow \ \omega_n = 1.26$$

Therefore, the poles in the s-plane are

$$s = -\zeta\omega \pm j\omega_n\sqrt{1-\zeta^2} = -0.63 \pm j1.0912$$

From the map (2.12), we get the location of the poles in the z-plane, which is displayed in the Fig. 2.12.

$$z = e^{Ts} = 0.25 \pm j0.47$$

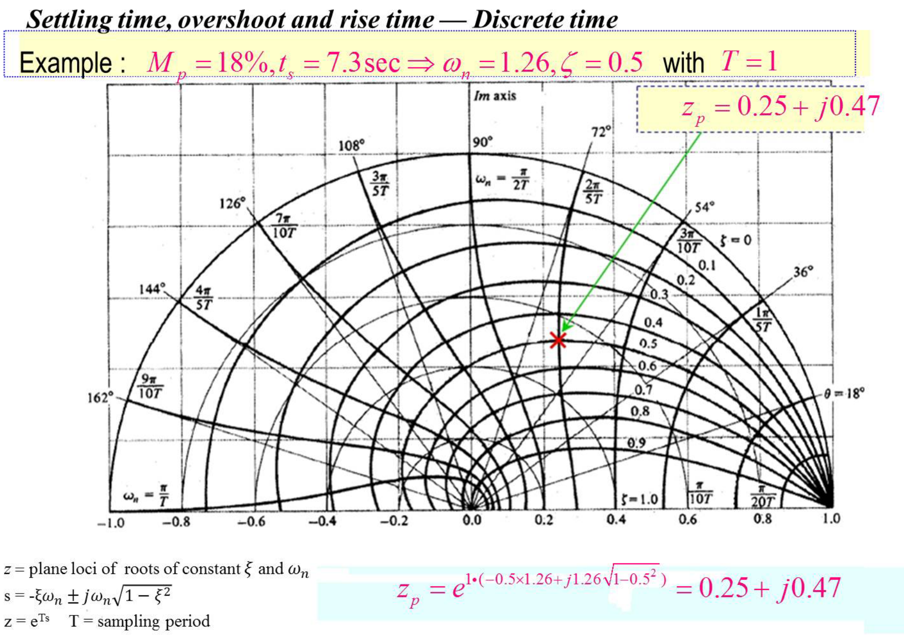


Figure 2.12: Position of the poles in z-plane to satisfy the time domain requirement.

Example 2.3 Find out the poles in the discrete-time system (with unity sampling period) for example 2.1.

From example 2.1, we have
$\omega_n \geq 2$, $\zeta \geq 0.59$, and $\zeta\omega_n \geq \frac{4}{3}$
We now need to convert these specifications into guidelines on the placement of poles in the z-plane in order to guide the design of digital controllers. We do so by mapping via $z = e^{sT}$. Thus the restriction on percent overshoot has been expressed as a restriction on damping ratio, ζ. In the z-plane, curves of pole locations for constant ζ are logarithmic spirals, as shown in Fig. 2.12 and 2.13. The restriction on rise time is the requirement that the natural frequency be greater than a certain value. In the z-plane the curves of constant ω_n are lines drawn at right angles to the constant ζ spirals as shown in Fig. 2.12 and 2.13. The final time-domain specification was in terms of settling time. In this case, the real parts of the roots, $-\zeta\omega_n$, were restricted. Because the $s - z$ mapping has the z-plane root radius at $r = e^{-\zeta\omega_n T}$, we see at once that a setting-time restriction maps into a restriction that the z-plane poles should be inside a circle given by $r = e^{-4/3} = 0.2636$.

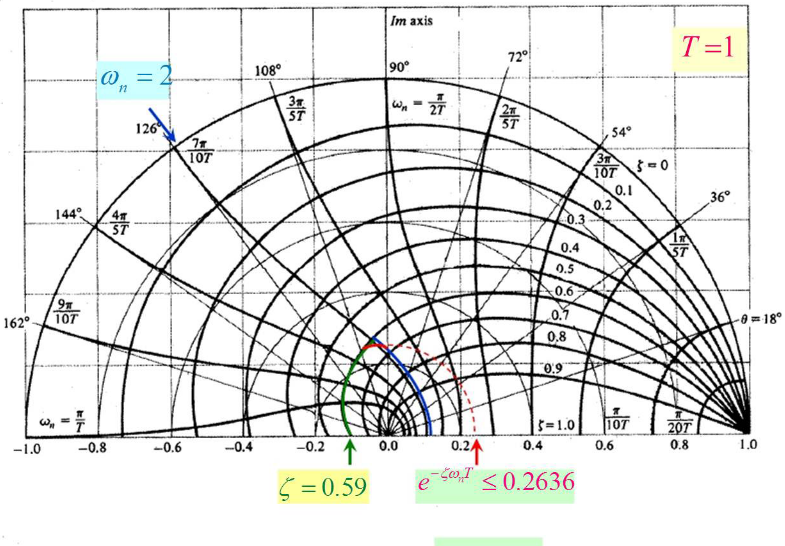


Figure 2.13: Plot of acceptable region for poles of a second-order system to satisfy dynamic response specifications.

The final effect is drawn in Fig. 2.13. The final design must be checked by simulation and/or experiment, and modifications must be made as indicated by the manner in which the first design fails to meet the specifications. For example, if a trial design has a settling time that is too long, then the radius of the poles should be reduced. Likewise, if corrections to overshoot and rise time are needed, Fig. 2.13 indicates how the specified z-plane pole locations should be modified to accomplish the desired result.

2.4.4 How to Deal with Higher Order Systems?

The time-domain performance specifications are normally given for standard second-order system. However, most of the practical systems have higher orders, and we need to know how to characterize the transient behavior of the higher-order system. In the following, we are going to use one example to illustrate the idea on how to approximate the dynamics of the higher-order system by first or second-order systems.

Consider a system described by the transfer function

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

Step One: Compute the Poles

The poles of the above system can be easily obtained as $s = 2$ and $s = -0.5 \pm j\frac{\sqrt{3}}{2}$

Step Two: identify the fast and slow modes

For higher-order system, we need to know the fast and slow modes of the dynamics. Since the fast mode decreases to zero very fast, the dynamics of the system is dominated by the slow mode of the system. The fast and slow modes can by be easily checked by the real parts of the poles as follows. Generally speaking, if p is the pole of the system, then $\frac{1}{s-p}$ must be a part of the response of the system in s-domain using the partial fraction expansion. Hence the corresponding time-domain response: e^{pt}, must be a part of the response of the system.

For pole $p = -2$, its time-domain response is: e^{-2t}.
For poles $s = -0.5 \pm j\frac{\sqrt{3}}{2}$, the time domain responses are $e^{-0.5t}e^{\pm j\frac{\sqrt{3}}{2}t}$
Obviously, e^{-2t} will decrease much faster to zero than $e^{-0.5t}e^{\pm j\frac{\sqrt{3}}{2}t}$. Generally, the bigger the absolute value of the real part, the faster its corresponding time-domain response goes to zero!

Step Three: approximate the system using the slow mode

In order to approximate the higher order system using the lower ones, we can ignore the fast mode and keep the slow mode only to approximate the dynamics of the system since the fast mode converges to zero very quickly. Therefore, we can approximate this system as:

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

How good is the approximation? The step responses of the original higher order-system $H_1(s)$ and the reduced second order system $H_2(s)$ are plotted out in Fig. 2.14. It can be easily seen that the approximation is quite reasonable. There are only minor differences in the transient part, while the steady state responses are almost the same! Hence we can conclude that the higher order system can be approximated reasonably well by second order or even first order system!

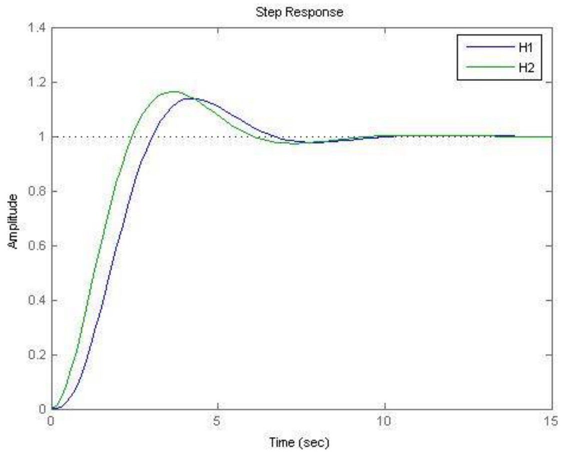


Figure 2.14: Comparison of step responses for the third order system and its approximation.

2.4.5 Model Reference Control

The significance of studying 2nd order systems is:

1. Transient behavior of engineering systems above 2nd order cannot be simply modeled, and the common way is to approximate by a 2nd order by identifying the fast and slow modes of the system.

2. We can design a desired reference model $H_{desired}(s/z)$, which satisfies all time domain specifications. Then the controller design task renders to the selection of an appropriate controller $C(s/z)$ such that the actual closed loop transfer function matches or close to the desired one.

In our daily life, we learn from a good example, as a reference model, and adjust our behavior or manner accordingly (control actions). This idea can be easily adopted into control system design, which is illustrated in the following Fig. 2.15.

What is the ultimate goal of the control system? To make the output of the system follow some desired trajectory! But how to specify the desired

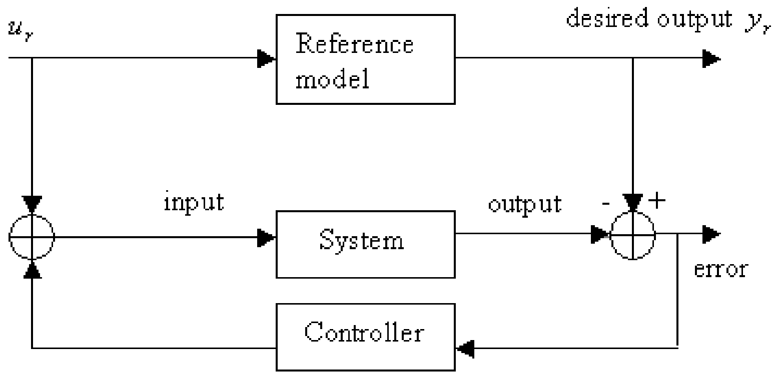


Figure 2.15: Model reference control system

output? One way of doing this is to generate the desired output through a well defined reference model to meet various system requirements. And then the next step is to design the controller such that the behavior of the overall system approaches that of the reference model. In other words, we will try to make the transfer function of the closed loop system as close to the reference model as possible. This approach is called model reference control.
How to specify the appropriate reference model in practice?
It depends upon the performance specifications. In many cases, the reference model can be specified by a second order system which satisfies all the performance requirements,

$$H_{desired}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

where ω_n and ζ are chosen to satisfy the performance specifications. That is the reason that we need to understand the behavior of the second-order system completely.
For Digital Control System, the reference model can be converted from the continuous reference model using the formula

$$H_{desired}(z) = (1 - z^{-1})Z[\frac{H_{desired}(s)}{s}]$$

The conversion of various transfer functions from continuous time to discrete-time is also listed in Table 2.1.
For easy reference, the Laplace transforms and Z-transforms of common functions are also listed in Table 2.2.

Table 2.1 Zero-order hold sampling of a continuous-time system, $G(s)$. The table gives the zero-order-hold equivalent of the continuous-time system, $G(s)$, preceded by a zero-order hold. The sampled system is described by its pulse-transfer operator. The pulse-transfer operator is given in terms of the coefficients of

$$H(q) = \frac{b_1 q^{n-1} + b_2 q^{n-2} + \ldots + b_n}{q^n + a_1 q^{n-1} + \ldots + a_n}$$

$G(s)$	$H(q)$ or the coefficients in $H(q)$
$\frac{1}{s}$	$\frac{h}{q-1}$
$\frac{1}{s^2}$	$\frac{h^2(q+1)}{2(q-1)^2}$
$\frac{1}{s^m}$	$\frac{q-1}{q} \lim_{a \to 0} \frac{(-1)^m}{m!} \frac{\partial^m}{\partial a^m} \left(\frac{q}{q - e^{-ah}} \right)$
e^{-sh}	q^{-1}
$\frac{a}{s+a}$	$\frac{1 - \exp(-ah)}{q - \exp(-ah)}$
$\frac{a}{s(s+a)}$	$b_1 = \frac{1}{a}(ah - 1 + e^{-ah}) \qquad b_2 = \frac{1}{a}(1 - e^{-ah} - ahe^{-ah})$ $a_1 = -(1 + e^{-ah}) \qquad a_2 = e^{-ah}$
$\frac{a^2}{(s+a)^2}$	$b_1 = 1 - e^{-ah}(1 + ah) \qquad b_2 = e^{-ah}(e^{-ah} + ah - 1)$ $a_1 = -2e^{-ah} \qquad a_2 = e^{-2ah}$
$\frac{s}{(s+a)^2}$	$\frac{(q-1)he^{-ah}}{(q - e^{-ah})^2}$
$\frac{ab}{(s+a)(s+b)}$ $a \neq b$	$b_1 = \frac{b(1 - e^{-ah}) - a(1 - e^{-bh})}{b - a}$ $b_2 = \frac{a(1 - e^{-bh})e^{-ah} - b(1 - e^{-ah})e^{-bh}}{b - a}$ $a_1 = -(e^{-ah} + e^{-bh})$ $a_2 = e^{-(a+b)h}$

Table 2.1 continued

$G(s)$	$H(q)$ or the coefficients in $H(q)$
$\dfrac{(s+c)}{(s+a)(s+b)}$ $a \neq b$	$b_1 = \dfrac{e^{-bh} - e^{-ah} + (1-e^{-bh})c/b - (1-e^{-ah})c/a}{a-b}$ $b_2 = \dfrac{c}{ab}\, e^{-(a+b)h} + \dfrac{b-c}{b(a-b)}\, e^{-ah} + \dfrac{c-a}{a(a-b)}\, e^{-bh}$ $a_1 = -e^{-ah} - e^{-bh} \qquad a_2 = e^{-(a+b)h}$
$\dfrac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$	$b_1 = 1 - \alpha\left(\beta + \dfrac{\zeta\omega_0}{\omega}\gamma\right) \qquad \omega = \omega_0\sqrt{1-\zeta^2} \qquad \zeta < 1$ $b_2 = \alpha^2 + \alpha\left(\dfrac{\zeta\omega_0}{\omega}\gamma - \beta\right) \qquad \alpha = e^{-\zeta\omega_0 h}$ $a_1 = -2\alpha\beta \qquad \beta = \cos(\omega h)$ $a_2 = \alpha^2 \qquad \gamma = \sin(\omega h)$
$\dfrac{s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$	$b_1 = \dfrac{1}{\omega}\, e^{-\zeta\omega_0 h}\sin(\omega h) \qquad b_2 = -b_1$ $a_1 = -2e^{-\zeta\omega_0 h}\cos(\omega h) \qquad a_2 = e^{-2\zeta\omega_0 h}$ $\omega = \omega_0\sqrt{1-\zeta^2}$
$\dfrac{a^2}{s^2 + a^2}$	$b_1 = 1 - \cos ah \qquad b_2 = 1 - \cos ah$ $a_1 = -2\cos ah \qquad a_2 = 1$
$\dfrac{s}{s^2 + a^2}$	$b_1 = \dfrac{1}{a}\sin ah \qquad b_2 = -\dfrac{1}{a}\sin ah$ $a_1 = -2\cos ah \qquad a_2 = 1$
$\dfrac{a}{s^2(s+a)}$	$b_1 = \dfrac{1-\alpha}{a^2} + h\left(\dfrac{h}{2} - \dfrac{1}{a}\right) \qquad \alpha = e^{-ah}$ $b_2 = (1-\alpha)\left(\dfrac{h^2}{2} - \dfrac{2}{a^2}\right) + \dfrac{h}{a}(1+\alpha)$ $b_3 = -\left[\dfrac{1}{a^2}(\alpha - 1) + \alpha h\left(\dfrac{h}{2} + \dfrac{1}{a}\right)\right]$ $a_1 = -(\alpha+2) \qquad a_2 = 2\alpha + 1 \qquad a_3 = -\alpha$

Table 2.2 Table of Laplace and Z Transforms

Entry #	Laplace Domain	Time Domain	Z Domain (t=kT)
1	1	$\delta(t)$ unit impulse	1
2	$\frac{1}{s}$	$u(t)$ unit step	$\frac{z}{z-1}$
3	$\frac{1}{s^2}$	t	$\frac{Tz}{(z-1)^2}$
4	$\frac{1}{s+a}$	e^{-at}	$\frac{z}{z-e^{-aT}}$
5		$b^k \quad (b=e^{-aT})$	$\frac{z}{z-b}$
6	$\frac{1}{(s+a)^2}$	te^{-at}	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
7	$\frac{1}{s(s+a)}$	$\frac{1}{a}(1-e^{-at})$	$\frac{z(1-e^{-aT})}{a(z-1)(z-e^{-aT})}$
8	$\frac{b-a}{(s+a)(s+b)}$	$e^{-at}-e^{-bt}$	$\frac{z(e^{-aT}-e^{-bT})}{(z-e^{-aT})(z-e^{-bT})}$
9	$\frac{1}{s(s+a)(s+b)}$	$\frac{1}{ab}-\frac{e^{-at}}{a(b-a)}-\frac{e^{-bt}}{b(a-b)}$	
10	$\frac{1}{s(s+a)^2}$	$\frac{1}{a^2}(1-e^{-at}-ate^{-at})$	
11	$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}$	
12	$\frac{b}{s^2+b^2}$	$\sin(bt)$	$\frac{z\sin(bT)}{z^2-2z\cos(bT)+1}$
13	$\frac{s}{s^2+b^2}$	$\cos(bt)$	$\frac{z(z-\cos(bT))}{z^2-2z\cos(bT)+1}$
14	$\frac{b}{(s+a)^2+b^2}$	$e^{-at}\sin(bt)$	$\frac{ze^{-aT}\sin(bT)}{z^2-2ze^{-aT}\cos(bT)+e^{-2aT}}$
15	$\frac{s+a}{(s+a)^2+b^2}$	$e^{-at}\cos(bt)$	$\frac{z^2-ze^{-aT}\cos(bT)}{z^2-2ze^{-aT}\cos(bT)+e^{-2aT}}$

2.4.6 Disturbance Rejection

The system is always subject to various disturbances. It is customary to distinguish among different types of disturbances, such as load disturbances, measurement errors, and parameter variations.

Load disturbances

Load disturbances influence the process variables. They may represent disturbance forces in a mechanical system – for example, wind gusts on a stabilized antenna, waves on a ship, load on a motor. In process control, load disturbances may be quality variations in a feed flow or variations in demanded flow. In thermal systems, the load disturbances may be variations in surrounding temperature. Load disturbances typically vary slowly. They may also be periodic – for example, waves in ship-control systems.

Measurement errors

Measurement errors enter in the sensors. There may be a steady-state error in some sensors due to errors in calibration. However, measurement errors typically have high-frequency components. There may also be dynamic errors because of sensor dynamics. There may also be complicated dynamic interaction between sensors and the process. Typical examples are gyroscopic measurements and measurement of liquid level in nuclear reactors. The character of the measurement errors often depends on the filtering in the instruments. It is often a good idea to look at the instrument and modify the filtering so that it fits the particular problem.

Parameter variations

Linear theory is used throughout this module. The load disturbance and the measurement noise then appear additively. Real systems are, however, often nonlinear. This means that disturbances can enter in a more complicated way. Because the linear models are obtained by linearizing the nonlinear models, some disturbances then also appear as variations in the parameters of the linear model.

Simple Disturbance Models

There are four different types of disturbances-impulse, step, ramp, and sinusoid-that are commonly used in analyzing control systems. These dis-

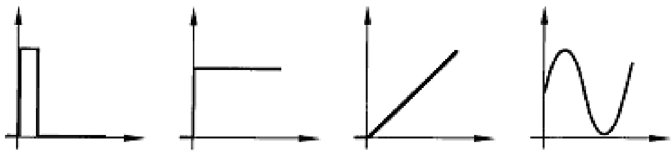


Figure 2.16: Idealized models of simple disturbances.

turbances are illustrated in Fig. 2.16 and a discussion of their properties follows.

The impulse and the pulse

The impulse and the pulse are simple idealizations of sudden disturbances of short duration. They can represent load disturbances as well as measurement errors. For continuous systems, the disturbance is an impulse (a delta function); for sampled systems, the disturbance is modeled as a pulse with unit amplitude and a duration of one sampling period.

The pulse and the impulse are also important for theoretical reasons because the response of a linear continuous-time system is completely specified by its impulse response and a linear discrete-time system by its pulse response.

The step

The step signal is another prototype for a disturbance (see Fig. 2.16). It is typically used to represent a load disturbance or an offset in a measurement.

The ramp

The ramp is a signal that is zero for negative time and increases linearly for positive time (see Fig. 2.16). It is used to represent drifting measurement errors and disturbances that suddenly start to drift away. In practice, the disturbances are often bounded; however, the ramp is a useful idealization.

The sinusoid

The sine wave is the prototype for a periodic disturbance. Choice of the frequency makes it possible to represent low-frequency load disturbances, as well as high-frequency measurement noise.

Disturbance Rejection

The effectiveness of the system in disturbance rejection is readily studied with the topology of the unity feedback system as shown below.
We set the reference $r = 0$ for simplicity and for the study of the effects of

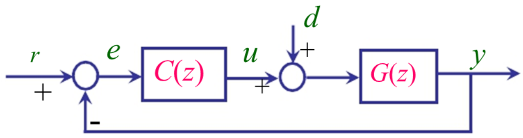


Figure 2.17: A unity feedback system.

the disturbance d (unwanted signal) on the system output.
Note that

$$Y(z) = G(z)[D(z) - C(z)Y(z)]$$

We get

$$Y(z) = \frac{G(z)}{1 + G(z)C(z)} D(z)$$

Consider the case when d is a unit step signal, we have

$$y(\infty) = \lim_{z \to 1}(z-1)Y(z) = \lim_{z \to 1}(z-1)\frac{G(z)}{1 + G(z)C(z)}\frac{z}{z-1} = \frac{G(1)}{1 + G(1)C(1)}$$

In order to eliminate the effect of the constant disturbance, i.e. make $y(\infty) = 0$, we need to let either $G(1) = 0$ or $C(1) = \infty$. The first method of making $G(1) = 0$ may not be practical as we cannot change the plant directly. The second choice of letting $C(1) = \infty$ is much easier as we just need to put an integrator in the controller.

2.4.7 Control Effort

The control effort required to perform a control task is important on several counts. Because all physical variables are bounded, the device that provides the control, such as the motor that drives the tracking antenna, can put out only a certain maximum torque even when turned on fully. It is pointless to try to get 100 N-m out of 10 mN-m motor! Conversely, after completion of a design meeting dynamic-response specifications, we can simulate a worst-case transient; and from the size of control signal required, we can determine the size of the motor necessary to meet these specifications. In addition to peak control, $\|u\|$, we are sometimes interested in the total heat generated by the drive motor. This, too, will influence the size and design (and expense) of the motor. Usually this number is proportional to $\int u^2 dt$ over a typical transient period. Another measure of control effort arises in gas jets used for attitude control of satellites, where the total fuel used is a proper measure of control effort, and where the fuel expenditure is proportional to $\int_0^\infty u^2 dt$. The theory and applications of optimal control are an effort to include these objectives directly in the design.

2.4.8 Sensitivity to Parameter Changes

Sensitivity to parameter changes needs to be studied separately for changes in plant parameters and for changes in controller parameters. As to changes in the parameters of the plant, the situation is very much like the disturbance rejection, and both features are contained in the concept of robustness. The larger the gain of the feedback loop around the offending parameter, the lower the sensitivity of the transfer function to changes in that parameter. Because in the most common cases we have very slowly varying parameters, we are led to design for high gain in the vicinity of z=1, which corresponds to very slow or constant signals. If this high gain is in front of the disturbance, then we will also achieve good disturbance rejection. The second aspect concerns the effects of changes in the controller, $C(z)$. Here, we have control over the topology, and a design choice can be made to minimize the effects of parameter changes in $C(z)$.

2.4.9 Other Considerations

The designer's job is to meet the specifications. This can typically be done in many different ways. There are different kinds of actuators that can be used, different sensors and sensed quantities are possible for selection, choices in the plant design are a design variable, and the control law can be selected. The designer must pick the most cost-effective combination of these options to meet the desired system performance.

Chapter 3

PID Control

3.1 Introduction to PID control

PID control is widely used in process control and most of industrial control systems. Statistics show that more than 90% of industrial processes are actually controlled by PID type of controllers. PID control consists of three essential components, namely, P (proportional control), I (integral control) and D (derivative control).
Two main advantages offered by PID controllers are:

1. They are very easy to implement.
2. They are model-free.

3.2 Implementation of Digital PID Control

Digital P (proportional control), I (integral control) and D (derivative control) can be derived from the continuous-time counterpart by discretization using the discrete equivalents.

3.2.1 Proportional Control

A discrete implementation of proportional control is identical to continuous; that is, where the continuous is

$$u(t) = K_p e(t) \Rightarrow C(s) = K_p$$

the discrete is

$$u(k) = K_p e(k) \Rightarrow C(z) = K_p$$

where $e(t)$ is the error signal.

3.2.2 Derivative Control

For continuous systems, derivative control has the form

$$u(t) = K_d \dot{e}(t) \Rightarrow C(s) = K_d s$$

Differentiation can be approximated in the discrete domain as the first difference, that is,

$$\dot{e}(t) \approx \frac{e(t) - e(t-T)}{T}$$

It follows that

$$u(k) = K_d \frac{e(k) - e(k-1)}{T} \Rightarrow C(z) = K_d \frac{1 - z^{-1}}{T} = K_d \frac{z-1}{Tz}$$

3.2.3 Integral Control

For continuous systems, we integrate the error to arrive at the control,

$$u(t) = K_i \int_0^t e(\tau) d\tau \Rightarrow C(s) = K_i \frac{1}{s}$$

It follows that

$$u(t) = K_i \int_0^t e(\tau) d\tau = K_i \int_0^{t-T} e(\tau) d\tau + K_i \int_{t-T}^t e(\tau) d\tau = u(t-T) + K_i \int_{t-T}^t e(\tau) d\tau.$$

Then the question is: how to approximate $\int_{t-T}^t e(\tau) d\tau$? The easiest way is to use

$$\int_{t-T}^t e(\tau) d\tau \approx T e(t)$$

Hence, the discrete equivalent of the integral control can be derived as

$$u(k) = u(k-1) + K_i T e(k) \Rightarrow C(z) = K_i \frac{T}{1 - z^{-1}} = K_i \frac{Tz}{z-1}$$

Just as for continuous systems, the primary reason for integral control is to reduce or eliminate steady-state errors, but this typically occurs at the cost of reduced stability.

3.2.4 PID Control

Combining all the above yields the "textbook" version of the PID controller

$$C(z) = K_p + K_i \frac{Tz}{z-1} + K_d \frac{z-1}{Tz} \tag{3.1}$$

This form of control law is able satisfactorily to meet the performance specifications for a large portion of control problems and is therefore packaged commercially and sold for general use. The user simply has to determine the best values of K_p, K_i, and K_d.
Different combination of the three components also leads to other commonly used controllers such as PI control

$$C(z) = K_p + K_i \frac{Tz}{z-1} \tag{3.2}$$

and PD control

$$C(z) = K_p + K_d \frac{z-1}{Tz}\,. \tag{3.3}$$

3.3 Digital PID Control System Design using Emulation

In this approach, we design a continuous-time controller that meets all design specifications and then discretize it using a bilinear transformation or other discretization technique to obtain an equivalent digital controller.

This method works if the sampling rate is 20-30 times faster than the system bandwidth. Further refinement is necessary for the case where the sampling rate is 6 (or lower) times the bandwidth.
Emulation design procedure consists of the following 5 steps:

1. From time domain performance specifications, determine the parameters ζ and ω_n, as well as the desired closed loop transfer function.
2. Choose an appropriate controller, e.g. P, PI, PD, PID, etc.
3. Comparing characteristic equations of desired and actual CLTF, determine the controller parameters, and complete the design for the continuous systems.
4. From the system bandwidth, select a sampling period T.
5. Discretize $C(s)$ to $C(z)$ via Tustin rule (replace s by $\frac{2}{T}\frac{z-1}{z+1}$).

3.3.1 Design Example 3.1: Position Control with P or PD

Consider a vehicle, which has a weight m = 1000 kg. Assuming the average friction coefficient b= 100, design a position control system such that the vehicle can move 100 m in 7.3 s with an overshoot less than 18%.
The position of the vehicle, x, is affected by the engine force u, and friction force. Applying Newton's Second Law leads to

$$m\ddot{x} = u - b\dot{x}$$

which results in the following open-loop transfer function

$$G(s) = \frac{X(s)}{U(s)} = \frac{1}{ms^2 + bs} = \frac{1}{1000s^2 + 100s}$$

Step One: determining ζ and ω_n from the design specifications

From the requirement on overshoot, using (2.6), we have

$$e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} = 0.18 \Longrightarrow \zeta = 0.5$$

From the requirement on settling time, using (2.9), we have

$$\frac{4.6}{\zeta\omega_n} = 7.3 \Longrightarrow \ \omega_n = 1.2603$$

Therefore, the desired closed-loop transfer function is

$$H_d(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1.5883}{s^2 + 1.2603s + 1.5883}$$

Step 2. Choose an appropriate controller

In Chapter One, we have shown that for the simplest configuration of the feedback control system
the closed loop transfer function from the reference signal $R(s)$ to output

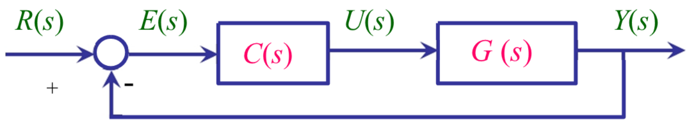


Figure 3.1: Feedback Control System

$Y(s)$:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{3.4}$$

Which controller should be our first choice?
It is always a good strategy to start with the simplest one first. Therefore, let's choose the Proportional Controller first,

$$C(s) = K_p.$$

It follows that

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} = \frac{0.001K_p}{s^2 + 0.1s + 0.001K_p}$$

However the desired CE and actual CE are respectively

$$s^2 + 1.2603s + 1.5883 \Longrightarrow s^2 + 0.1s + 0.001K_p$$

which obviously cannot match each other.
Let's choose the standard PD controller,

$$C(s) = K_p + K_d s.$$

It follows that

$$H(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} = \frac{0.001K_p + 0.001K_d s}{s^2 + (0.1 + 0.001k_d)s + 0.001K_p}$$

Since the zeros of above CLTF do not match the standard form of the second order system, we still cannot meet the performance specification. Therefore, we need to choose a more sophisticated controller, a modified PD controller, with $C_1(s) = K_p$ and $C_2(s) = 1 + K_d s$, as shown in the following block diagram 3.2.

The corresponding closed loop transfer function is then

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G(s)C_1(s)}{1 + G(s)C_1(s)C_2(s)} = \frac{0.001K_p}{s^2 + (0.1 + 0.001K_pK_d)s + 0.001K_p}$$

Step 3. Calculate controller parameters

Compare the above CLTF with the desired CLTF

$$H_d(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1.5883}{s^2 + 1.2603s + 1.5883}$$

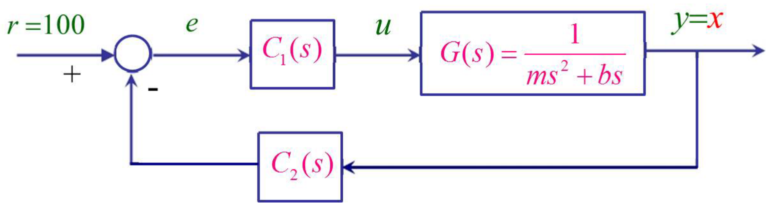


Figure 3.2: Modified PD controller

to match them completely, we have

$$0.001K_p = 1.5883 \Longrightarrow \ K_p = 1588.3$$

and

$$0.1 + 0.001K_pK_d = 1.2603 \Longrightarrow \ K_d = 0.73.$$

The MATLAB Simulink block diagram and the step response of the closed loop are shown in Fig. 3.3. The overshoot is around 16% and the settling time is about 7s, which meet the performance specifications.

But that is the result by the analog controller. We need to convert it into

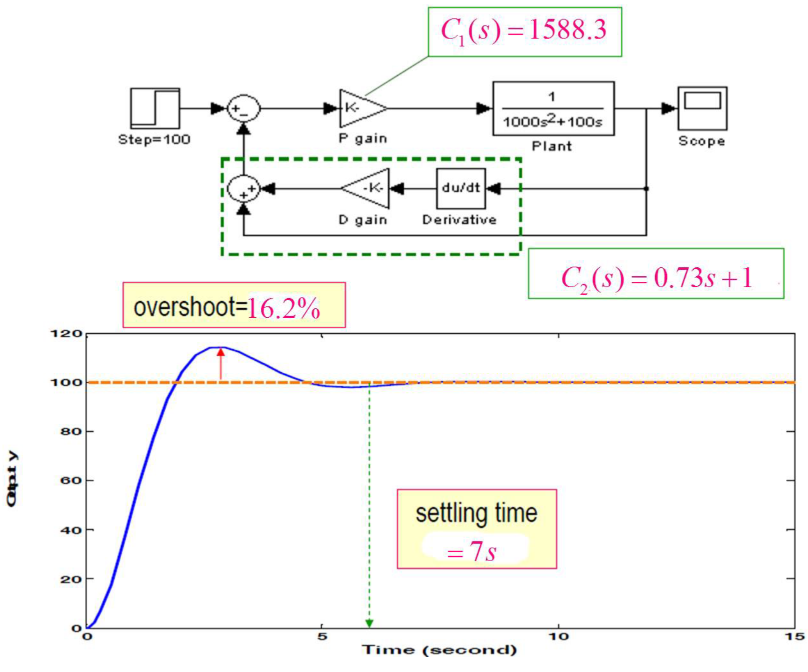


Figure 3.3: Step Response with Modified PD controller

digital controller. We need to decide the sampling rate first.

Step 4. From the system bandwidth, select a sampling period T

First use the Bode plot to determine the bandwidth of the closed-loop system, which is defined to be the frequency whose corresponding magnitude drops by -3dB from the DC gain. As shown in Fig. 3.4, the bandwidth is 1.6 rad/sec = 0.2546 HZ.

As mentioned before, for emulation to work well, the sampling rate is at

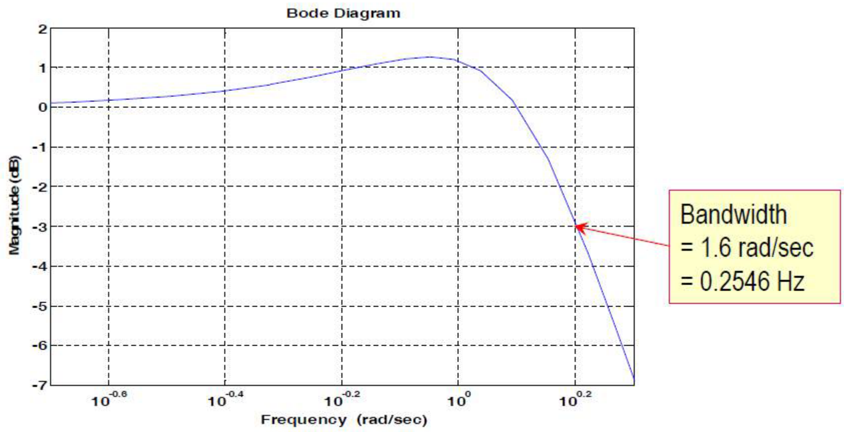


Figure 3.4: Bandwidth of the closed-loop system

least 30 times the bandwidth (30x0.2546=7.64). In this case, let's choose $T = 0.1$, which is smaller than 1/7.64.

Step 5. Discretize C(s) with the selected sampling period T

Discretize the continuous-time PD control law with T =0.1 seconds using Tustin rule,

$$C_2(z) = C_2(s)|_{s=\frac{2}{T}\frac{z-1}{z+1}} = \frac{15.6z - 13.6}{z+1}$$

The resulting step response of the closed-loop system is shown in Fig. 3.5. The overshoot is around 17.9% and the settling time is about 7s, which still meet the performance specifications.

What would happen if we use a smaller sampling rate, say, just round 6 times the bandwidth. In this case, let T=0.5s. Discretize the continuous-

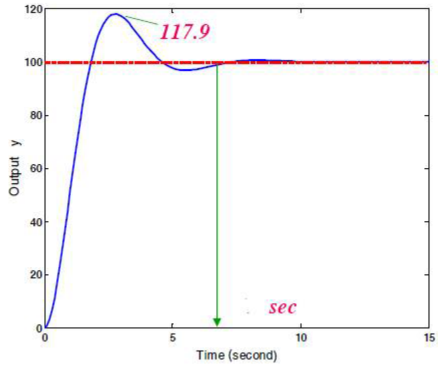


Figure 3.5: Step Response with Modified Digital PD controller using high sampling rate

time PD control law with T =0.5s using Tustin rule,

$$C_2(z) = C_2(s)|_{s=\frac{2}{T}\frac{z-1}{z+1}} = \frac{3.92z - 1.92}{z+1}$$

The resulting step response of the closed-loop system is shown in Fig. 3.6. The overshoot is above 20% and the settling time is around 8s, which is slightly below the performance specifications. The result would be even worse if the sampling rate is below 6 times the bandwidth.

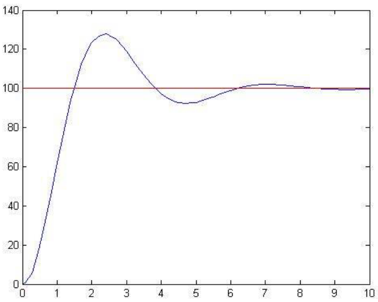


Figure 3.6: Step Response with Modified Digital PD controller using smaller sampling rate

Next, let's consider another example of emulation design.

3.3.2 Design Example 3.2. Speed Control with PI Control

Consider the vehicle, which has a weight m = 1000 kg. Assuming the average friction coefficient b= 100, design a speed control system such that the vehicle can reach 100 km/h from 0 km/h in 8 s with an overshoot less than 5%.

The speed of the vehicle, v, is affected by the engine force u, and friction force. Applying Newton's Second Law leads to

$$m\dot{v} = u - bv$$

which results in the following open-loop transfer function

$$G(s) = \frac{V(s)}{U(s)} = \frac{1}{ms + s} = \frac{1}{1000s + 100}$$

Step One: determining ζ and ω_n from the design specifications

From the requirement on overshoot, using (2.6), we have

$$e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 0.05 \Longrightarrow \zeta \geq 0.6901 \Longrightarrow \zeta = 0.7$$

From the requirement on settling time, using (2.9), we have

$$\frac{4.6}{\zeta\omega_n} = 8 \Longrightarrow \ \omega_n = 0.82$$

Therefore, the desired closed-loop transfer function is

$$H_d(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{0.67}{s^2 + 1.15s + 0.67} \tag{3.5}$$

Step 2. Choose an appropriate controller

The closed loop transfer function from the reference signal $R(s)$ to output $Y(s)$:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G(s)C(s)}{1 + G(s)C(s)} \tag{3.6}$$

Let's choose the Proportional Integral Controller since we want to assure the steady state accuracy.

$$C(s) = K_p + K_i\frac{1}{s}.$$

It follows that

$$H(s)=\frac{G(s)C(s)}{1+G(s)C(s)}=\frac{0.001K_ps+0.001K_i}{s^2+(0.1+0.001K_p)s+0.001K_i}$$

Step 3. Calculate controller parameters

Compare the above CLTF with the desired CLTF

$$H_d(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}=\frac{0.67}{s^2+1.15s+0.67}$$

to match the poles (the denominators of the TF) completely, we have

$$0.1+0.001K_p=1.15 \Longrightarrow \; K_p=1050$$

and

$$0.001K_i=0.67 \Longrightarrow \; K_i=670$$

The resulting closed-loop transfer function is

$$H(s)=\frac{1.05s+0.67}{s^2+1.15s+0.67} \tag{3.7}$$

The MATLAB Simulink block diagram and the step response of the closed loop are shown in Fig. 3.7.

Simulation results show a rather large overshoot, around 18%. Why?
If we compare the actual CLTF with the desired one, we can see that the poles are matched, while the zeros are different. Let's decompose the actual CLTF in terms of the desired TF and an extra term,

$$H(s)=\frac{1.05s+0.67}{s^2+1.15s+0.67}=\frac{0.67}{s^2+1.15s+0.67}+\frac{1.05s}{s^2+1.15s+0.67}$$

The large over shoot is due to the presence of the TF $\frac{1.05s}{s^2+1.15s+0.67}$, which is clearly shown in Fig. 3.7.

Is it possible to match both the poles and zeros? It is possible by increasing the complexity of the controller, which will be discussed later in this module.

Step 4. From the system bandwidth, select a sampling period T

As shown in Fig. 3.7, the bandwidth is 0.25 HZ.

As mentioned before, for emulation to work well, the sampling rate is at least 30 times the bandwidth (30x0.25=7.5). In this case, let's choose $T=0.1s$, which is smaller than 1/7.5.

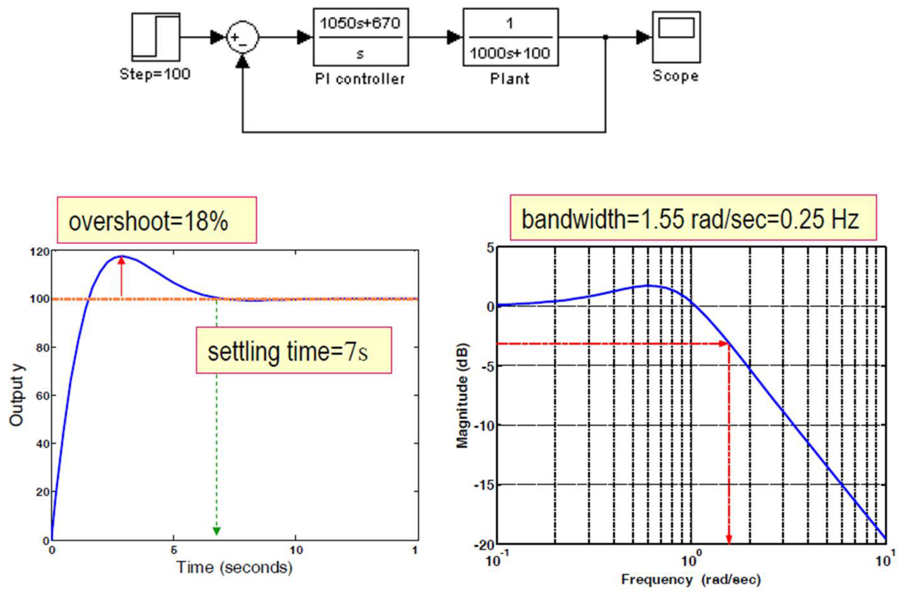


Figure 3.7: Step Response with analog PI controller

Step 5. Discretize C(s) with the selected sampling period T

Discretize the continuous-time PD control law with T =0.1s using Tustin rule,

$$C_{(}z) = C_{(}s)|_{s=\frac{2}{T}\frac{z-1}{z+1}} = \frac{1083.5z - 1016.5}{z-1}$$

The resulting step response of the closed-loop system is shown in Fig. 3.9. The overshoot is around 20% and the settling time is about 8s, which is about the same as that of the analog controller.

What would happen if we use the smaller sampling rate, say, just round 6 times the bandwidth. In this case, let T=0.6s. Discretize the continuous-time PD control law with T =0.5s using Tustin rule,

$$C(z) = C(s)|_{s=\frac{2}{T}\frac{z-1}{z+1}} = \frac{1251z - 849}{z-1}$$

The resulting step response of the closed-loop system is shown in Fig. 3.6. The performance is not as good as that for the analog controller. The overshoot is even larger than 20%. In practice, emulation based design can be used directly without validation when the sampling frequency is sufficiently high in comparing with the system bandwidth. It is important

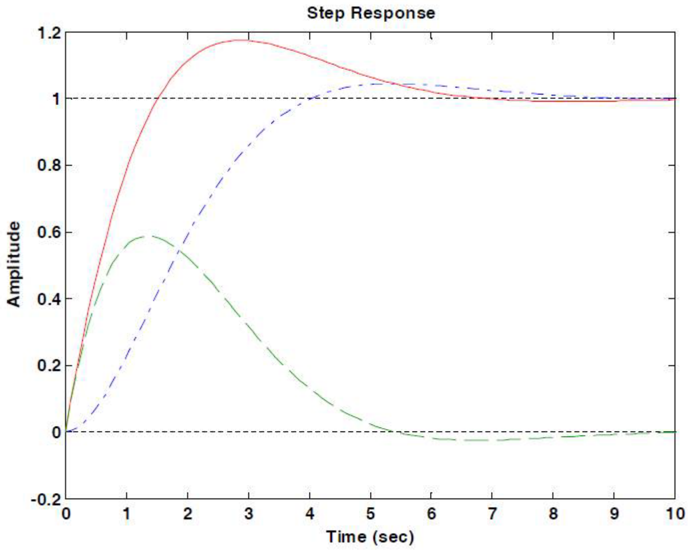


Figure 3.8: Comparing the Step Responses between actual CLTF and desired CLTF

to note that the system bandwidth is referred to the closed-loop system bandwidth instead of the open loop plant.
Next, we are going to design the digital controller directly without relying on the analog controller.

3.4 Digital PID Controller Design based on Discrete-time System

We can discretize the continuous-time plant first or directly work on a discrete-time plant to design a digital controller.

3.4.1 Design Example 3.3: Speed Control with PI

Consider the vehicle, which has a weight m = 1000 kg. Assuming the average friction coefficient b= 100, design a speed control system such that the vehicle can reach 100 km/h from 0 km/h in 8 s with an overshoot less than 5%.

Assuming the sampling period T = 0.6 seconds, design a digital PI con-

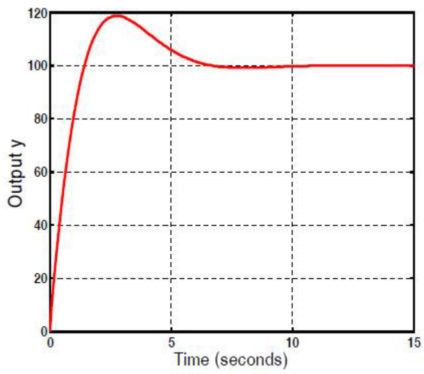


Figure 3.9: Step Response with Digital PI controller using high sampling rate

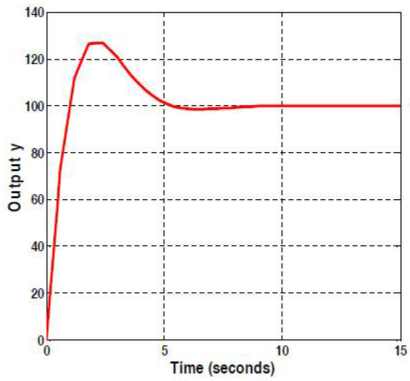


Figure 3.10: Step Response with Digital PI controller using smaller sampling rate

troller that achieves the above specifications. From the previous emulation design, we know that the sampling period T=0.6 is too large to get a good performance. Let's see if the direct design method can work better.
Let's first try to get the discrete-time T.F. from the continuous time T.F.

$$G(z) = (1 - z^{-1})Z[\frac{G(s)}{s}]$$

The open loop TF has been derived as

$$G(s) = \frac{1}{ms + b} = \frac{1}{1000s + 100}$$

It follows that

$$G(z) = (1-z^{-1})Z[\frac{G(s)}{s}] = (1-z^{-1})Z[\frac{0.001}{s(s+0.1)}] = \frac{z-1}{z}Z[\frac{0.01}{s} - \frac{0.01}{s+0.1}]$$

From the z-transform table 2.2, we get

$$Z[\frac{1}{s}] = \frac{z}{z-1}$$

and

$$Z[\frac{1}{s+0.1}] = \frac{z}{z-e^{-0.1\times 0.6}}$$

Overall, we have

$$G(z) = 0.01\frac{z-1}{z}(\frac{z}{z-1} - \frac{z}{z-e^{-0.1\times 0.6}}) = \frac{0.00058}{z-0.942}$$

The block diagram for the PI controller is shown in Fig. 3.11.

The closed loop transfer function from the reference signal, r(k), to the

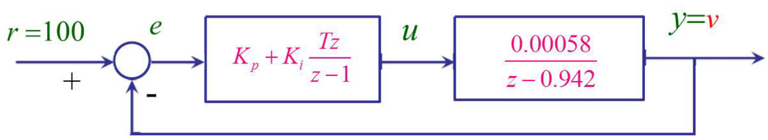


Figure 3.11: Speed Control with Digital PI controller

output, y(k), is

$$H(z) = \frac{Y(z)}{R(z)} = \frac{G(z)C(z)}{1+G(z)C(z)} \tag{3.8}$$

The PI controller is

$$C(z) = K_p + K_i\frac{Tz}{z-1}$$

We can obtain

$$H(z) = \frac{0.00058(K_p+0.6K_i)z - 0.00058K_p}{z^2 + (0.00058(K_p+0.6K_i) - 1.942)z + (0.942 - 0.00058K_p)}$$

From the previous design example, we have already obtained the desired damping ratio, $\zeta = 0.7$ and the desired natural frequency, $\omega_n = 0.82$ from the performance specifications. The desired transfer function for the continuous system is

$$H_d(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{0.67}{s^2 + 1.15s + 0.67}$$

Its corresponding desired TF in the discrete-time system with sampling period $T = 0.6$, can be obtained from the table 2.1 as follows

$$H_d(z) = \frac{0.09521z + 0.07557}{z^2 - 1.331z + 0.5016}$$

Comparing the denominator of the actual closed-loop transfer function with the desired one, we have

$$0.00058(K_p + 0.6K_i) - 1.942 = -1.331$$

and

$$0.942 - 0.00058K_p = 0.5016$$

The two controller gains K_p and K_i can be easily obtained as $K_p = 759.31$ and $K_i = 490.23$. The resulting digital PI controller is

$$C(z) = 759.31 + 490.23\frac{0.6z}{z-1} = 759.31 + \frac{294.14z}{z-1}$$

We simulate the digital PI control system response with the discretized plant to see whether the specifications are fulfilled in the discrete-time setting. The block diagram and the output is shown in Fig. 3.12. It is seen that the overshoot is slightly larger than the continuous-time system, but the settling time meets the specification. The system performance can be fine- tuned by re-selecting the desired pole locations in Z-plane.

Why do we still get large overshoot just like that in continuous time?
If we compare the numerators of the actual closed loop transfer function and the desired one, we can see that they do not match each other. So we only succeed in matching the poles, not the zeros.
Now let's simulate the digital PI control system response with actual plant, the result is shown in Fig. 3.13. When a digital control law is implemented onto the actual continuous- time plant, the overshoot and the settling time is about the same as those obtained with the discretized systems. The response is smooth, even though the control input is updated with a sampling period of T=0.6 seconds.

Disturbance Rejection

From the analysis in 2.4.6, it was shown that a step disturbance can be rejected when the controller has an integral action. Since we are using a PI controller, step disturbance should be rejected in our design although we

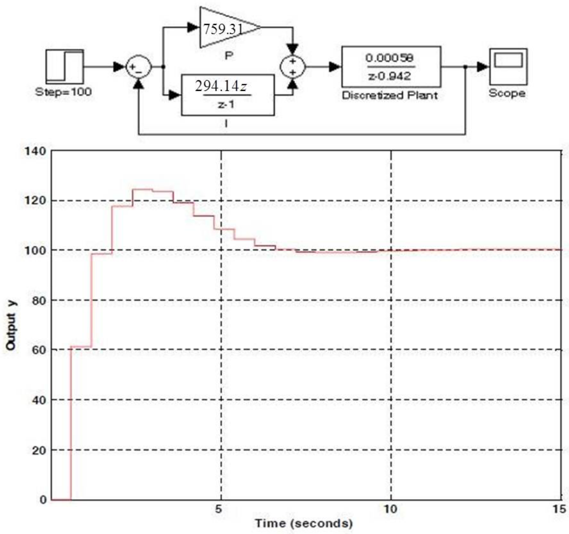


Figure 3.12: Simulated Output in Discrete Setting

haven't explicitly considered such a property in our design. We verify this through simulation by injecting a step function into the plant input. To see the effect of the disturbance on the system output, we let the reference to be zero. The result is shown in Fig. 3.14. It is shown that a step disturbance with magnitude of 100 is nicely rejected from the output.

3.4.2 Higher Order Systems

When the given plant has a dynamic order higher than 1 and/or a general PID controller is used, the overall closed-loop transfer function from r to y will have an order larger than 2. However, the desired closed loop transfer function is normally designed from the standard second order system. In addition to the two desired poles specified by the desired second order system, where should we place the other poles in the higher-order system?
From the previous discussion in section 2.4.4, the higher-order system can be approximated by lower-order system by identifying the fast and slow modes of the system. Hence, one natural way is to place all the rest poles close to the origin, which is the fastest location in digital control. Eventually, dynamics associated with the poles close to the origin will die out very fast and the overall system is dominated by the pair left.

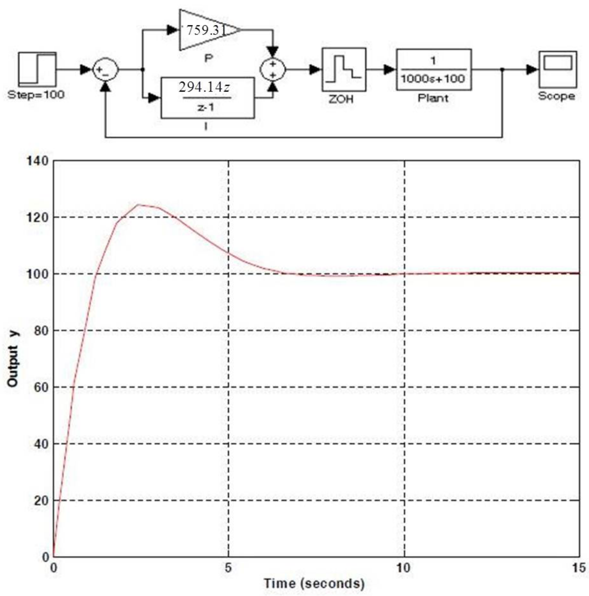


Figure 3.13: Simulated Output in Actual Plant

3.4.3 Deadbeat Controller Design

A unique feature of digital control is that we can design a control law such that the resulting system output is capable of following the reference input in a finite number of steps, i.e., in finite time interval, which can never be achieved in continuous-time setting. Such a control law is called deadbeat controller, which in fact places all the closed-loop system poles at the origin. Thus, the desired closed-loop transfer function under the deadbeat control is

$$H_{desired} = \frac{*}{z^n}$$

The deadbeat control can only guarantee the system output to reach the target reference in n steps. The resulting overshoot could be huge and the control input could be very high (which is equivalent to that the energy required to achieve such a performance is large). As such, deadbeat control is generally impractical and rarely used in practical situations.

Design Example 3.4: Deadbeat Control

We now design a deadbeat PI controller for the speed control system. Recall that the closed loop transfer function is

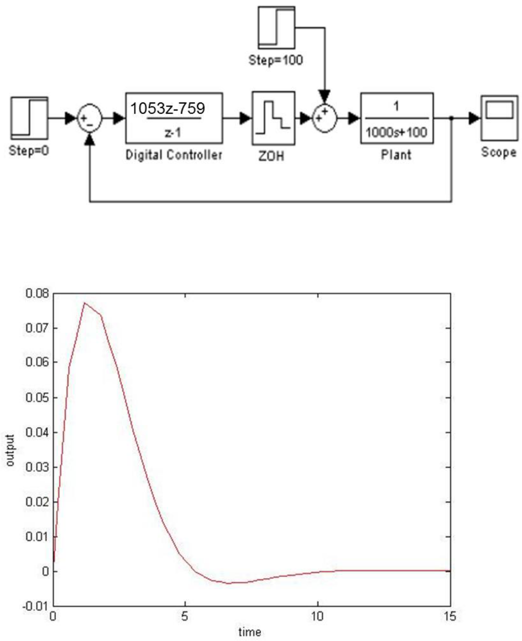


Figure 3.14: Disturbance Rejection in Actual Plant

$$H(z) = \frac{0.00058(K_p + 0.6K_i)z - 0.00058K_p}{z^2 + (0.00058(K_p + 0.6K_i) - 1.942)z + (0.942 - 0.00058K_p)}$$

Compared with the desired deadbeat TF: $\frac{*}{z^2}$, we have

$$0.00058(K_p + 0.6K_i) - 1.942 = 0$$

and

$$0.942 - 0.00058K_p = 0$$

which results in $K_p = 1624.14$ and $K_i = 2873.56$. The corresponding controller is

$$C(z) = 1624.14 + 2873.56\frac{0.6z}{z-1} = \frac{3348.28z - 1624.14}{z-1}$$

The simulated input and output signals are shown in Fig. 3.15. The system output settles down to the target reference in 2 samples only, i.e. $2 \times 0.6 = 1.2$ seconds (very fast). But, it has about 100% overshoot and a huge control input. Such a controller is very hard to be implemented in real life! However, there could be applications (such as military applications) that deadbeat control might be desirable.

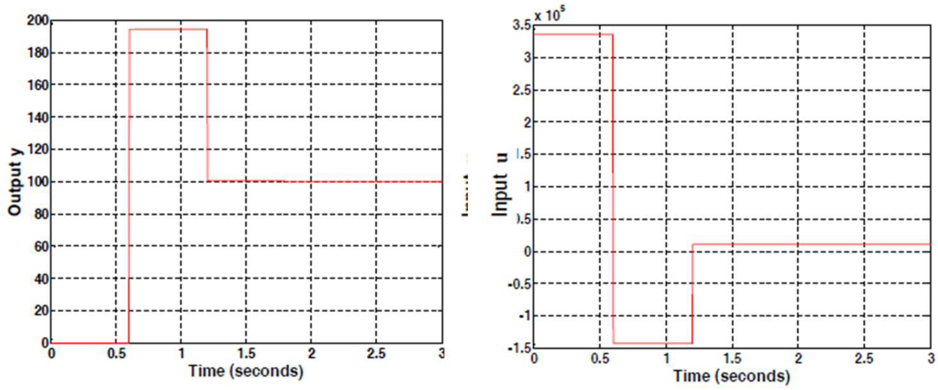


Figure 3.15: Deadbeat Control

3.5 Ziegler-Nicholes PID Tuning

The parameters in the PID controller could be selected by various design methods previously discussed. However, these methods require a dynamic model of the process which is not always readily available. Ziegler-Nichols tuning is a method for picking the parameters based upon fairly simple experiments on the process and thus bypasses the need to determine a complete dynamic model. This is also a big reason why PID control is so popular in industry. It can be model-free in some sense.
There are two classic tuning methods.

3.5.1 1st ZN method – Transient Response Method

The first, called the transient-response method, requires that a step response of the open-loop system is obtained which looks something like that in Fig. 3.16. Three parameters are obtained from the step response: dead-time L, DC gain K, time constant T and the steepest slope, R=K/T, which are shown in the Fig. 3.16. In order to achieve a damping ratio of about $\zeta = 0.2$, the parameters are selected according to those in Table 3.1.
Those heuristic rules were obtained through intensive simulations on many different plant models. From the step response in Fig. 3.16, the plant is essentially a first order system, which limits the applicability of this tuning rule.

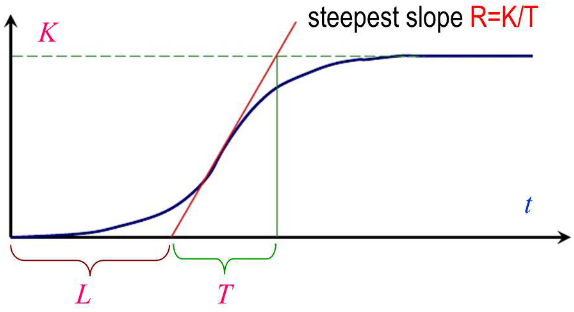


Figure 3.16: Process open-loop step response

Table 3.1 Ziegler-Nichols tuning
Parameters using transient response.

	K_p	K_i	K_d
P	$\frac{1}{RL}$		
PI	$\frac{0.9}{RL}$	$\frac{0.3}{RL^2}$	
PID	$\frac{1.2}{RL}$	$\frac{0.6}{RL^2}$	$\frac{0.6}{R}$

3.5.2 2nd ZN Method – Stability Limit Method

The second method is called stability-limit method. In this method, a proportional controller is first used to form a negative feedback loop. The proportional gain is slowly increasing until continuous (sustained) oscillations occur. At that point, we define two parameters: the critical gain (ultimate gain) K_u, and the critical period (ultimate period) P_u = oscillation period. Then the P, PI and PID control gains can be determined from Table 3.2.

Those heuristic rules were obtained through intensive simulations on many plant models. It pushes the process to the stability limit (at least a pair of poles on the imaginary axis). Such test may not be acceptable for a number of industrial processes such as power plant, aircraft, etc.

The rules are based on continuous system and will apply to the discrete case for very fast sampling (more than 20 times the bandwidth). For slower sampling, a response degradation should be expected, and the gains should

Table 3.2 Ziegler-Nichols tuning Parameters using stability limit.

	K_p	K_i	K_d
P	$0.5K_u$		
PI	$0.45K_u$	$\frac{0.54K_u}{P_u}$	
PID	$0.6K_u$	$\frac{1.2K_u}{P_u}$	$0.075K_uP_u$

be fine tuned.

3.5.3 Design Example 3.5

Consider a process $G(s) = \frac{1}{s(s+1)(s+5)}$,which has an integrator, thus only the second ZN method is applicable.
Apply proportional control and increase the gain such that continuous oscillations result as shown in Fig. 3.17.
The two parameters: $K_u = 30$, and $P_u = 2.81$, can be easily obtained from

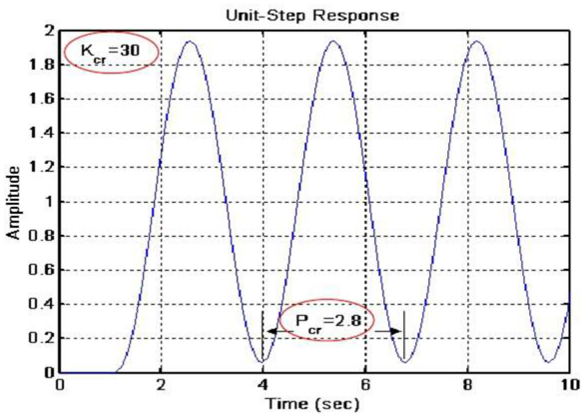


Figure 3.17: Continuous oscillations with proportional control.

Fig. 3.17. Then the control gains can be found out by Table 3.2. as follows.

For PID control, $K_p = 0.6K_u = 18$, $K_i = \frac{1.2K_u}{P_u} = 12.8$, and $K_d = 0.075K_uP_u = 6.32$, and the closed loop step response is shown in Fig. 3.18.

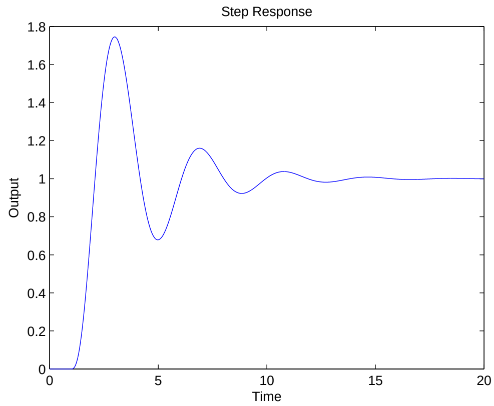


Figure 3.18: Step response with PID control using Z-N Stability Limit Method.

The overshoot is a bit large, whereas the rise time is short. ZN methods tend to produce a fast response. ZN methods provide primary tuning results and the starting point for further fine tuning. Many refined ZN auto-tuning methods were developed in the past 70 years, highlighting more on overshoot and settling time. For instance, after fine tuning, the step response of system with the PID controller is shown in Fig. 3.19.

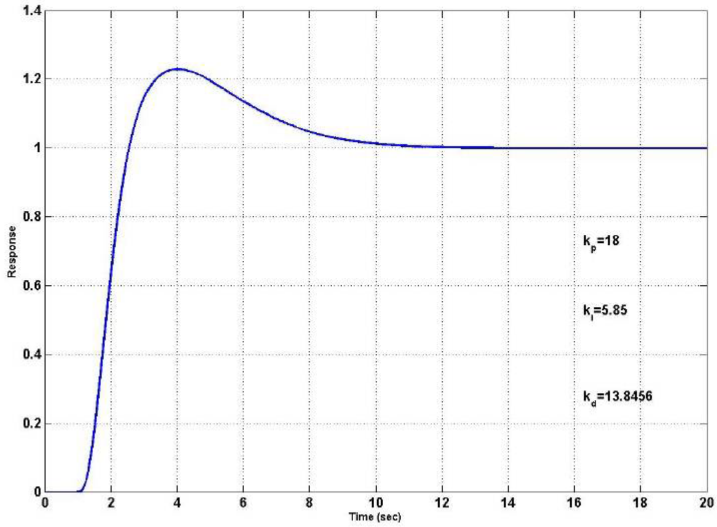


Figure 3.19: Step response with PID control using fine tuned gains.

Chapter 4

Frequency Domain Design

The frequency-response methods for control-system design were developed from the original work of Bode (1945) on feedback-amplifier techniques. Their attractiveness for design depends on several ideas.

1. The gain and phase curves can be easily plotted by hand.

2. Nyquist's stability criterion can be applied, and dynamic response specifications can be readily interpreted in terms of gain and phase margins, which are easily seen on the plot of log gain and phase versus log frequency.

3. The system error constants, i.e. the position error constant introduced in section 2.4.2, can be read directly from the low frequency asymptote of the gain plot.

4. The corrections to the gain and phase curves (and thus the gain and phase margins) introduced by a trial pole or zero of a compensator can be quickly and easily computed, using the gain curve alone.

5. The effect of pole, zero, or gain changes of a compensator on the speed of response can be quickly and easily determined using the gain curve alone.

We will briefly review these points here as they apply to continuous systems. Then we will discuss how to apply frequency domain design methods to discrete systems.

4.1 What is Frequency Response?

By the term *frequency response*, we mean the steady-state response of a system to a sinusoidal input. In frequency-response methods, we vary the frequency of the input signal over a certain range and study the resulting response. One advantage of the frequency-response approach is that we can use the data obtained from experimental data on the physical system without deriving its mathematical model.
It can be shown that for stable, linear, time-invariant system with the transfer function $H(s)$, when the input is a sinusoid, the output is also a sinusoid with the same frequency, but with different amplitude and phase. More specifically, let the input signal be

$$x(t) = X\sin(\omega t)$$

the steady-state response is

$$y_{ss}(t) = Y\sin(\omega t + \phi)$$

where the magnitude $Y = X|H(j\omega)|$, and the phase shift $\phi = \angle H(j\omega)$. Hence the function $H(j\omega)$ completely determines the steady state response to sinusoidal input,

- $|H(j\omega)|$ is the amplitude ratio of the output sinusoid to the input sinusoid.
- $\angle H(j\omega)$ is the phase shift of the output sinusoid with respect to the input sinusoid.

A positive phase angle is called phase lead, and a negative phase angle is called phase lag. A network that has phase-lead characteristics is called a lead network, while a network that has phase-lag characteristics is called a lag network.
The function $H(j\omega)$ is called the sinusoidal transfer function, which is represented by the magnitude and phase angle with frequency as a parameter. There are two commonly used representations of frequency response:

1. Bode Diagram
2. Nyquist plot or polar plot

We shall briefly discuss these representations in the following sections.

4.2 Bode Diagrams

A Bode diagram consists of two graphs: one is a plot of the logarithm of the magnitude of a sinusoidal transfer function $H(j\omega)$; the other is a plot of the phase angle; both are plotted against the frequency on a logarithmic scale.

The standard representation of the logarithmic magnitude of $H(j\omega)$ is $20log|H(j\omega)|$, where the base of the logarithm is 10. The unit used in this representation is decibel, usually abbreviated dB. The main advantage of using logarithm in Bode diagram is that multiplication of magnitudes can be converted into addition. Furthermore, a simple method for sketching an approximate log-magnitude curve is available. It is based upon asymptotic approximations. Such approximation by straight-line asymptotes is sufficient if only rough information on the frequency-response characteristics is needed. Expanding the low frequency range by use of a logarithmic scale is also highly advantageous since characteristics at low frequencies are most important in practical systems.
Since transfer function can be represented as multiplication of basic factors like gains, first and second-order systems, we only need to be familiar with the logarithmic plots of these basic factors. Then it is possible to utilize them in contructing a composite logarithmic plot for any general form of transfer functions by sketching the curves for each factor and adding individual curves graphically, because adding the logarithms of the gains corresponds to multiplying them together. The Bode diagrams for typical first and second order systems are shown in Figures 4.1 and 4.2.
Consider a standard unity feedback system shown in Fig. 4.3, which has a closed-loop transfer function $H(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$. Bode diagram is obtained with s replaced by $j\omega$.

If the open-loop transfer function $G(s)C(s)$ is stable, then its frequency responses (or Bode diagram), $G(j\omega)C(j\omega)$, can be easily used to determine the stability of the closed-loop system $H(s)$. Therefore, it is often to plot the Bode diagram for the open-loop system $G(s)C(s)$, with magnitude response $|G(j\omega)C(j\omega))|$, and phase response $\angle G(j\omega)C(j\omega)$.

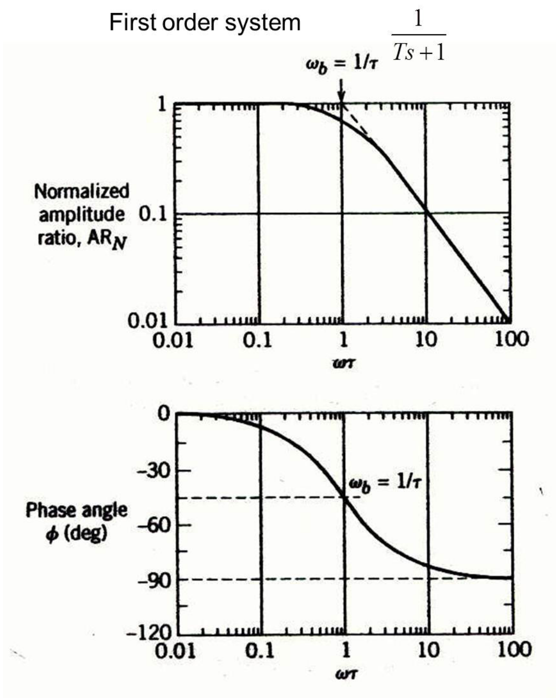


Figure 4.1: Bode diagram for first order system.

Example 4.1: Bode Diagram of the Open-loop System in the Speed Control System Example

The open loop transfer function of the speed control system is

$$G(s)C(s) = \frac{1050s + 670}{1000s^2 + 100s}$$

The corresponding Bode Diagram for the open-loop system is shown in Fig. 4.4.

4.3 Nyquist Plot

The second way to represent the frequency response is the Nyquist plot, i.e. polar plot. Nyquist plot is to draw the frequency response of the open-loop system in a single complex plane instead of separating the magnitude and phase responses into two individual diagrams as in the Bode plot. An example of such as plot is shown in Fig. 4.5. Each point on the polar plot of a sinusoidal transfer function $G(j\omega)$ represents the terminal point of a

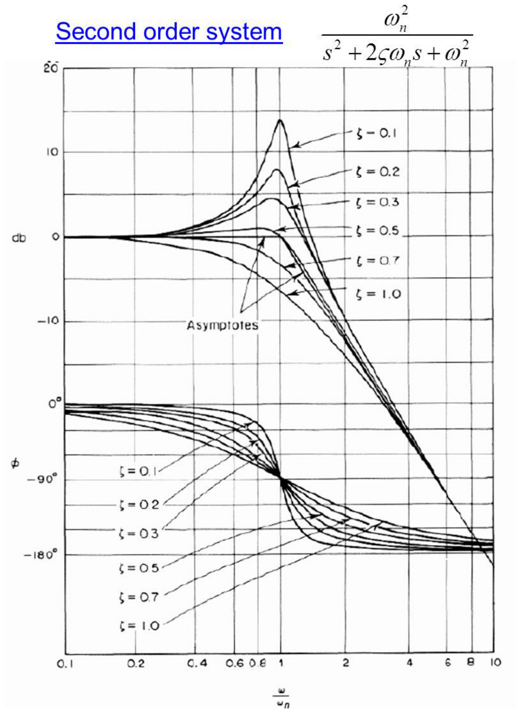


Figure 4.2: Bode diagram for second order system.

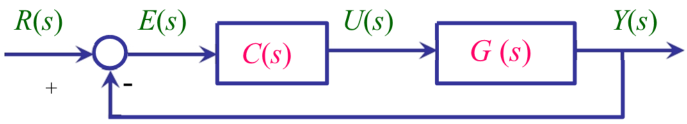


Figure 4.3: Unity Feedback Control System

vector at a particular value of ω. In the polar plot it is important to show the frequency graduation of the locus. The projections of $G(j\omega)$ on the real and imaginary axes are its real and imaginary components. An advantage in using a polar plot is that it depicts the frequency-response characteristics of a system over the entire frequency range in a single plot. One disadvantage is that the plot does not indicate the contributions of each individual factor of the open-loop transfer function.

The major advantage of polar plot is that the stability of the closed-loop system can be easily determined by the polar plot of the open-loop system from the following Nyquist Stability Criterion.

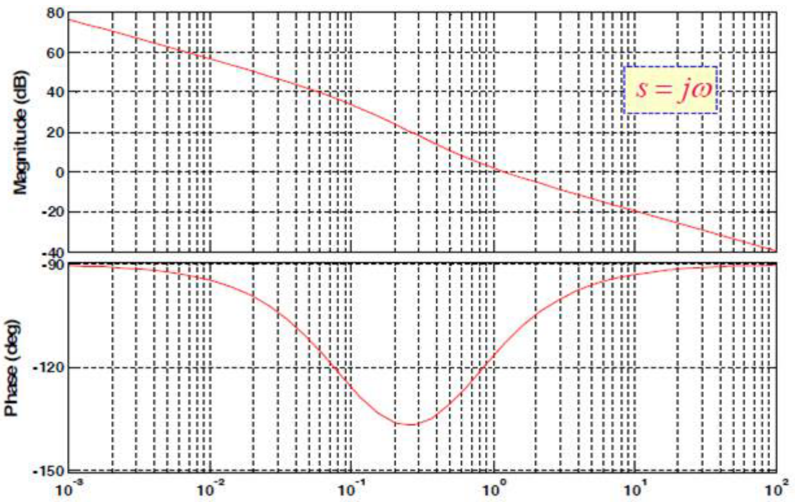


Figure 4.4: Bode Diagram for the open-loop system of the speed control system

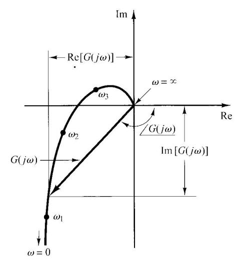


Figure 4.5: An example of polar plot

Nyquist Stability Criterion

Let N be the number of clockwise encirclements of the point $(-1, 0)$ in the Nyquist plot, P be the number of unstable poles of the open-Loop system $G(s)C(s)$. Then the number of unstable poles of the closed-loop system $H(s)$, say Z, is given by Z = P+ N.
If $G(s)C(s)$ is stable, P = 0 , then N has to be zero in order to guarantee the stability of the closed-loop system $H(s)$.

As mentioned previously, frequency response is only meaningful for stable system. If the open-loop system is stable, then there should be no encirclement around the critical point $(-1, 0)$ to assure that the closed-loop system is stable. If the polar plot is close to the critical point $(-1, 0)$, then the closed-loop system may become unstable due to small changes in the parameters. In some sense, we say that the system is less stable. This is the subject of relative stability.

4.4 Relative Stability

In designing a control system, we require that the system be stable. Furthermore, it is necessary that the system has adequate relative stability.

To illustrate the basic concept of relative stability, consider the proportional control system with the open-loop transfer function of $KG(s)$. Fig. 4.6 shows the Nyquist plot (polar plot) for three different values of the open loop gain K.

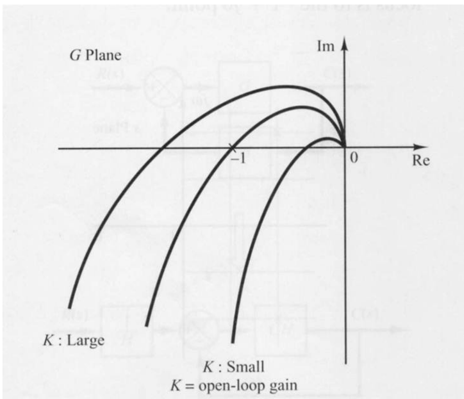


Figure 4.6: Polar plots with different gains

For a large value of the gain K, the system is unstable. As the gain is decreased to a certain value, the plot passes through the critical point (-1,0). This means that with this gain value the system is on the verge of instability, and the system will exhibit sustained oscillations. For a small value of the gain K, the system is stable.

In general, the closer the Nyquist plot to the point (-1,0), the more oscillatory is the system response. Therefore, the closeness of the plot to the point (-1,0) can be used as the measure of the margin of stability. Then the natural question is: How to measure the closeness of the Nyquist plot to the critical point (-1,0)?

4.5 Gain and Phase Margins

It is common practice to represent the closeness in terms of phase margin and gain margin.
Phase margin: The phase margin is that amount of additional phase lag at the gain crossover frequency required to bring the system to the verge of instability. The gain crossover frequency is the frequency as which $|G(j\omega)C(j\omega)|$, the magnitude of the open-loop transfer function, is unity. The phase margin is 180^o plus the phase angle of the open-loop transfer function at the gain crossover frequency.

Gain margin: The gain margin is the reciprocal of the magnitude $|G(j\omega)C(j\omega)|$ at the frequency at which the phase angle is -180^o.
The phase and gain margins of a control system are a measure of the closeness of the Nyquist plot to the (-1,0) point. Therefore, these margins may be used as design criteria.
It should be noted that either the gain margin alone or the phase margin alone does not give a sufficient indication of the relative stability. Both should be given in the determination of relative stability.

4.5.1 Gain and Phase Margins in the Nyquist Plot

Assuming the open-loop system is stable, the gain margin and phase margin can be found from the Nyquist plot by zooming in the region in the neighborhood of the critical point (-1,0) as shown in Fig. 4.7.
Gain margin is the maximum additional gain that can be applied to the closed-loop system such that it remains stable. Similarly, phase margin is the maximum phase lag that the closed-loop system can tolerate without losing the stability.

Mathematically, the gain margin is

$$GM = \frac{1}{|G(j\omega_p)C(j\omega_p)|}$$

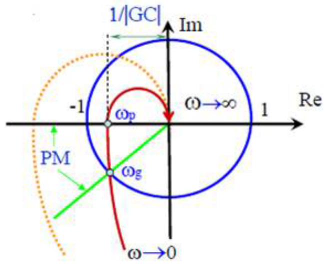


Figure 4.7: Stability margins in Nyquist plots

where the phase crossover frequency, ω_p, is the frequency at which the phase angle is -180^o, i.e.

$$\angle G(j\omega_p)C(j\omega_p) = -180^o$$

The phase margin is

$$PM = 180^o + \angle G(j\omega_g)C(j\omega_g)$$

where the gain crossover frequency, ω_g, is the frequency at which the magnitude is one, i.e.

$$|G(j\omega_g)C(j\omega_g)| = 1$$

4.5.2 Gain and Phase Margins in the Bode Diagram

The gain and phase margins can also be easily obtained by Bode diagram. We first identify the two crossover frequencies and then mark out the margins as shown in Fig. 4.8.
Proper phase and gain margins ensure us against variations in the system components. For satisfactory performance, the phase margin should be between 30^o and 60^o and the gain margin should be greater than 6dB.
The major roles of Bode Diagram in control system design are summarized below:

1. From open-loop response, obtain closed-loop stability condition from PM and GM.

2. From PM and GM, tell how stable the closed-loop system is.

3. Obtain the frequency domain response shown by Bode diagram.

4. The low-frequency region (the region far below the gain crossover frequency) indicates the steady-state behavior of the closed-loop system. The higher DC gain, the better the steady state performance.

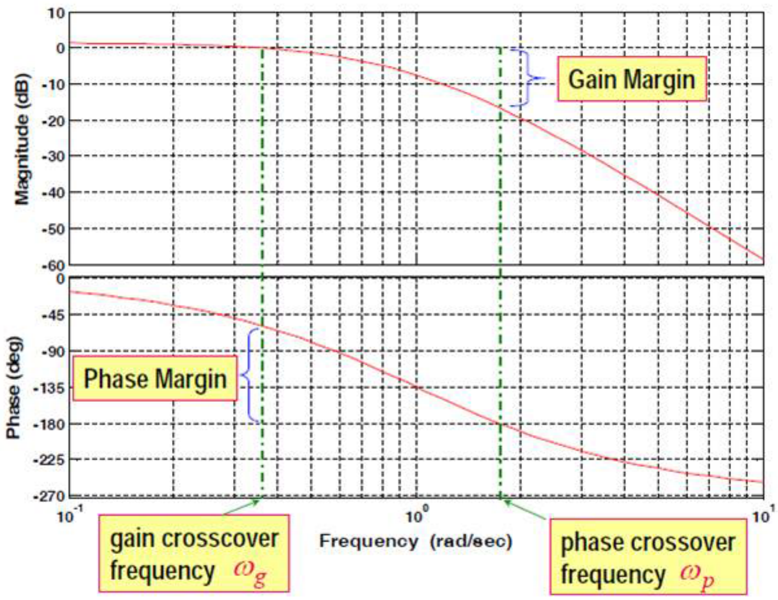


Figure 4.8: Stability margins in Bode diagram

5. The medium-frequency region (the region near the (-1,0) point) indicates the relative stability.

6. The high-frequency region indicates the complexity of the system.

4.6 Frequency Response and Bode Diagram for Discrete Systems

If a sinusoid at frequency ω, $e^{j\omega t}$, is applied to a stable, linear time-invariant continuous system described by the transfer function $H(s)$, it can be shown that the response is a transient (which decreases to zero quickly) plus a sinusoidal steady state response, $H(j\omega)e^{j\omega t}$, as shown in Fig. 4.9.

$$e^{j\omega t} \longrightarrow \boxed{H(s)} \longrightarrow H(j\omega)e^{j\omega t}$$

Figure 4.9: Block Diagram of Frequency Response for Continuous System.

We can say almost exactly the same respecting the frequency response of a stable, linear time-invariant discrete system. If the system has a transfer function $H(z)$, then its frequency response is $H(e^{j\omega T})$, where T is the sampling period. If a sinusoid, $e^{j\omega Tk}$, is applied to the system, $H(z)$, then in the steady state, the response can be represented as $H(e^{j\omega T})e^{j\omega Tk}$ as shown in Fig. 4.10.

$$e^{j\omega Tk} \longrightarrow \boxed{H(z)} \longrightarrow H\left(e^{j\omega T}\right)e^{j\omega Tk}$$

Figure 4.10: Block Diagram of Frequency Response for Discrete System.

The basic concepts of frequency response are the same as for continuous systems, but the evaluation of the magnitude and phase of a transfer function, $H(z)$, is accomplished by letting z taking on values around the unit circle, $z = e^{j\omega T}$, that is,

$$magnitude = |H(z)|_{z=e^{j\omega T}}$$

$$phase = \angle H(z)_{z=e^{j\omega T}}$$

These relationships make useless the hand-plotting procedures developed by Bode and his proof relating the phase to the magnitude curve on a log-log plot. The inability to use these ideas detracts from the ease with which a designer can predict the effect of pole and zero changes on the frequency response. Therefore, frequency domain design is rarely used in z-domain directly.

For discrete systems represented in the z-plane, the hand-plotting rules do not apply because z takes on values around the unit circle instead of the imaginary axis evaluation of s that is the basis for the hand-plotting rules. The lack of hand-plotting capability has little impact if computer-based tools are available to perform the task. However, it is important for the designer to retain the ability to perform hand plotting of continuous-system frequency response because the general nature of the curves is similar to the discrete case, and the intuition gained can be used as a check. In fact, for fast sampling, the curves are virtually identical. As the sample rate slows to four times the frequency of interest, the phase curve departs from that of an equivalent continuous system.

Example 4.2. Bode Diagram for Discrete System

To illustrate the ideas above, Fig. 4.11 shows the magnitude and phase of

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

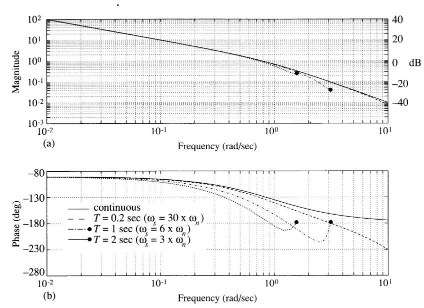


Figure 4.11: Frequency Response for Discrete System.

for s taking on values from $0 \leq \omega < \infty$. Also shown are the magnitude and phase of its ZOH discrete equivalent at three sample rates, that is

$$H(z) = 0.0187 \frac{(z + 0.9355)}{(z-1)(z-0.8187)}, \quad T = 0.2s$$

$$H(z) = 0.368 \frac{(z + 0.718)}{(z-1)(z-0.368)}, \quad T = 1s$$

$$H(z) = 1.135 \frac{(z + 0.523)}{(z-1)(z-0.135)}, \quad T = 2s$$

with z taking on values at $z = e^{j\omega T}$ for $0 \leq \omega T \leq \pi$.
The H(s) magnitude curve could be approximated by two straight line asymptotes intersecting at the corner frequency of $\omega = 1$. No analogous methods exist for plotting the $G(z)$ curves; they need to be done by computer (using command bode in MATLAB). As should be expected, the discrete and continuous curves are very similar in nature because they all represent the response of the same physical process but at different sample rates. The fastest sampling case is extremely close to the continuous one, whereas the slower sampling cases progressively degrade from it. Note from this example that the primary effect of sampling is to cause an additional phase lag, whereas the amplitude response is affected very little.
The Gain Margin and Phase Margin are defined in the same way as that for continuous system.

- The Gain Margin (GM) is the inverse of the amplitude of $H(z)$ when its phase is 180^o and is a measure of how much the gain of the system can increase before instability results.

- The Phase Margin (PM) is the difference between 180^o and the phase of $H(z)$ when its amplitude is 1. It is a measure of how much phase lag or time delay can be tolerated before instability results.

The GM and PM can be easily determined using the Bode diagram of the open-loop system.

Example 4.3. Stability Margins for Discrete System

Consider the speed control system with the open loop transfer function

$$G(s)C(s) = \frac{1050s + 670}{1000s^2 + 100s}$$

the ZOH discrete equivalent with sampling period $T = 0.6s$ is

$$G(z)C(z) = \frac{0.73z - 0.496}{z^2 - 1.942z + 0.942}.$$

The GM and PM can be easily determined in the same fashion as that for continuous case, as shown in Fig. 4.12. The gain margin is 10 dB, and the phase margin is 47^o.

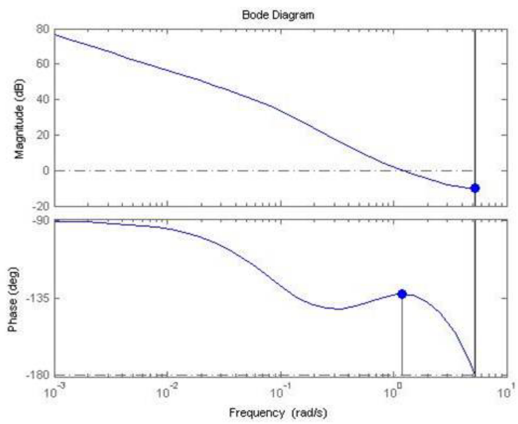


Figure 4.12: Stability margins of the digital speed control system

4.7 Frequency Domain Design Specifications

It is important to note that transient-response performance (overshoot, rise time and settling time etc) is usually most important for control system design. The time domain design specifications can be translated into the requirement on phase margin and some other frequency domain properties. For continuous systems, it is often noted that the phase margin is related to the damping ratio, ζ, for a second-order system by the approximate relation, $\zeta \approx \frac{PM}{100}$. This relationship is examined in Fig. 4.13 for the continuous case and for discrete systems with two values of the sample rate. For second order systems without zeros, the relationship between ζ and PM , i.e. $\zeta \approx \frac{PM}{100}$, is equally valid for continuous and discrete systems. For higher order systems, the damping of the individual modes needs to be determined using other methods. For linear, time-invariant, higher-order systems having a domi-

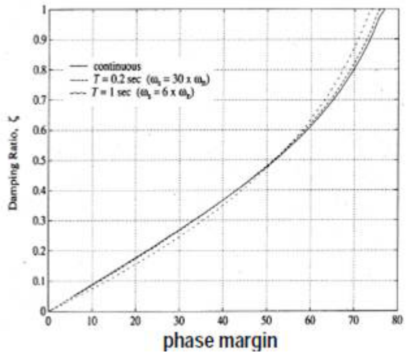


Figure 4.13: Damping ratio of a second-order system versus phase margin (PM)

nant pair of complex conjugate closed-loop poles, the following relationships generally exist between the step transient response and frequency response:

1. The value of resonant peak magnitude, M_r, is indicative of the relative stability. Satisfactory transient performance is usually obtained if the value of M_r is in the range $1.0 < M_r < 1.4(0dB < M_r < 3dB)$, which corresponds to an effective damping ratio of $0.4 < \zeta < 0.7$. In general, a large value of M_r corresponds to a large overshoot in the step transient response.

2. The magnitude of the resonant frequency ω_r is indicative of the speed of the transient response. The larger the value of ω_r, the faster the time response is. In terms of the open-loop frequency response, the damped natural frequency in the transient response is somewhere between the gain crossover frequency and phase crossover frequency.

3. The resonant peak frequency $\omega_r(\omega_r = \omega_n\sqrt{1-2\zeta^2})$ and the damped natural frequency $\omega_d(\omega_d = \omega_n\sqrt{1-\zeta^2})$ are very close to each other for the lightly damped systems.

The three relationships just listed are useful for correlating the step transient response with the frequency response of higher-order systems, provided that they can be approximated by the standard second-order system or a pair of complex-conjugate closed-loop poles. If a higher-order system satisfies this condition, a set of time-domain specifications may be translated into frequency-domain specifications. This simplifies greatly the design work or compensation work of higher-order systems.

In the frequency-response approach, we specify the transient-response performance in an indirect manner. That is, the transient-response performance is specified in terms of the phase margin , gain margin, resonant peak magnitude (they give a rough estimate of the system damping); the gain crossover frequency, resonant frequency, bandwidth (they give a rough estimate of the speed of transient response); and static error constants (they give the steady-state accuracy). Although the correlation between the transient response and frequency response is indirect, the frequency domain specifications can be conveniently met in the Bode diagram approach.

After the open loop has been designed by the frequency-response method, the closed-loop poles and zeros can be determined. The transient-response characteristics must be checked to see whether the designed system satisfies the requirements in the time domain. If it does not, then the compensator must be modified and the analysis repeated until a satisfactory result is obtained.

Design in the frequency domain is simple and straightforward. The Bode diagram indicates clearly the manner in which the system should be modified, although the exact quantitative prediction of the transient-response characteristics cannot be made. Although the correlation between the transient response and frequency response is indirect, the frequency domain specifications can be conveniently met in the Bode diagram approach.

The frequency-response approach can be applied to systems whose dynamic characteristics are given in the form of experimental frequency-response data. Therefore it can be model-free in some sense. Note also that in dealing with high-frequency noises we find that the frequency-response approach is more convenient than other approaches.

In many applications, compensation is essentially a compromise between steady-state accuracy and relative stability. To have a high value of the position or velocity error constants and yet satisfactory relative stability, it is necessary to reshape the open-loop frequency-response curve. For satisfactory performance, the phase margin should be between 30^o and 60^o and the gain margin should be greater than 6dB. The gain in the low frequency region, K, should be large enough to assure good steady state accuracy, since the steady state error is given as $1/(1+K)$. For the high-frequency region, the gain should be attenuated as rapidly as possible to minimize the effects of noise.

4.8 Frequency Domain Design Methods

There are basically two approaches in the frequency-domain design. One is the polar plot approach and the other is the Bode diagram approach. When a compensator is added, the polar plot does not retain the original shape, and therefore, we need to draw a new polar plot, which will take time and is thus inconvenient. On the other hand, a Bode diagram of the compensator can be simply added to the original Bode diagram, and thus plotting the complete Bode diagram is a simple matter. Also, if the open-loop gain is varied, the magnitude curve is simply shifted up or down without changing the shape of the curve, and the phase curve remains the same. For design purpose, therefore, it is best to work with the Bode diagram.

A common approach to the frequency domain design is that we first adjust the open-loop gain so that the requirement on the steady-state accuracy is met. Then the magnitude and phase curves of the uncompensated open loop (with the open-loop gain just adjusted) is plotted. If the specifications on the phase margin and gain margin are not satisfied, then a suitable compensator that will re-shape the open-loop transfer function is determined. Finally, if there are any other requirements to be met, we try to satisfy them, unless some of them are mutually contradictory.

There are three commonly used frequency domain design methods, namely

1. Lead Compensator, which can be viewed as filtered PD control,

2. Lag Compensator, which can be viewed as modified PI control,

3. Lag-Lead Compensator

We shall start with the most commonly used one: lead compensator.

4.9 Lead Compensator

A PD controller is $C(s) = K_p + K_d s$. Adding a first order low pass filter, $\frac{1}{cs+1}$, leads to the following controller,

$$C(s) = \frac{K_p + K_d s}{cs + 1} = k\frac{\tau s + 1}{\alpha\tau s + 1} \tag{4.1}$$

where $k = K_p$, $\tau = \frac{K_d}{K_p}$, and $\alpha = \frac{K_p c}{K_d}$. This controller becomes a lead compensator if the attenuation factor, $\alpha < 1$. Note that PD controller stabilize the closed-loop. Therefore we can expect a similar role from a lead compensator.

What is the main difference between a PD controller and a Lead Compensator?
PD controller amplifies non-smooth signal, whereas lead compensator will produce a smoother signal, which can be easily seen in the frequency domain.

4.9.1 The Frequency Response of PD Controller

The frequency response of the PD controller is

$$C(j\omega) = K_p + K_d j\omega = K_p(\tau j\omega + 1)$$

The Bode diagram of a typical PD controller $(Ts + 1)$ (assuming the proportional gain $K_p = 1$) is shown in Fig. 4.14. It is obvious that the PD controller will amplify the high frequency noise! This is the main reason that PD controller alone is rarely used in practice. We need to add a low pass filter to filer out the noise! This is the motivation for the lead-lag compensators.

4.9.2 The Frequency Response of Lead Compensator

The frequency response of the lead compensator is

$$C(j\omega) = k\frac{\tau j\omega + 1}{\alpha\tau j\omega + 1}, \alpha < 1$$

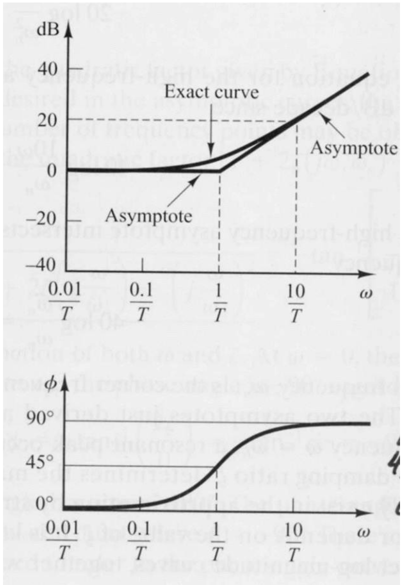


Figure 4.14: Frequency response of a typical PD controller $Ts + 1$

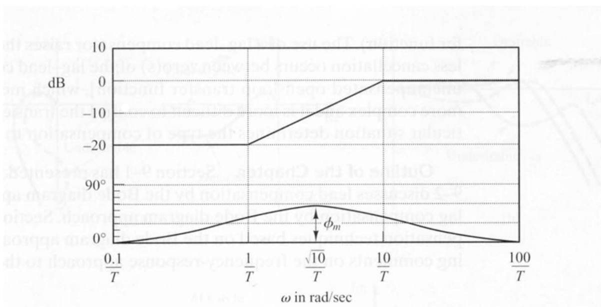


Figure 4.15: Bode Diagram of a lead compensator $\alpha(Tj\omega + 1)/(\alpha Tj\omega + 1)$ where $\alpha = 0.1$.

Let's try to use the asymptotes to draw the Bode diagram. The first step is to identify the corner frequencies by putting the above sinusoidal transfer function in the following form,

$$C(j\omega) = k\frac{\tau j\omega + 1}{\alpha\tau j\omega + 1} = k\frac{\frac{j\omega}{1/\tau} + 1}{\frac{j\omega}{1/\alpha\tau} + 1}, \alpha < 1.$$

Then we can obtain the following two corner frequencies,

$$\omega_1 = \frac{1}{\tau}, \quad \omega_2 = \frac{1}{\alpha\tau}, \quad \alpha < 1.$$

Because $\alpha < 1$, we have $\omega_1 < \omega_2$. The first corner frequency, ω_1, corresponds to a zero, and the second one, ω_2, corresponds to a pole. Then we can apply the following rule to draw the asymptotes:

- If the corner frequency is associated with a zero, the asymptote has positive slope.
- If the corner frequency is associated with a pole, the asymptote has negative slope.

The Bode diagram is approximated using the asymptotes as shown in Fig. 4.15, which is actually very close to the real one as shown in Fig. 4.16. As can be seen from the frequency response plot, the lead compensator

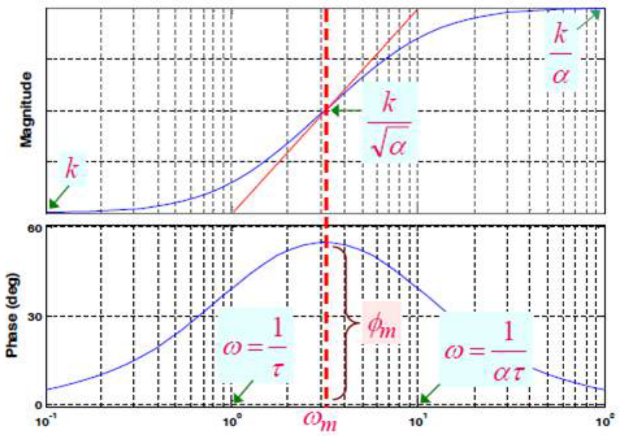


Figure 4.16: Bode Diagram of a lead compensator $k(\tau j\omega + 1)/(\alpha\tau j\omega + 1)$.

is basically a high-pass filter. (The high frequencies are passed, but low frequencies are attenuated.) The maximum phase lead angle, ϕ_m, is related to the attenuation factor, α, by

$$\sin\phi_m = \frac{1-\alpha}{1+\alpha} \tag{4.2}$$

and

$$\alpha = \frac{1-\sin\phi_m}{1+\sin\phi_m} \tag{4.3}$$

while the frequency at which the maximum phase occurs, ω_m, is related to both α and τ as

$$\omega_m = \frac{1}{\tau\sqrt{\alpha}} \tag{4.4}$$

$$\tau = \frac{1}{\omega_m\sqrt{\alpha}} \tag{4.5}$$

The magnitude of the lead compensator at the frequency, ω_m, is related to the attenuation factor, α, as

$$|C(j\omega)|_{\omega=\omega_m} = \frac{k}{\sqrt{\alpha}} \tag{4.6}$$

From the Bode Diagram 4.16, it is obvious that the lead compensator lifts up both the magnitude and the phase curves. Therefore the primary function of the lead compensator is to reshape the frequency-response curve to provide sufficient phase-lead angle to offset the excessive phase lag associated with the components of the fixed system.

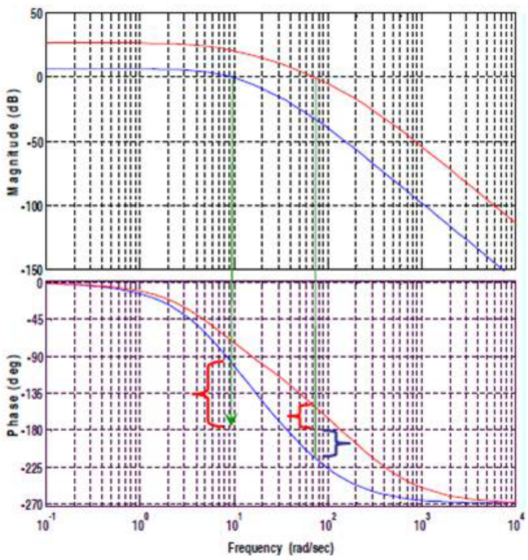


Figure 4.17: Comparison between uncompensated and lead-compensated systems

Let's compare the uncompensated and lead-compensated systems as shown in Fig. 4.17. The uncompensated system has a large steady state error due to small open-loop gain. If we just increase the gain without re-shaping the phase curve, the phase margin would be negative, i.e. the system becomes unstable. with the help of the lead compensator, both the phase and gain are increased such that overall performance is improved, i.e. the compensated system has a smaller steady state error, while retains a sufficient phase margin.

The key idea of lead compensation is

1. enlarge the gain crossover frequency ω_g or increasing the gain k,
2. add additional phase to improve phase margin.

The performance of the system will improve from the following three perspectives:

1. the steady state error is reduced by increasing the gain of the open-loop;
2. the speed of the system is increased due to the increase of the gain crossover frequency (which leads to higher bandwidth);
3. the system is more stable because of higher phase margin.

4.9.3 Lead Compensation Design Procedure

Consider the unity feedback control system shown in Fig. 4.3. Assume that the Bode diagram of the plant, $G(j\omega)$, is given, and the performance specifications are given in terms of phase margin, gain margin, steady state error, and so on. The procedure for designing a lead compensator may be stated as follows:

1. Determine open-loop gain k to satisfy requirements on steady-state performance.
2. Find new open-loop crossover frequency ω_g from $kG(j\omega)$
3. Evaluate the PM of $kG(j\omega)$ at the new crossover frequency ω_g, i.e.
$$\phi = 180^o + \angle G(j\omega_g)$$
4. Determine the necessary phase-lead angle to be added to the system. Allow for some extra angle (5^o to 12^o), because the addition of the

lead compensation shifts the gain crossover frequency to the right and decrease the phase margin further. Therefore, choose the maximum phase of the lead compensator ϕ_m to satisfy

$$\phi_m = \phi_{desired} - \phi + (5^o \rightarrow 12^o)$$

5. Compute the attenuation factor, α, of the lead compensator

$$\alpha = \frac{1 - \sin \phi_m}{1 + \sin \phi_m}$$

6. Determine the frequency where the magnitude of the uncompensated system $kG(j\omega)$ is equal to $-20log(1/\sqrt{\alpha})$. Select this frequency as the new gain crossover frequency, which also corresponds to the frequency for the maximum phase of the lead compensator, ω_m. Therefore, we can compute the parameter τ as

$$\tau = \frac{1}{\omega_m \sqrt{\alpha}}$$

7. Verify the design using MATLAB. Redo if necessary.

It is important to note that

- In practice the desired PM is from 30^o to 60^o. Too low a PM leads to oscillatory or unstable closed-loop response, and too large a PM leads sluggish response.
- Extra PM (5^o to 12^o) is chosen trial and error. Thus this is a trial and error design.
- The open-loop cross over frequency ω_g from $kG(j\omega)$ will be lower than the actual one with $C(j\omega)G(j\omega)$ because $C(j\omega)$ is a high pass filter and will increase bandwidth, i.e. ω_g. Thus the actual design will be done iteratively. That is the reason why Step 7 is needed.

Design Example 4.4: Lead Compensator for Continuous System

Consider the system with the open-loop transfer function

$$G(s) = \frac{4}{s(s+2)}$$

It is desired to design a compensator for the system such that the static velocity error constant is 20, the phase margin is at least 50^o, and the gain margin is at least 10 dB.

We shall use a lead compensator of the form

$$G_c(s) = k\frac{\tau s + 1}{\alpha\tau s + 1}$$

The compensated system will have the open-loop transfer function $G_c(s)G(s)$.

Define

$$G_1(s) = kG(s) = \frac{4k}{s(s+2)}$$

The first step in the design is to adjust the gain k to meet the steady-state performance specification. The system is type I system with one integrator, so the closed-loop system will have zero steady-state error for a step input, and a non-zero, finite steady-state error for a ramp input (assuming that the closed-loop system is stable). In this case, the required velocity error constant is given as 20, so we have

$$\beta_v = \lim_{s\to 0} skG(s) = \lim_{s\to 0}\frac{4k}{(s+2)} = 2k = 20$$

which results in $k = 10$.

Fig. 4.18 shows the Bode diagram of $G_1(j\omega)$. From this plot, the phase and

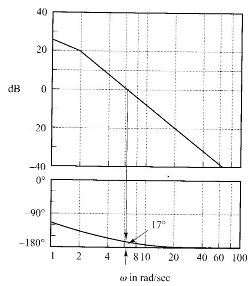


Figure 4.18: Bode Diagram of the uncompensated system $kG(j\omega)$.

gain margins are found to be 17^o and ∞. The specification calls for a phase margin of at least 50^o. We thus find the additional phase lead necessary to satisfy the relative stability requirement is 33^o.

Noting that the addition of a lead compensator modifies the magnitude curve, we realize that the gain crossover frequency will be shifted to the right. We must offset the increased phase lag due to this increase in the gain crossover frequency. In this case, the extra phase is first chosen as 5^o. Thus we have

$$\phi_m = \phi_{desired} - \phi + 5^o = 50^o - 17^o + 5^o = 38^o$$

Then the attenuation factor can be immediately obtained as

$$\alpha = \frac{1 - \sin\phi_m}{1 + \sin\phi_m} = 0.24$$

From Fig. 4.16, we note that the magnitude of the lead compensator at the frequency, ω_m, is $\frac{1}{\sqrt{\alpha}}$. Note that

$$\frac{1}{\sqrt{\alpha}} = \frac{1}{\sqrt{0.24}} = \frac{1}{0.49} = 6.2dB$$

Since at the new gain crossover frequency, i.e. ω_m, the magnitude of the compensated system is 0dB, the corresponding magnitude of the uncompensated system $kG(j\omega)$ should be equal to $-20log(1/\sqrt{\alpha}) = -6.2dB$. From the Bode plot of $kG(j\omega)$, we can find out the corresponding frequency $\omega_m = 9$. The the parameter τ can be computed as

$$\tau = \frac{1}{\omega_m\sqrt{\alpha}} = \frac{1}{4.41} = 0.227$$

Overall, the lead compensator is

$$G_c(s) = k\frac{\tau s + 1}{\alpha\tau s + 1} = 10\frac{0.227s + 1}{0.054s + 1}$$

The magnitude and phase curves for the lead compensator $G_c(j\omega)/10$ are shown in Fig. 4.19(dash-dotted line). The compensated system has the following open-loop transfer function:

$$G_c(s)G(s) = 41.7\frac{s + 4.41}{s + 18.4}\frac{4}{s(s + 2)}$$

The solid curves in Fig. 4.19 show the magnitude curve and phase curve for the compensated system. The lead compensator causes the gain crossover frequency to increase from 6.3 to 9. The increase in this frequency means an increase in bandwidth, which also implies an increase in the speed of response. The phase and gain margins are seen to be approximately 50^o and ∞, respectively. Therefore, the compensated system meets both the steady-state and the relative stability requirements.

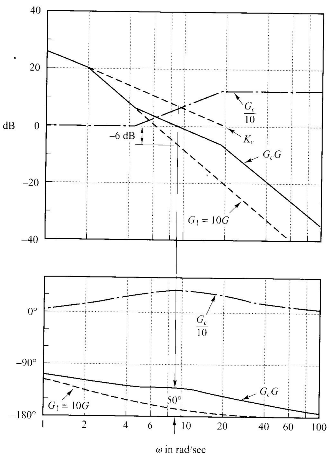


Figure 4.19: Bode Diagram of the compensated system $G_c(j\omega)G(j\omega)$.

4.10 Discrete-time Lead Compensation

So far, everything we talked about the frequency domain design is for continuous time system, what about the digital control system?

The frequency domain design methods for continuous-time systems and sampled discrete-time systems are basically the same. The real plant is continuous in nature. If the continuous-time plant model $G(s)$ is known, then the controller $C(s)$ can be designed first, and then use bilinear transformation to convert it into digital controller. But what if the continuous-time model G(s) is not known, and only the discrete-time model G(z) is given?

For discrete-time systems G(z), the idea is to use a so-called w-transformation (actually it is a bilinear transformation) to transform the discrete-time system into a w-domain system, which is pretty much the same as a continuous-time system.

Everything we have learnt from the compensation design for continuous-time systems can be directly applied to yield a necessary compensator in w-domain, which can be transformed back to a discrete-time version through an inverse w-transformation (an inverse bilinear transformation).

The flow chart of the w-transform method is shown in Fig. 4.20.

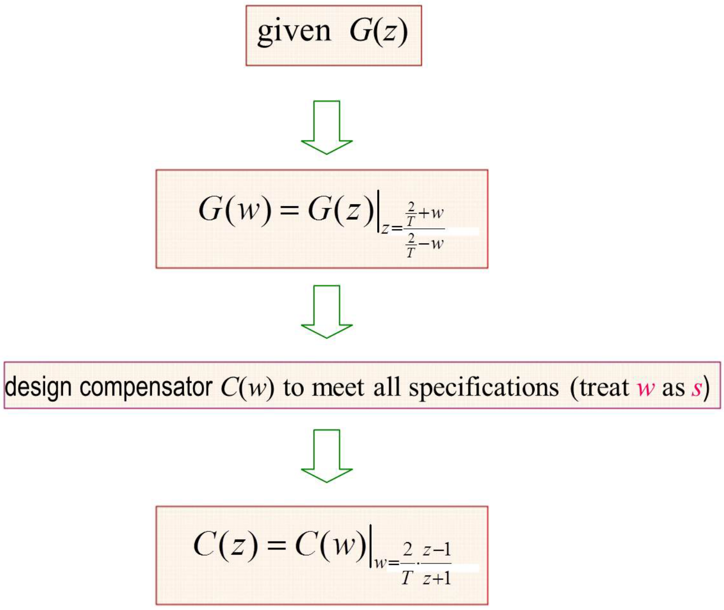


Figure 4.20: Flow Chart of the w-transform method for frequency domain design.

Design example 4.5: Frequency Domain Design for Discrete System

An electric motor can be modeled with a sampling period T = 0.1s as follows:

$$G(z) = \frac{0.0001z + 0.0001}{(z - 0.99)(z - 0.995)}$$

Design a digital control system with a lead compensator such that the resulting system output tracks a step reference, with an overshoot less than 20% and steady state error less than 3% .
Since the continuous TF is not available, we need to use the w-transform to convert the discrete transfer function $G(z)$ into $G(w)$,

$$G(w) = G(z)_{z=\frac{2/T+w}{2/T-w}} = \frac{-0.004w + 0.08}{3.97w^2 + 0.598w + 0.02}$$

From now on, we will just treat G(w) as the continuous TF G(s), and design the analog controller first, and then convert it into digital controller using bilinear transformation.

We shall use a lead compensator of the form

$$C(w) = k\frac{\tau w + 1}{\alpha\tau w + 1}$$

In frequency domain, by replacing w by $j\omega$, the compensator is

$$C(j\omega) = k\frac{\tau j\omega + 1}{\alpha\tau j\omega + 1}$$

The compensated system will have the open-loop transfer function $C(w)G(w)$.

The first step in the design is to adjust the gain k to meet the steady-state performance specification. First check the adequacy of the plant DC gain,

$$DCgain = G(w)|_{w=0} = \frac{0.08}{0.02} = 4$$

Steady state error is then

$$\frac{1}{1 + DCgain} = \frac{1}{1+4} = 0.2$$

Additional DC gain is needed to decrease the steady state error. Let the desired position error constant be β_p, then the steady state error is

$$\frac{1}{1+\beta_p} = 0.03$$

We get $\beta_p = 32$. From the definition of position error constant,

$$\beta_p = \lim_{w\to 0} kG(w) = 4k = 32$$

which results in $k = 8$.

Fig. 4.18 shows the Bode diagram of $kG(j\omega)$. From this plot, the phase margin is 20.5^o. In order to get 20% (or lower) overshoot, the desired damping ratio is around 0.5, which can be found by the relation,

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

Then from the approximate relation between damping ratio and phase margin, $\zeta \approx \frac{PM}{100}$, we can obtain the desired PM to be at least 50^o. We thus find the additional phase lead necessary to satisfy the relative stability requirement is,

$$\phi_m = \phi_{desired} - \phi + 7^o = 50^o - 20.5^o + 7^o = 36.5^o$$

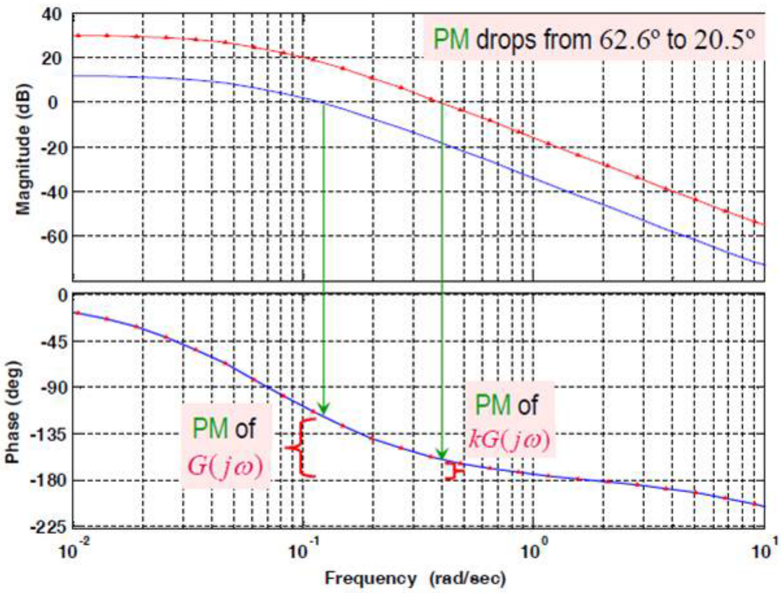


Figure 4.21: Bode Diagram of the uncompensated system $kG(j\omega)$.

where the extra phase is chosen as 7^o. Then the attenuation factor can be immediately obtained as

$$\alpha = \frac{1 - \sin\phi_m}{1 + \sin\phi_m} = 0.254$$

From Fig. 4.16, we note that the magnitude of the lead compensator at the frequency, ω_m, is $\frac{1}{\sqrt{\alpha}}$. Note that

$$\frac{1}{\sqrt{\alpha}} = \frac{1}{\sqrt{0.254}} = \frac{1}{0.504} = 6.0dB$$

Since at the new gain crossover frequency, i.e. ω_m, the magnitude of the compensated system is 0dB, the corresponding magnitude of the uncompensated system $kG(j\omega)$ should be equal to $-20log(1/\sqrt{\alpha}) = -6.0dB$. From the Bode plot of $kG(j\omega)$, we can find out the corresponding frequency $\omega_m = 0.56$. The the parameter τ can be computed as

$$\tau = \frac{1}{\omega_m\sqrt{\alpha}} = 3.54$$

Overall, the lead compensator is

$$C(w) = k\frac{\tau w + 1}{\alpha\tau w + 1} = 8\frac{3.54w + 1}{0.9w + 1}$$

After compensation, PM is improved to 50^o, and bandwidth is also improved, as shown in Fig. 4.22

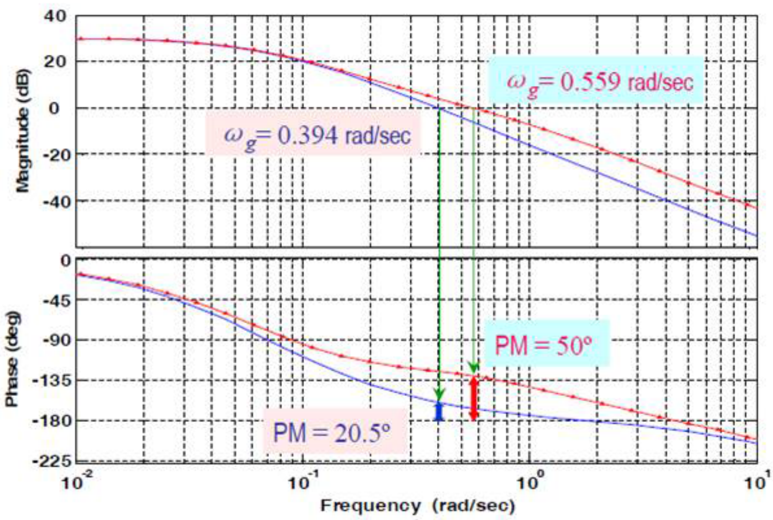


Figure 4.22: Bode Diagram of the compensated system $C(j\omega)G(j\omega)$.

The above compensator can be converted back in z-domain using the inverse bilinear transform,

$$C(z) = C(w)|_{w=\frac{2}{T}\frac{z-1}{z+1}} = 30\frac{z-0.97}{z-0.89}$$

The above is the digital lead compensator. However, nothing is certain without verification. We need to first verify our design in frequency domain, which is shown in Fig. 4.22. More importantly, it should also meet the design specifications in time domain. Fig. 4.23 shows the step response of the closed-loop system with the digital lead compensator. The steady state error is 3.0%, settling time is 14 s, and overshoot is 20%. All the design specifications are met.

4.11 Lag Compensation

Lag compensation has the same type of transfer function as Lead compensator

$$C(j\omega) = k\frac{\tau j\omega + 1}{\alpha\tau j\omega + 1}, \alpha > 1$$

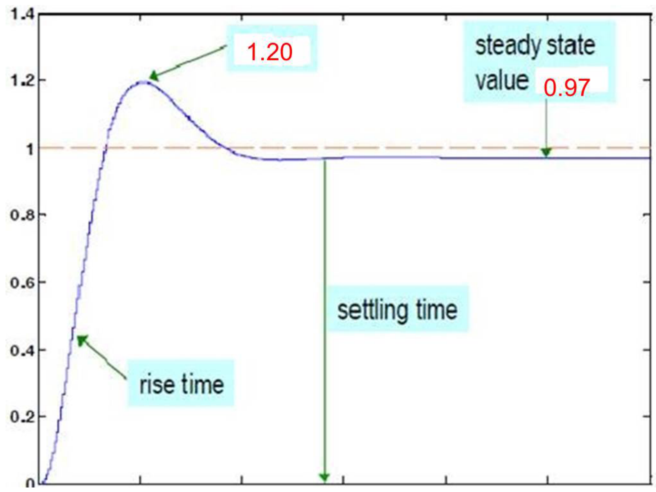


Figure 4.23: Step response of the closed-loop system with the digital lead compensator.

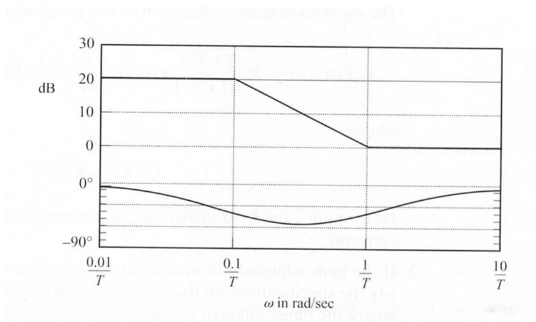


Figure 4.24: Bode Diagram of a lag compensator $\alpha(Tj\omega + 1)/(\alpha Tj\omega + 1)$ where $\alpha = 10$.

The only difference is that $\alpha > 1$, which results in a totally different frequency response.

Let's try to use the asymptotes to draw the Bode diagram. The first step is to identify the corner frequencies by putting the above sinusoidal transfer function in the following form,

$$C(j\omega) = k\frac{\tau j\omega + 1}{\alpha\tau j\omega + 1} = k\frac{\frac{j\omega}{1/\tau} + 1}{\frac{j\omega}{1/\alpha\tau} + 1}, \alpha > 1.$$

Then we can obtain the following two corner frequencies,

$$\omega_1 = \frac{1}{\tau}, \quad \omega_2 = \frac{1}{\alpha\tau}, \quad \alpha > 1.$$

Because $\alpha > 1$, we have $\omega_2 < \omega_1$. The first corner frequency, ω_2, corresponds to a pole, and the second one, ω_1, corresponds to a zero. Then we can apply the following rule to draw the asymptotes:

- If the corner frequency is associated with a zero, the asymptote has positive slope.
- If the corner frequency is associated with a pole, the asymptote has negative slope.

The Bode diagram is approximated using the asymptotes as shown in Fig. 4.24, which is actually very close to the real one as shown in Fig. 4.25. From

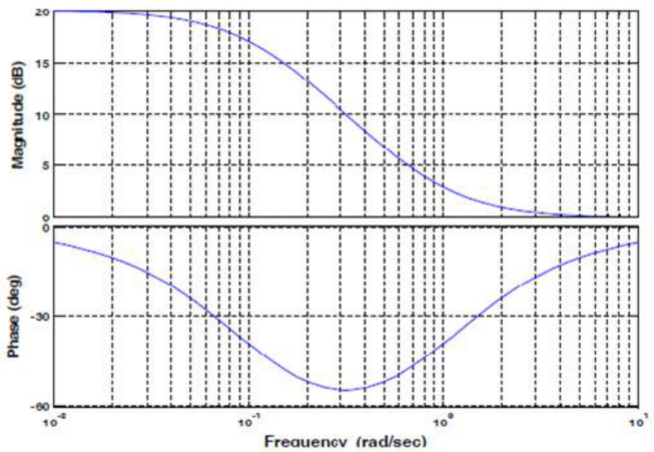


Figure 4.25: Bode Diagram of a lag compensator $k(\tau j\omega + 1)/(\alpha\tau j\omega + 1)$.

the frequency response, we can see that lag compensator is essentially a low pass filter. Therefore it can permit high gain at low frequencies and reduces gain in higher critical range of frequencies so as to improve the phase margin.

When do we need this? Look at the example shown in Fig. 4.26. The lag compensator reduces the gain crossover frequency to where the phase margin is required, while it can retain the high gain at low frequencies. It mainly relies on the property of reduces gain in higher critical range of frequencies (low pass) while the phase-lag is of no use for compensation.

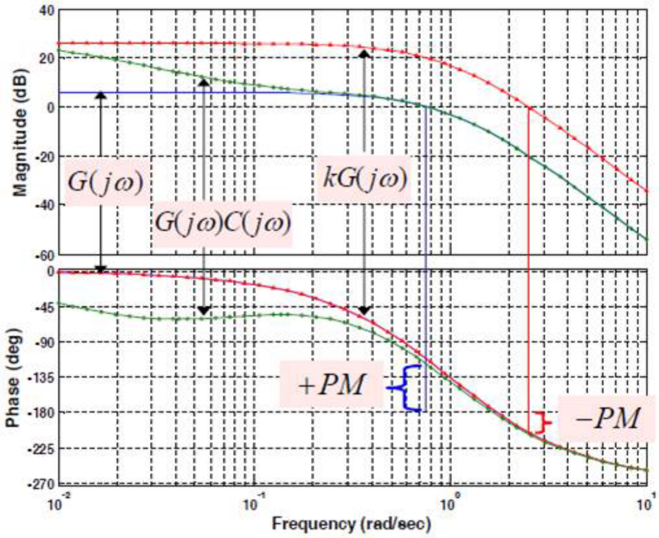


Figure 4.26: Bode diagram of a system with and without lag compensation.

4.12 Lag-Lead Compensation

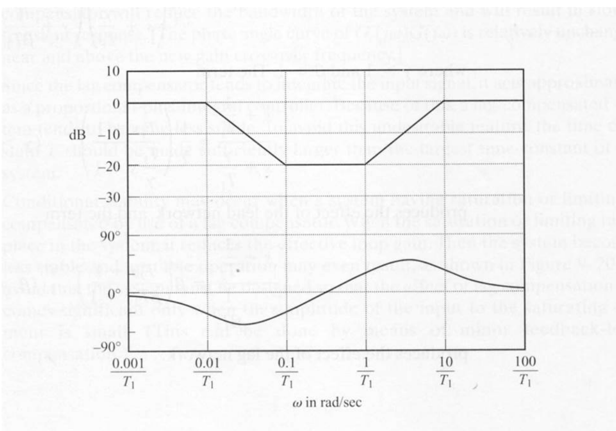


Figure 4.27: Bode diagram of lag-lead compensator.

Note that the lead compensation is the approximation of PD control and the lag compensation approximates PI control. Lag-Lead compensation is nothing more than the combination of lead compensation (PD) and lag compensation (PI), which has the following transfer function,

$$C(s) = k\frac{T_1 s + 1}{\alpha T_1 s + 1}\frac{T_2 s + 1}{\frac{1}{\alpha}T_2 s + 1}, \alpha < 1, T_1 < T_2$$

and frequency responses:

$$C(j\omega) = k\frac{T_1 j\omega + 1}{\alpha T_1 j\omega + 1}\frac{T_2 j\omega + 1}{\frac{1}{\alpha}T_2 j\omega + 1}$$

The Bode diagram is shown in Fig. 4.27.

Comparison of Lead, Lag and Lag-Lead Compensation

1. Lead compensation achieves the desired result through the merits of its phase-lead contribution, where as lag compensation accomplishes the result through the merits of its attenuation property at high frequencies. In some design problems both lag and lead compensation may satisfy the specifications.

2. Lead compensation is commonly used for improving the stability margins. Lead compensation yields a higher gain crossover frequency, which results in larger bandwidth, and hence faster response. If fast response is desired, lead compensation should be employed. If noise signals are present, then a large bandwidth may not be suitable.

3. Lag compensation reduces the system gain at higher frequencies without affecting the gain at lower frequencies. Since the bandwidth is reduced, the system has a slower speed to respond. The total DC gain can be increased, and thereby steady-state accuracy can be improved. Also any high-frequency noises can be attenuated.

4. If both fast responses and good steady-state accuracy are desired, a lag-lead compensator may be employed. By use of the lag-lead compensator, the low-frequency gain can be increased (which means an improvement in steady-state accuracy), while at the same time the system bandwidth and stability margin can be increased.

5. Although a large number of practical compensation tasks can be achieved by the simple compensators, more advanced controllers are needed for complicated systems.

There are cases where a controller can also be viewed as a filter, such as the lead and lag compensators which are high-pass and low pass filters respectively. P control can also be regarded as an all-pass filter. However a controller is in general not a filter, because of several reasons.

A filter is designed to meet frequency domain specifications such as cut-off, bandwidth, low-pass, high-pass, etc., whereas a controller are often designed to meet time-domain specifications such as overshoot, settling time, rise time, etc., in addition to frequency domain requirement.

A filter is designed to shape open-loop characteristics, thus there is no stability problem. The first issue of a feedback controller is the closed-loop stability. A filter is only concerned of the frequency-magnitude property, whereas the frequency-phase property is equally important in the controller design.

On the other hand, frequency domain specifications on magnitude for a filter are more demanding than that for a controller. A digital filter is more concerned of bit rate than a digital controller.

Chapter 5

Pole-Placement Controller

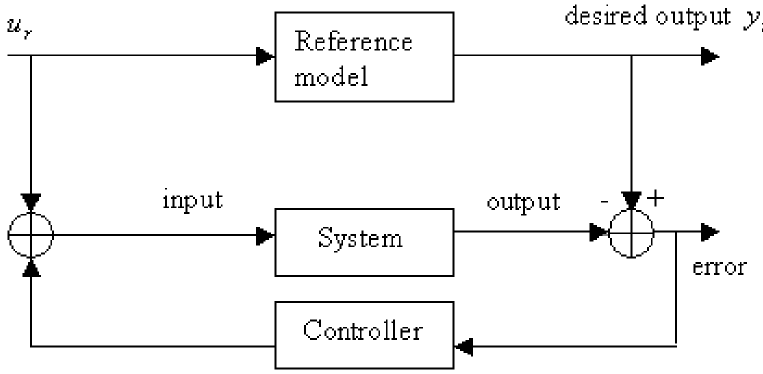


Figure 5.1: Model reference control system

In the previous lectures, we have discussed some simple controller design methods such as PID controller and frequency domain design. Both of them are widely used in industry. One reason is that the exact knowledge on the transfer function is not required to implement these two types of controller. If the mathematical model is completely known, then we have plenty of other model-based approaches.
What is the ultimate goal of the control system? To make the output of the system follow some desired trajectory! But how to specify the desired output? There are normally two ways. The first one is just to specify the desired output arbitrarily without any constraint of the behavior. You can imagine that it might be extremely hard to make the system behave in any way you want in practical problems.
Usually, we have to be more modest and try to make the desired output

more attainable. One way of doing this is to generate the desired output through a well defined reference model to meet various system requirements. And then the next step is to design the controller such that the behavior of the overall system approaches that of the reference model. In other words, we will try to make the transfer function of the closed loop system as close to the reference model as possible. This approach is called model reference control. In practice, perfect matching of these two transfer functions is often impossible. Which task is more important then, matching the zeros or the poles? As discussed in the previous lectures, since stability is of fundamental importance for control system design, which is decided by the location of the poles, everyone would agree that we should try to match the poles first. This is so called the pole placement problem: design the controller to place the poles of the closed loop at the desired positions that are given by the reference model.

5.1 Pole-Placement Problem – the Basic Idea

In many problems, the input-output representation, i.e. the transfer function, is available

$$Y(z) = G(z)U(z) = \frac{B(z)}{A(z)}U(z) \tag{5.1}$$

The poles of the system is decided by the roots of $A(z)$. If the poles of $A(z)$ are not at ideal places, then we may want to place the poles at proper positions, which are specified by the reference model:

$$H_{desired}(z) = \frac{B_m(z)}{A_m(z)}$$

5.1.1 Error-Feedback Control

Let's start with the simple design first — the one degree of freedom controller. The control input will depend upon the error signal only, so called error feedback controller. The block diagram for error-feedback controller is shown in Fig. 5.2.

The pure error feedback control is in the form of

$$U(Z) = \frac{S(z)}{R(z)}E(Z) = \frac{S(z)}{R(z)}(U_c(z) - Y(z)) \tag{5.2}$$

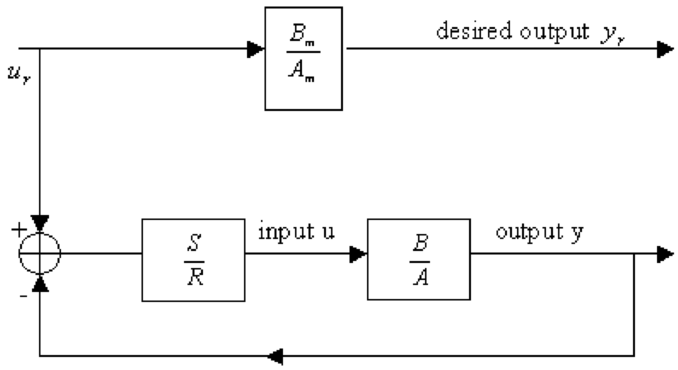


Figure 5.2: Pure Error-Feedback Controller

The closed loop transfer function from the command signal $u_c(k)$ to the output y(k) can be easily obtained as

$$Y(z) = \frac{B(z)S(z)}{A(z)R(z) + B(z)S(z)} U_c(z) \tag{5.3}$$

Suppose we can design the R(z) and S(z) such that $A(z)R(z) + B(z)S(z)$ matches the desired C.P., then there is no room left for us to match the zeros since there is no freedom to manipulate the numerator in above transfer function.
Therefore, error feedback controller is sufficient only for stabilization of the process. In order to match the zeros of the reference model, we have to increase the degree of freedom in the controller.

5.1.2 Two-degree-of-freedom Controller

Practical control systems often have specifications that involve both servo (tracking) and regulation (stabilization) properties. This is traditionally solved using a two-degree-of-freedom structure, as shown in Fig. 5.3. This configuration has the advantage that the servo and regulation problems are separated. The feedback controller $H_{fb} = \frac{S(z)}{R(z)}$, is designed to obtain a closed-loop system that is insensitive to process disturbances, measurement noise, and process uncertainties. The feed-forward compensator $H_{ff} = \frac{T(z)}{R(z)}$ is then designed to obtain the desired servo properties.
The two-degree-of-freedom controller can be designed in following form

$$U(z) = \frac{T(z)}{R(z)} U_c(z) - \frac{S(z)}{R(z)} Y(z) \tag{5.4}$$

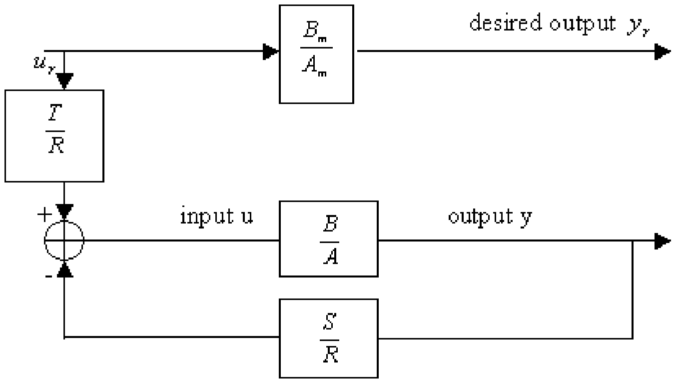


Figure 5.3: Two-Degree-of-Freedom Controller

The block diagram is shown in the Fig. 5.3.

The closed loop transfer function from the command signal $u_c(k)$ to the output y(k) can be readily obtained as

$$Y(z) = \frac{B(z)T(z)}{A(z)R(z) + B(z)S(z)} U_c(z) \tag{5.5}$$

If the desired C.P. is A_{cl}, then let

$$A_{cl}(z) = A(z)R(z) + B(z)S(z) \tag{5.6}$$

Question: Can we solve $R(z)$ and $S(z)$ from above polynomial equation? Yes, under certain conditions.

$$Y(z) = \frac{B(z)T(z)}{A_{cl}(z)} U_c(z)$$

The C.P. for the overall closed-loop system can be decomposed into

$$A_{cl}(z) = A_m(z)A_o(z) \tag{5.7}$$

So

$$Y(z) = \frac{B(z)T(z)}{A_m(z)A_o(z)} U_c(z) \tag{5.8}$$

How to choose $T(z)$?
$T(z)$ can be chosen to satisfy other design requirements.

A simple way to choose $T(z)$ is

$$T(z) = t_o A_o(z) \tag{5.9}$$

Such that

$$Y(z) = \frac{t_o B(z)}{A_m(z)} U_c(z) \tag{5.10}$$

Note that the dynamics of the closed-loop system is low-order now by canceling out the stable poles and zeros.

t_o is chosen to obtain the desired static gain of the system. Note that the static gain of the closed loop is $\frac{t_o B(1)}{A_m(1)}$. For example, to have unit static gain, we simply let

$$t_o = \frac{A_m(1)}{B(1)} \tag{5.11}$$

But you may not be satisfied with above choice of $T(z)$ as the zeros of the closed loop do not match the zeros of the reference model. If you want to achieve perfect match, it is possible under certain conditions which will be discussed later in this chapter.

Keep in mind that there are always multiple ways to control a system, once the poles are properly assigned, $T(z)$ can be chosen to satisfy other design specifications.

Rules to follow:

1. The order of R, S and T should be as low as possible! This is so called ***Occam's razor***, the simpler, the better.

2. The controller has to be causal: the input cannot depend upon the future values of the output.

Example 5.1:

$$R = z + 1, \;\; S = z^2 + 1$$

then we have

$$\begin{aligned}
&Ru = -Sy \Rightarrow \\
&u(k+1) + u(k) = -y(k+2) - y(k)
\end{aligned}$$

What is the problem with such design? Such kind of controllers cannot be implemented, although mathematically it is fine.

Causality condition:

$$DegR \geq DegS \tag{5.12}$$

Example 5.2: Double integrator

$$\begin{aligned} A(z)y &= B(z)u \\ A(z) &= (z-1)^2 \\ B(z) &= \frac{T^2}{2}(z+1) \end{aligned}$$

Let's try the simplest controller, the proportional control,

$$R = 1 \;\; S = s_0$$

So

$$AR + BS = (z-1)^2 + \frac{T^2}{2}s_0(z+1)$$

If the desired C.P. is

$$A_{cl} = z^2 + p_1 z + p_2$$

Can we match the desired C.P.?
Two equations with one parameter:—-mission impossible!

Therefore let's try the first-order controller,

$$\begin{aligned} R &= z + r_1 \\ S &= s_0 z + s_1 \end{aligned}$$

Then

$$\begin{aligned} AR + BS &= (z-1)^2(z+r_1) + \frac{T^2}{2}(z+1)(s_0 z + s_1) \\ &= z^3 + (r_1 + \frac{T^2}{2}s_0 - 2)z^2 + (1 - 2r_1 + \frac{T^2}{2}(s_0+s_1))z + r_1 + s_1\frac{T^2}{2} \end{aligned}$$

If the desired C.P. is

$$A_{cl} = z^3 + p_1 z^2 + p_2 z + p_3$$

Compare the coefficients

$$
\begin{aligned}
& r_1 + \frac{T^2}{2} s_0 - 2 = p_1 \\
& 1 - 2r_1 + \frac{T^2}{2}(s_0 + s_1) = p_2 \\
& r_1 + s_1 \frac{T^2}{2} = p_3
\end{aligned}
$$

We have

$$
\begin{aligned}
r_1 &= \frac{3 + p_1 + p_2 - p_3}{4} \\
s_0 &= \frac{5 + 3p_1 + p_2 - p_3}{2T^2} \\
s_1 &= -\frac{3 + p_1 - p_2 - 3p_3}{2T^2}
\end{aligned}
\tag{5.13}
$$

Next question is then how to choose $T(z)$?

We notice that A_{cl} is of third order, let

$$A_{cl}(z) = A_m(z)A_o(z)$$

Since the third order polynomial always has a real root, we can choose this as $A_o(z)$. Let

$$T(z) = t_o A_o(z)$$

where

$$t_o = \frac{A_m(1)}{B(1)}$$

Remark: The controller must have sufficient degree of freedom. In this example, first order dynamical controller is used instead of the proportional controller. Increasing the order of the controller by one will give an increase of two parameters. But the degree of the desired C.P. is increased by one, and hence the total number of equations for solving the design parameters is increased by one. For instance, if a second order controller is used in the double integrator example, then there may be infinite number of solutions to achieve the same goal.

Now it is time to ask the following question:

Given polynomials $A(z)$ and $B(z)$ whose maximum degree is n, and a third polynomial $A_{cl}(z)$, under what conditions does the solution exist for

$$A(z)R(z) + B(z)S(z) = A_{cl}(z) \tag{5.14}$$

Diophantine equation

First of all, let's see under what condition the solution may not exist?

If $A(z)$ and $B(z)$ have a common factor $(z-c)$, then $A_{cl}(z)$ must contain the same factor $(z-c)$, otherwise, the solution does not exist.

Is this the only condition? Let's examine the simple case of second order system.

$$\begin{aligned}
A(z) &= a_0 z^2 + a_1 z + a_2 \\
B(z) &= b_0 z^2 + b_1 z + b_2 \\
A_{cl}(z) &= p_0 z^3 + p_1 z^2 + p_2 z + p_3
\end{aligned}$$

Design a first order controller

$$\begin{aligned}
R(z) &= r_0 z + r_1 \\
S(z) &= s_0 z + s_1
\end{aligned}$$

From

$$AR + BS = A_{cl}$$

and compare the coefficients, we have

$$\begin{array}{rcc}
z^3: & p_0 & a_0 r_0 + b_0 s_0 \\
z^2: & p_1 & a_1 r_0 + a_0 r_1 + b_1 s_0 + b_0 s_1 \\
z: & p_2 & a_2 r_0 + a_1 r_1 + b_2 s_0 + b_1 s_1 \\
1: & p_3 & a_2 r_1 + b_2 s_1
\end{array}$$

Rewrite it in a compact matrix form, we have

$$\begin{bmatrix} a_0 & 0 & b_0 & 0 \\ a_1 & a_0 & b_1 & b_0 \\ a_2 & a_1 & b_2 & b_1 \\ 0 & a_2 & 0 & b_2 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

Sylvester matrix

If Sylvester matrix is nonsingular, then the solution exists.

What is the condition for nonsingularity of Sylvester matrix?

Sylvester Theorem:

Two polynomials $A(z)$ and $B(z)$ have maximum order of n. They are relatively prime, i.e. they have no common factors, if and only if the corresponding Sylvester matrix M_e is nonsingular, where M_e is defined to be the following $2n \times 2n$ matrix:

$$M_e = \underbrace{\begin{bmatrix} a_0 & 0 & \cdots & 0 & b_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \ddots & \vdots & b_1 & b_0 & \ddots & \vdots \\ \vdots & a_1 & \ddots & 0 & \vdots & b_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & a_0 & \vdots & \vdots & \ddots & b_0 \\ a_n & \vdots & \vdots & a_1 & b_n & \vdots & \vdots & b_1 \\ 0 & \ddots & \vdots & \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_n & 0 & \cdots & 0 & b_n \end{bmatrix}}_{<------n------><------n------>} \tag{5.15}$$

where

$$\begin{aligned} A(z) &= a_0 z^n + a_1 z^{n-1} + \cdots + a_n \\ B(z) &= b_0 z^n + b_1 z^{n-1} + \cdots + b_n \end{aligned} \tag{5.16}$$

The proof is not included here due to time constraint.

Using ***Sylvester Theorem***, we can solve the simplest pole-place problem as follows, which may provide a solid basis for tackling more complicated design problem.

Given a system described by

$$Y(z) = \frac{B(z)}{A(z)} U(z)$$

Let

$$\begin{aligned} &Deg\ \ A = n, \ \ Deg\ \ B \leq n \\ &Deg\ \ A_{cl} = 2n - 1 \end{aligned} \tag{5.17}$$

Design a controller of degree $n - 1$,

$$\begin{aligned} Deg\ \ R &= n - 1 \\ Deg\ \ S &= n - 1 \end{aligned}$$

Then if the $A(z)$ and $B(z)$ have no common factors, $R(z)$ and $S(z)$ can be uniquely determined from the Diophantine equation

$$A(z)R(z) + B(z)S(z) = A_{cl}(z)$$

by

$$\begin{bmatrix} r_0 \\ \vdots \\ r_{n-1} \\ s_0 \\ \vdots \\ s_{n-1} \end{bmatrix} = M_s^{-1} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ \vdots \\ p_{2n-1} \end{bmatrix} \tag{5.18}$$

You may notice that the condition to assure the existence of inverse of the Sylvester matrix, M_s^{-1}, has nothing to do with the degrees of the polynomials. But what about the condition of the degree of the desired polynomial, $A_{cl}(z)$? Why is it required to be 2n-1? Or in other words, the first coefficient, p_0 cannot be zero? What would happen if we let $p_0 = 0$?

If $p_0 = 0$, the solution in above equation (5.18) is still available. But it may not be what you want. We will use the double-integrator to illustrate this point.

If the order of the controller is allowed to be larger than $n - 1$, then the number of solution to the Diophantine equation may be infinity. The extra freedom can be used to satisfy other design specifications.

Example 5.3: Double Integrator

$$\begin{aligned} A(z) &= (z-1)^2 = z^2 - 2z + 1 \\ B(z) &= \frac{T^2}{2}(z+1) \end{aligned}$$

$n = 2$, hence the degree of A_{cl} should be $2n - 1 = 3$,

$$A_{cl} = z^3 + p_1 z^2 + p_2 z + p_3$$

Then

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & \frac{T^2}{2} & 0 \\ 1 & -2 & \frac{T^2}{2} & \frac{T^2}{2} \\ 0 & 1 & 0 & \frac{T^2}{2} \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{5.19}$$

and we can obtain

$$\begin{aligned}
r_0 &= 1 \\
r_1 &= \frac{3 + p_1 + p_2 - p_3}{4} \\
s_0 &= \frac{5 + 3p_1 + p_2 - p_3}{2T^2} \\
s_1 &= -\frac{3 + p_1 - p_2 - 3p_3}{2T^2}
\end{aligned}$$

which is the same answer as those obtained by direct calculation.
As mentioned before, the degree of the desired polynomial, 2n-1=3, is important. Let's see what would happen if the degree of A_{cl} is 2, and hence in the form of

$$A_{cl} = z^2 + p_2 z + p_3 = 0z^3 + z^2 + p_2 z + p_3$$

A simple check of the equation (5.19), by replacing 1 with 0, and $p_1 = 1$ in the right hand would lead to

$$r_0 = 0$$

and the degree of $R(z)$ will be lower than that of $S(z)$, which cannot be implemented in reality (violation of causality condition) as discussed before. That's why if the degree of the denominator in the reference model, $A_m(z)$ is lower than 2n-1, we have to increase its order by multiplying with $A_o(z)$ rather than setting higher order terms to be zero. This extra term of $A_o(z)$ can be canceled out later by properly choosing $T(z)$, and the goal of matching $A_m(z)$ can still be achieved.

5.2 Pole-Placement Problem— More Realistic Situations

We have already given the solution to pole-placement problem for the input-output model (5.1). In practical problems, we have to consider other realistic factors. For instance, we may wish to decrease the order of the controller and hence the overall system in order to simplify the design. Sometimes, we also have to consider how to deal with the external disturbances. To achieve the goal of lowering the order of the system, pole-zero cancellation can be used.

5.2.1 Pole-Zero Cancellation

Consider the system described by

$$Y(z) = \frac{B(z)}{A(z)}U(z)$$

Let the pole-placement controller be

$$U(z) = \frac{T(z)}{R(z)}U_c(z) - \frac{S(z)}{R(z)}Y(z)$$

Then we have

$$Y(z) = \frac{B(z)T(z)}{A_{cl}}U_c(z)$$

where

$$A_{cl}(z) = A(z)R(z) + B(z)S(z)$$

The idea is to reduce the order of the closed-loop system as low as possible. To achieve this, we may properly choose $T(z)$ to cancel out $A_o(z)$. Besides this, is there any other way to further reduce the degree?

Another way is to cancel out the stable zeros in $B(z)$! To achieve this, $A_{cl}(z)$ must have the stable zeros as the factors.

If $A(z)$ has stable poles, to simplify the controller, we may even leave those stable poles of $A(z)$ alone, which means $A_{cl}(z)$ have those stable poles as factors too.

More precisely, let

$$\begin{cases} A = A^+A^- \\ B = B^+B^- \end{cases} \tag{5.20}$$

Where A^+ and B^+ are the stable factors, and A^- and B^- are the unstable factors. Then let

$$A_{cl}(z) = A^+B^+\bar{A}_{cl} \tag{5.21}$$

and

$$T(z) = A^+\bar{T} \tag{5.22}$$

Then the closed loop transfer function becomes

$$\frac{B(z)T(z)}{A_{cl}(z)} = \frac{B^+B^-A^+\bar{T}}{A^+B^+\bar{A}_{cl}} = \frac{B^-\bar{T}}{\bar{A}_{cl}} \tag{5.23}$$

The corresponding Diophantine equation

$$A(z)R(z) + B(z)S(z) = A_{cl}(z)$$

becomes

$$A^+A^-R + B^+B^-S = A^+B^+\bar{A}_{cl} \tag{5.24}$$

Obviously

$$\begin{aligned} R &= B^+\bar{R} \\ S &= A^+\bar{S} \end{aligned} \tag{5.25}$$

And we have another Diophantine equation of lower order polynomials which is easier to solve.

$$\bar{A}\bar{R} + \bar{B}\bar{S} = \bar{A}_{cl} \tag{5.26}$$

The canceled factors must correspond to stable modes. In practice, it is useful to have more stringent requirements on allowable cancellations. Sometimes cancellation may not be desirable at all. In other cases, it may be reasonable to cancel zeros that are sufficiently well damped. As we discussed before, the canceled mode still affect the transient response. If it is well-damped, then the effects will decay to zero very fast any way. Otherwise, the performance of the system may be affected by these canceled modes. For instance, if the canceled modes are not well-damped, and are negative or complex, then undesirable oscillations may be introduced to the system.

Example 5.4: Cancellation of zeros

Let the open loop transfer function be

$$H(z) = \frac{k(z-b)}{(z-1)(z-a)}$$

where $b < 1$ and $a > 1$

Assume the desired close-loop system is characterized by the transfer function

$$H_m(z) = \frac{z(1+p_1+p_2)}{z^2 + p_1 z + p_2}$$

$z = b$ is a stable zero, which may be canceled out.

The order of the system is 2, therefore,

$$\begin{aligned} Deg\ \ R &= 1, Deg\ \ S = 1 \\ Deg\ \ A_{cl} &= 2n - 1 = 3 \end{aligned}$$

But the desired closed-loop is of second order. One possible way to reduce the order of the system is to cancel out the stable zero $z - b$,
Then let

$$R = z - b$$

And

$$A_{cl} = (z - b)(z^2 + p_1 z + p_2)$$

From the Diophantine equation we have

$$AR + BS = A(z - b) + k(z - b)S = A_{cl} = (z - b)(z^2 + p_1 z + p_2)$$

we have

$$A + kS = z^2 + p_1 z + p_2$$

$$(z-1)(z-a) + k(s_0 z + s_1) = z^2 + (ks_0 - a - 1)z + a + ks_1 = z^2 + p_1 z + p_2$$

Therefore

$$\begin{aligned} ks_0 - a - 1 &= p_1 \\ a + ks_1 &= p_2 \end{aligned}$$

and

$$\begin{aligned} s_0 &= \frac{a + 1 + p_1}{k} \\ s_1 &= \frac{p_2 - a}{k} \end{aligned}$$

Then the transfer function for the closed loop is

$$\frac{k(z-b)T}{(z-b)(z^2 + p_1 z + p_2)} = \frac{kT}{(z^2 + p_1 z + p_2)}$$

We still have the freedom to choose proper T.

Compare the above form to the desired transfer function H_m, we simply let

$$kT = z(1 + p_1 + p_2)$$

or

$$T = \frac{z(1 + p_1 + p_2)}{k} = t_0 z$$

and the resulting controller is then

$$U(z) = \frac{T(z)}{R(z)}U_c(z) - \frac{S(z)}{R(z)}Y(z)$$

$$(z-b)u = t_0 z u_c - (s_0 z + s_1)y$$

or

$$u(k) = bu(k-1) + t_0 u_c(k) - s_0 y(k) - s_1 y(k-1)$$

5.2.2 Disturbance Rejection Problem

Consider a system described by

$$Y(z) = \frac{B(z)}{A(z)}(U(z) + D(z))$$

where $d(k)$, i.e. $D(z)$, is the load disturbance. In this module, we only consider the case when the disturbance is an unknown constant. Let the two-degree-of-freedom controller be designed as

$$U(z) = \frac{T(z)}{R(z)}U_c(z) - \frac{S(z)}{R(z)}Y(z)$$

The transfer function from the disturbance to the output can be derived as

$$\frac{Y(z)}{D(z)} = \frac{R(z)B(z)}{A(z)R(z) + B(z)S(z)}$$

To eliminate the effect of the constant disturbance, we should design the controller such that the static gain of above transfer function is zero. Since the static gain is simply

$$\frac{R(1)B(1)}{A(1)R(1) + B(1)S(1)}$$

all we need to do is letting $R(z)$ have $(z-1)$ as a factor,

$$R(z) = R'(z)(z-1) \tag{5.27}$$

Then

$$R(1) = 0 \tag{5.28}$$

This design is essentially putting the integrator in the controller, which follows the same idea discussed earlier in section 2.4.6.

Example 5.5:

$$y(k+1) = 3y(k) + u(k) + d(k)$$

where $d(k)$ is a constant disturbance. In order to reject the disturbance, $R(z)$ must contain the factor $(z-1)$.

$$A(z) = z - 3$$
$$B(z) = 1$$

The simplest controller is the first order controller

$$R(z) = z - 1$$
$$S(z) = s_0 z + s_1$$

$$AR + BS = (z-3)(z-1) + s_0 z + s_1 = z^2 + (s_0 - 4)z + s_1 + 3$$

If the desired C.P. is

$$A_{cl}(z) = z^2 + p_1 z + p_2$$

we have

$$s_0 - 4 = p_1$$
$$s_1 + 3 = p_2$$

and

$$s_0 = 4 + p_1$$
$$s_1 = p_2 - 3$$

The closed loop transfer function is then

$$\frac{T(z)}{z^2 + p_1 z + p_2}$$

where $T(z)$ can be chosen to meet other requirements.

5.2.3 Tracking Problem — Command Signal Following

If the desired transfer function is

$$\frac{B_m(z)}{A_m(z)}$$

Can we make the transfer function of the closed-loop match this?
We know by output feedback control

$$U(z) = \frac{T(z)}{R(z)} U_c(z) - \frac{S(z)}{R(z)} Y(z)$$

We can have

$$Y(z) = \frac{B(z)T(z)}{A_{cl}(z)} U_c(z)$$

Question: is it possible to make

$$\frac{B(z)T(z)}{A_{cl}(z)} = \frac{B_m(z)}{A_m(z)}$$

Obviously, there is no problem to make the denominators match each other. The problem is how to deal with $B(z)$?

- If the zeros of $B(z)$ are all stable, then we can make A_{cl} contains B such that B will be canceled out, and then let $T = B_m$.
- If the some zeros of $B(z)$ are unstable, it is impossible to make the transfer function match the ideal one unless B_m contains all the unstable zeros of B as factors. That's why tracking an arbitrary signal is possible only for system with stable zeros.

Is it really impossible to follow a reference model perfectly for system with unstable zero? Can we overcome this problem by increasing the freedom of the controller? For instance, in the two-degree-of-freedom controller as shown in Fig. 5.3, the control input is the summation of the feedforward control signal, u_{ff} and u_{fb},

$$u(k) = u_{ff}(k) + u_{fb}(k)$$

where

$$U_{ff}(z) = \frac{T(z)}{R(z)} U_c(z)$$

and

$$U_{fb}(z) = -\frac{S(z)}{R(z)} Y(z)$$

Note that the transfer functions for generating the two inputs share the same denominator $R(z)$. In order to gain extra freedom, it is also possible to relax this constraint, and let

$$U_{ff}(z) = H_{ff}(z) U_c(z)$$

where the feedforward transfer function $H_{ff}(z)$ gives you additional freedom.

What would be the advantages by doing this?

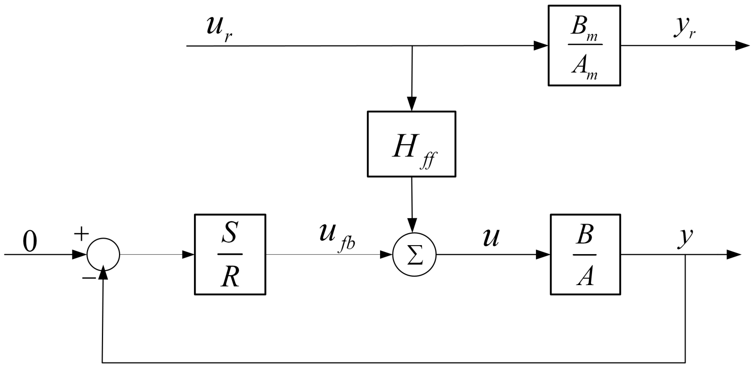


Figure 5.4: Another type of two-degree-of-freedom controller

Easy calculation will give us the following transfer functions if T/R is replaced by $H_{ff}(z)$,

$$\frac{Y(z)}{U_c(z)} = \frac{BR}{AR+BS} H_{ff}$$

$$\frac{U(z)}{U_c(z)} = \frac{AR}{AR+BS} H_{ff}$$

In order to match the reference model, design the feedback controller $S(z)/R(z)$ such that

$$AR + BS = A_m$$

and choose the feedforward controller,

$$H_{ff}(z) = \frac{B_m}{RB}$$

we will have

$$\frac{Y(z)}{U_c(z)} = \frac{B_m}{A_m}$$

So the transfer function indeed matches the reference model perfectly. But how about the input signal? We have to make sure the input signals are all bounded. For this purpose, we have to check the transfer function from the command signal to the input,

$$\frac{U(z)}{U_c(z)} = \frac{AB_m}{BA_m}$$

Still, B(z) is required to be stable to keep the input signal bounded. This shows that the stable zeros condition is essential for perfect tracking, which cannot be simply got rid of by adding more design freedoms.

In summary, we have following rules of thumb for pole placement controller,

$$U(z) = \frac{T(z)}{R(z)}U_c(z) - \frac{S(z)}{R(z)}Y(z)$$

1. Occam's razor — the simpler, the better.
2. Based on the design specifications, determine the constraints on R. For instance,
   - If zero cancellation is needed, then R must contain the stable zero factor B^+.
   - If constant disturbance rejection is required, then R must contain $(z-1)$.
3. Causal conditions: $DegR \geq DegS$
4. Design R and S first by solving the Diophantine equation $AR+BS = A_{cl}$
5. Choose T or H_{ff} at the final stage, to satisfy other design requirements, such as zero placement.
6. For the simplest case, the order of the controller is limited to $n-1$. However, the order of the controller can be increased in order to meet other design requirements.

5.3 Summary and Further Beyond

In any control system design problem one can distinguish five important considerations: stability, transient response, tracking performance, constraints, and robustness.

1. *Stability.* This is concerned with stability of the system, including boundedness of inputs, outputs, and states. This is clearly a prime consideration in any control system.

2. *Transient response.* Roughly this is concerned with how fast the system responds. For linear systems, it can be specified in the time domain in terms of rise time, settling time, percent overshoot, and so on, and in the frequency domain in terms of bandwidth, damping, resonance, and so on.

3. *Tracking performance.* This is concerned with the ability of the system to reproduce desired output values. A special case is when the desired outputs are constant, in which case we often use the term output regulation, in lieu of tracking.

4. *Constraints.* It turns out that in the linear case if the system model is known exactly and there is no noise, then arbitrarily good performance can be achieved by use of a sufficiently complex controller. However one must bear in mind that there are usually physical constraints that must be adhered to, such as limits in the magnitude of the allowable control effort, limits in the rate of change of control signals, limits on internal variables such as temperatures and pressures, and limits on controller complexity. These factors ultimately place an upper limit on the achievable performance.

5. *Robustness.* This is concerned with the degradation of the performance of the system depending on certain contingencies, such as unmodeled dynamics, including disturbances, parameter variations, component failure, and so on.

In this module, we primarily treat sampled data control systems so that our inputs and outputs are sequences of numbers. Typically, the actual process will be continuous in nature. Suitable data models can then be obtained by the techniques discussed in previous lectures. It must be borne in mind that a design based on discrete-time models refers only to the sampled values of the input and output. Usually, the corresponding continuous response will also be satisfactory, but in certain cases it may be important to check that the response between samples is acceptable.

Now I hope you have mastered some of the fundamental knowledge in controller designs for improving the performance of the system. Of course, we have made many assumptions in order to construct the controller properly. In reality, these assumptions may be invalid, which stimulate us to explore more in coping with the complex real world.

We have made three critical assumptions through out the module:

1. The system is linear.
2. The mathematical model is completely known.
3. The dynamic equations governing the system do not change with time. In other words, the system is time-invariant.

Regarding the first assumption, it is not a big issue in many of the practical problems, where we are only interested in controlling small deviations from certain operating point. If the deviation is big, and the linearization does not work any more, you may have to resort to nonlinear control technology, which still has a lot of open questions. The biggest challenge of nonlinear system is that we don't have superposition principle to guide us any more! The most often used trick in solving the nonlinear control problem is to choose the control input properly such that the closed loop system is linear just as I showed you in the introduction of this module.

If the model is unknown, or partially known, the first option is to use model-free controller such as PID control. If PID controller does not work well, then you may have to estimate the model parameters and use the adaptive control scheme to solve the problem.

We only examine the single-input, single-output system in this module. If there are multiple inputs and multiple outputs, there are other interesting phenomena arising.

We have formulated our control problem in the framework of model reference control, in which the objective is to track a reference signal. If there are many other constraints to consider in the problem, like the cost of the control inputs, it is better to formulate it as an optimization problem.

If you are interested in exploring more on how to control more complex systems, please register for other advanced modules such as EE4302 Advanced Control Systems, EE4305 Introduction to Fuzzy / Neural Systems, EE4307 Control Systems Design and Simulation, EE4306 Distributed Autonomous Robotic Systems and ME4245 Robot Kinematics, Dynamics and Control.

For the rest of the semester, we are going to show how the control theory we have learned can be applied to real world applications by examining a case study in great details.

So stay tuned!