EE5103/ME5403 Computer Control Systems
Part I

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. For the first half of this module, our focus will be only upon the control theory, and I will leave the practical applications and design issues to the later stage. Therefore, 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 is. 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:

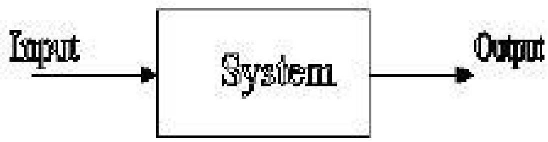


Figure 1.1: The simplest block diagram of a system

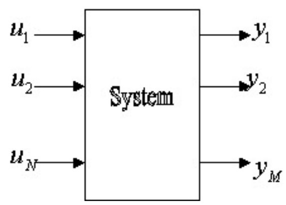


Figure 1.2: A block diagram of a system by variety of inputs and outputs

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.

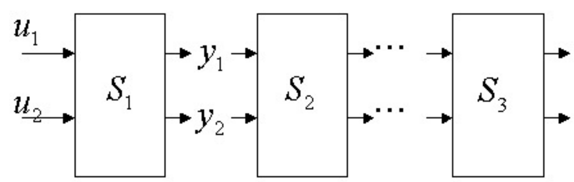


Figure 1.3: A block diagram of a multi-stage system

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:

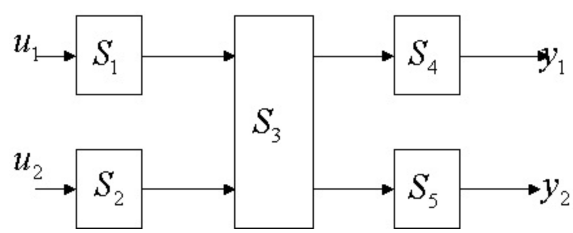


Figure 1.4: A block diagram of systems in atmospheric physics

Fig1.4 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.

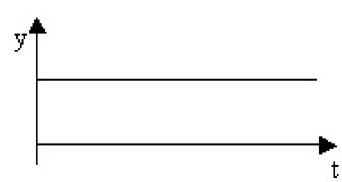
1. Build a new system;
2. Replace or change the subsystems;
3. Change the inputs.

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.

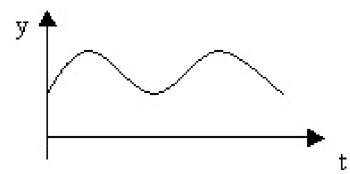
Control: influence the behavior of the system to achieve a desired goal.

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

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

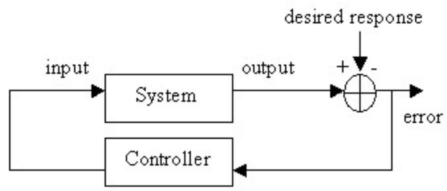


Figure 1.6: A closed-loop system

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.

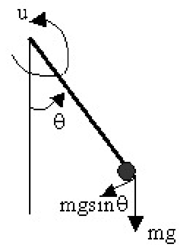


Figure 1.7: A single-link rotational joint

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 finger tip, 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 systems 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.

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.

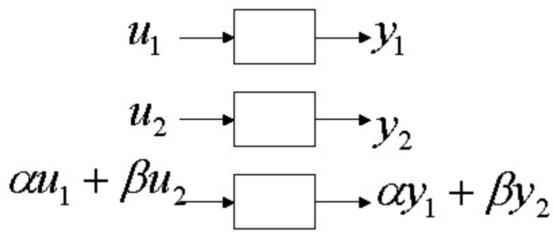


Figure 1.8: Superposition Principle

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 signal $u(t)$ 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 $a_\lambda, \lambda = 1, 2,$ ... , 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

$$A(u) = a_1 A(\phi_1) + a_2 A(\phi_2) + ...$$

Where $A(\phi_\lambda)$ represents the response of the system to the component $\phi_\lambda, \lambda = 1, 2,$ ... . 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.

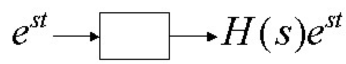


Figure 1.9: The response function for exponential input e^{st}

Try the response function as $H(s)e^{st}$, and plug it into the pendulum Eqn. (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

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

   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 Eqn. (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 — $H(s)$

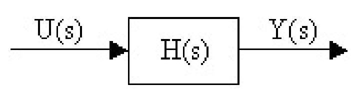


Figure 1.10: Transfer Function

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 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 undergraduate course on 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 can be used for disturbance rejection, i.e. the effect of the disturbance on the output is minimized.

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

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

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 made 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:

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

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?

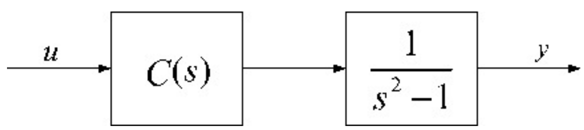


Figure 1.11: Open Loop Control

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 are 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

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

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?

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

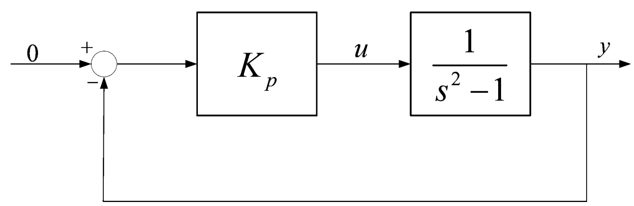


Figure 1.12: Proportional Control

1. If $K_p < 1$, then all of the solutions diverge to infinity.
2. If $K_p = 1$, then it becomes a double integrator, which is unstable.
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 need 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,

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

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

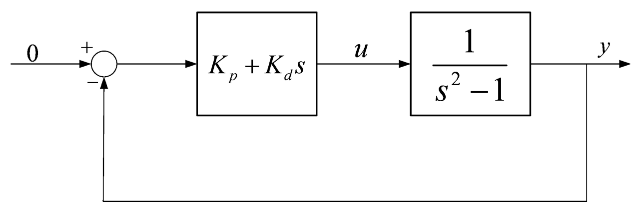


Figure 1.13: Proportional and Derivative Control

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.

In the next section, we will see how to solve the control problem using a different approach.

1.3 State-Space approach

For many of you, I believe that today is not the first time you hear of the concept of state for dynamic system. What is the state of a system?

Historical background: the concept of state has been around since the end of 19th century, (Poincare introduced it to study dynamical systems) but it was only till the late 1950s that it was introduced into the control theory. And it marks the new era of control theory and modern control theory usually means the control theory formulated in the state space. And classical control theory means the transfer function approach.

<u>State at time *t*</u> (present):

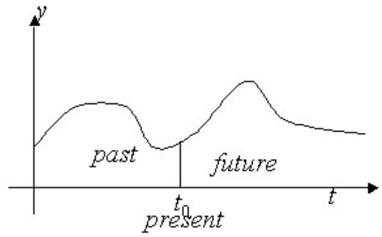


Figure 1.14: State at time t

state at time t_0 (present) separates the future from the past by providing all the information about the past of the system that is relevant to the determination of the response of the system to any input starting from t_0. In other words, the future of the system can be completely determined by the knowledge of current state combining with the present and future inputs.

To find the right state variable, basically we have to answer the following question: What information is really needed for predicting the future assuming we have the complete information about the past and the present of the system combining with the inputs information?

1.3.1 Examples

Pendulum problem

$$\ddot{y}(t) - y(t) = u(t) \tag{1.20}$$

At present time t_0, what do we have to specify in order to determine $y(t)$ in the future? Obviously the knowledge of the initial condition $y(0)$, $y'(0)$ would suffice. So the state variable of the pendulum at any time t can be defined as

$$x(t) = [y(t), y'(t)] \tag{1.21}$$

Order: The dimension of the state. In the pendulum case, it is second order system.

Perfect delay problem

$$y(t) = y(t-d) \tag{1.22}$$

At time t_0 , state $x(t_0) = y_{[t_0-d,t_0]}$, since the value of $y(t)$ after t_0 is completely determined by state $x(t_0)$. The order of the system is infinity! Since the state of the system is specified by the output signal for the period of $[t_0 - d, t_0]$. This is one of the reasons for the difficulty of dealing with

time-delay in continuous time system.
Question: what is the state for the pendulum with delayed input as shown below?

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

1.3.2 Properties

By definition of state we have following map:

$$\begin{aligned} x(t) &= x(x(t_0); u_{[t_0,t]}) \\ y(t) &= y(x(t_0); u_{[t_0,t]}) \end{aligned} \tag{1.23}$$

There are many properties associate with the concept of state, and the most important is the

Separation property

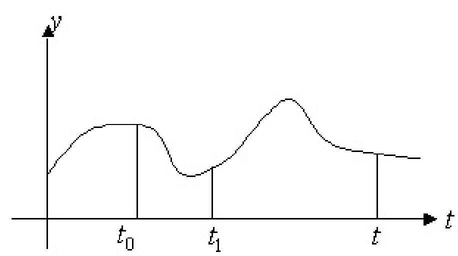


Figure 1.15: Separation property

The present state separates the past from the future, and the future only depends upon the present state. In other words, all the past information relevant for determining the future of the system is compacted into the current state.

More specifically, for any input consisting of a segment $u_{[t_0,t_1]}$ followed by a segment $u_{[t_1,t]}$, the response of the system to the whole segment $u_{[t_0,t]}$ starting from initial state $x(t_0)$ consists of the response segment $Y(x(t_0); u_{[t_0,t_1]})$ followed by the response segment $Y(x(t_1); u_{[t_1,t]})$, where $x(t_1)$ is the state of the system at time t_1. $x(t_1)$ in a sense separates the past response $Y(x(t_0); u_{[t_0,t_1]})$ from the future response $Y(x(t_1); u_{[t_1,t]})$ at time t_1 . And

$$x(t) = x(x(t_0); u_{[t_0,t]}) = x(x(t_1); u_{[t_1,t]}) = x(x(x(t_0); u_{[t_0,t_1]}); u_{[t_1,t]}) \tag{1.24}$$

Decomposition property of linear system

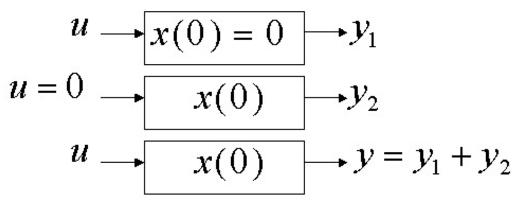


Figure 1.16: System Response= zero-input response + zero-state response

$$
\begin{aligned}
x(t) &= x(x(t_0); u_{[t_0,t]}) = x(x(t_0); 0) + x(0; u_{[t_0,t]}) \\
y(t) &= y(x(t_0); u_{[t_0,t]}) = y(x(t_0); 0) + y(0; u_{[t_0,t]})
\end{aligned}
\tag{1.25}
$$

If the system is stable, then the zero-input response can be safely ignored for steady state behavior because this component will decay to zero in the long run.

Let's return to our pendulum example.

Using the state vector $x(t) = [y(t), \dot{y}(t)]$, we can derive the state-space representation of the pendulum.

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_1(t) + u(t) \\
y(t) &= x_1(t)
\end{aligned}
$$

or in matrix form

$$
\begin{aligned}
\dot{x} &= Ax + bu \\
y &= cx
\end{aligned}
\tag{1.26}
$$

where

$$
A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 0 \end{bmatrix}
$$

Note: the original second order differential equation has been transformed into a first order differential equation of state vectors.

In general, most of the linear systems can be described by equation (1.26)

If x is a scalar, then the solution to above equation can be easily obtained by Laplace transform as follows.

$$\begin{aligned}
&sX(s) - x(0) = AX(s) + bU(s) \\
&X(s) = \frac{1}{s-A}x(0) + \frac{bU(s)}{s-A} \\
&x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}bu(\tau)d\tau \\
&y = cx = ce^{At}x(0) + c\int_0^t e^{A(t-\tau)}bu(\tau)d\tau
\end{aligned} \tag{1.27}$$

Now if x is a vector, can we still write the solution in such forms?

Fortunately, even if x is a vector, the solution still takes above form, where

$$e^{At} = I + At + ... + A^k\frac{t^k}{k!} + ... \tag{1.28}$$

How to calculate e^{At}?

Using the infinite series obviously is not an option.

To calculate e^{At}, let's see if Laplace transform can do the trick again.

$$\begin{aligned}
&sX(s) - X(0) = AX(s) + bU(s) \\
&Y(s) = CX(s)
\end{aligned}$$

we have

$$\begin{aligned}
&(sI - A)X(s) = X(0) + bU(s) \\
&X(s) = (sI - A)^{-1}X(0) + (sI - A)^{-1}bU(s) \\
&Y(s) = C(sI - A)^{-1}X(0) + C(sI - A)^{-1}bU(s)
\end{aligned}$$

Compared to

$$\begin{aligned}
&x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}bu(\tau)d\tau \\
&y = cx = ce^{At}x(0) + c\int_0^t e^{A(t-\tau)}bu(\tau)d\tau
\end{aligned}$$

it is obvious that $(sI - A)^{-1}$ is e^{At} in the s-domain. Therefore e^{At} can be easily obtained by inverse Laplace transform of $(sI - A)^{-1}$.
There are some other ways to compute e^{At}. Using Laplace transform is the easiest way.

What's the corresponding transfer function?

Setting $X(0) = 0$,

We have

$$Y(s) = H(s)U(s)$$

And

$$H(s) = C(sI - A)^{-1}b = \frac{CB(s)b}{d(s)} \tag{1.29}$$

1.3.3 State-feedback controller

Input depends upon the states, not on time.

$$u = g(x) \tag{1.30}$$

If we use the linear state-feedback controller:

$$u = -kx = -k_1x_1 - k_2x_2 + ... \tag{1.31}$$

The closed loop equations are:

$$\begin{aligned} \dot{x} &= Ax + bu = Ax - bkx = (A - bk)x \\ y &= cx \end{aligned} \tag{1.32}$$

The poles of the system can be determined by solving

$$\det(sI - (A - bk)) = 0 \tag{1.33}$$

Now the question becomes an algebraic one: given pair (A, b), can we choose proper k such that the eigenvalues of $(A - bk)$ can match any set of n eigenvalues?

We will examine this question in greater detail later in this module. Let's just take a look at the pendulum example.

The pendulum example:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

If we choose to use the state feedback controller as

$$u = -k_1x_1 - k_2x_2 = -kx \tag{1.34}$$

where

$$k = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

Remark: the previous PD (proportional and derivative) controller is the same as this, just with a different notation.

Plug this control input into the equation, we have

$$\dot{x} = Ax + bu = (A - bk)x \tag{1.35}$$

where

$$A - bk = \begin{bmatrix} 0 & 1 \\ 1 - k_1 & -k_2 \end{bmatrix}$$

The characteristic equation of the closed loop systems

$$\det(sI - (A - bk)) = s^2 + k_2 s + (k_1 - 1) \tag{1.36}$$

So if the desired characteristic equation is

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{1.37}$$

where ζ and ω_n are the desired damping ratio and natural frequency of the closed loop system.
Just let

$$\begin{aligned} k_2 &= 2\zeta\omega_n \\ k_1 - 1 &= \omega_n^2 \end{aligned} \tag{1.38}$$

1.4 Computer Control System

In all the previous discussion, the system is continuous-time. All the signals are usually handled by analog technology. If we want to implement the controller using the digital computers which are discrete in nature, we have to sample the signals of the continuous-time system.

There are many issues involved such as:

- How do we sample the continuous-time system? what are the related issues? After sampling, the continuous-time system will become discrete-time system.
- What are the analytical tools for discrete-time system?
- Is there a similar transform for discrete-time system just as the Laplace transform for continuous time?

- What is the transfer function for discrete-time system?
- What is the meaning of poles and zeros?
- How to determine the stability?

All the issues will be discussed in the future lectures.

1.5 Notes and References

Most of the materials covered in this chapter can be found in following references:

1. R. Bellman, *Introduction to the Mathematical Theory of Control Processes, vol. 1*, Academic Press, 1967.
2. L. A. Zadeh and C. A. Desoer, *Linear System Theory*, McGraw-Hill, 1963.
3. E. D. Sontag, *Mathematical Control Theory,* second edition, Springer-Verlag, 1998.

Chapter 2

Computer Control and Discrete-Time Systems

2.1 Introduction

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

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

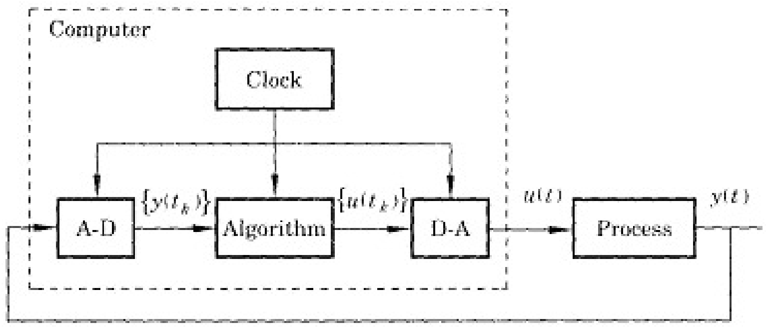


Figure 2.1: Schematic diagram of a computer-controlled system.

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 h. 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. How to choose the sampling period properly is a big issue, which you can find out more in the textbooks.

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

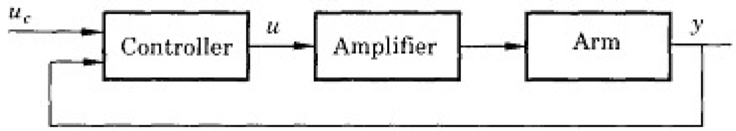


Figure 2.2: A system for controlling the position of the arm of a disk drive.

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.

We will give examples that illustrate the differences and the similarities of analog and computer-controlled systems. It will be shown that essential new phenomena that require theoretical attention do indeed occur.

2.1.1 A Naive Approach to Computer-Controlled Systems

We may expect that a computer-controlled system behaves as a continuous-time system if the sampling period is sufficiently small. This is true under very reasonable assumptions. We will illustrate this with an example.

Example 2.1 Controlling the arm of a disk drive

A schematic diagram of a disk-drive assembly is shown in Fig. 2.2.

Let J be the moment of inertia of the arm assembly. The dynamics relating the position y of the arm to the voltage u of the drive amplifier is approximately described by the Newton's law,

$$J\ddot{y}(t) = u(t)$$

the corresponding transfer function is

$$G(s) = \frac{k}{Js^2} \tag{2.1}$$

where k is a constant. The purpose of the control system is to control the position of the arm so that the head follows a given track and that it can be rapidly moved to a different track. It is easy to find the benefits of improved control. Better trackkeeping allows narrower tracks and higher packing density. A faster control system reduces the search time. In this example we will focus on the search problem, which is a typical servo problem. Let u_c , be the command signal and denote Laplace transforms with capital letters. A simple servo controller can be described by

$$U(s) = \frac{bK}{a} U_c(s) - K\frac{s+b}{s+a} Y(s) \tag{2.2}$$

Here we ignore the details on the procedure of how to design above analog controller due to time limitation. If the controller parameters are chosen as

$$a = 2\omega_0$$

$$b = \omega_0/2$$

$$K = 2\frac{J\omega_0^2}{k}$$

a closed system with the characteristic polynomial

$$F(s) = s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3$$

is obtained. This system has a reasonable behavior with a settling time to 5% of $5.52/\omega_0$. See Fig. 2.3.

Now the question is how to convert this analog controller to computer control system? To obtain an algorithm for a computer-controlled system, the control law given by (2.2) is first written as

$$U(s) = \frac{bK}{a} U_c(s) - KY(s) + K\frac{a-b}{s+a} Y(s) = K(\frac{b}{a} U_c(s) - Y(s) + X(s))$$

This control law can be written as

$$u(t) = K(\frac{b}{a} u_c(t) - y(t) + x(t)) \tag{2.3}$$

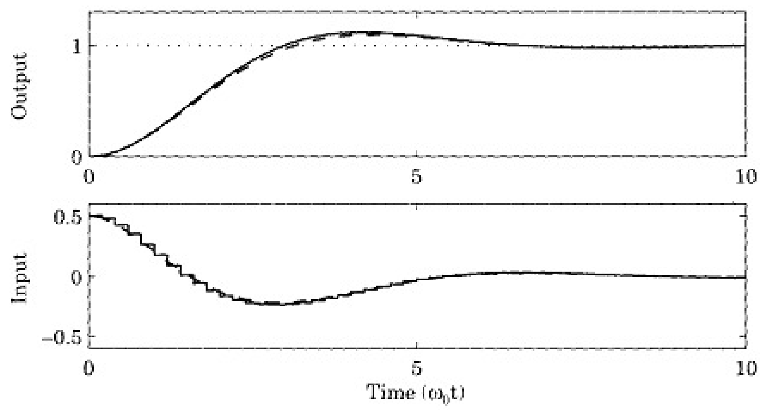


Figure 2.3: Simulation of the disk arm servo with analog (dashed) and computer control (solid). The sampling period is $h = 0.2/\omega_0$.

where

$$\frac{dx}{dt} = -ax + (a-b)y$$

How to calculate $x(t)$? The derivative dx/dt may be simply approximated with a difference. This gives

$$\frac{x(t+h) - x(t)}{h} = -ax(t) + (a-b)y(t)$$

The following approximation of the continuous algorithm (2.3) is then obtained:

$$u(t_k) = K(\frac{b}{a}u_c(t_k) - y(t_k) + x(t_k))$$
$$x(t_k + h) = x(t_k) + h((a-b)y(t_k) - ax(t_k))$$

Arm position y is read from an analog input. Its desired value is assumed to be given digitally. The algorithm has one state, variable x, which is updated at each sampling instant. The control law is computed and the value is converted to an analog signal. The program is executed periodically with period h by a scheduling program, as illustrated in Fig. 2.2.

Because the approximation of the derivative by a difference is good if the interval h is small, we can expect the behavior of the computer-controlled system to be close to the continuous-time system. This is illustrated in Fig. 2.3, which shows the arm positions and the control signals for the systems with $h = 0.2/\omega_0$. Notice that the control signal for the computer-controlled

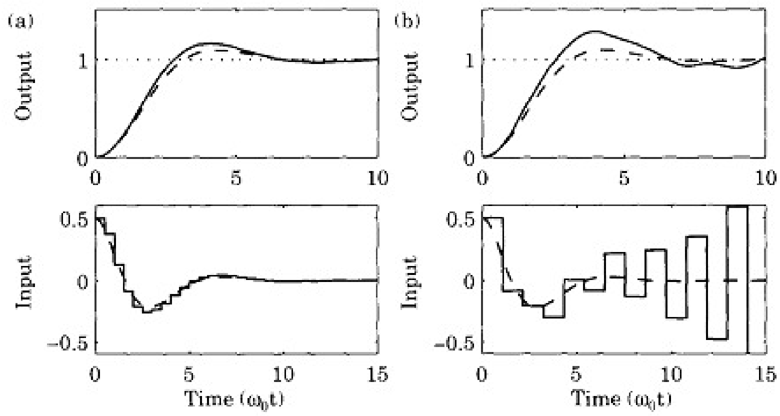


Figure 2.4: Simulation of the disk arm servo with computer control having sampling rates $(a) h = 0.5/\omega_0$ and $(b) h = 1.08/\omega_0$. For comparison, the signals for analog control are shown with dashed lines.

system is constant between the sampling instants. Also notice that the difference between the outputs of the systems is very small. The computer-controlled system has slightly higher overshoot and the settling time to 5% is a little longer, $5.7/\omega_0$ instead of $5.5/\omega_0$. The difference between the systems decreases when the sampling period decreases. When the sampling period increases the computer-controlled system will, however, deteriorate. This is illustrated in Fig. 2.4, which shows the behavior of the system for the sampling periods $h = 0.5/\omega_0$ and $h = 1.08/\omega_0$. The response is quite reasonable for short sampling periods, but the system becomes unstable for long sampling periods.

We have thus shown that it is straightforward to obtain an algorithm for computer control simply by writing the continuous-time control law as a differential equation and approximating the derivatives by differences. The example indicated that the procedure seemed to work well if the sampling period was sufficiently small. The overshoot and the settling time are, however, a little larger for the computer-controlled system.

2.1.2 Deadbeat Control

Example 2.1 seems to indicate that a computer-controlled system will be inferior to a continuous-time system. We will now show that this is not necessarily the case. The periodic nature of the control actions can be

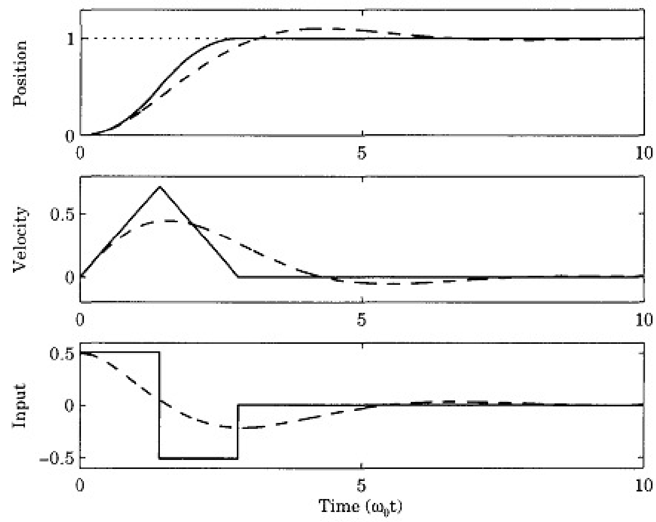


Figure 2.5: Simulation of the disk arm servo with deadbeat control (solid). The sampling period is $h = 1.4/\omega_0$. The analog controller from Example 1.2 is also shown (dashed).

actually used to obtain control strategies with superior performance.

Example 2.2 Disk drive with deadbeat control

Consider the disk drive in the previous example. Fig. 2.5 shows the behavior of a computer-controlled system with a very long sampling interval $h = 1.4/\omega_0$. For comparison we have also shown the arm position, its velocity, and the control signal for the continuous controller used in Example 2.1. Notice the excellent behavior of the computer-controlled system. It settles much quicker than the continuous-time system. The 5% settling time is $2.34/\omega_0$, which is much shorter than the settling time $5.5/\omega_0$ of the continuous system. The output also reaches the desired position without overshoot and it remains constant when it has achieved its desired value, which happens in finite time. This behavior cannot be obtained with continuous-time systems because the solutions to such systems are sums of functions that are products of polynomial and exponential functions. The behavior obtained can be also described in the following way: The arm accelerates with constant acceleration until it is halfway to the desired position and it then decelerates with constant retardation. The example shows that control strategies with different behavior can be obtained with computer control. In the particular example the response time can be reduced by a factor of

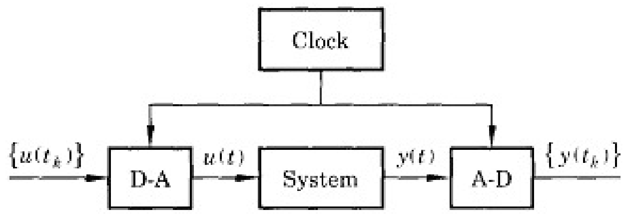


Figure 2.6: Block diagram of a continuous-time system connected to $A-D$ and $D-A$ converters.

2. The control strategy in Example 2.2 is called deadbeat control because the system is at rest when the desired position is reached. Such a control scheme cannot be obtained with a continuous-time controller. We will discuss how to design deadbeat controller in greater details later in this course. Therefore, we cannot simply regard computer-control system as an approximation of the analog controller. It is more powerful than analog controller, and that's why we need to study it as a separate subject for one semester!

2.2 Sampling a Continuous-Time State-Space System

The computer receives measurements from the process at discrete times and transmits new control signals at discrete times. The goal then is to describe the change in the signals from sample to sample and disregard the behavior between the samples. The use of difference equations then becomes a natural tool. It should be emphasized that computer-oriented mathematical models only give the behavior at the sampling points, and the physical process is still a continuous-time system. Looking at the problem this way, however, will greatly simplify the treatment.

Assume that the continuous-time system is given in the following state-space form:

$$\begin{aligned}\frac{dx}{dt} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{2.4}$$

The system has r inputs, p outputs, and is of order n.
A common situation in computer control is that the $D-A$ converter is so constructed that it holds the analog signal constant until a new conversion is commanded. This is often called a zero-order-hold circuit. Given the

state at the sampling time t_k, the state at some future time t is obtained by solving (2.4). The state at time t, where $t_k \leq t \leq t_{k+1}$, is thus given by

$$
\begin{aligned}
x(t) &= e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-s')}Bu(s')ds' \\
&= e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-s')}ds'Bu(t_k) \\
&= e^{A(t-t_k)}x(t_k) + \int_{0}^{t-t_k} e^{As}dsBu(t_k) \\
&= \Phi(t, t_k)x(t_k) + \Gamma(t, t_k)u(t_k)
\end{aligned}
\tag{2.5}
$$

The second equality follows because u is constant between the sampling instants. The state vector at time t is thus a linear function of $x(t_k)$ and $u(t_k)$. The system equation of the sampled system at the sampling instants is then

$$
\begin{aligned}
x(t_{k+1}) &= \Phi(t_{k+1}, t_k)x(t_k) + \Gamma(t_{k+1}, t_k)u(t_k) \\
y(t_k) &= Cx(t_k) + Du(t_k)
\end{aligned}
\tag{2.6}
$$

where

$$
\begin{aligned}
\Phi(t_{k+1}, t_k) &= e^{A(t_{k+1}-t_k)} \\
\Gamma(t_{k+1}, t_k) &= \int_{0}^{t_{k+1}-t_k} e^{As}dsB
\end{aligned}
$$

The relationship between the sampled signals thus can be expressed by the linear difference equation (2.6). The model in (2.6) is therefore called a zero-order-hold sampling of the system in (2.4). The system in (2.6) can also be called the zero-order-hold equivalent of (2.4).

The state vector at times between sampling points is given by (2.5). This makes it possible to investigate the intersample behavior of the system. Notice that the responses between the sampling points are parts of step responses, with initial conditions, for the system. This implies that the system is running in open loop between the sampling points.

For periodic sampling with period h, we have $t_k = kh$ and the model of (2.6) simplifies to the time-invariant system

$$
\begin{aligned}
x(kh + h) &= \Phi x(kh) + \Gamma u(kh) \\
y(kh) &= Cx(kh) + Du(kh)
\end{aligned}
\tag{2.7}
$$

where

$$\begin{aligned}\Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{As} ds B\end{aligned} \tag{2.8}$$

By the way, how to get e^{Ah}? Use Laplace transform to get e^{At} first! Check out section 1.3.2 in your lecture notes!

2.3 State-space Model of Discrete-Time Systems

In most of the remaining part of this course we will disregard how the difference equation representing the discrete-time system has been obtained. Instead we will concentrate on the properties of difference equations. Time-invariant discrete-time systems can be described by the difference equation

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned} \tag{2.9}$$

For simplicity the sampling period is used as the time unit, i.e., $h = 1$.

2.3.1 Solution of the System Equation

To analyze discrete-time systems it is necessary to solve the system Eq (2.9). Assume that the initial condition $x(k_0)$ and the input signals $u(k_0), u(k_0 + 1), ...$ are given. How is the state then evolving? It is possible to solve (2.9) by simple iterations.

$$\begin{aligned}x(k_0+1) &= \Phi x(k_0) + \Gamma u(k_0) \\ x(k_0+2) &= \Phi x(k_0+1) + \Gamma u(k_0+1) \\ &= \Phi^2 x(k_0) + \Phi\Gamma u(k_0) + \Gamma u(k_0+1) \\ &\ \vdots \\ x(k) &= \Phi^{k-k_0} x(k_0) + \Phi^{k-k_0-1}\Gamma u(k_0) + ... + \Gamma u(k-1) \\ &= \Phi^{k-k_0} x(k_0) + \sum_{j=k_0}^{k-1} \Phi^{k-j-1}\Gamma u(j)\end{aligned} \tag{2.10}$$

The solution consists of two parts: One depends on the initial condition, and the other is a weighted sum of the input signals.

2.4 Input-Output Models

A dynamic system can be described using either internal models or external models. Internal models – for instance, the state-space models – describe all internal couplings among the system variables. The external models give only the relationship between the input and the output of the system. For instance the ARMA (Auto-Regressive Moving Average) model,

$$y(k+1) = \sum_{i=0}^{n-1} a_i y(k-i) + b_i u(k-i) \tag{2.11}$$

In order to describe such difference equation, it is convenient to introduce the time shift-operator.

2.4.1 Shift-Operator calculus

The forward-shift operator is denoted by q. It has the property

$$qf(k) = f(k+1) \tag{2.12}$$

The inverse of the forward-shift operator is called the backward-shift operator or the delay operator and is denoted by q^{-1}. Hence

$$q^{-1}f(k) = f(k-1) \tag{2.13}$$

Operator calculus gives compact descriptions of systems and makes it easy to derive relationships among system variables, because manipulation of difference equations is reduced to a purely algebraic problem.

The shift operator is used to simplify the manipulation of higher-order difference equations. Consider the equation

$$y(k+n_a)+a_1 y(k+n_a-1)+\cdots+a_{n_a} y(k) = b_0 u(k+n_b)+\cdots+b_{n_b} u(k) \tag{2.14}$$

where $n_a \geqslant n_b$. Use of the shift operator gives

$$(q^{n_a} + a_1 q^{n_a-1} + ... + a_{n_a})y(k) = (b_0 q^{n_b} + ... + b_{n_b})u(k) \tag{2.15}$$

With the introduction of the polynomials

$$A(z) = z^{n_a} + a_1 z^{n_a-1} + ... + a_{n_a} \tag{2.16}$$

and

$$B(z) = b_0 z^{n_b} + b_1 z^{n_b-1} + ... + b_{n_b} \tag{2.17}$$

the difference equation can be written as

$$A(q)y(k) = B(q)u(k) \tag{2.18}$$

Or in a more compact form

$$y(k) = \frac{B(q)}{A(q)}u(k) = H(q)u(k)$$

and we have a name for $H(q)$—**pulse-transfer operator**. Table 2.1. lists the pulse-transfer operators for various discrete-time systems obtained by Zero-order hold sampling of continuous-time systems, $G(s)$.

Eq (2.14) can be also expressed in terms of the backward-shift operator. Notice that (2.14) can be written as

$$y(k)+a_1y(k-1)+...+a_{n_a}y(k-n_a) = b_0u(k-d)+...+b_{n_b}u(k-d-n_b) \tag{2.19}$$

where $d = n_a - n_b$ is the pole excess of the system. The system in (2.14) can be written as

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k-d) \tag{2.20}$$

where

$$A^*(z^{-1}) = A(z)/z^{n_a} = 1 + a_1z^{-1} + \cdots + a_{n_a}z^{-n_a} \tag{2.21}$$

$$B^*(z^{-1}) = B(z)/z^{n_b} = b_0 + b_1z^{-1} + \cdots + b_{n_b}z^{-n_b} \tag{2.22}$$

2.5 The z-Transform

In the analysis of continuous-time systems the Laplace transform plays an important role. The Laplace transform turns the differential equations into algebraic ones, and makes it possible to introduce the transfer function and the frequency interpretation of a system. The discrete-time analogy of the Laplace transform is the z-transform — a convenient tool to study linear difference equations.

Definition 2.1 z-Transform

Consider the discrete-time signal $\{ f(k) : k = 0, 1, ... \}$. The z-transform of $f(k)$ is defined as

$$z\{f(k)\} = F(z) = \sum_{k=0}^{\infty} f(k)z^{-k} \tag{2.23}$$

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

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

$G(s)$	$H(q)$ 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}
$\frac{a}{s+a}$	$\frac{1 - \exp(-ah)}{q - \exp(-ah)}$
$\frac{a}{s(s+a)}$	$b_1 = \frac{1}{a}(ah - 1 + e^{-ah})$ $b_2 = \frac{1}{a}(1 - e^{-ah} - ahe^{-ah})$ $a_1 = -(1 + e^{-ah})$ $a_2 = e^{-ah}$
$\frac{a^2}{(s+a)^2}$	$b_1 = 1 - e^{-ah}(1 + ah)$ $b_2 = e^{-ah}(e^{-ah} + ah - 1)$ $a_1 = -2e^{-ah}$ $a_2 = e^{-2ah}$
$\frac{s}{(s+a)^2}$	$\frac{(q-1)he^{-ah}}{(q - e^{-ah})^2}$
$\frac{ab}{(s+a)(s+b)}$ $a \neq b$	$b_1 = \frac{b(1 - e^{-ah}) - a(1 - e^{-bh})}{b - a}$ $b_2 = \frac{a(1 - e^{-bh})e^{-ah} - b(1 - e^{-ah})e^{-bh}}{b - a}$ $a_1 = -(e^{-ah} + e^{-bh})$ $a_2 = e^{-(a+b)h}$

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}	$\frac{Tze^{-aT}}{\left(z-e^{-aT}\right)^2}$
7	$\frac{1}{s(s+a)}$	$\frac{1}{a}\left(1-e^{-at}\right)$	$\frac{z\left(1-e^{-aT}\right)}{a(z-1)\left(z-e^{-aT}\right)}$
8	$\frac{b-a}{(s+a)(s+b)}$	$e^{-at}-e^{-bt}$	$\frac{z\left(e^{-aT}-e^{-bT}\right)}{\left(z-e^{-aT}\right)\left(z-e^{-bT}\right)}$
9	$\frac{1}{s(s+a)(s+b)}$	$\frac{1}{ab}-\frac{e^{-at}}{a(b-a)}-\frac{e^{-bt}}{b(a-b)}$	
10	$\frac{1}{s(s+a)^2}$	$\frac{1}{a^2}\left(1-e^{-at}-ate^{-at}\right)$	
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	$\frac{s+a}{(s+a)^2+b^2}$	$e^{-at}\cos(bt)$	$\frac{z^2-ze^{-aT}\cos(bT)}{z^2-2ze^{-aT}\cos(bT)+e^{-2aT}}$

where z is a complex variable. It is interesting to note that the z-transform bears a form similar to Laplace transform, in which the integration is replaced by summation now. The z-transform of f is denoted by Zf or $F(z)$.

The inverse transform is given by

$$f(k) = \frac{1}{2\pi j}\oint F(z)z^{k-1}dz \tag{2.24}$$

where the contour of integration encloses all singularities of $F(z)$. How we derive this inverse transform is not our concern here, and this inverse z-transform is rarely used in practise just as the case that the inverse Laplace transform is seldom used in the calculation of the time-domain signal. Instead, a more convenient way is simply by looking up the transform table.

Example 2.3 Transform of a ramp

Consider a ramp signal defined by $y(k) = k$ for $k \geq 0$. Then

$$Y(z) = 0 + z^{-1} + 2z^{-2} + ... = z^{-1} + 2z^{-2} + ...) = \frac{z}{(z-1)^2}$$

The most important property of z-transform is,

$$Z\{x(k+1)\} = zX(z) - zx(0).$$

Compare with the forward shift operator $q\{x(k)\} = x(k+1)$, we have following correspondence with zero initial condition,

$$Z - - - q$$

just as the correspondence in the continuous-time case:

$$s - - - d/dt$$

Other properties of the z-transform are collected in Table 2.3.

Table 2.3 Some properties of the z-transform.

1. Definition.
   $F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}$
2. Inversion.
   $f(k) = \frac{1}{2\pi j}\oint F(z)z^{k-1}dz$

3. Linearity.

   $Z\{\alpha f + \beta g\} = \alpha Z f + \beta Z g$

4. Time shift.

   $Z\{q^{-n} f\} = z^{-n} F$

   $Z\{q^n f\} = z^n (F - F_1)$ where $F_1(z) = \displaystyle\sum_{j=0}^{n-1} f(j) z^{-j}$

5. Initial-value theorem.

   $f(0) = \lim_{z \to \infty} F(z)$

6. Final-value theorem.

   If $(1 - z^{-1})F(z)$ does not have any poles on or outside the unit circle, then

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

7. Convolution.

   $Z\{f * g\} = Z\left\{\sum_{n=0}^{k} f(n) g(k-n)\right\} = (Zf)(Zg)$

Shift-Operator Calculus and z-transforms

There are strong formal relations between shift-operator calculus and calculations with z-transforms. When manipulating difference equations we can use either. The expressions obtained look formally very similar. In many textbooks the same notation is in fact used for both. The situation is very similar to the difference between the differential operator $p = d/dt$ and the Laplace transform s for continuous-time systems.

The z-transform can be used to solve difference equations; for instance,

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= C x(k) + D u(k) \end{aligned}$$

If the z-transform of both sides is taken,

$$\begin{aligned} z(X(z) - x(0)) &= \Phi X(z) + \Gamma U(z) \\ X(z) &= (zI - \Phi)^{-1} (z x(0) + \Gamma U(z)) \end{aligned}$$

and

$$Y(z) = C(zI - \Phi)^{-1} z x(0) + (C(zI - \Phi)^{-1} \Gamma + D) U(z)$$

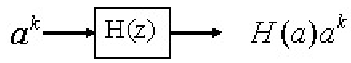


Figure 2.7: Steady state response of system $H(z)$ to input a^k

The transfer function can now be introduced.

$$H(z) = C(zI - \Phi)^{-1}\Gamma + D \tag{2.25}$$

The time sequence $y(k)$ can now be obtained using the partial fraction expansion and looking up the z-transform table as shown in table 2.2.

Just as in the continuous-time case, the transfer function is an alternative and convenient mathematical description of the system. Given any difference equation, the transfer function can be easily obtained by replacing the time shift operator "q" with "z". And given any transfer function, the corresponding difference equation can be found by simply replacing "z" with "q". For instance, The transfer functions of the systems listed in table 2.1. can be obtained by simply replacing "q" with "z".

It can be easily verified that $H(a)a^k$ is the steady state response of the system $H(z)$ subject to input a^k as shown in Figure 2.7.

In the special case of unit step input ($a = 1$), the response is simply $H(1)$, which is the steady state gain, or, the static gain. Note that in the continuous-time case, the static gain is given by $H(0)$, which is different from that of the discrete-time system.

2.6 Frequency Response 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. 2.8.

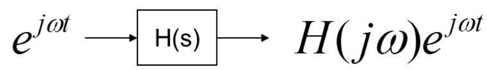


Figure 2.8: Block Diagram of Frequency Response for Continuous System.

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. 2.9.

The frequency response can be easily derived using the knowledge that the steady state response of the system, $H(z)$, to the input signal, a^k, is $H(a)a^k$. Therefore, consider the special case of sinusoid input $e^{j\omega Tk}$, where $a = e^{j\omega T}$, the response is simply

$$H(a)a^k = H(e^{j\omega T})e^{j\omega Tk}$$

We are going to use this knowledge when we deal with disturbances later in this module.

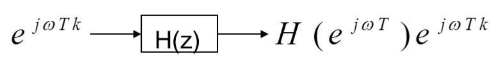


Figure 2.9: Block Diagram of Frequency Response for Discrete System.

2.7 Transfer Function from Continuous Time to Discrete Systems

The transfer function can be obtained from the state space model using the formula (2.25). It can also be determined directly from the continuous-time transfer function, $G(s)$. We know that the transfer function is the impulse response. Therefore what we need to do is just to figure out the impulse response for the sampled system.

The unit impulse signal in discrete-time is $\delta(k)$. Due to zero-order hold, the corresponding input signal in the continuous time is just a square pulse signal as follows,

$$u(t) = \begin{cases} 1, t \in [0, h] \\ 0, elsewhere \end{cases}$$

which can easily expressed as combination of two step input signals as

$$u(t) = U(t) - U(t-h)$$

The step response in continuous-time is simply $\frac{G(s)}{s}$ (where $G(s)$ is the transfer function of the continuous-time process.) in s-domain. In time domain, the step response can be obtained by inverse Laplace Transform as

$$L^{-1}(\frac{G(s)}{s})$$

We need to get the step response in z-domain by z-transform as

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

Therefore, the formula to convert continuous-time TF to discrete-time TF is

$$H(z) = Z[L^{-1}(\frac{G(s)}{s})] - z^{-1}Z[L^{-1}(\frac{G(s)}{s})] = (1-z^{-1})Z[L^{-1}(\frac{G(s)}{s})] \quad (2.26)$$

The table 2.1 lists the transformation formulae for the commonly used transfer functions.

2.8 Poles and Zeros

For single-input-single-output finite-dimensional systems, poles and zeros can be conveniently obtained from the denominator and numerator of the transfer function. Poles and zeros have good system-theoretic interpretation. A pole $z = a$ corresponds to a free mode of the system associated with the time function $z(k) = a^k$ (corresponding to the component $\frac{1}{z-a}$ in the transfer function). Poles are also the eigenvalues of the system matrix Φ.

The zeros are related to how the inputs and outputs are coupled to the states. Zeros can also be characterized by their signal blocking properties. A zero $z = a$ means that the transmission of the input signal $u(k) = a^k$ is blocked by the system. In other words, the system does not respond to the input signal a^k at all if a is one of the zeros of the system.

Poles

As discussed in the first lecture, the stability concerns the behavior of the system under initial deviations from the equilibrium point (zero states), and can be examined from the impulse response. For the discrete-time system, the unit impulse signal is $\delta(k)$, where $\delta(0) = 1$, and $\delta(k) = 0$, for $k \geq 1$. The corresponding z-transform is 1, hence the unit impulse response of the system is $Y(z) = H(z)U(z) = H(z)$. A pole $z = a = \rho e^{j\theta}$ corresponds to a component of the impulse response of the system associated with the time function $a^k = \rho^k e^{j\theta k}$. Obviously, if the magnitude of the pole, $|a| = \rho > 1$, this component will blow up, and the system is unstable. Otherwise, the system will be stable.

When would the output be oscillatory? In continuous-time, we know that the system would exhibit oscillation if the imaginary part of the pole is non-zero. Similarly, since $a^k = \rho^k e^{j\theta k}$, the oscillation is definitely related to the imaginary part. However, there is an extra twist here. Even if the pole is purely negative real, the resulting output is also oscillatory! For instance, if the pole is -1, then the corresponding free mode is {1, -1,1,-1, ...}.

Remark: Please note the difference between the stability criterion of the discrete-time and continuous-time systems. In the continuous-time system, since the corresponding component function is $e^{at} = e^{\sigma t} e^{j\omega t}$ corresponding to the pole of $a = \sigma + \omega j$, the stability is determined by the real part of the pole, σ. While in the discrete-time, the stability is decided by the magnitude of the poles as shown above.

In the following we will study how the poles of the continuous-time systems are transferred into poles of the discrete-time system under zero-order held. Consider a continuous-time system described by the nth-order state-space model

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.27}$$

The poles of the system are the eigenvalues of A, which we denote by $\lambda_i(A), i = 1, \ \ldots \ n$. The zero-order-hold sampling of (2.27) gives the discrete-time system

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) \end{aligned}$$

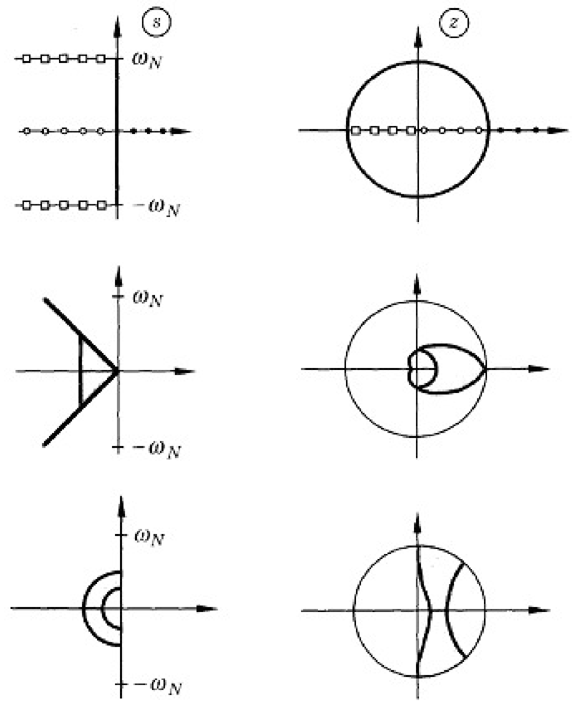


Figure 2.10: The conformal map $z = exp(sh)$.

Its poles are the eigenvalues of $\Phi, \lambda_i(\Phi), i = 1...n$. Because $\Phi = exp(Ah)$ it follows (please try to verify by yourself) that

$$\lambda_i(\Phi) = e^{\lambda_i(A)h} \tag{2.28}$$

Eq (2.28) gives the mapping from the continuous-time poles to the discrete-time poles. Fig. 2.10 illustrates a mapping of the complex s-plane into the z-plane, under the map

$$z = e^{sh}$$

For instance, the left half of the s-plane is mapped into the unit disc of the z-plane. The map is not bijective (one to one) — several points in the s-plane are mapped into the same point in the z-plane (see Fig. 2.10).
Remark: please note that the map $z = e^{sh}$ describes the relationship between the poles of the continuous-time system and its sampled system. It is NOT the map between the transfer function of the continuous-time system and its counter-part in the discrete-time!

Example 2.4

Consider the continuous-time system

$$\frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{2.29}$$

The poles of the corresponding discrete-time system are given by the characteristic equation by looking up Table 2.1.

$$z^2 + a_1 z + a_2 = 0$$

where

$$\begin{aligned} a_1 &= -2e^{-\zeta\omega_0 h}\cos(\sqrt{1-\zeta^2}\omega_0 h) \\ a_2 &= e^{-2\zeta\omega_0 h} \end{aligned}$$

It can be easily verified that the poles of the continuous-time system is

$$s = -\zeta\omega_0 \pm j\sqrt{1-\zeta^2}\omega_0$$

and the poles of the discrete-time system can be found by applying the map $z = e^{sh}$,

$$z = e^{-\zeta\omega_0 h \pm j\sqrt{1-\zeta^2}\omega_0 h}$$

Zeros

Can we also find a straightforward map between the zeros of the continuous-time and discrete-time systems? The answer is unfortunately NO. It is not possible to give a simple formula for the mapping of zeros. The following example shows that a continuous time system without any zeros may introduce zeros to the corresponding discrete-time system by sampling.

Example 2.5 Second-order system

Consider the continuous-time transfer function

$$\frac{2}{(s+1)(s+2)}$$

Using Table 2.1 gives the zero of the transfer function

$$z = -\frac{(1-e^{-2h})e^{-h} - 2(1-e^{-h})e^{-2h}}{2(1-e^{-h}) - (1-e^{-2h})}$$

Table 2.4 Numerator polynomials, Z_d, when sampling the continuous-time system $\frac{1}{s^d}$ with sampling period $h \to 0$.

d	Z_d
1	1
2	$z + 1$
3	$z^2 + 4z + 1$
4	$z^3 + 11z^2 + 11z + 1$
5	$z^4 + 26z^3 + 66z^2 + 26z + 1$

Definition 2.3 Systems with stable and unstable inverses

A discrete-time system has a stable inverse if all the zeros are stable (inside the unit disc, or on the unit disc for single ones). Otherwise, the system has an unstable inverse.

We will show later in this course that the condition of stable inverse plays a vital role in designing perfect tracking system.

A continuous-time with stable inverse may become a discrete-time system with unstable inverse when it is sampled. For example, it follows from Table 2.4 that the sampled system always has an unstable inverse if the pole excess of the continuous- time system is larger than 2, and if the sampling period is sufficiently short.

Further, a continuous-time system with unstable inverse will not always become a discrete-time system with unstable inverse, as shown in the following example.

Example 2.6 stable and unstable inverse changes with sampling

The transfer function

$$G(s) = \frac{6(1-s)}{(s+2)(s+3)}$$

has an unstable zero $s = 1$, and it has an unstable inverse. Sampling the system gives a discrete-time transfer function with a zero:

$$z_1 = -\frac{8e^{-2h} - 9e^{-3h} + e^{-5h}}{1 - 9e^{-2h} + 8e^{-3h}}$$

For $h \approx 1.25, z_1 = -1$; for larger h, the zero is always inside the unit circle and the sampled system has a stable inverse.

2.9 Notes and References

Most of the materials covered in Chapters 2 to 5, are extracted from the main text book of this module:

- K. J. Astrom and B. Wittenmark, *Computer-controlled Systems, Theory and Design*, Prentice Hall, 1997.

Chapter 3

Analysis of Discrete - Time Systems

3.1 Introduction

Chapter Two has shown how continuous-time systems are transformed into discrete-time systems when sampled. Z-transform has also been introduced to deal with the discrete-time system, which can easily transform the difference equations into algebraic equations. And transfer function can be readily derived using z-transform. In this lecture we will continue to develop the key tools for analyzing discrete-time systems. Stability is introduced first in Sec 3.2. The concepts of controllability, observability and realizability, which are useful for understanding discrete-time systems, are discussed in Sec 3.3. Simple feedback loops and their properties are treated in Sec 3.4.

3.2 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.

3.2.1 Definitions

In the previous lectures, we have touched upon the concept of stability several times without giving any precise definition. Now it is time to give a mathematical definition for stability.
There are many types of stability defined in the literature. It is usually defined with respect to changes in the initial conditions. Consider the discrete-time state-space equation (possibly nonlinear and time-varying)

$$x(k+1) = f(x(k), k) \tag{3.1}$$

Let $x_o(k)$ and $x(k)$ be solutions of (3.1) when the initial conditions are $x_o(0)$ and $x(0)$, respectively. Further, let denote $\|.\|$ a vector norm.

Definition 3.1 Stability

The solution $x_o(k)$ of (3.1) is stable if for any given $\varepsilon > 0$, there exists a $\delta(\varepsilon) > 0$, such that all solutions with $||x(0) - x_o(0)|| < \delta$ are such that $||x(k) - x_o(k)|| < \varepsilon$ for all $k \geq 0$.

Definition 3.2 Asymptotic Stability

The solution $x_o(k)$ of (3.1) is asymptotically stable if it is stable and if δ can be chosen such that $||x(0) - x_o(0)|| < \delta$ implies that $||x(k) - x_o(k)|| \to 0$ when $k \to \infty$.

From the definitions, it follows that stability in general is defined for a particular solution and not for the system. The definitions also imply that stability, in general, is a local concept. When we talk about stability, it normally refers to stability around certain solutions such as set points and equilibrium positions etc. Sometimes, the reference solution can also be time-varying function. The interpretation of Definitions 3.1 and 3.2 is that the system is (asymptotically) stable if the trajectories do not change much if the initial condition is changed by a small amount, which is consistent with our previous discussions.
In the following, we are going to apply this definition to linear systems, which leads to something very special about the stability of linear systems. It is no longer a local concept! It is global for linear systems!

3.2.2 Stability of Linear Discrete-Time Systems

Consider the linear system

$$\begin{aligned} x_o(k+1) &= \Phi x_o(k) \\ x_o(0) &= a_o \end{aligned} \tag{3.2}$$

To investigate the stability of the solution of (3.2), $x_o(k)$, the initial value is perturbed. Hence

$$\begin{aligned} x(k+1) &= \Phi x(k) \\ x(0) &= a \end{aligned}$$

The deviation of the solutions, $\tilde{x} = x - x_o$, satisfies the equation

$$\begin{aligned} \tilde{x}(k+1) &= \Phi \tilde{x}(k) \\ \tilde{x}(0) &= a - a_o \end{aligned} \tag{3.3}$$

Because the form of above equation (3.3) is independent of the reference solution $x_o(k)$, it implies that if the solution $x_o(k)$ is stable, then every other solution is also stable. For linear, time-invariant systems, stability is thus a property of the system and not of a special solution.

The system (3.3) has the solution

$$\tilde{x}(k) = \Phi^k \tilde{x}(0)$$

If it is possible to diagonalize Φ, then obviously the solution is a linear combination of terms λ_i^k, where $\lambda_i, i = 1, \ \ldots \ , n$ are the eigenvalues of Φ. In the general case, when Φ cannot be diagonalized, the solution is instead a linear combination of the terms λ_i^k, $k\lambda_i^k$,...,$k^{m-1}\lambda_i^k$, where m is the multiplicity of the corresponding eigenvalue λ_i (i.e, the eigenvalue is repeated roots of the characteristic equation). If you are interested in the theoretical proof of this, you may try to use the knowledge from Linear algebra regarding the Jordan forms of matrices. For the time being, we will simply accept this and move on. To get asymptotic stability, the deviations $\tilde{x}(k)$ must go to zero as k increases to infinity. The eigenvalues of Φ then have the property

$$|\lambda_i| < 1, i = 1, \ \ldots \ , n$$

If one of the eigenvalues $|\lambda_i| > 1$, then the term λ_i^k will blow up, and the system is unstable.

What would happen if some of the eigenvalues lie on the unit disc, i.e. $|\lambda_i| = 1$? Things will be more complicated and it depends upon whether the corresponding eigenvalue is simple or multiple. If the eigenvalue is simple, then the corresponding component solution is λ_i^k with a magnitude of one. In this case, the system is still stable (verify it by yourself!) , but not asymptotic stable as it does not converge to zero. If the eigenvalue is multiple (repeated m times) , there exist components like $k\lambda_i^k$,...,$k^{m-1}\lambda_i^k$, which will increase to infinity.

Another way of expressing the solution of $\tilde{x}(k) = \Phi^k \tilde{x}(0)$ is using Z-transform. It can be easily shown that the z-transform of Φ^k is $z(zI - \Phi)^{-1}$. Then by decomposing the solution through partial fraction expansion, we can reach the same conclusion regarding the relationship between poles (eigenvalues) and the stability as discussed above.

All the above discussions might be summarized into the following theorem.

Theorem 3.1 Stability of Linear Systems

Discrete-time linear time-invariant system (3.2) is stable if all eigenvalues of Φ, i.e. poles, are either inside or on the unit disc, and all the eigenvalues with unit magnitude are simple (not repeated poles). The system is asymptotically stable if all eigenvalues of Φ are strictly inside the unit disk.

It is noted that input signal is assumed to be zero or a constant in above discussions on stability. As mentioned earlier, there are many types of stability. Stability with respect to disturbances in the initial value has already been defined. Sometimes, we might be keen in knowing whether the inputs and outputs are bounded or not.

3.2.3 Input-Output Stability

Definition 3.3 Bounded-Input Bounded-Output Stability

A linear time-invariant system is defined as bounded-input-bounded-output (BIBO) stable if a bounded input gives a bounded output for every initial value.

Since BIBO stability is not defined with respect to any particular reference solution, it is not a local concept. From the definition it follows that asymptotic stability is the strongest concept. The following theorem is a result.

Theorem 3.2 Relation between Stability Concepts

Asymptotic stability implies stability and BIBO stability for linear systems.

When the word stable is used without further qualification in this module, it normally means asymptotic stability.

It is noted that for nonlinear systems, asymptotic stability might not imply BIBO stability. For instance, consider the following example,

$$x(k+1) = \frac{0.5x(k)}{(1-u(k))}$$

Obviously the system is asymptotic stable when the input is zero. But it is unstable if input u(k)=0.8. Therefore the system is not BIBO stable.

It is easy to give examples showing that stability does not imply BIBO stability even for linear systems, and vice versa.

Example 3.1 Harmonic oscillator

Consider the sampled harmonic oscillator

$$x(k+1) = \begin{pmatrix} \cos\omega h & \sin\omega h \\ -\sin\omega h & \cos\omega h \end{pmatrix} x(k) + \begin{pmatrix} 1 - cos\omega h \\ \sin\omega h \end{pmatrix} u(k)$$

$$y(k) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(k)$$

The magnitude of the eigenvalues is one. The system is stable because $\|x(k)\| = \|x(0)\|$ if $u(k) = 0$, although not asymptotic stable. Let the input be a square wave with the frequency ω rad/s. By using the z-transform, it is easily seen that the output contains a sinusoidal function with growing amplitude and the system is not BIBO stable. Fig. 3.1 shows the input and output of the system. The input signal is exciting the system at its undamped frequency and the output amplitude is growing, which is the well-known phenomenon of resonant oscillation. Similar phenomenon exists for the continuous -time system. What would happen if feeding a sinusoid to the undamped system where the frequency of the input signal matches the natural frequency? The system becomes unstable!

There are also simple examples that the system is BIBO stable, but not stable. A good example is a ball rolling in the surface shown in Fig. 3.2.

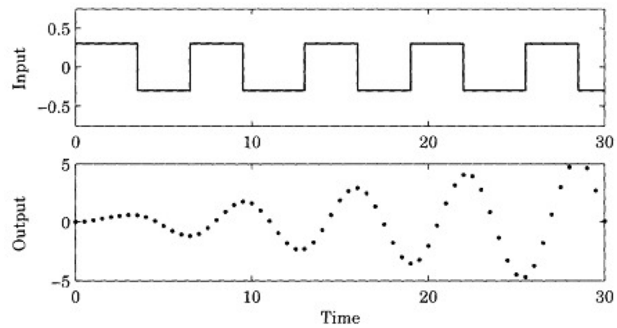


Figure 3.1: Input and output of the system in Example 3.1 when $\omega = 1$, $h = 0.5$, and the initial state is zero.

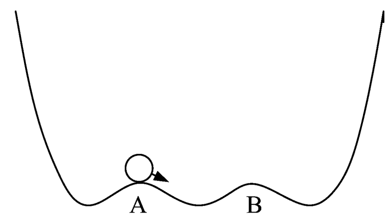


Figure 3.2: A system is BIBO stable, but not stable.

Obviously, the ball is unstable around the point A and B, but overall, it is BIBO stable.
Since above example is a nonlinear system, you may still wonder whether BIBO stability might imply stability for linear system. Consider the following example

$$
\begin{aligned}
x(k+1) &= \begin{pmatrix} 0.5 & 0 \\ 1 & 2 \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(k) \\
y(k) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(k)
\end{aligned}
$$

It can be easily verified that the system is unstable since it has a pole at 2. But it is BIBO stable since the observation of the output y(k) cannot reveal the dynamics of the second state variable $x_2(k)$. Such kind of system is unobservable (the information of the state cannot be recovered by observing the inputs and outputs), which will be discussed in details later in this module.

3.2.4 Stability Tests

It follows from Theorem 3.1 that a straightforward way to test the stability of a given system is to calculate the eigenvalues of the matrix Φ. There are good numerical algorithms for doing this. The routines are also included in packages like MATLAB®. The eigenvalues of a matrix then can be calculated with a single command (eig in MATLAB).

It is, however, also important to have algebraic or graphical methods for investigating stability. These methods make it possible to understand how parameters in the system or the controller will influence the stability. The following are some of the ways of determining the stability of a discrete-time system:

- Direct numerical or algebraic computation of the eigenvalues of Φ

- Methods based on properties of characteristic polynomials
- The root locus method
- The Nyquist criterion
- Lyapunov's method

Explicit calculation of the eigenvalues of a matrix cannot be done conveniently by hand for systems of order higher than 2. In some cases it is easy to calculate the characteristic equation

$$A(z) = a_0 z^n + a_1 z^{n-1} + \ \dots \ + a_n = 0 \tag{3.4}$$

and investigate its roots. Stability tests can be obtained by investigating conditions for the zeros of a polynomial to be inside the unit disc.

It is also useful to have algebraic or graphical conditions that tell directly if a polynomial has all its zeros inside the unit disc. Such a criterion, which is the equivalent of the ***Routh-Hurwitz criterion*** for the continuous-time case, was developed by Schur, Cohn, and Jury. This test will be described in detail in the following section.

3.2.5 Jury's Stability Criterion

The following test is useful for determining if (3.4) has all its zeros inside the unit disc. Form the table

where

$$a_i^{k-1} = a_i^k - \alpha_k a_{k-i}^k$$
$$\alpha_k = a_k^k / a_0^k$$

The first and second rows are the coefficients in Eq (3.4) in forward and reverse order, respectively. The third row is obtained by multiplying the second row by $\alpha_n = a_n/a_0$ and subtracting this from the first row. The last element in the third row is thus zero. The fourth row is the third row in reverse order. The scheme is then repeated until there are $2n+1$ rows. The last row consists of only one element. The following theorem results.

Theorem 3.3 Jury's Stability Test

If $a_0 > 0$, then Eq (3.4) has all roots inside the unit disc if and only if all $a_0^k, k = 0, 1, \ \dots \ , n-1$ are positive. If no a_0^k is zero, then the number of negative a_0^k is equal to the number of roots outside the unit disc.

$$
\begin{array}{ccccc|l}
a_0 & a_1 & \cdots & a_{n-1} & a_n & \\
a_n & a_{n-1} & \cdots & a_1 & a_0 & \alpha_n = \dfrac{a_n}{a_0} \\
\hline
a_0^{n-1} & a_1^{n-1} & \cdots & a_{n-1}^{n-1} & & \\
a_{n-1}^{n-1} & a_{n-2}^{n-1} & \cdots & a_0^{n-1} & & \alpha_{n-1} = \dfrac{a_{n-1}^{n-1}}{a_0^{n-1}} \\
\hline
\vdots & & & & & \\
a_0^0 & & & & &
\end{array}
$$

Example 3.2 Stability of a second-order system

Let the characteristic equation be

$$A(a) = z^2 + a_1 z + a_2 = 0 \tag{3.5}$$

Jury's scheme is
All the roots of Eq (3.5) are inside the unit circle if

$$
\begin{aligned}
&1 - a_2^2 > 0 \\
&\frac{1-a_2}{1+a_2}\left((1+a_2)^2 - a_1^2\right) > 0
\end{aligned}
$$

This gives the conditions

$$
\begin{cases}
a_2 < 1 \\
a_2 > -1 + a_1 \\
a_2 > -1 - a_1
\end{cases}
$$

The stability area for the second-order equation is a triangle shown in Fig. 3.3.

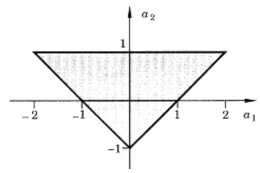


Figure 3.3: The stability area for the second-order Eq (3.5) of the coefficients a_1 and a_2 .

3.3 Controllability, Observability and Realizability

In this section, two fundamental questions for dynamic systems are discussed first. The first is whether it is possible to steer a system from a given initial state to any other state. The second is how to determine the state of a dynamic system from observations of inputs and outputs. Toward the end of the section, we will also discuss how to construct a state space model for a system whose behavior is described by input-output model.

3.3.1 Controllability

Definition 3.4 Controllability

The system is controllable if it is possible to find a control sequence such that an arbitrary state can be reached from any initial state in finite time. For nonlinear system, it is difficult to check whether the system is control-

$$
\begin{array}{ccc|c}
1 & a_1 & a_2 & \\
a_2 & a_1 & 1 & \alpha_2 = a_2 \\
\hline
1 - a_2^2 & a_1(1 - a_2) & & \\
a_1(1 - a_2) & 1 - a_2^2 & & \alpha_1 = \dfrac{a_1}{1 + a_2} \\
\hline
1 - a_2^2 - \dfrac{a_1^2(1 - a_2)}{1 + a_2} & & &
\end{array}
$$

lable or not. But for linear system, we have a neat result.

Consider the system

$$
\begin{aligned}
x(k+1) &= \Phi x(k) + \Gamma u(k) \\
y(k) &= C x(k)
\end{aligned}
\tag{3.6}
$$

Assume that the initial state $x(0)$ is given. The state at time n, where n is the order of the system, is given by

$$
\begin{aligned}
x(n) &= \Phi^n x(0) + \Phi^{n-1}\Gamma u(0) + ... + \Gamma u(n-1) \\
&= \Phi^n x(0) + W_c U
\end{aligned}
\tag{3.7}
$$

where

$$
\begin{aligned}
W_c &= \begin{pmatrix} \Gamma & \Phi\Gamma & \cdots & \Phi^{n-1}\Gamma \end{pmatrix} \\
U &= \begin{pmatrix} u(n-1) & \cdots & u(0) \end{pmatrix}^T
\end{aligned}
$$

Now the question regarding controllability is following: starting from arbitrary initial state x(0) is it possible to find control sequence U such that any final state x(n) can be reached?

If W_c has rank n, then it is possible to obtain n equations from which the control signals can be found such that the initial state is transferred to the desired final state $x(n)$.

But what would happen if W_c is not of full rank? Can we apply more inputs to drive the state into the target state? If we try it in one more step by applying input u(n), then we have

$$\begin{aligned} x(n+1) &= \Phi^{n+1}x(0) + \Phi^n\Gamma u(0) + \Phi^{n-1}\Gamma u(1) ... + \Gamma u(n-1) \\ &= \Phi^{n+1}x(0) + WU \end{aligned}$$

where

$$\begin{aligned} W &= \begin{pmatrix} \Gamma & \Phi\Gamma & \cdots & \Phi^n\Gamma \end{pmatrix} \\ U &= \begin{pmatrix} u(n) & \cdots & u(0) \end{pmatrix}^T \end{aligned}$$

If the rank of W is of full rank, then it is possible to find out the control sequence. Unfortunately, the rank of W will be the same as that of W_c! This fact can be shown by the following famous ***Cayley-Hamilton theorem***.

Cayley-Hamilton theorem

Given any $n \times n$ matrix A, and its characteristic equation

$$det[zI - A] = z^n + a_1 z^{n-1} + \ldots + a_n$$

Then

$$A^n + a_1 A^{n-1} + \ldots + a_n I = 0$$

In other words, A^n is a linear combination of $A^{n-1}, A^{n-2}, \ldots, A, I$.
Applying ***Cayley-Hamilton theorem*** to our problem, it follows that $\Phi^n\Gamma$ is a linear combination of $\Phi^{n-1}\Gamma, \Phi^{n-2}\Gamma, \ldots, \Gamma$. Hence the rank of W cannot be increased by applying more inputs!
The following theorem follows from the preceding definition and calculations.

Theorem 3.4 Controllability

The system (3.6) is controllable if and only if the matrix W_c has rank n.
Remark. The matrix W_c is referred to as the controllability matrix.

3.3.2 Controllable Canonical Form

Assume that Φ has the characteristic equation

$$\det(\lambda I - \Phi) = \lambda^n + a_1\lambda^{n-1} + ... + a_n = 0 \tag{3.8}$$

and that W_c, is nonsingular. Then there exists a transformation, T, such that the transformed system, $z(k) = Tx(k)$, is described by

$$z(k+1) = \begin{pmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} z(k) + \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u(k) \tag{3.9}$$

$$y(k) = (b_1 \; ... \; b_n)z(k)$$

which is called the controllable canonical form. There are a number of advantages in expressing the model in this form. First, it is easy to compute the input-output model. It can be verified directly by z-transform that the corresponding input-output model, i.e. transfer function, is determined by the first row of elements in $\tilde{\Phi}$ and $\tilde{C}$ as follows.

$$H(z) = \frac{b_1 z^{n-1} + ... + b_n}{z^n + a_1 z^{n-1} + ... + a_n}$$

The second advantage is that the state feedback-control law can be easily derived from controllable canonical form. We will elaborate this point later when we discuss the control system design in the future.

But how to find out the transformation T, such that it can transform any controllable system into its canonical form?

Assume that new coordinates are introduced by a nonsingular transformation matrix T

$$z(k) = Tx(k) \tag{3.10}$$

then

$$z(k+1) = Tx(k+1) = T(\Phi x(k) + \Gamma u(k)) = T\Phi T^{-1}z(k) + T\Gamma u(k) \tag{3.11}$$

In the new coordinates,

$$
\begin{aligned}
\tilde{W}_C &= \begin{pmatrix} \tilde{\Gamma} & \tilde{\Phi}\tilde{\Gamma} & \cdots & \tilde{\Phi}^{n-1}\tilde{\Gamma} \end{pmatrix} \\
&= \begin{pmatrix} T\Gamma & T\Phi T^{-1}T\Gamma & \cdots & T\Phi^{n-1}T^{-1}T\Gamma \end{pmatrix} \\
&= TW_c
\end{aligned}
\tag{3.12}
$$

For a single-input system the transformation matrix to the controllable canonical form is

$$T = \tilde{W}_c W_c^{-1}$$

where $\tilde{W}_c$ is the controllability matrix for the representation (3.9).

3.3.3 Trajectory Following

From the preceding definitions and calculations, it is possible to determine a control sequence such that a desired state can be reached after at most n steps of time. Does controllability also imply that it is possible to follow a given trajectory in the state space at every step? The answer is usually no. Note that for a single-input-single-output system it is, in general, possible to reach desired states only at each n-th sample point (not every step!), provided that the desired points are known n steps ahead.

It is easier to make the output follow a given trajectory. Assume that the desired trajectory is given by $u_c(k)$.
A naive approach would be the following. The control signal u then should satisfy

$$y(k) = \frac{B(q)}{A(q)} u(k) = u_c(k)$$

or

$$u(k) = \frac{A(q)}{B(q)} u_c(k) \tag{3.13}$$

Assume that there are d steps of delay in the system. The generation of $u(k)$ is then causal only if the desired trajectory is known d steps ahead. The control signal then can be generated in real time. The control signal thus is obtained by sending the desired output trajectory through the inverse system A/B. The signal u is bounded if u_c is bounded and if the system has a stable inverse, i.e., $B(z)$ is stable. Now you should realize that the condition of stable inverse is important in tracking. But does $y(k)$ really track the command signal $u_c(k)$? We have to check out the transfer function

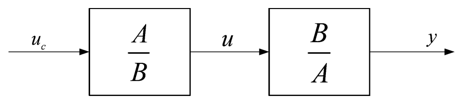


Figure 3.4: Trajectory Following

from the command signal to the output. It can be shown from the open loop transfer function and the inverse controller (3.13), that

$$y(k) = \frac{A(q)}{A(q)} u_c(k)$$

Therefore we can conclude that another important condition for perfect tracking is that the original system has to be stable, in which case the poles and zeros can be perfectly canceled out.

Overall, if the system and its inverse are stable, it is possible to use very simple open-loop control to follow any desired trajectory.

3.3.4 Observability

The concept of observability is regarding whether the state information can be extracted from the observations of the inputs and output. This is an important issue since only the inputs and outputs are measurable in many practical situations. The system in (3.6) is observable if there is a finite k such that knowledge of the inputs $u(0), \ \dots \ , u(k-1)$ and the outputs $y(0), \ \dots \ , y(k-1)$ is sufficient to determine the initial state of the system. Once the initial state information is obtained, the states at every step can then be calculated.

Consider the system in (3.6). The effect of the known input signal always can be determined, and there is no loss of generality to assume that $u(k) = 0$. Assume that $y(0), y(l), \ \dots \ , y(n-1)$ are given. This gives the following set of equations:

$$
\begin{aligned}
y(0) &= Cx(0) \\
y(1) &= Cx(1) = C\Phi x(0) \\
&\vdots \\
y(n-1) &= C\Phi^{n-1}x(0)
\end{aligned}
$$

Using vector notation gives

$$
\begin{pmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{pmatrix} x(0) = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{pmatrix} \tag{3.14}
$$

The state $x(0)$ can be obtained from (3.14) if and only if the observability matrix

$$
W_o = \begin{pmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{pmatrix} \tag{3.15}
$$

is of full rank. The state $x(0)$ is unobservable if $x(0)$ is in the null space of W_o. If two states are unobservable, then any linear combination is also unobservable; that is, the unobservable states form a linear subspace.

Theorem 3.5 Observability

The system (3.6) is observable if and only if W_o is of full rank.

Definition 3.5 Detectability

A system is detectable if the only unobservable states are such that they decay to the origin. That is, the corresponding eigenvalues are stable.
Question: consider the example given at the end of section 3.2.3, which shows that BIBO stability might not imply stability. Is this system detectable?
In general, if the linear system is observable or detectable, then BIBO stability implies stability. If the system is undetectable, then the unobservable state variables will blow up, which will not be revealed by the observation of the inputs and outputs. Hence BIBO stability is not sufficient to guarantee the stability of the system.

Example 3.3 A system with unobservable states

Consider the system

$$x(k+1) = \begin{pmatrix} 1.1 & -0.3 \\ 1 & 0 \end{pmatrix} x(k)$$
$$y(k) = (1 \quad -0.5)x(k)$$

The observability matrix is

$$W_o = \begin{pmatrix} C \\ C\Phi \end{pmatrix} = \begin{pmatrix} 1 & -0.5 \\ 0.6 & -0.3 \end{pmatrix}$$

The rank of W_o is 1, and the unobservable states belong to the null space of W_o, that is, [0.5 1]. Fig. 3.5 shows the output for four different initial states. All initial states that lie on a line parallel to [0.5 1] give the same output (see Fig. 3.5 (b) and (d)). Therefore, the initial state cannot be determined uniquely by observing the inputs and outputs.

3.3.5 Observable Canonical Form

Similar to the controllable canonical form, there is also an observable canonical form, which would make the control system design more convenient. Assume that the characteristic equation of Φ is (3.8) and that the observability matrix W_o is nonsingular. Then there exists a transformation matrix such that

the transformed system is

$$z(k+1) = \begin{pmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} & 0 & 0 & \cdots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{pmatrix} z(k) + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} u(k) \tag{3.16}$$
$$y(k) = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} z(k)$$

which is called the observable canonical form. This form has the advantage that it is easy to find the input-output model and to determine a suitable observer. Following the similar argument as that in the section on controllability, the transformation in this case can be found as

$$T = \tilde{W}_o^{-1} W_o$$

where W_o is the observability matrix for the representation (3.16).
Remark. The observable and controllable canonical forms are also called companion forms.

3.3.6 Loss of Controllabilty and Observability Through Sampling

Sampling of a continuous-time system gives a discrete-time system with system matrices that depend on the sampling period. How will that influence the controllability and observability of the sampled system? To get controllable discrete-time system, it is necessary that the continuous-time system also be controllable, because the allowable control signals for the sampled system-piecewise — constant signals — are a subset of the allowable control signals for the continuous-time system.

However, it may happen that the controllability and/or observability is lost for some sampling periods. The harmonic oscillator can be used to illustrate the preceding discussion.

Example 3.4 Loss of controllability and observability

The discrete-time model of the harmonic oscillator is given by

$$x(k+1) = \begin{pmatrix} \cos\omega h & \sin\omega h \\ -\sin\omega h & \cos\omega h \end{pmatrix} x(k) + \begin{pmatrix} 1 - cos\omega h \\ \sin\omega h \end{pmatrix} u(k)$$

$$y(k) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(k)$$

The determinants of the controllability and observability matrices are

$$det\ \ W_c = -2\sin\omega h(1-\cos\omega h)$$

and

$$det\ \ W_o = \sin\omega h$$

Both controllability and observability are lost for $\omega h = n\pi$, although the corresponding continuous-time system is both controllable and observable.

The example shows one obvious way to lose observability and/or controllability. If the sampling period is half the period time (or a multiple thereof) of the natural frequency of the system, then this frequency will not be seen in the output. This happens only for isolated values of the sampling period. A change in sampling period will make the system observable and/or controllable again.

3.3.7 Realizability

The concepts of controllability and observability are defined only for state-space model of the system. An interesting question would arise: how to determine whether the system is controllable or observable if only the input-output model, i.e. transfer function, is available? In order to address this question, another important concept in control theory has to be introduced — the realizability.
Given a sequence of observations of inputs and outputs, what possible systems could give rise to it? In other words, if we start with an input-output model which describes how the inputs affect the outputs, how does one deduce the underlying state-space model that is responsible for this behavior?
Let's first start with the simplest case — the first order system. Assume the input-output model is

$$y(k+1) = -a_1 y(k) + b_1 u(k)$$

then the natural choice of the state variable is

$$x(k) = y(k)$$

which leads to the state-space model

$$x(k+1) = -a_1 x(k) + b_1 u(k)$$

This procedure is trivial. However, it has one important trick: one can always choose the output $y(k)$ as the state variable! Let's follow this idea to deal with the second-order system

$$y(k+1) = -a_1 y(k) - a_2 y(k-1) + b_1 u(k) + b_2 u(k-1)$$

If we still let one of the state be the output,

$$x_1(k) = y(k)$$

then the input-output model becomes

$$x_1(k+1) = -a_1 x_1(k) - a_2 y(k-1) + b_1 u(k) + b_2 u(k-1) = -a_1 x_1(k) + b_1 u(k) - a_2 y(k-1) + b_2 u(k-1)$$

Note that above equation contains terms in the past step k-1, $-a_2 y(k-1) + b_2 u(k-1)$, which are not supposed to appear in the right hand of the standard form of the state-space model like equation (3.6), which only consists of present state $x(k)$, and input $u(k)$. How to deal with the past terms? This is the most important step! Once you understand this trick, then realization problem would be trivial to you.

This trick is very simple: you just introduce another state variable which summarizes all the past information!

Let
$$x_2(k) = -a_2 y(k-1) + b_2 u(k-1)$$
then we have
$$\begin{aligned} x_1(k+1) &= -a_1 x_1(k) + b_1 u(k) + x_2(k) = -a_1 x_1(k) + x_2(k) + b_1 u(k) \\ x_2(k+1) &= -a_2 y(k) + b_2 u(k) = -a_2 x_1(k) + b_2 u(k) \end{aligned}$$
Now let's try above procedure for the third order system
$$y(k+1) = -a_1 y(k) - a_2 y(k-1) - a_3 y(k-2) + b_1 u(k) + b_2 u(k-1) + b_3 u(k-2)$$
First step: let
$$x_1(k) = y(k)$$
and obtain the first equation
$$x_1(k+1) = -a_1 x_1(k) + b_1 u(k) - a_2 y(k-1) - a_3 y(k-2) + b_2 u(k-1) + b_3 u(k-2)$$
Second step: introduce one state variable to summarize the past information
$$x_2(k) = -a_2 y(k-1) - a_3 y(k-2) + b_2 u(k-1) + b_3 u(k-2)$$
then we have obtained two equations
$$\begin{aligned} x_1(k+1) &= -a_1 x_1(k) + b_1 u(k) + x_2(k) = -a_1 x_1(k) + x_2(k) + b_1 u(k) \\ x_2(k+1) &= -a_2 x_1(k) + b_2 u(k) - a_3 y(k-1) + b_3 u(k-1) \end{aligned}$$
How to deal with the past information in the second equation? I hope everyone here in the class would give me the right answer: introduce another state variable!

$$x_3(k) = -a_3 y(k-1) + b_3 u(k-1)$$
That would be the end of the story. Overall, we have obtained the following state-space equation,
$$\begin{aligned} x_1(k+1) &= -a_1 x_1(k) + b_1 u(k) + x_2(k) = -a_1 x_1(k) + x_2(k) + b_1 u(k) \\ x_2(k+1) &= -a_2 x_1(k) + b_2 u(k) + x_3(k) = -a_2 x_1(k) + x_3(k) + b_2 u(k) \\ x_3(k+1) &= -a_3 x_1(k) + b_3 u(k) \end{aligned}$$

where the state variables are defined as combinations of the present and past outputs and inputs:

$$\begin{aligned}
x_1(k) &= y(k) \\
x_2(k) &= -a_2 y(k-1) - a_3 y(k-2) + b_2 u(k-1) + b_3 u(k-2) \\
x_3(k) &= -a_3 y(k-1) + b_3 u(k-1)
\end{aligned}$$

Note that the state variables are completely determined by the observations of the inputs and outputs. Therefore, the system must be observable. If you still have any doubt regarding this, let's put the above state-space model in the matrix form

$$\begin{aligned}
x(k+1) &= \begin{pmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} u(k) \\
y(k) &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x(k)
\end{aligned} \tag{3.17}$$

Is there anything special about this form? Does it look familiar to you? This is the observable canonical form!
Now it is time to state the following realizability result for realizability.

Theorem 3.6 Realizability

Given a system which is described by an input-output model

$$H(z) = \frac{b_1 z^{n-1} + \dots + b_n}{z^n + a_1 z^{n-1} + \dots + a_n} \tag{3.18}$$

If the system is observable, then the system can be realized by the observable canonical form (3.16), and the corresponding state variables are defined as

$$\begin{aligned}
x_1(k) &= y(k) \\
x_2(k) &= -a_2 y(k-1) + \dots - a_n y(k-n+1) + b_2 u(k-1) + \dots + b_n u(k-n+1) \\
x_3(k) &= -a_3 y(k-1) + \dots - a_n y(k-n+2) + b_3 u(k-1) + \dots + b_n u(k-n+2) \\
&\vdots \\
x_n(k) &= -a_n y(k-1) + b_n u(k-1)
\end{aligned}$$

Once one set of state-space model is deduced, other state-space model can be obtained by various linear transformations.

The following example shows that the controllability and observability sometimes cannot be decided by checking the input-output model only.

Example 3.5 Realization of uncontrollable and/or unobservable system

Consider the following input-output model

$$H(z) = \frac{z+1}{(z+1)(z+2)} = \frac{z+1}{z^2+3z+2} \tag{3.19}$$

If the system is observable, applying above realizability theorem results in following state-space model

$$\begin{aligned} x(k+1) &= \begin{pmatrix} -3 & 1 \\ -2 & 0 \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(k) \\ y(k) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(k) \end{aligned} \tag{3.20}$$

and the corresponding state variables are defined as

$$\begin{aligned} x_1(k) &= y(k) \\ x_2(k) &= -2y(k-1) + u(k-1) \end{aligned}$$

Now let's check if the system is controllable. The controllability matrix is

$$W_c = \begin{pmatrix} 1 & -2 \\ 1 & -2 \end{pmatrix}$$

Obviously, this system is not controllable as W_c is singular.

Notice that the above realization is under the assumption that the system is observable. You may wonder whether this system can be realized using the following state-space model in controllable canonical form

$$\begin{aligned} z(k+1) &= \begin{pmatrix} -3 & -2 \\ 1 & 0 \end{pmatrix} z(k) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(k) \\ y(k) &= \begin{pmatrix} 1 & 1 \end{pmatrix} z(k) \end{aligned} \tag{3.21}$$

You may also wonder whether this system is observable or not. Check it out by calculating W_o! It is unobservable! But is it really the realization of (3.19)?

In the following, we will try to show that the above controllable canonical form (3.21) is NOT the realization of the input-output model (3.19). It follows immediately from equation (3.21) by adding the two state equations that

$$z_1(k+1) + z_2(k+1) = -2(z_1(k) + z_2(k)) + u(k)$$

Since $y(k) = z_1(k) + z_2(k)$, it follows that

$$y(k+1) = -2y(k) + u(k)$$

and hence the corresponding transfer function is

$$G(z) = \frac{1}{z+2}$$

Therefore, the state-space model (3.21) is in fact the realization of the transfer function $\frac{1}{z+2}$ instead of $\frac{z+1}{(z+1)(z+2)}$. However, we also know that a more natural realization of the transfer function, $\frac{1}{z+2}$ is simply the first-order system

$$x(k+1) = -2x(k) + u(k)$$

Now we end up in an interesting situation where we have two realizations of the same input-output model $\frac{1}{z+2}$. But one system is a first order system which is both controllable and observable, and the other one is a second-order system which is controllable but unobservable. I guess everyone in this class is eager to know: which realization is correct?

The answer is we don't know based upon the transfer function only. The first order system is called minimal-realization of the system, in which the order of the state-space model is equal to the order of the input-output model. There are other infinite possible realizations of the system which are of higher-order. But those higher order realizations are all unobservable since the additional state variables cannot be determined by the observation of the outputs.

In general, given a system which is described by an input-output model without common poles and zeros, the minimal-realization of the system is both controllable and observable. If there are common poles and zeros, the minimal-realization of the system is observable but uncontrollable. The controllability and observability can not be determined by transfer function only as there are infinite number of non-minimal-realizations of the same transfer function.

3.4 Analysis of Simple Feedback Loops

Several advantages are obtained by using feedback in continuous-time as well as in discrete-time systems. Feedback, for instance, can do the following:

- Stabilize the system if the open-loop is unstable.
- Improve the transient behavior of the system
- Increase the robustness 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

3.4.1 Character of 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.

3.4.2 Simple Disturbance Models

There are four different types of disturbances-impulse, step, ramp, and sinusoid-that are commonly used in analyzing control systems. These disturbances are illustrated in Fig. 3.6 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. 3.6). 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. 3.6). 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.

3.4.3 Generation of disturbances

It is convenient to view disturbances as being generated by dynamic systems (see Fig. 3.7). It is assumed that the input to the dynamic system is a unit pulse δ_k that is,

$$u_c(k) = H_r(q)\delta_k \tag{3.22}$$

In order to generate a step, use $H_r(q) = q/(q-1)$; to generate a ramp, use $H_r(q) = q/(q-1)^2$, and a sinusoid from a harmonic oscillator. From an input-output viewpoint, disturbances may be described as impulse responses. Disturbances also may be regarded as the responses of dynamic systems with zero inputs but nonzero initial conditions. In both cases the major characteristics of the disturbances are described by the dynamic systems that generate them. The approach, of course, can be applied to continuous-time, as well as discrete-time systems.

3.4.4 Steady-State Values

When analyzing control systems, it is important to calculate steady-state values of the output and of the error of the system. To get the steady state value of the step response, it is very simple. Just put $z = 1$ of the transfer function involved, which was discussed in the previous lecture.

Example 3.5 Steady-state errors for step inputs

Consider the system described by

$$y(k) = H(z)u(k) = \frac{z-0.5}{(z-0.8)(z-1)}u(k)$$

Closing the system with unit feedback, as in Fig. 3.7, gives

$$y(k) = \frac{H(z)}{1+H(z)}u_c(k) = \frac{z-0.5}{(z-0.8)(z-1)+(z-0.5)}u_c(k)$$

and

$$e(k) = \frac{1}{1+H(z)}u_c(k) = \frac{(z-0.8)(z-1)}{(z-0.8)(z-1)+(z-0.5)}u_c(k)$$

Assume that u_c is a unit step, the steady state gain can be obtained simply by putting $z = 1$ in the corresponding transfer functions. Notice that the open-loop system contains one integrator, that is, a pole in $+1$, which makes $H(1) = \infty$. Therefore the steady state gain for the output is $\frac{H(1)}{1+H(1)} = 1$, and the steady state gain for the error is $\frac{1}{1+H(1)} = 0$, which means the steady state error of the step response is zero.

3.4.5 Simulation

Simulation is a good way to investigate the behavior of dynamic systems – for example, the intersample behavior of computer-controlled systems. Computer simulation is a very good tool, but it should be remembered that simulation and analysis have to be used together. When making simulations, it is not always possible to investigate all combinations that are unfavorable, for instance, from the point of view of stability, observability, or reachability. These cases can be found through analysis.

It is important that the simulation program be so simple to use that the person primarily interested in the results can be involved in the simulation and in the evaluation of the simulation results.

In the beginning of the 1960s, several digital simulation packages were developed. These packages were basically a digital implementation of analog simulation. The programming was done using block diagrams and fixed-operation modules. Later programs were developed in which the models were given directly as equations. It is important to have good user-machine communication for simulations; the user should be able to change parameters and modify the model easily. Most simulation programs are interactive, which means that the user interacts with the computer and decides the next step based on the results obtained so far. One way to implement interaction is to let the computer ask questions and the user select from predefined answers. This is called menu-driven interaction. Another possibility is command-driven interaction, which is like a high-level problem-solving language in which the user can choose freely from all commands available in the system. This is also a more flexible way of communicating with the computer, and it is very efficient for the experienced user, whereas a menu-driven program is easier to use for an inexperienced user. In a simulation package, it is also important to have a flexible way of presenting the results, which are often curves. Finally, to be able to solve the type of problems of interest in this module, it is important to be able to mix continuous- and discrete-time systems.

Examples of simulation packages are MATLAB® with SIMULINK® Because these packages are readily available we will not describe any of them in detail. However, we urge the students to use simulation to get a good feel for the behavior of the computer-controlled systems that are described in the module. For the figures in the lectures we have used MATLAB® with

Control of the Double Integrator

The double integrator will be used as the main example to show how the closed-loop behavior is changed with different controllers. The transfer function of the double integrator for the sampling period $h = 1$ is

$$H_0(z) = \frac{0.5(z+1)}{(z-1)^2} \tag{3.23}$$

Assume that the purpose of the control is to make the output follow changes in the reference value. Let's start with the simplest feedback controller, proportional feedback, that is,

$$u(k) = K(u_c(k) - y(k)) = Ke(k) \quad K > 0$$

where u_c is the reference value. The characteristic equation of the closed-loop system is

$$(q-1)^2 + 0.5K(q+1) = q^2 + (0.5K-2)q + 1 + 0.5K = 0 \tag{3.24}$$

Jury's stability test gives the following conditions for stability:

$$\begin{aligned} &1 + 0.5K < 1 \\ &1 + 0.5K > -1 + 0.5K - 2 \\ &1 + 0.5K > -1 - 0.5K + 2 \end{aligned}$$

The closed-loop system is unstable for all values of the gain K. The root locus (the graphic plot of the poles against different gains) is shown in Fig. 3.8.

To get a stable system, the controller must be modified. It is known from continuous-time synthesis that derivative action improves stability, so proportional and derivative feedback can be tried also for the discrete-time system. We now assume that it is possible to measure and sample the velocity $\dot{y}$ and use that for feedback; that is,

$$u(k) = K(e(k) - T_d\dot{y}(k)) \tag{3.25}$$

To find the input-output model of the closed-loop system with controller (3.25), observe that

$$\ddot{y} = \frac{d\dot{y}}{dt} = u$$

Because u is constant over the sampling intervals,

$$\dot{y}(k+1) - \dot{y}(k) = u(k)$$

or

$$\dot{y}(k) = \frac{1}{q-1} u(k) \tag{3.26}$$

Equations (3.23), (3.25), and (3.26) give the closed-loop system

$$y(k) = \frac{0.5K(q+1)}{(q-1)(q-1+T_d K) + 0.5K(q+1)} u_c(k) \tag{3.27}$$

The system is of second order, and there are two free parameters, K and T_d, that can be used to select the closed-loop poles. The closed-loop system is stable if $K > 0, T_d > 0.5$, and $T_d K < 2$. The root locus with respect to K of the characteristic equation of (3.27) is shown in Fig. 3.10 when $T_d = 1.5$.

Let the reference signal be a step. Fig. 3.11 shows the continuous-time output for four different values of K. The behavior of the closed-loop system varies from an oscillatory to a well-damped response. When $K = 1$, the poles are in the origin and the output is equal to the reference value after two samples. This is called deadbeat control and will be discussed further later in this module. When $K > 1$, the output and the control signal oscillate because of the discrete-time pole on the negative real axis. Remember that in addition to complex poles, negative poles can also generate oscillation for discrete-time system, as discussed in section 2.6. The poles are inside the unit circle if $K < 4/3$. When $K > 4/3$, the system becomes unstable.

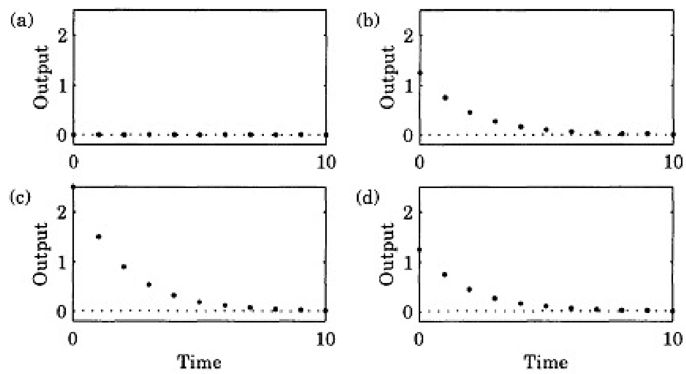


Figure 3.5: The output of the system in Example 3.3 for the initial states (a) $[0.5 \;\; 1]$, (b) $[1.5 \;\; 0.5]$, (c) $[2.5 \;\; 0]$, and (d) $[1 \;\; -0.5]$.

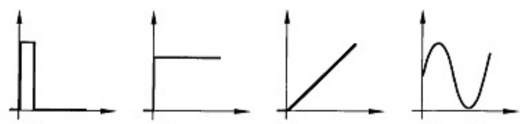


Figure 3.6: Idealized models of simple disturbances.

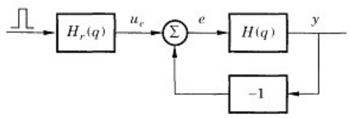


Figure 3.7: Generation of the reference value using a dynamic system with a pulse input.

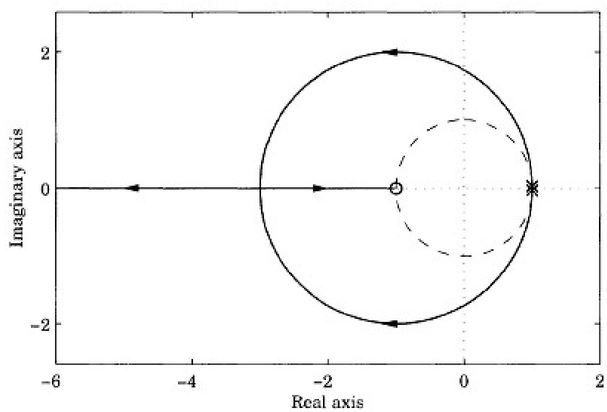


Figure 3.8: The root locus of (3.24) when $K > 0$.

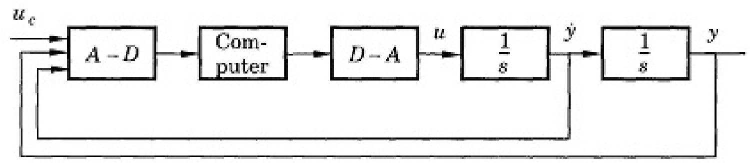


Figure 3.9: Discrete-time controller with feedback from position and velocity of the double integrator.

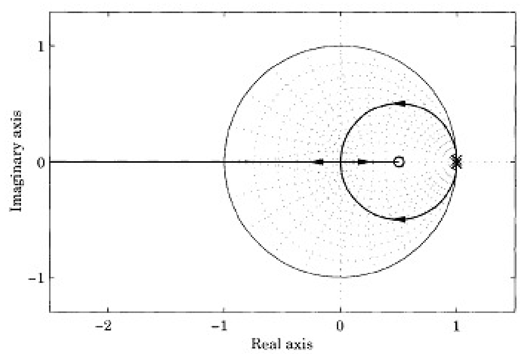


Figure 3.10: The root locus of the characteristic equation of the system in (3.27) with respect to the parameter K when $T_d = 1.5$.

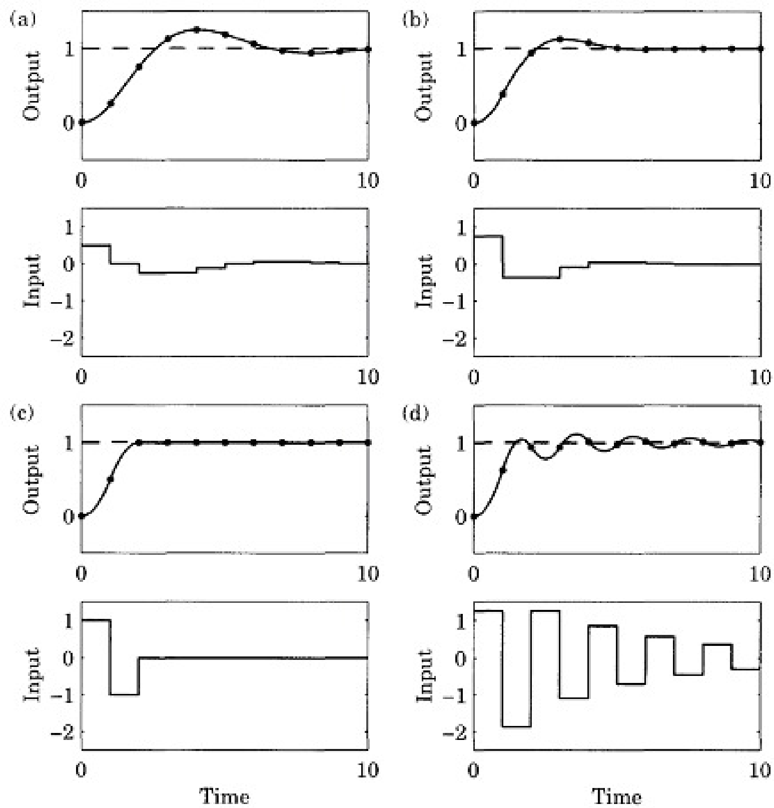


Figure 3.11: The continuous-time output of the system in Fig. 3.8 when $T_d = 1.5$ and (a) $K = 0.5$, (b) $K = 0.75$, (c) $K = 1$, and (d) $K = 1.25$.

Chapter 4

Pole-Placement Problem — State-Space Approach

In the previous lectures, we have discussed some key tools and concepts for analyzing discrete-time systems. Now it is time to apply these tools to computer control system design, which is supposed to be the core of this course. 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. We will discuss this type of problem later under certain conditions.

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 what 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.

4.1 Pole placement problem

We will try to solve the pole placement problem for the simplest case first: both the state-space model and the state variables are available. When you do research, it is always a good strategy to attack the simplest case first, then build it up.

Consider a linear system described by

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= cx \end{aligned} \tag{4.1}$$

Discretize the system using sampling period h, and we have

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= cx(k) \end{aligned} \tag{4.2}$$

where

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{As} b ds \end{aligned}$$

What's the simplest feedback-controller in the robotic manipulator example discussed in the first class? Proportional control.

Similarly, we will try the simplest one first. Let the state feedback controller be

$$u(k) = -Lx(k) \tag{4.3}$$

and the desired characteristic polynomial (C.P.), i.e., the denominator of the reference model, $H_m(z) = \frac{B_m(z)}{A_m(z)}$, is

$$A_m(z) = z^n + p_1 z^{n-1} + \cdots + p_n \tag{4.4}$$

Question: Can we choose L such that the poles of the closed loop system match desired poles given by above characteristic polynomial?

Example: Double-integrator:

$$\ddot{y}(t) = u(t) \tag{4.5}$$

Discretize it into

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) \end{aligned} \tag{4.6}$$

Let

$$u(k) = -Lx(k) = -l_1 x_1(k) - l_2 x_2(k) \tag{4.7}$$

and the closed-loop system becomes

$$x(k+1) = \begin{bmatrix} 1 - \frac{l_1 h^2}{2} & h - \frac{l_2 h^2}{2} \\ -l_1 h & 1 - l_2 h \end{bmatrix} x(k) \tag{4.8}$$

The corresponding characteristic polynomial is

$$z^2 + (\frac{l_1 h^2}{2} + l_2 h - 2)z + (\frac{l_1 h^2}{2} - l_2 h + 1)$$

Assume that the desired polynomial is

$$A_m(z) = z^2 + p_1 z + p_2 \tag{4.9}$$

Then just let

$$\begin{cases} \frac{l_1 h^2}{2} + l_2 h - 2 = p_1 \\ \frac{l_1 h^2}{2} - l_2 h + 1 = p_2 \end{cases}$$

and we obtain

$$\begin{cases} l_1 = \frac{1}{h^2}(1 + p_1 + p_2) \\ l_2 = \frac{1}{2h}(3 + p_1 - p_2) \end{cases} \tag{4.10}$$

Conclusion:

The state-feedback control can not only stabilize the system, but also place the poles at any desired locations!

Question:

Is this true for any linear system?
The answer to this question turns out to be directly connected with another fundamental concept in control theory: *Controllability*

Now let's take a look at another example.

$$x(k+1) = \Phi x(k) + \Gamma u(k) \tag{4.11}$$

where

$$\Phi = \begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix}, \Gamma = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

What's the C.P. of this system?

$$\det\{zI - \Phi\} = \det\left\{\begin{bmatrix} z + a_1 & +a_2 \\ -1 & z \end{bmatrix}\right\} = z^2 + a_1 z + a_2 \tag{4.12}$$

Note that the first row of Φ determines the C.P.!

Let's check the controllability matrix,

$$W_c = \begin{bmatrix} \Gamma & \Phi\Gamma \end{bmatrix} = \begin{bmatrix} 1 & -a_1 \\ 0 & 1 \end{bmatrix} \tag{4.13}$$

which is nonsingular. Therefore, the system is controllable.

Let's then try to design a state-feedback controller to place the poles at some desired places.

$$u(k) = -Lx(k) = -\begin{bmatrix} l_1 & l_2 \end{bmatrix} x(k) \tag{4.14}$$

The closed loop system then becomes

$$x(k+1) = \Phi x(k) + \Gamma u(k) = (\Phi - \Gamma L)x(k)$$

where

$$\Gamma L = \begin{bmatrix} l_1 & l_2 \\ 0 & 0 \end{bmatrix}$$

and

$$\Phi - \Gamma L = \begin{bmatrix} -a_1 - l_1 & -a_2 - l_2 \\ 1 & 0 \end{bmatrix}$$

whose C.P. is

$$z^2 + (a_1 + l_1)z + (a_2 + l_2)$$

If the desired C.P. is

$$z^2 + p_1 z + p_2$$

We simply choose the control gains L as

$$\begin{cases} l_1 = p_1 - a_1 \\ l_2 = p_2 - a_2 \end{cases}$$

and hence

$$u(k) = -Lx(k) = -\left[\begin{array}{cc} p_1 - a_1 & p_2 - a_2 \end{array}\right] x(k)$$

Note that the control parameters are the difference between the coefficients of the C.P. of the system and the desired polynomial. The pole-placement problem can be easily solved for such special kind of system. And the above result can be extended to general case as follows.

$$z(k+1) = \tilde{\Phi} z(k) + \tilde{\Gamma} u(k)$$

where

$$\tilde{\Phi} = \left[\begin{array}{cccc} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{array}\right]$$

$$\tilde{\Gamma} = \left[\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \end{array}\right]$$

and a_1, a_2, ... , a_n are the coefficients of the C.P.

$$z^n + a_1 z^{n-1} + \cdots + a_n$$

If the desired C.P. is

$$A_m(z) = z^n + p_1 z^{n-1} + \cdots + p_n$$

then

$$u(k) = -\tilde{L} z(k) = -\left[\begin{array}{cccc} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{array}\right] z(k) \qquad (4.15)$$

Remark: The pole-placement problem can be easily solved because of its special form. If you look at this form carefully, can you recognize what form it is? It is controllable companion /canonical form.
Question: How about systems in other forms?

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$

If the system is controllable, then it is similar to its corresponding controllable form. Or in other words, it can be transformed into controllable companion form by

$$
\begin{aligned}
&z = Tx \\
&x(k+1) = \Phi x(k) + \Gamma u(k) \\
&z(k+1) = Tx(k+1) = T\Phi x(k) + T\Gamma u(k) = T\Phi T^{-1} z(k) + T\Gamma u(k) = \tilde{\Phi} z(k) + \tilde{\Gamma} u(k)
\end{aligned}
$$

where

$$\tilde{\Phi} = T\Phi T^{-1}$$

and

$$\tilde{\Gamma} = T\Gamma$$

and the state-feedback controller is then

$$
\begin{aligned}
u(k) &= -\begin{bmatrix} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{bmatrix} z(k) \\
&= -\begin{bmatrix} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{bmatrix} Tx(k)
\end{aligned}
$$

Question: what is T?

$$T = \tilde{W}_c W_c^{-1} \tag{4.16}$$

Please refer to section 3.4.2 for further details about equation (4.16).

Overall, we have

$$
\begin{aligned}
&u(k) = -\begin{bmatrix} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{bmatrix} \tilde{W}_c W_c^{-1} x(k) = -Lx(k) \\
&L = \begin{bmatrix} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{bmatrix} \tilde{W}_c W_c^{-1}
\end{aligned}
$$

Using ***Cayley-Hamilton theorem***, it can be shown that

$$L = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} W_c^{-1} A_m(\Phi) \tag{4.17}$$

where the polynomial $A_m(z)$ is the desired C.P.
This is called the ***Ackermann's formula***

Remark: The pole-placement problem can be formulated as following algebraic problem:
Given pair $\{\Phi, \Gamma\}$, find L, such that $\{\Phi - \Gamma L\}$ has any desired C.P. $A_m(z)$.

Example: double-integrator

$$x(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u(k)$$

the controllability matrix

$$W_c = \begin{bmatrix} \frac{h^2}{2} & \frac{3h^2}{2} \\ h & h \end{bmatrix}$$

is nonsingular. Therefore, it is controllable. We will use ***Ackermann's formula*** to calculate the feedback gains L.

$$W_c^{-1} = \begin{bmatrix} -\frac{1}{h^2} & \frac{1.5}{h} \\ \frac{1}{h^2} & \frac{-0.5}{h} \end{bmatrix}$$

$$A_m(\Phi) = \Phi^2 + p_1\Phi + p_2 I = \begin{bmatrix} 1+p_1+p_2 & 2h+p_1 h \\ 0 & 1+p_1+p_2 \end{bmatrix}$$

Then applying ***Ackermann's formula*** gives

$$L = \begin{bmatrix} 0 & 1 \end{bmatrix} W_c^{-1} A_m(\Phi) = \begin{bmatrix} \frac{1+p_1+p_2}{h^2} & \frac{3+p_1-p_2}{2h} \end{bmatrix}$$

which is the same result as what we obtained before.

But what if the system is not controllable?

Example:

$$x(k+1) = \begin{bmatrix} a & 1 \\ 0 & b \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k)$$

The controllability matrix

$$W_c = \begin{bmatrix} 1 & a \\ 0 & 0 \end{bmatrix}$$

is singular. Can we still use Ackermann's Formula? No!

Let

$$u(k) = -l_1 x_1(k) - l_2 x_2(k)$$

then

$$x(k+1) = \begin{bmatrix} a-l_1 & 1-l_2 \\ 0 & b \end{bmatrix} x(k)$$

The C.P. is

$$(z - a + l_1)(z - b)$$

Hence, the pole $z = a$ can be changed, but the pole $z = b$ can not be changed!

- If $|b| < 1$, this uncontrollable mode is stable, we call such kind of system <u>stabilizable</u>.
- If $|b| > 1$, since this mode is unstable, there's nothing we can do to control this system.

4.1.1 Dead beat control

Let the desired poles are all zero. i.e.

$$A_m(z) = z^n \tag{4.18}$$

By ***Cayley-Hamilton theorem***

$$\Phi_c^n = (\Phi - \Gamma L)^n = 0$$

immediately we have

$$x(n) = (\Phi - \Gamma L)^n x(0) = 0$$

The state can be driven into zero at most n steps.

Example: Double-integrator

Let

$$p_1 = p_2 = 0$$
$$L = \left[\begin{array}{cc} \frac{1}{h^2} & \frac{3}{2h} \end{array}\right]$$
$$\Gamma L = \left[\begin{array}{c} \frac{h^2}{2} \\ h \end{array}\right] \left[\begin{array}{cc} \frac{1}{h^2} & \frac{3}{2h} \end{array}\right] = \left[\begin{array}{cc} \frac{1}{2} & \frac{3h}{4} \\ \frac{1}{h} & \frac{3}{2} \end{array}\right]$$
$$(\Phi - \Gamma L) = \left[\begin{array}{cc} \frac{1}{2} & \frac{h}{4} \\ -\frac{1}{h} & -\frac{1}{2} \end{array}\right]$$
$$(\Phi - \Gamma L)^2 = 0$$

and

$$x(2) = (\Phi - \Gamma L)^2 x(0) = 0$$

4.1.2 Summary

The state-feedback controller

$$u(k) = -Lx(k)$$

can place the poles of the system at any desired location if the system is controllable. Note that the state vector $x(k)$ is involved in the calculation of control input $u(k)$ at every instant k. But in many practical problems, only the output $y(k)$ is available, how to solve this problem?

We have to estimate the state x by the observations of the output y and input u.

4.2 Observer

Consider the system

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= cx(k) \end{aligned}$$

where $x(k)$ is not accessible. Since the state feedback controller, $u(k) = -Lx(k)$, needs the state information to calculate the input, it is necessary to estimate the state. This state estimator is usually called observer. Let's first try the simplest state estimator you might have in mind.

4.2.1 The Naive Observer

The naive observer takes the same form as the state-space model by replacing the state vector $x(k)$ (not available) with its estimate $\hat{x}(k)$.

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) \\ \hat{y}(k) &= c\hat{x}(k) \end{aligned}$$

Since the observer has the same dynamic equation as that of the state-space model, does that mean that the response of the observer will be the same as that of the original system subject to the same input sequence $u(k)$? The answer is no! Different initial conditions will lead to different solutions!

Our main concern is: would the state estimation error decrease to zero? If yes, then this will be a good observer.
Let the estimation error be

$$e(k) = x(k) - \hat{x}(k)$$

it easily follows that

$$e(k+1) = \Phi(e(k))$$

Therefore, the estimation error will blow up if the system is unstable. It might work if the system is stable.
What is missing in the naive observer? The output y! It did not use any information of the output! Let's see next if we can improve the performance of the observer by feeding back the output estimation error, $y(k) - \hat{y}(k)$.

4.2.2 Observer

Construct an observer by feeding back the output estimation error,

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k)) \\ \hat{y}(k) &= c\hat{x}(k) \end{aligned} \tag{4.19}$$

Then

$$x(k+1) - \hat{x}(k+1) = \Phi(x(k) - \hat{x}(k)) - K(y(k) - \hat{y}(k))$$

Let the estimation error be defined by

$$e(k) = x(k) - \hat{x}(k)$$

then

$$e(k+1) = \Phi e(k) - Kce(k) = (\Phi - Kc)e(k)$$

Question: Can we chose K such that $\Phi - Kc$ has any desired C.P., $A_o(z)$?

Recall that we have already solved a very similar problem: Given pair $\{\Phi, \Gamma\}$, find L, such that $\{\Phi - \Gamma L\}$ has any desired C.P. And the solution is given by the ***Ackermann's formula***:

$$L = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} W_c^{-1} A_m(\Phi)$$

We cannot directly apply this formula here since matrix multiplication is not commutable. But we know that the C.P. of a matrix in invariant to the operation of transpose. Let

$$(\Phi - Kc)^T = \Phi^T - c^T K^T$$

Then the problem becomes: given pair of $\{\Phi^T, c^T\}$, find K^T, such that $\Phi - Kc$ has any desired C.P.. Now we can apply the previous Ackermann's formula by identifying following

$$\begin{aligned} \Phi &\Rightarrow \Phi^T \\ \Gamma &\Rightarrow c^T \\ L &\Rightarrow K^T \end{aligned} \tag{4.20}$$

The corresponding W_c in the ***Ackermann's formula*** is then

$$W_c = \begin{bmatrix} c^T & \Phi^T c^T & \cdots & (\Phi^{n-1})^T c^T \end{bmatrix} = \begin{bmatrix} c \\ c\Phi \\ \vdots \\ c\Phi^{n-1} \end{bmatrix}^T = W_o^T \tag{4.21}$$

Then applying the ***Ackermann's formula*** gives

$$K^T = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} (W_o^T)^{-1} A_o(\Phi^T)$$

which can be rewritten as

$$K = A_o(\Phi)(W_o)^{-1} \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \tag{4.22}$$

Example: Double-Integrator

$$x(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k)$$

$$c = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}$$

$$W_o = \begin{bmatrix} c \\ c\Phi \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & h \end{bmatrix}$$

$$W_o^{-1} = \frac{1}{h} \begin{bmatrix} h & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{h} & \frac{1}{h} \end{bmatrix}$$

$$A_o(\Phi) = \Phi^2 + \lambda_1 \Phi + \lambda_2 = \begin{bmatrix} 1 + \lambda_1 + \lambda_2 & 2h + \lambda_1 h \\ 0 & 1 + \lambda_1 + \lambda_2 \end{bmatrix}$$

$$K = A_o(\Phi)(W_o)^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 + \lambda_1 \\ \frac{1+\lambda_1+\lambda_2}{h} \end{bmatrix}$$

4.2.3 Dead-beat observer

Choose $A_o(z) = z^n$. Since $A_o(z)$ is the C.P. of $(\Phi - Kc)$, it follows that

$$e(k+n) = (\Phi - Kc)^n e(k) = 0$$

Then the observation error goes to zero at most n steps.

4.3 Output-Feedback Controller

Summarizing the previous results, we can now give the complete solution to pole-placement problem when only outputs are available. Consider system

$$
\begin{aligned}
x(k+1) &= \Phi x(k) + \Gamma u(k) \\
y(k) &= c x(k)
\end{aligned}
$$

Step 1: Assume the state variables $x(k)$ is available, design a state-feedback controller

$$
u(k) = -Lx(k) \tag{4.23}
$$

such that $(\Phi - \Gamma L)$ has desired C.P.

$$
A_m(z) = z^n + p_1 z^{n-1} + \cdots + p_n
$$

L can be calculated by ***Ackermann's formula*** if the system is controllable.

$$
L = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} W_c^{-1} A_m(\Phi) \tag{4.24}
$$

Step 2: Build an observer using output feedback

$$
\begin{aligned}
\hat{x}(k+1) &= \Phi \hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k)) \\
\hat{y}(k) &= c\hat{x}(k)
\end{aligned} \tag{4.25}
$$

$$
e(k+1) = (\Phi - Kc)e(k)
$$

Choose K such that $(\Phi - Kc)$ has desired C.P.

$$
A_o(z) = z^n + \lambda_1 z^{n-1} + \cdots + \lambda_n
$$

K can be calculated by

$$
K = A_o(\Phi)(W_o)^{-1} \begin{bmatrix} 0 & \cdots & 1 \end{bmatrix}^T \tag{4.26}
$$

if the system is observable.

Step 3: Replace the state $x(k)$ in *step 1* by the estimate $\hat{x}(k)$

$$
u(k) = -L\hat{x}(k) \tag{4.27}
$$

The overall closed loop system becomes

$$
\begin{cases}
x(k+1) = \Phi x(k) - \Gamma L \hat{x}(k) \\
e(k+1) = (\Phi - Kc)e(k)
\end{cases}
$$

$$\begin{cases} x(k+1) = \Phi x(k) - \Gamma L x(k) + \Gamma L(x(k) - \hat{x}(k)) = (\Phi - \Gamma L)x(k) + \Gamma L e(k) \\ e(k+1) = (\Phi - Kc)e(k) \end{cases}$$

If both $(\Phi - \Gamma L)$ and $(\Phi - Kc)$ is stable, the overall system is stable, and

$$e(k) \to 0, \quad x(k) \to 0$$

The above design has a very nice property:
Separation property: the poles for the controller and observer can be designed separately.

Example: Double-integrator

$$x(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k)$$

Step 1:

$$u(k) = -Lx(k)$$
$$L = \begin{bmatrix} \frac{1+p_1+p_2}{h^2} & \frac{3+p_1-p_2}{2h} \end{bmatrix}$$

Step 2:

$$\hat{x}(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \hat{x}(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u(k) + K(y(k) - \hat{y}(k))$$
$$\hat{y}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{x}(k)$$

where

$$K = \begin{bmatrix} 2 + \lambda_1 \\ \frac{1+\lambda_1+\lambda_2}{h} \end{bmatrix}$$

Step 3: Output feedback controller:

$$u(k) = -L\hat{x}(k)$$

4.4 Disturbance Rejection Problem

The results we have derived are only useful for impulse disturbances such as

when each impulse disturbance occurs, the states will be driven away from equilibrium position, and the feedback controller would drive them back to equilibrium position as shown in Fig. 4.6.

In general if there are disturbances acting upon the system, the controller has to cancel out the effects of the disturbances. This is called disturbance rejection problem.

Let's consider a system where disturbance $\omega(k)$ affects the system as follows.

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Phi_{x\omega}\omega(k) + \Gamma u(k) \\ \omega(k+1) &= \Phi_\omega \omega(k) \\ y(k) &= cx(k) \end{aligned} \tag{4.28}$$

What would happen simply ignoring the disturbance term? Let the controller still be the state feedback controller

$$u(k) = -Lx(k)$$

then

$$x(k+1) = (\Phi - \Gamma L)x(k) + \Phi_{x\omega}\omega(k)$$

Obviously, the response will be heavily affected by the disturbance. Hence the disturbance cannot be ignored.

What's the state of the system? We'd better consider the disturbance $\omega(k)$ as part of the state instead of the input, since we cannot change the $\omega(k)$ itself.

Introduce the augmented state vector,

$$z(k) = \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} \tag{4.29}$$

then

$$\begin{aligned} z(k+1) &= \begin{bmatrix} x(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Phi_{x\omega} \\ 0 & \Phi_\omega \end{bmatrix} z(k) + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} c & 0 \end{bmatrix} z(k) \end{aligned} \tag{4.30}$$

Is this system controllable?

Of course not. How can you control the disturbance! But we can control $x(k)$, which is the original state vector by estimating the disturbance as shown below.

Step 1: Assume the disturbances and state variables are measurable, design the controller to stabilize the system and cancel out the disturbances. Let

$$u(k) = -Lx(k) - L_\omega \omega(k) \tag{4.31}$$

The closed loop system becomes

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ \omega(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & \Phi_{x\omega} \\ 0 & \Phi_\omega \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} - \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} \begin{bmatrix} L & L_\omega \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} \\
&= \begin{bmatrix} \Phi - \Gamma L & \Phi_{x\omega} - \Gamma L_\omega \\ 0 & \Phi_\omega \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} \\
y(k) &= \begin{bmatrix} c & 0 \end{bmatrix} z(k)
\end{aligned}$$

The we can choose L such that $(\Phi - \Gamma L)$ has desired poles as before.

But how to choose L_ω?

Choose L_ω to reduce the effect of the disturbance ω_k. This control action is particularly effective if $\Phi_{x\omega} - \Gamma L_\omega = 0$, which implies that the disturbance can be completely canceled out. But this can be hardly achieved for most cases, and selecting proper L_ω remains a difficult problem. If the disturbance ω is constant, then it is possible to choose L_ω such that the steady state gain from the disturbance to the output is zero, which implies that the effect of the disturbance can be completely filtered out in steady state.

Step 2: If the disturbances are not accessible, which is usually the case in practice, then build an observer to estimate the disturbances, as well as the states.

$$\begin{aligned}
\hat{z}(k+1) &= \begin{bmatrix} \Phi & \Phi_{x\omega} \\ 0 & \Phi_\omega \end{bmatrix} \hat{z}(k) + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k) + K(y(k) - \hat{y}(k)) \\
\hat{y}(k) &= \begin{bmatrix} c & 0 \end{bmatrix} \hat{z}(k)
\end{aligned} \tag{4.32}$$

Is the system observable? It is indeed observable in many cases.

Then K can be chosen such that the C.P. of

$$\left(\begin{bmatrix} \Phi & \Phi_{x\omega} \\ 0 & \Phi_\omega \end{bmatrix} - K \begin{bmatrix} c & 0 \end{bmatrix} \right)$$

match any desired one. Then we can make sure that the estimation error of the augmented state vector $z(k)$ converge to zero. It can be made to be zero after finite number of steps if dead-beat observer is used.

Step 3: Replace the disturbance variable and state variables by their corresponding estimates in the controller designed in *Step 1.* i.e.

$$u(k) = -L\hat{x}(k) - L_\omega \hat{\omega}(k) \tag{4.33}$$

The block diagram of the overall system is shown in Fig. 4.7.

Example: Double-Integrator

A constant but unknown disturbance $\omega(k)$ acting on the input as follows,

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} (u(k) + \omega(k)) \\ y(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) \\ \omega(k+1) &= \omega(k) \end{aligned}$$

Step 1: If both $x(k)$ and $\omega(k)$ are known, design a state-feedback controller

$$u(k) = -Lx(k) - \omega(k)$$

Step 2: Add $\omega(k)$ into the state vector

$$z(k) = \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix}$$

then

$$\begin{bmatrix} x(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} \frac{h^2}{2} \\ h \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix}$$

The observability matrix for the augmented system is

$$W_o = \begin{bmatrix} c \\ c\Phi \\ c\Phi^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & h & \frac{h^2}{2} \\ 1 & 2h & 2h^2 \end{bmatrix}$$

which is nonsingular, hence the system is observable! Therefore, the augmented state vector including the state $x(k)$ and disturbance $v(k)$ can be estimated.

Build the observer as follows,

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{\omega}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{\omega}(k) \end{bmatrix} + \begin{bmatrix} \frac{h^2}{2} \\ h \\ 0 \end{bmatrix} u(k) + K(y(k) - \hat{y}(k))$$

$$\hat{y}(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{\omega}(k) \end{bmatrix}$$

K can be calculated based on the desired C.P. for the observer

$$A_o(z) = z^3 + \lambda_1 z^2 + \lambda_2 z + \lambda_3$$

then

$$K = \lambda(\Phi)(W_o)^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Step 3: Replace the true values by the estimated in *step 1*:

$$u(k) = -L\hat{x}(k) - \hat{\omega}(k)$$

4.5 The Tracking Problem — Zero Placement

So far only the pole placement problem has been completely solved. In the following we will try to match the zeros too. This can be called zero placement problem.

4.5.1 A Naive Approach

A simple way to obtain the desired response to command signals is to replace the regular state feedback $u(k) = -L\hat{x}(k)$ by

$$u(k) = -L\hat{x}(k) + L_c u_c(k) \tag{4.34}$$

where $u_c(k)$ is the command signal. To investigate the response of such a controller, we consider the closed-loop system that is described by

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= cx(k) \\ \hat{x}(k+1) &= \Phi \hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k)) \\ \hat{y}(k) &= c\hat{x}(k) \\ u(k) &= -L\hat{x}(k) + L_c u_c(k) \end{aligned}$$

A block diagram of the system is shown in the Fig. 4.8. Eliminating u and introducing the estimation error e we find that the closed-loop system can be described by

$$
\begin{aligned}
x(k+1) &= (\Phi - \Gamma L)x(k) + \Gamma L e(k) + \Gamma L_c u_c(k) \\
e(k+1) &= (\Phi - Kc)e(k) \\
y(k) &= cx(k)
\end{aligned}
$$

It follows from above equations that the transfer function from the command signal $u_c(k)$ to the process output y is given by

$$
H_{cl}(z) = c(zI - \Phi + \Gamma L)^{-1}\Gamma L_c = L_c \frac{B(z)}{A_m(z)} \tag{4.35}
$$

This can be compared with the transfer function of the process

$$
H(z) = c(zI - \Phi)^{-1}\Gamma = \frac{B(z)}{A(z)} \tag{4.36}
$$

The fact that the polynomial $B(z)$ appears in the numerator of both transfer functions can be seen by transforming both systems to controllable canonical form.

The closed-loop system obtained with this naive control law has the same zeros as the plant and its poles are the eigenvalues of the matrix $(\Phi - \Gamma L)$, which can be placed at any positions by properly choosing L is the system is controllable.

Let the transfer function of the reference model be $H_m(z) = \frac{B_m(z)}{A_m(z)}$, then the tracking problem is to design the controller such that the transfer function of the closed loop $H_{cl}(z)$ match the reference model $H_m(z)$ as close as possible.

$$
L_c \frac{B(z)}{A_m(z)} \xrightarrow{\textit{choose} \quad L_c} \frac{B_m(z)}{A_m(z)}
$$

It is impossible to achieve this if L_c is limited to some constant vector as in the naive controller. However, what if L_c is another transfer function?

4.5.2 Two-Degree-of-Freedom Controller

Practical control systems often have specifications that involve both servo and regulation properties. This is traditionally solved using a two-degree-of-freedom structure, as shown in Fig. 4.9. This configuration has the advantage that the servo and regulation problems are separated. The feedback controller H_{fb}, 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} is then designed to obtain the desired servo properties. Obviously, the naive controller discussed before is just the simple case of two-degree-of-freedom controller with $H_{ff} = L_c$. If instead, we choose

$$H_{ff} = L_c = \frac{B_m(z)}{B(z)} \tag{4.37}$$

Then we have the closed loop transfer function from the command signal u_c to the output,

$$\frac{Y(z)}{U_c(z)} = L_c \frac{B(z)}{A_m(z)} = \frac{B_m(z)}{B(z)} \frac{B(z)}{A_m(z)} = \frac{B_m(z)}{A_m(z)} = H_m(z) \tag{4.38}$$

and mission accomplished!

However, this is only true if all the zeros of $B(z)$ are stable or all the unstable zeros of $B(z)$ are also the zeros for $B_m(z)$! This is due to the fact that the closed-loop transfer function from the command signal u_c to the input signal is

$$\frac{U(z)}{U_c(z)} = \frac{A(z)B_m(z)}{B(z)A_m(z)} \tag{4.39}$$

which requires that $B(z)$ has to be stable. Otherwise, the input signal may blow up!

The condition of all the zeros of $B(z)$ to be stable is called the
Stable inverse condition which was discussed in section 2.6.

In general, only system with stable inverse can be controlled to perfectly track arbitral command signals. If the system has an unstable inverse, how to design H_{ff} would depend upon the nature of the specific command signal, which is outside the scope of this module.

4.5.3 Putting It All Together

By combining the solutions to the regulation and servo problems we have a powerful controller, which is described by

$$
\begin{aligned}
x(k+1) &= \Phi x(k) + \Phi_{x\omega}\omega(k) + \Gamma u(k) \\
\omega(k+1) &= \Phi_\omega \omega(k) \\
y(k) &= cx(k) \\
\hat{x}(k+1) &= \Phi \hat{x}(k) + \Phi_{x\omega}\hat{\omega}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k)) \\
\hat{\omega}(k+1) &= \Phi_\omega \hat{\omega}(k) + K_\omega(y(k) - \hat{y}(k)) \\
\hat{y}(k) &= c\hat{x}(k) \\
u(k) &= u_{ff}(k) + u_{fb}(k) \\
u_{fb}(k) &= -L\hat{x}(k) - L_\omega \hat{\omega}(k) \\
u_{ff}(k) &= H_{ff} u_c(k)
\end{aligned}
\tag{4.40}
$$

This controller captures many aspects of a control problem such as load-disturbance attenuation, reduction of effects of measurement noise, and command signal following. The responses to load disturbances, command signals, and measurement noise are completely separated. The command signal response is determined by the reference model. The response to load disturbances and measurement noise is influenced by the observer and the state feedback. A block diagram of the closed-loop system is shown in Fig. 4.10.

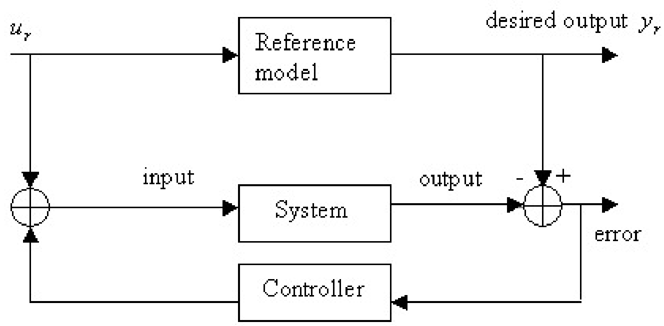


Figure 4.1: Model reference control system

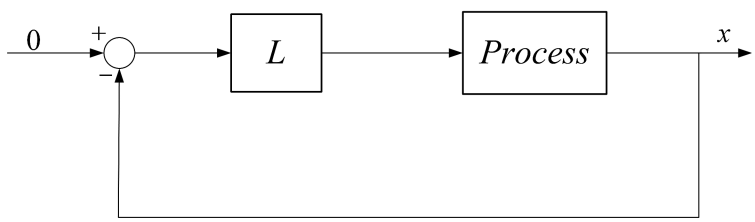


Figure 4.2: State feedback controller

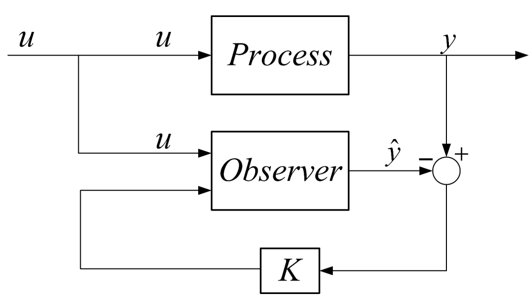


Figure 4.3: observer by output error feedback

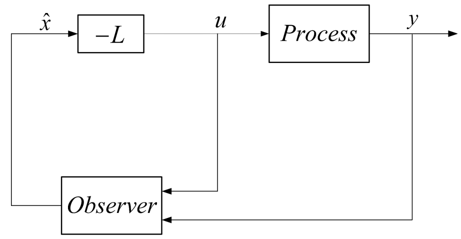


Figure 4.4: output feedback controller by combining state feedback with observer.

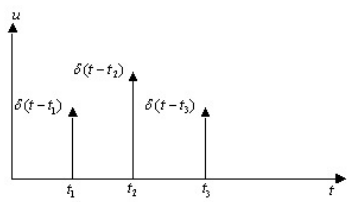


Figure 4.5: Impulse Disturbance

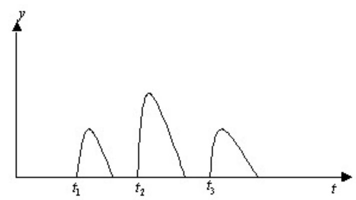


Figure 4.6: Response of the system to impulse disturbance

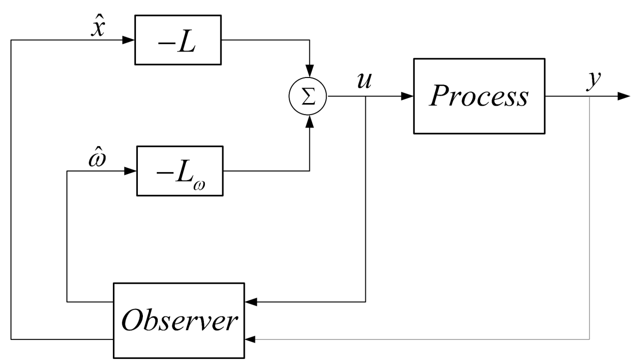


Figure 4.7: output feedback controller with estimated disturbance and states.

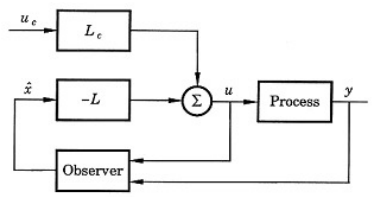


Figure 4.8: Naive approach for tracking problem

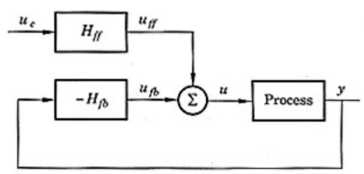


Figure 4.9: Two-degree-of-freedom Controller

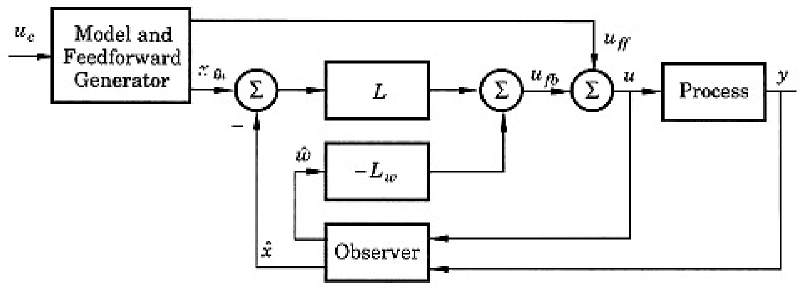


Figure 4.10: The state-space approach to control system design

Chapter 5

Pole-Placement — Polynomial Approach

5.1 Review: Pole placement problem - State-space approach

Consider system

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = cx(k)$$

Step 1: Assume that the state variables $x(k)$ are available, design a state-feedback controller

$$u(k) = -Lx(k)$$

such that $(\Phi - \Gamma L)$ has desired C.P. and L can be calculated by Ackermann's formula.

What if the system is not controllable, i.e., W_c is singular? Can we still assign the poles to any desired locations?

When the system is not controllable, it implies there are some poles that cannot be changed by the state-feedback controller. The situation is not completely hopeless provided these uncontrollable poles are stable! Such systems are called *stabilizable*.

Step 2: In many cases, only output $y(k)$ is available, we need to build an observer using output feedback

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k))$$
$$\hat{y}(k) = c\hat{x}(k)$$

$$e(k+1) = (\Phi - Kc)e(k)$$

Choose K such that $(\Phi - Kc)$ has desired C.P. K can be calculated by ***Ackermann's formula***.

Again, what if the system is unobservable? It implies that some poles for the observer cannot be changed by output feedback. But if these unobservable poles are all stable, then the estimation error can be made to converge to zero asymptotically. We call such kind of system *detectable*.

When a system is detectable but unobservable, the state variables can still be estimated. But unobservability implies that the state variables cannot be determined uniquely by the observation of the outputs, do we have a contradiction here?

No. Convergence of estimation error to zero does not imply that the state variables can be determined exactly. For instance, consider two convergent series, $\{sin(t)/t\}$ and $\{1/t\}$. Both of them converge to zero, but they are not equal to each other.

Step 3: Output-feedback controller–Replacing the state $x(k)$ by the estimate

$$u(k) = -L\hat{x}(k)$$

Disturbance Rejection

The standard way of dealing with disturbance is adding them into the state vector, and building an observer to estimate the whole augmented state vector.

Question: If $x(k)$ available, we have $u(k) = -Lx(k)$. If $x(k)$ not available, we use $u(k) = -L\hat{x}(k)$. What if $x(k)$ is partially available?

$$\underbrace{x_1(k), \cdots, x_l(k)}_{Yes}, \underbrace{x_{l+1}(k), \cdots, x_n(k)}_{No}$$

Should we still use $u(k) = -L\hat{x}(k)$, or try the combination.

$$u(k) = -L \begin{bmatrix} x_1(k) \\ \vdots \\ x_l(k) \\ \hat{x}_{l+1}(k) \\ \vdots \\ \hat{x}_n(k) \end{bmatrix}$$

Rule of Thumb: always use the more accurate information.

5.2 Relation between State-Space Model and Input-Output Model

As discussed in the last lecture, the pole-placement problem can be completely solved by state-space approach. Then why do we need another approach? Because there are a number of difficulties associated with this approach: first of all, choosing proper L_ω for disturbance rejection is quite a headache. To make things worse, the state-space model may not be available in real world problems! In real problems, many of the systems are directly described by input-output models. It is natural to explore if the pole-placement problem can be done directly by polynomial calculation.

But before we address this problem, let's discuss the relationship between the state-space model and the input-output model.

Given a state-space model

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= cx(k) \end{aligned}$$

How do we deduce the input-output model or transfer function? What's the tool? The z-transform! Apply z-transform and assume zero initial conditions, we have

$$\begin{aligned} zX(z) &= \Phi X(z) + \Gamma U(z) \\ Y(z) &= cX(z) \end{aligned}$$

From the above equation,

$$\begin{aligned} (zI - \Phi)X(z) &= \Gamma U(z) \\ X(z) &= (zI - \Phi)^{-1}\Gamma U(z) \end{aligned}$$

Finally, we have

$$Y(z) = cX(z) = c(zI - \Phi)^{-1}\Gamma U(z)$$

Transfer function:

$$c(zI - \Phi)^{-1}\Gamma = \frac{B(z)}{A(z)}$$

Difference Equation:

$$A(q)y(k) = B(q)u(k)$$

Now the inverse problem: let a system be described by difference equation

$$A(q)y(k) = B(q)u(k)$$

Or equivalently a transfer function

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

Can we find the corresponding state-space representation? This is the so called

Realization Problem:

Let

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

How many state-space models can be derived? There are infinite number of realizations. For instance, the most famous one is

the ***Controllable Companion/Canonical Form:*** $\{\Phi_c, \Gamma_c, c_c\}$

$$\Phi_c = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad \Gamma_c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$c_c = \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}$$

Or

The ***Observable Companion/Canonical form:*** $\{\Phi_o, \Gamma_o, c_o\}$, which is simply the dual system of the controllable companion form $\{\Phi_c^T, \Gamma_c^T, c_c^T\}$

$$\Phi_o = \Phi_c^T = \begin{bmatrix} -a_1 & 1 & \cdots & 0 \\ -a_2 & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ -a_n & \cdots & 0 & 0 \end{bmatrix}, \quad \Gamma_o = c_c^T = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$$c_o = \Gamma_c^T = \begin{bmatrix} 1 & \cdots & 0 \end{bmatrix}$$

Are these two representations similar to each other? Not always!

If the system is both controllable and observable, then these two companion forms are similar to each other.

5.3 Pole-Placement Problem – Polynomial Approach

In many problems, only input-output representation is available

$$A(q)y(k) = B(q)u(k) \tag{5.1}$$

The poles of the system is decided by the roots of $A(z)$. If the poles of $A(q)$ are not at ideal places, then we may want to place the poles at proper positions. — The same objective as before.

One obvious approach: realize the system with a state-space representation, and then design the output-feedback controller.

Or is there a direct way?
Yes.

5.3.1 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. 5.1.

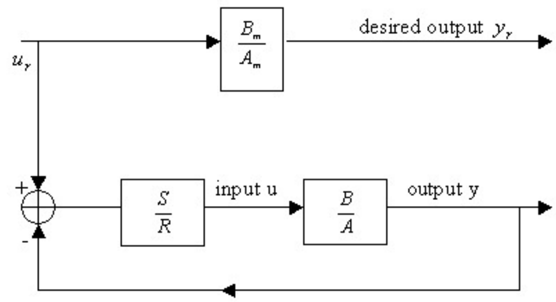


Figure 5.1: Pure Error-Feedback Controller

The pure error feedback control is in the form of

$$R(q)u(k) = S(q)e(k) = S(q)(u_c(k) - y(k)) \tag{5.2}$$

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, just as we did for the state-space approach.

5.3.2 Two-degree-of-freedom controller

The two-degree-of-freedom controller can be designed in following form

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k) \tag{5.4}$$

The block diagram is shown in the Fig. 5.2.
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.

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

In the state-space approach, there are two desired polynomials, one for the state-feedback controller A_c, and one for the output-feedback observer, A_o, and the C.P. for the overall system is

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

So

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

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

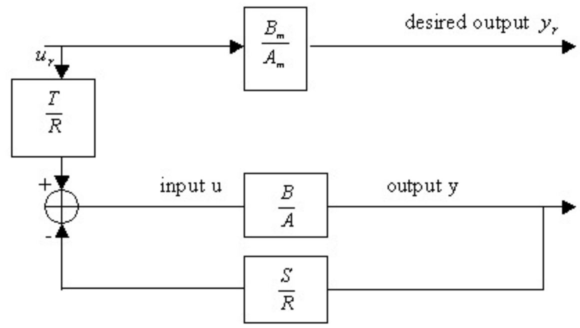


Figure 5.2: Two-Degree-of-Freedom Controller

If perfect match of the reference model is not required, one simple way to choose $T(z)$ is

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

Such that

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

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 $\frac{t_o B(1)}{A_c(1)}$. For example, to have unit static gain, we simply let

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

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 class.

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:

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

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 can not be implemented, although mathematically it is fine.

Causality condition:

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

Example: Double integrator

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

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

$$R = 1 \quad S = s_0$$

So

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

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

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

If the desired C.P. is

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

Compare the coefficients

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

We have

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

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

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

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

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

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

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 conditions the solution may not exist?

Hint: In the state space approach, if the system is uncontrollable, then the problem cannot be solved. Translating that into the input-output model, we may state that:

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.

And we know, if $A(z)$ and $B(z)$ have a common factor, then this pole cannot be changed by linear state-feedback controller, and the minimal realization of the system is uncontrollable.

Do we need other conditions to guarantee the existence of solution?

Hint: if $A(z)$ and $B(z)$ have no common factors, then the minimal realization of the system is both controllable and observable. Therefore, the solution exists at least for the minimal realization of the system. So our intuition tells us that we don't need extra conditions.

Let's see if our intuition is true. 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. I hope you can figure it out after the lecture.

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

$$A(z)y(k) = B(z)u(k)$$

Let

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

Design a controller of degree $n-1$,

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

Then if the minimal realization of the system is both controllable and observable (i.e., A and B have no common factors), R and S 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 controllable and observable condition assures the existence of inverse of the Sylvester matrix, M_s^{-1}, in above solution. 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: Double Integrator

$$A(z) = (z-1)^2 = z^2 - 2z + 1$$
$$B(z) = \frac{h^2}{2}(z+1)$$

$n = 2$, 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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & \frac{h^2}{2} & 0 \\ 1 & -2 & \frac{h^2}{2} & \frac{h^2}{2} \\ 0 & 1 & 0 & \frac{h^2}{2} \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{5.19}$$

and we can obtain

$$
\begin{aligned}
r_0 &= 1\\
r_1 &= \frac{3+p_1+p_2-p_3}{4}\\
s_0 &= \frac{5+3p_1+p_2-p_3}{2h^2}\\
s_1 &= -\frac{3+p_1-p_2-3p_3}{2h^2}
\end{aligned}
$$

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.

5.4 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 cancelation can be used.

5.4.1 Pole-Zero Cancelation

Consider the system described by

$$A(q)y(k) = B(q)u(k)$$

Let the output feedback controller be

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

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 cancelations. Sometimes cancelation 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: Cancelation 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

$$H_m(z) = \frac{z(1+p_1+p_2)}{z^2+p_1z+p_2}$$

$z = b$ 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

$$ks_0 - a - 1 = p_1$$
$$a + ks_1 = p_2$$

and

$$s_0 = \frac{a + 1 + p_1}{k}$$
$$s_1 = \frac{p_2 - a}{k}$$

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

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

$$(z - b)u = t_0 z u_c - (s_0 z + s_1)y$$

or

$$u(k) = bu(k - 1) + t_0 u_c(k) - s_0 y(k) - s_1 y(k - 1)$$

5.4.2 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. 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 closed-loop transfer function from the disturbance to the output, $D_{cl}(z)$, can be derived as

$$D_{cl}(z) = \frac{Y(z)}{D(z)} = \frac{R(z)B(z)}{A(z)R(z) + B(z)S(z)} \tag{5.27}$$

The block diagram is shown in Fig. 5.3.
Let's first consider the case when the disturbance is an unknown constant. To eliminate the effect of the constant disturbance, we should design the controller such that the static gain of the closed-loop transfer function, $D_{cl}(z)$, relating the disturbance to the output, is zero. Since the static gain is simply

$$D_{cl}(1) = \frac{R(1)B(1)}{A(1)R(1) + B(1)S(1)}$$

all we need to do is letting $R(z)$ have $(z-1)$ as a factor,

$$R(z) = R'(z)(z-1) \tag{5.28}$$

Then

$$R(1) = 0 \tag{5.29}$$

This design is essentially putting the integrator in the controller.

Example:

$$y(k+1) = 3y(k) + u(k) + d(k)$$

where $d(k)$ is a constant disturbance. In order to cancel out the disturbance R must contain the factor $(z-1)$.

$$A(z) = z - 3$$
$$B(z) = 1$$

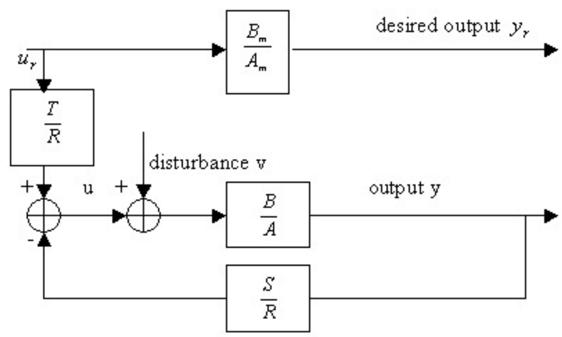


Figure 5.3: Two-Degree-of-Freedom Controller with load disturbance.

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.

Other disturbances

What if the disturbance is not a constant?
If the disturbance $d(k)$ is deterministic rather than purely random, then it is still possible to reject the effect of the disturbance. If the frequency of the disturbance, ω_d, can be estimated, then we can use the frequency response of the transfer function, $D_{cl}(z)$, to design the controller.

If $d(k)$ is a sinusoid with frequency ω_d, then the frequency response due to this sinusoid is expressed by

$$D_{cl}(e^{j\omega_d h}) \tag{5.30}$$

From the expression of $D_{cl}(z)$, if we can design $R(z)$ such that

$$R(e^{j\omega_d h}) = 0$$

then the frequency response due to this sinusoid will be zero at steady state. In order to achieve this, let's construct a polynomial as follows

$$(z - e^{j\omega_d h})(z - e^{-j\omega_d h}) = z^2 - 2\cos\omega_d h z + 1$$

Then let $R(z)$ contain above polynomial as a factor, the frequency response due to the sinusoid disturbance will be zero. If there are multiple frequencies in the disturbance, we just add more factors into $R(z)$. It is as simple as that.

5.4.3 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

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

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 inverse (all zeros are stable). This is the same conclusion as we found out for the state space approach.

Is it really impossible to follow a reference model perfectly for system with unstable inverse? 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. 5.2, 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

$$u_{ff}(k) = \frac{T(q)}{R(q)}u_c(k)$$

and

$$u_{fb}(k) = -\frac{S(q)}{R(q)}y(k)$$

Note that the transfer functions for generating the two inputs share the same denominator $R(q)$. In order to gain extra freedom, it is also possible to relax this constraint, and let

$$u_{ff}(k) = H_{ff}(q)u_c(k)$$

where the feedforward transfer function $H_{ff}(q)$ gives you additional freedom.

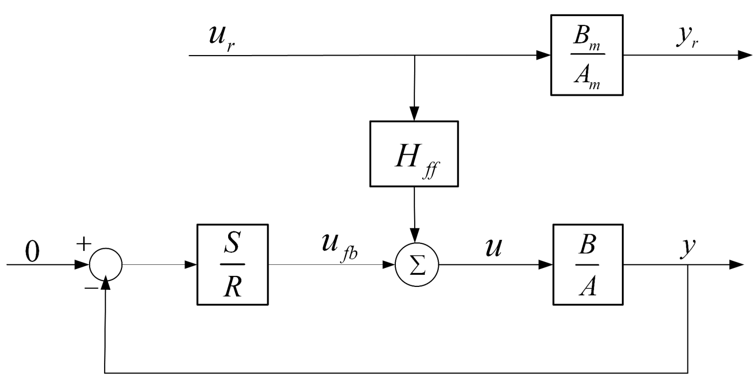


Figure 5.4: Another type of two-degree-of-freedom controller

What would be the advantages by doing this?

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 inverse 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 polynomial design,

$$R(q)u = T(q)u_c - S(q)y$$

1. Occam's razor — the simpler, the better.
2. Based on the design specifications, determine the constraints on R. For instance,
   - If zero cancelation is needed, then R must contain the stable zero factor B^+
   - If Disturbance rejection is required, then R must contain certain factors to reject the disturbance.
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.

Chapter 6

Predictive Control — Solution to Tracking Problem

6.1 Introduction

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.

We shall begin our discussion by emphasizing tracking performance and introduce a very simple form of control law which we call a one-step-ahead controller. The basic principle behind this controller is that the control input is determined at each point in time so as to bring the system output at a future time instant to a desired value. This is a very simple idea but, nonetheless, has considerable intuitive appeal. We shall show that the scheme works well not only for linear systems but can also be applied in a straightforward fashion to a large class of nonlinear systems. However, as with all design techniques, one must be aware of its limitations as well as its applications. Potential difficulties with this approach are that an excessively large effort may be called for to bring the output to the desired value in one step.

We are therefore motivated to look at alternative control schemes which allow us to limit the control effort needed to achieve the objective. This will then lead us to consider weighted one-step-ahead controllers wherein a penalty is placed on excessive control effort. We shall subsequently show that the one-step-ahead controller is a special case of the weighted one-step-ahead controller. Again, we shall point to its limitations and show that it may still lead to difficulty in some cases.

6.2 Minimum Prediction Error Controllers

We can think of a model as providing a way of predicting the future outputs of a system based on past outputs and on past and present inputs. It therefore seems reasonable to turn this around and ask what control action at the present instant of time would bring the future output to a desired value. This will be especially simple if the future outputs are a linear function of the present control, since then the determination of the control effort involves the solution of a set of linear equations.

We consider first the case of linear dynamical system. We have seen in previous lectures that the input-output properties of these systems (for arbitrary initial states) can be described by either state-space model or transfer function.

Let a system be described by difference equation

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n) = b_0u(k-d)+b_1u(k-d-1)+\cdots+b_{n_1}u(k-d-n_1) \tag{6.1}$$

which can be expressed as

$$A(q^{-1})y(k) = B(q^{-1})u(k) \tag{6.2}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \cdots a_nq^{-n} \\ B(q^{-1}) &= q^{-d}(b_0 + b_1q^{-1} + \cdots b_{n_1}q^{-n_1}), \qquad b_0 \;\neq 0 \\ &= q^{-d}B'(q^{-1}) \end{aligned} \tag{6.3}$$

6.2.1 One-Step-Ahead Control: The Case of $d = 1$

We first consider the case of time delay $d = 1$. Here we will consider one-step-ahead control, which brings $y(k+1)$ to a desired value $r(k+1)$ in one step. Let's rewrite the model (6.1) in a predictor form as (6.4).

$$y(k+1) = -a_1y(k)+\cdots-a_ny(k-n+1)+b_0u(k)+b_1u(k-1)+\cdots+b_{n_1}u(k-n_1) \tag{6.4}$$

or equivalently,

$$y(k+1) = \alpha(q^{-1})y(k) + \beta(q^{-1})u(k) \tag{6.5}$$

where

$$\begin{aligned} \alpha(q^{-1}) &= -a_1 + \cdots - a_nq^{-(n-1)} \\ \beta(q^{-1}) &= B'(q^{-1}) = (b_0 + b_1q^{-1} + \cdots b_{n_1}q^{-n_1}), \qquad b_0 \;\neq 0 \end{aligned} \tag{6.6}$$

Obviously, the simple solution of tracking problem is to replace $y(k+1)$, the future output at the next step, in model (6.4) with $r(k+1)$, the desired output at the next step,

$$r(k+1) = -a_1 y(k) + \cdots - a_n y(k-n+1) + b_0 u(k) + b_1 u(k-1) + \cdots + b_{n_1} u(k-n_1) \tag{6.7}$$

and then solve for $u(k)$ as

$$u(k) = \frac{1}{b_0}[r(k+1) + a_1 y(k) + \cdots + a_n y(k-n+1) - b_1 u(k-1) + \cdots - b_{n_1} u(k-n_1)] \tag{6.8}$$

If control law (6.8) is used, it follows immediately by comparing (6.4) and (6.7) that

$$y(k+1) = r(k+1)$$

It appears that the above controller design can be directly extended to the case of $d > 1$, by writing the model (6.1) in the form of

$$y(k+d) = -a_1 y(k+d-1) + \cdots - a_n y(k+d-n) + b_0 u(k) + b_1 u(k-1) + \cdots + b_{n_1} u(k-n_1) \tag{6.9}$$

and solve for $u(k)$ from the equation below

$$r(k+d) = -a_1 y(k+d-1) + \cdots - a_n y(k+d-n) + b_0 u(k) + b_1 u(k-1) + \cdots + b_{n_1} u(k-n_1)$$

Unfortunately, this type of controller violates the law of causality since the future inputs $y(k+d-1)$ is not available at time k!

Therefore, we need to put the input-output model (6.1) in a d-step-ahead prediction form which is more suitable for design of controllers.

6.2.2 Prediction with Known Models

If a model for the evolution of the time series of interest is known, it is possible to construct suitable predictors from the model. In fact, in the case that the model is linear and finite-dimensional, suitable predictors can be obtained by simple algebraic manipulations of the model. We show how this is achieved below.

Consider a linear discrete-time system described by model (6.2) as

$$A(q^{-1})y(k) = B(q^{-1})u(k) = q^{-d}B'(q^{-1})u(k) \tag{6.10}$$

Then the output of the system at time $t+d$ can be expressed in the following predictor form:

$$y(k+d) = \alpha(q^{-1})y(k) + \beta(q^{-1})u(k) \tag{6.11}$$

where

$$\begin{aligned} \alpha(q^{-1}) &= G(q^{-1}) \\ \beta(q^{-1}) &= F(q^{-1})B'(q^{-1}) \end{aligned} \tag{6.12}$$

and $F(q^{-1})$ and $G(q^{-1})$ are the unique polynomials satisfying

$$\begin{aligned} 1 &= F(q^{-1})A(q^{-1}) + q^{-d}G(q^{-1}) \\ F(q^{-1}) &= 1 + f_1q^{-1} + \cdots + f_{d-1}q^{-d+1} \\ G(q^{-1}) &= g_0 + g_1q^{-1} + \cdots + g_{n-1}q^{-n+1} \end{aligned} \tag{6.13}$$

Proof

Multiplying (6.10) by $F(q^{-1})$ gives

$$F(q^{-1})A(q^{-1})y(k) = F(q^{-1})B(q^{-1})u(k) = q^{-d}F(q^{-1})B'(q^{-1})u(k)$$

Using (6.13), we have

$$(1 - q^{-d}G(q^{-1}))y(k) = q^{-d}F(q^{-1})B'(q^{-1})u(k)$$

which can be rewritten as

$$y(k) = q^{-d}G(q^{-1})y(k) + q^{-d}F(q^{-1})B'(q^{-1})u(k)$$

or

$$y(k+d) = \alpha(q^{-1})y(k) + \beta(q^{-1})u(k)$$

where

$$\begin{aligned} \alpha(q^{-1}) &= G(q^{-1}) \\ \beta(q^{-1}) &= F(q^{-1})B'(q^{-1}) \end{aligned}$$

How to solve equation (6.13)? Does this equation look familiar to you? If you compare it with the equation (5.14), you may realize that it is also a Diophantine equation, and you know how to solve it.

There is also another natural way to convert the original input-output model into predictor form as shown in the following example.

Example 6.1

Consider the system

$$y(k+2) = y(k+1) + u(k)$$

as mentioned before, since y(k+1) appears in the right hand, it is not in the proper form of predictor, in which all the terms in the right hand have to be from the present or the past.
So how to deal with the "future term", y(k+1)?
A simple answer is just expressing y(k+1) in terms of information in the past!

$$y(k+1) = y(k) + u(k-1)$$

Then plug it into the original equation, we have

$$y(k+2) = y(k) + u(k) + u(k-1)$$

What if the delay $d > 2$? like

$$y(k+3) = y(k+2) + y(k+1) + u(k)$$

You just repeat above procedure! First express y(k+2)in terms of y(k+1), y(k), u(k) etc. Then express y(k+1) in terms of y(k), y(k-1), u(k),u(k-1), etc.

From this example, you can see how easy it is to convert the equation into predictor form! Personally, I would rely on this natural way to do the job instead of solving the Diophantine equation.

6.2.3 One-Step-Ahead Control: General Case

Now we are ready to use the predictor model (6.11) to design the feedback controller, which brings the output at time $k+d$, $y(k+d)$, to some desired bounded value $r(k+d)$ as follows. Set

$$r(k+d) = \alpha(q^{-1})y(k) + \beta(q^{-1})u(k) \tag{6.14}$$

in which $u(k)$ can be easily solved.

The resulting closed-loop system is described by

$$y(k) = r(k); \quad k \geqslant d \tag{6.15}$$

$$B(q^{-1})u(k) = A(q^{-1})r(k); \qquad k \geqslant d+n \tag{6.16}$$

The resulting closed-loop system has bounded inputs and outputs provided that the system has a stable inverse, i.e., the zeros of the polynomial lie inside the closed unit disk.

Example 6.2

Consider a simple system with disturbance described by the following state-space equations:

$$\begin{aligned}
&\omega(k+1) = \omega(k); \qquad && k \geqslant 0 \quad \texttt{(disturbance model)}\\
&x(k+1) = u(k); && k \geqslant 0 \quad \texttt{(system model)}\\
&y(k) = x(k) + \omega(k) && k \geqslant 0 \quad \texttt{(output)}
\end{aligned}$$

Let's first derive the input-output model

$$\begin{aligned}
y(k+1) &= x(k+1) + \omega(k+1)\\
&= u(k) + \omega(k)\\
&= u(k) + (y(k) - x(k))\\
&= y(k) + u(k) - u(k-1)
\end{aligned}$$

The one-step-ahead control law (6.14) is then

$$u(k) = r(k+1) - y(k) + u(k-1)$$

and this gives a closed-loop system characterized by

$$\begin{aligned}
&y(k) = r(k)\\
&u(k) - u(k-1) = r(k+1) - r(k)
\end{aligned}$$

Note that although the zero is not strictly inside the unit disk, the inputs will still be bounded, which implies that the uncontrollable pole (z=1 in this example) can lie on the unit disk.

6.2.4 Weighted One-Step-Ahead Control

We note in passing that the one-step-ahead feedback control law (6.14) minimizes the following cost function comprising the squared prediction error:

$$J_1(k+d) = \frac{1}{2}[y(k+d) - r(k+d)]^2 \tag{6.17}$$

As noted in Section 6.1, excessive control effort may be called for to bring $y(k+d)$ to $r(k+d)$ in one step. We therefore consider a slight generalization of the cost function (6.17) to the form (6.18) given below, which aims to achieve a compromise between bringing $y(k+d)$ to $r(k+d)$ and the amount of effort expended.

$$J_2(k+d) = \frac{1}{2}[y(k+d) - r(k+d)]^2 + \frac{\lambda}{2}u(k)^2 \tag{6.18}$$

We shall call the control law minimizing (6.18) at each time instant a weighted one-step-ahead controller. The term "one-step-ahead" is still used because we aim to minimize (6.18) on a one-step basis rather than in an average sense over an extended time horizon.

Substituting (6.11) into (6.18) gives

$$J_2(k+d) = \frac{1}{2}[\alpha(q^{-1})y(k) + \beta(q^{-1})u(k) - r(k+d)]^2 + \frac{\lambda}{2}u(k)^2 \tag{6.19}$$

Differentiating with respect to $u(k)$ and setting the result to zero gives

$$\beta_0[\alpha(q^{-1})y(k) + \beta(q^{-1})u(k) - r(k+d)] + \lambda u(k) = 0 \tag{6.20}$$

the control input $u(k)$ can be easily obtained as

$$u(k) = \frac{\beta_0[r(k+d) - \alpha(q^{-1})y(k) - \beta'(q^{-1})u(k-1)]}{\beta_0^2 + \lambda} \tag{6.21}$$

where

$$\beta'(q^{-1}) = q(\beta(q^{-1}) - \beta_0) = \beta_1 + \beta_2 q^{-1} + \cdots$$

The control law is (6.20). This equation can also be written using the predictor representation (6.11) as

$$\beta_0[y(k+d) - r(k+d)] + \lambda u(k) = 0 \tag{6.22}$$

Multiplying by $A(q^{-1})$ gives

$$\beta_0[A(q^{-1})y(k+d) - A(q^{-1})r(k+d) + \frac{\lambda}{\beta_0}A(q^{-1})u(k)] = 0$$

and using (6.10) results in

$$\beta_0[B'(q^{-1})u(k) - A(q^{-1})r(k+d) + \frac{\lambda}{\beta_0}A(q^{-1})u(k)] = 0$$

It follows that

$$[B'(q^{-1}) + \frac{\lambda}{\beta_0} A(q^{-1})]u(k) = A(q^{-1})r(k+d) \tag{6.23}$$

Similarly, multiplying (6.22) by $B'(q^{-1})$ gives

$$[B'(q^{-1}) + \frac{\lambda}{\beta_0} A(q^{-1})]y(k+d) = B'(q^{-1})r(k+d) \tag{6.24}$$

Obviously, the resulting closed-loop system has bounded inputs and outputs provided that the zeros of the polynomial $[B'(z^{-1}) + \frac{\lambda}{\beta_0} A(z^{-1})]$ lie inside the unit disk.

It is important to note that the closed-loop system is stable if $[B'(z^{-1}) + \frac{\lambda}{\beta_0} A(z^{-1})]$ is stable. This represents a slight relaxation of the requirement for the one-step-ahead controller. We now have an additional degree of freedom. If $\lambda = 0$, the result reduces to the one-step-ahead result.
But there is no free lunch, in one-step-ahead control, we can achieve perfect tracking, i.e., $y(k+1) = r(k+1)$. If we want to keep lower control efforts, the weighted one-step-ahead controller will in general lead to a steady-state tracking error. This is because a compromise is made between bringing $y(k+d)$ to $r(k+d)$ and keeping $u(k)$ small.

Example 6.3

Consider a system in the form of (6.10) with $d = 1$ and

$$A(q^{-1}) = 1 - 2q^{-1}$$
$$B(q^{-1}) = 1 + 3q^{-1}$$

This system does not satisfy the condition of stable inverse, and thus $u(k)$ will not be bounded for a one-step-ahead control law. The weighted one-step-ahead controller gives a closed-loop system whose modes are the zeros of

$$1 + 3z^{-1} + \lambda(1 - 2z^{-1}) = 0$$

And the pole of the closed loop system is $z = \frac{2\lambda - 3}{\lambda + 1}$. Therefore the system is asymptotically stable for $\frac{2}{3} < \lambda < 4$.

The idea of bringing a predicted system output to a desired value is not restricted to linear models. The idea can be used with nonlinear models as well. This is particularly straightforward in the case of bilinear systems. For example, in the first-order case, the bilinear model has the simple form

$$y(k+1) = ay(k) + bu(k) + nu(k)y(k) \tag{6.25}$$

The corresponding one-step-ahead bilinear control law is simply

$$r(k+1) = ay(k) + bu(k) + nu(k)y(k) \tag{6.26}$$

which results in

$$u(k) = \frac{r(k+1) - ay(k)}{b + ny(k)}$$

The one-step-ahead control principle is very simple and may not be as general as other control laws, such as those based on pole assignment or linear optimal control. However, as we have seen above, the one-step-ahead control principle does have the advantage of being able to utilize the true nonlinear dynamical model of the process. Thus it may be better able to capture the intrinsic features of the control problem than a control law based on a linear approximation. And the strongest point of this type of controller is that it can track arbitrary desired output $r(k)$, whereas other controllers may not be able to accomplish.

As discussed earlier, the motivation for introducing the weighted one-step-ahead control is that we need to achieve a compromise between bringing y(k + d) to r(k + d) and the cost of the control effort in the real world control system applications. This actually is the starting point of the so called "Model Predictive Control", which will be covered in greater details in part II of this module.

6.3 Further Beyond

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 you are interested in exploring more in this topic, you can study our module: **EE6105 Non-linear Dynamics and Control**.

If the model is unknown, or partially known, the first option is to use model-free controller such as the classical 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, which is covered by our modules: **EE5104/EE6104 Adaptive control systems**.

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, which are covered by our module: **EE5102 Multivariable Control Systems**.

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. And we have another module devoted to this topic: **EE5105 Optimal Control Systems**.

How to deal with time-varying system? That is a very challenging problem, and currently we do not have a systematic way in dealing with this. In the past decade or so, a new field called hybrid systems, which involves multiple models, switching, and tuning, has emerged as a promising tool in coping with time-varying system. We will try to offer this module to you in the future.

6.4 Notes and References

Most of the materials covered in this chapter can be found in the following book

- G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control,* Prentice Hall, 1984.