Real-Time Computer Control

ECE411 Course Notes, January 2015

Contents

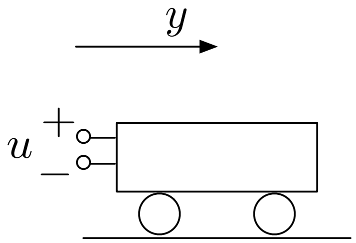
Chapter 0

Introduction

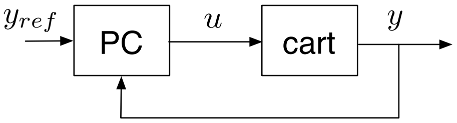
The course is mainly about digital control systems—the controllers are digital computers.

0.1 Example

Let's begin with a simple example from (ECE311 or ECE356). Consider a cart with a motor drive:

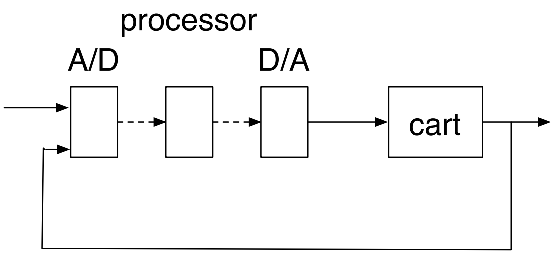


It has an input $u(t)$, the voltage to the motor, and an output $y(t)$, the position of the cart. Suppose this system is controlled by a PC with an A/D card:



Thus the position $y(t)$ is sampled and these sampled values are read in to the computer as a periodic data stream. The PC does some appropriate discrete-time control computations. Then discrete-time control values are output to become $u(t)$. The reference signal $y_{ref}(t)$ is shown as another continuous-time input to the PC, but it could also be generated in software.

Thus a digital control system involves both continuous-time signals and discrete-time signals. Let's see this in more detail by modelling the components of the PC from the control viewpoint:



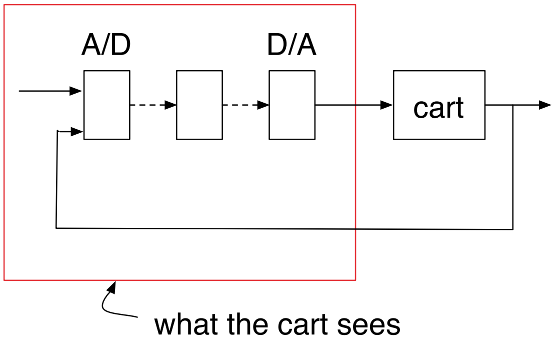
Continuous-time signals are shown as continuous lines and discrete-time signals as dashed lines. The simplest example would be this: The position $y(t)$ is sampled to produce $y(kT)$, where T is the sampling period and k an integer—the discrete-time independent variable. Likewise for $y_{ref}(kT)$. Then proportional error feedback is computed:

$$u(kT) = K[y_{ref}(kT) - y(kT)].$$

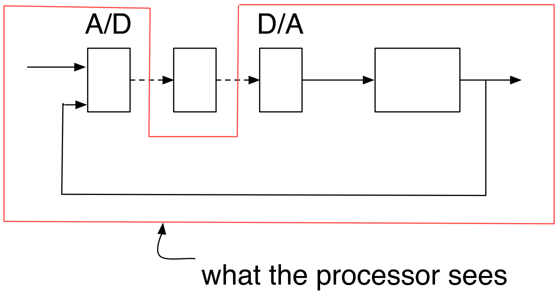
Here, K is a controller gain. Then $u(kT)$ is converted to $u(t)$ by interpolation, the simplest method being to hold the sampled value for the period T:

$$u(t) = u(kT), \quad kT \leq t < (k+1)T.$$

Notice that the cart sees a continuous-time system:



On the other hand, the processor sees a discrete-time system:



Thus in this course we will be dealing with both discrete-time and continuous-time systems and their interconnections.

0.2 Some Themes in the Course

1. Discrete-time signals and systems theory is similar to, and sometimes simpler than, the continuous-time theory.
   (a) Discrete time has difference equations instead of differential equations.
   (b) Discrete time has z-transforms instead of Laplace transforms.
   (c) Both domains have state models.
   (d) The concepts of controllability, stabilizability, observability, detectability are very similar in the two domains.
   (e) The problems of stabilization, tracking/regulation, and optimal control are very similar in the two domains.
2. It's the interface elements S (sampler) and H (hold) that complicate things. Sampled-data systems are periodically time varying, not time invariant.
3. Good block diagrams are very helpful. We'll use dotted arrows in discrete time and continuous arrows in continuous time.
4. Digital control design is not more difficult than analog unless there are hardware limitations.

0.3 General Comments

In the past, control systems were implemented using analog devices and circuits. Some disadvantages of analog controllers:

1) Inflexible: Changes in control or system require rewiring; difficult to maintain.

2) Difficult to implement sophisticated controllers.

When computers became cheap enough, they replaced analog controllers. They were referred to as digital controllers.

Nowadays, computer control refers to the use of computers in many aspects of system operation and control:

1. Sequencing: Most often in consumer electronics, e.g., your microwave oven. There is little dynamics or interaction with the environment.
2. Digital control: Carry out the actual control of the system, including control law implementation, control signal generation, signal acquisition, etc.
3. Distributed control: Implementation of microcontrollers at different locations as simple control loops to achieve an overall objective.
4. Supervision and control: Monitors the status of various processes (hardware and software) in addition to controlling the physical processes.

In addition, the computer may also be used for other tasks not directly related to control, e.g., data logging and presentation, interaction with the human operator, etc.

What is real-time?

This refers to situations where tasks must be executed and completed on time. For example, when a pilot hits a button to bring out the landing gears, the computer will need to complete that task within a specified period of time, regardless of what else it may be doing. Since the computer may be involved in doing many tasks, the computer program or the operating system would need to provide mechanisms to ensure satisfaction of real-time requirements.

Real-time computer control

To design a real-time computer control system requires some knowledge of

1. Discrete-time system analysis: No matter how fast a computer may operate, it executes at discrete instants of time. The computer uses only samples of signals. Therefore we need to understand sampling and its effects, how to generate a discrete-time system by sampling a continuous-time system. As in continuous time systems, we need to study discrete-time input-output models, transfer function models, and state space models.

2. Design and analysis of digital controllers: You have taken at least one basic course on design of controllers for continuous time systems, using frequency domain and/or state space techniques. What are the techniques for designing digital controllers and how do we analyze their performance? What are the important differences, if any, that distinguish digital control design from continuous time control design?

3. Implementation of digital controllers: We need to understand issues related to the selection of sampling rates, quantization effects associated with the processor, and controller realization forms to reduce numerical errors.

4. Real-time aspects: Real-time constraints, real-time operating systems, scheduling, priorities, analysis and measurement of latency, timing analysis.

In addition, in advanced applications, a computer controller sometimes executes logic decisions as well as digital control, especially when it performs both supervision and control functions. This gives rise to new types of dynamic systems called discrete event systems and hybrid systems, which are active areas of current research.

This course is mostly about digital control; 20% on real-time aspects.

0.4 A Bit of History

1. Pioneering period 1955-59: Owes much to chemical process industry. Computers at the time were very expensive. TRW (Thomson Ramo Woodridge) contacted Texaco to do a feasibility study on a polymerization plant, cost around \$1 M. Too expensive unless the process is complex and very expensive to run. Mainly used for finding optimal operating conditions and changing set-points on analog controllers. Computers were of course slow and unreliable (mean time between failures was just hours or days). Supported by computer manufacturers who saw a potential market.

2. Direct digital control (DDC) 1962: Imperial Chemical Industries in England replaced a complete analog implementation with a single computer. The computer simulated the analog controllers, but also added control loops and operator displays. Changing control laws was easy. Successful, as the structure of the problem had already been defined by the analog controllers. Unreliability of the central computer was an issue but a second computer could be installed as backup.

3. Transistors helped to bring in minicomputers 1967: Reduced costs greatly to $100,000. Allowed computer control to be implemented for less complex applications. Can be installed close to the process. Special process control computers were introduced. Considerable improvement in speed, data storage capacity, and reliability.

4. Microcomputers 1970s: Card-mounted computers reduced cost down to $1000 per unit. Spurred implementation of digital controllers

5. 1980s - 1990s: General use in all areas of control. Microcontrollers used in consumer electronics (CD players, etc.)

6. 1990s - 2000s: Embedded systems and distributed control, control networks consisting of many computers and devices communicating to each other through a communication network and protocol (e.g., BMWs).

Chapter 1

Analysis and Control of Continuous-Time Linear Systems

This chapter goes over some mathematical background and some continuous-time control theory. Ideally the chapter would be a review. Students who have taken (ECE311 or ECE356) and (ECE410 or ECE557) will have seen most of this chapter before; maybe not residues, unless from a complex variables course, and maybe not the Jordan form. Students who have taken only (ECE311 or ECE356) will find much that is new. Strictly speaking, you don't have to know continuous-time state-space theory to learn discrete-time state-space theory—but it certainly helps. Try to read the notes, then ask for help if needed.

1.1 Laplace Transforms

The Laplace transform is the fundamental tool used in control systems to get to the frequency domain.[1] Here we go over the definition, the conditions for existence, the inversion formula using residues, and the final-value theorem.

Let $x(t)$ be a continuous-time signal. Thus x is a function $\mathbb{R} \longrightarrow \mathbb{R}$ that maps time t to a value $x(t)$. Assume x is piecewise continuous and bounded by some exponential, that is,

$$|x(t)| \leq M\mathrm{e}^{at}, \quad t \geq 0$$

for some $M \geq 0$ and $a \in \mathbb{R}$. Its Laplace transform is then defined to be

$$X(s) = \int_0^\infty x(t)\mathrm{e}^{-st}dt,$$

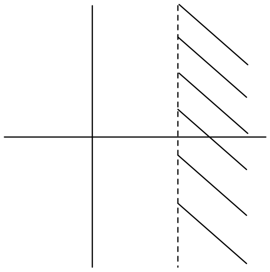
where s is a complex variable. The function $x(t)$ may or may not be zero for negative time. The Laplace transform is one-sided and ignores the value of $x(t)$ for $t < 0$. Letting $\Re$ denote real part,

[1] This is in contrast to communication theory where the Fourier transform dominates. Control problems frequently involve unstable systems, unbounded signals, and transient response requirements—all absent in communication problems.

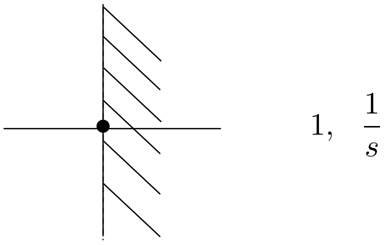
we have

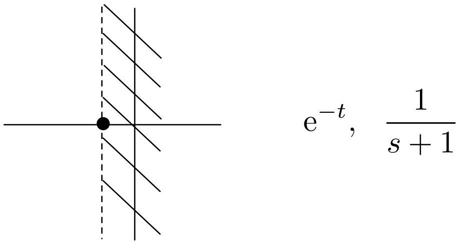
$$\begin{aligned} |X(s)| &\leq \int_0^\infty \left|x(t)\mathrm{e}^{-st}\right| dt \\ &\leq \int_0^\infty M \left|\mathrm{e}^{at}\mathrm{e}^{-st}\right| dt \\ &= \int_0^\infty M \left|\mathrm{e}^{-(s-a)t}\right| dt \\ &= \int_0^\infty M\mathrm{e}^{-((\Re s)-a)t} dt, \end{aligned}$$

Thus $X(s)$ exists if $\Re s - a > 0$, that is, $\Re s > a$. Thus the region of convergence (ROC) of the Laplace integral is a right half-plane, like this:



Familiar examples are the constant (or unit step) $x(t) = 1$ and the decaying exponential $x(t) = \mathrm{e}^{-t}$:





Notice that $X(s)$ has no poles inside the ROC, but it has a pole on the boundary. For example, if

$$X(s) = \frac{s^2+1}{(s-1)^2(2s+3)},$$

then the right-most pole is $s = 1$ and so the ROC must be $\Re s > 1$.

The preceding summary doesn't include the impulse $\delta(t)$; it is not piecewise continuous (indeed, it isn't a function $\mathbb{R} \longrightarrow \mathbb{R}$) and requires special handling. The proper way is via the theory of distributions due to Laurent Schwartz. In any case, in control systems it seems that discussion of the Laplace transform of $\delta(t)$ can be avoided altogether.

Now we turn to residues. This is a topic in complex variables related to Cauchy's theorem.

Example

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

This function has a pole at $s = -1$. At all other points it's perfectly well defined. For example, near $s = 0$ it has a Taylor series expansion:

$$F(s) = F(0) + F'(0)s + \frac{1}{2}F''(0)s^2 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!} F^{(k)}(0)s^k.$$

Near $s = 1$ it has a different Taylor series expansion:

$$F(s) = F(1) + F'(1)(s-1) + \frac{1}{2}F''(1)(s-1)^2 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!} F^{(k)}(1)(s-1)^k.$$

And so on. Only at $s = -1$ does it not have a Taylor series. Instead, it has a Laurent series expansion, where we have to take negative indices:

$$F(s) = \sum_{k=-\infty}^{\infty} c_k (s+1)^k.$$

In fact, equating

$$\frac{1}{s+1} = \sum_{k=-\infty}^{\infty} c_k (s+1)^k$$

and matching coefficients, we see that $c_k = 0$ for all k except $c_{-1} = 1$. The coefficient c_{-1} is called the **residue** of $F(s)$ at the pole $s = -1$. □

Example

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

This has a pole at $s = 0$ and another at $s = -1$. At all points except these two, $F(s)$ has a Taylor series. The Laurent series at $s = 0$ has the form

$$F(s) = \sum_{k=-\infty}^{\infty} c_k s^k.$$

To determine these coefficients, first do a partial-fraction expansion:

$$F(s) = \frac{1}{s(s+1)} = \frac{1}{s} - \frac{1}{s+1}.$$

Then do a Taylor series expansion at $s = 0$ of the second term:

$$F(s) = \frac{1}{s} - 1 + s + \cdots .$$

Thus the residue of $F(s)$ at $s = 0$ is $c_{-1} := 1$. Similarly, to get the residue at the pole $s = -1$, start with

$$F(s) = \frac{1}{s} - \frac{1}{s+1}$$

but now do a Taylor series expansion at $s = -1$ of the first term:

$$F(s) = -\frac{1}{s+1} - 1 - (s+1) - (s+1)^2 - \cdots .$$

Thus the residue of $F(s)$ at $s = -1$ is $c_{-1} := -1$. □

More generally, if p is a simple pole of $F(s)$, then the residue equals

$$\lim_{s \to p} (s-p)F(s).$$

Example

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

This has a pole at $s = 0$ of multiplicity 2 and a simple pole at $s = -1$. Partial-fraction expansion looks like

$$F(s) = \frac{1}{s^2(s+1)} = \frac{A}{s^2} + \frac{B}{s} + \frac{C}{s+1}.$$

We can get A and C by the usual coverup method, e.g.,

$$A = s^2 F(s)\big|_{s=0} = 1.$$

The formula for B is

$$B = \frac{d}{ds}\left(s^2 F(s)\right)\bigg|_{s=0} = -1.$$

Thus for this function, the residue at the pole $s = 0$ is $B = -1$. □

Residues are very useful in the **inversion of the Laplace transform**:

$$x(t) = \sum \text{ residues of } X(s)\mathrm{e}^{st} \text{ at all its poles}, \quad t \geq 0.$$

Example: $X(s) = \frac{1}{s(s-1)}$ has two poles and e^{st} has none; thus for $t \geq 0$

$$x(t) = \mathrm{Res}_{s=0} \frac{1}{s(s-1)}\mathrm{e}^{st} + \mathrm{Res}_{s=1} \frac{1}{s(s-1)}\mathrm{e}^{st} = -1 + \mathrm{e}^{t}.$$

Finally, the final-value theorem:

Theorem 1 *Suppose $X(s)$ is rational.*

1. *If $X(s)$ has no poles in $\Re s \geq 0$, then $x(t)$ converges to 0 as $t \to \infty$.*
2. *If $X(s)$ has no poles in $\Re s \geq 0$ except a simple pole at $s = 0$, then $x(t)$ converges as $t \to \infty$ and $\lim_{t\to\infty} x(t)$ equals the residue of $X(s)$ at the pole $s = 0$.*
3. *If $X(s)$ has a repeated pole at $s = 0$, then $x(t)$ doesn't converge as $t \to \infty$.*
4. *If $X(s)$ has a pole at $\Re s \geq 0, s \neq 0$, then $x(t)$ doesn't converge as $t \to \infty$.*

Some examples: $X(s) = \dfrac{1}{s+1}$: final value equals 0; $X(s) = \dfrac{2}{s(s+1)}$: final value equals 2; $X(s) = \dfrac{1}{s^2+1}$: no final value. Remember that you have to *know* that $x(t)$ has a final value, by examining the poles of $X(s)$, before you calculate the residue of $X(s)$ at the pole $s = 0$ and claim that that residue equals the final value.

1.2 Matrix Theory

Besides the Laplace transform, linear algebra and matrix theory are fundamental in linear control theory. We don't review linear algebra, but begin with eigenvalues.

First, some notation. Usually, a vector is written as a column vector, but sometimes to save space it is written as an n-tuple:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \text{ or } x = (x_1, \ldots, x_n).$$

The **spectrum** of a square real $n \times n$ matrix A, denoted $\sigma(A)$, is its set of eigenvalues. The spectrum consists of n numbers, in general complex, i.e., the zeros of the characteristic polynomial $\det(sI - A)$. Example:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 2 & -2 \end{bmatrix}.$$

The characteristic polynomial is $s^3(s+3)$, and therefore

$$\sigma(A) = \{0, 0, 0, -3\}.$$

Spectral theory is the study of the eigenvalues and eigenvectors, the deepest result being the Jordan form of A.

Let's begin with the simplest case, where A is 2×2 and has 2 distinct eigenvalues, λ_1, λ_2. It can be proved that there are then 2 linearly independent eigenvectors, say v_1, v_2 (maybe complex vectors). The equations

$$Av_1 = \lambda_1 v_1, \quad Av_2 = \lambda_2 v_2$$

are equivalent to the matrix equation

$$A \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

that is, $AV = VA_{JF}$, where

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix}, \quad A_{JF} = \text{diag}\ (\lambda_1, \lambda_2).$$

The latter matrix is the **Jordan form** of A. It is unique up to reordering of the eigenvalues. The mapping $A \longmapsto A_{JF} = V^{-1}AV$ is called a similarity transformation. Example:

$$A = \begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}, \quad A_{JF} = \begin{bmatrix} 0 & 0 \\ 0 & -3 \end{bmatrix}.$$

Corresponding to the eigenvalue $\lambda_1 = 0$ is the eigenvector $v_1 = (1, 1)$, the first column of V. All other eigenvectors corresponding to λ_1 have the form cv_1, $c \neq 0$. We call the subspace spanned by v_1 the eigenspace corresponding to λ_1. Likewise, $\lambda_2 = -3$ has a one-dimensional eigenspace.

These results extend from $n = 2$ to general n. Note that in the preceding result we didn't actually need distinctness of the eigenvalues — only linear independence of the eigenvectors.

The great thing about diagonalization is that the equation $\dot{x} = Ax$ can be transformed via $w = V^{-1}x$ into $\dot{w} = A_{JF}w$, that is, n **decoupled** equations:

$$\dot{w}_i = \lambda_i w_i, \quad i = 1, \ldots, n.$$

The latter equations are trivial to solve:

$$w_i(t) = \mathrm{e}^{\lambda_i t} w_i(0), \quad i = 1, \ldots, n.$$

In other words, the transition matrix $\mathrm{e}^{A_{JF}t}$ is the diagonal matrix

$$\mathrm{e}^{A_{JF}t} = \text{diag}\ \left(\mathrm{e}^{\lambda_1 t}, \ldots, \mathrm{e}^{\lambda_n t}\right).$$

Now we move to matrices with repeated eigenvalues where there are not n linearly independent eigenvectors. Examples are

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{1.1}$$

For both of these matrices, $\sigma(A) = \{0, 0, 0\}$. For the first matrix, the eigenspace is two-dimensional and for the second matrix, one-dimensional. These are examples of nilpotent matrices: A is **nilpotent** if $A^k = 0$ for some $k \geq 1$. The following statements are equivalent:

1. A is nilpotent.

2. All its eigs are 0.

3. Its characteristic polynomial is s^n.

4. It is similar to a matrix of the form (1.1), where all elements are 0's, except 0's or 1's on the first diagonal above the main one. This is called the Jordan form of the nilpotent matrix.

Example Here we do an example of transforming a nilpotent matrix to Jordan form. Suppose A is 3×3, $A^3 = 0, A^2 \neq 0$. There must exist a vector x such that $A^2 x \neq 0$. Then define

$$v_1 = A^2 x, \quad v_2 = Ax, \quad v_3 = x.$$

These three vectors are linearly independent (prove it), and

$$Av_1 = 0, \quad Av_2 = v_1, \quad Av_3 = v_2.$$

Define the matrix

$$V = \left[\begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \right].$$

Then

$$V^{-1}AV = A_{JF} = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right].$$

The vector v_1 is an eigenvector, while v_2, v_3 are called generalized eigenvectors. □

A nilpotent matrix has only the eigenvalue 0. Extending the previous example, now consider a matrix A that has only one eigenvalue, λ, i.e.,

$$\det(sI - A) = (s - \lambda)^n.$$

To simplify notation, suppose $n = 3$. Letting $r = s - \lambda$, we have

$$\det[rI - (A - \lambda I)] = r^3,$$

i.e., $A - \lambda I$ has only the zero eigenvalue, and hence $A - \lambda I =: N$, a nilpotent matrix. So the Jordan form of N must look like

$$\left[\begin{array}{ccc} 0 & \star & 0 \\ 0 & 0 & \star \\ 0 & 0 & 0 \end{array} \right],$$

where each star can be 0 or 1, and hence the Jordan form of A is

$$\left[\begin{array}{ccc} \lambda & \star & 0 \\ 0 & \lambda & \star \\ 0 & 0 & \lambda \end{array} \right], \tag{1.2}$$

To recap, if A has just one eigenvalue, λ, then its Jordan form is $\lambda I + N$, where N is a nilpotent matrix in Jordan form.

An extension of this analysis results in the **Jordan form** in general. Suppose A is $n \times n$ and $\lambda_1, \ldots, \lambda_p$ are the distinct eigenvalues of A and $m_1, \ldots, m_p$ are their multiplicities; that is, the characteristic polynomial is

$$\det(sI - A) = (s - \lambda_1)^{m_1} \cdots (s - \lambda_p)^{m_p}.$$

Then A is similar to

$$A_{JF} = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_p \end{bmatrix}, \quad A_i = \lambda_i I + N,$$

where A_i has only the eigenvalue λ_i, of multiplicity m_i. Thus A_i has the form $\lambda_i I + N_i$, where N_i is a nilpotent matrix in Jordan form.. Example:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 2 & -2 \end{bmatrix}$$

As we saw, the spectrum is $\sigma(A) = \{0, 0, 0, -3\}$. Thus the Jordan form must be of the form

$$A_{JF} = \begin{bmatrix} 0 & \star & 0 & 0 \\ 0 & 0 & \star & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}.$$

Since A has rank 2, so does A_{JF}. Thus only one of the stars is 1. Either is possible, for example,

$$A_{JF} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}.$$

This has two "Jordan blocks":

$$A_{JF} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_2 = -3.$$

1.3 State Models

Linear control theory is based on transfer function models and state models. We now focus on the latter. Systems that are linear, time-invariant, causal, finite-dimensional, and having proper transfer functions have state models,

$$\dot{x} = Ax + Bu, \quad y = Cx + Du.$$

Here u, x, y are vector-valued functions of t and A, B, C, D real constant matrices.

1.3.1 Deriving State Models

How to get a state model depends on what we have to start with.

Example n^{th} order ODE. Suppose we have the system

$$2\ddot{y} - \dot{y} + 3y = u.$$

The natural state vector is

$$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} =: \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Then

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{2}x_2 - \frac{3}{2}x_1 + \frac{1}{2}u \ , \end{aligned}$$

so

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{3}{2} & \frac{1}{2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0.$$

This technique extends to

$$a_n y^{(n)} + \cdots + a_1 \dot{y} + a_0 y = u.$$

What about derivatives on the right-hand side:

$$2\ddot{y} - \dot{y} + 3y = \dot{u} - 2u?$$

The transfer function is

$$Y(s) = \frac{s-2}{2s^2 - s + 3} U(s).$$

Introduce an intermediate signal v:

$$Y(s) = (s-2)\underbrace{\frac{1}{2s^2 - s + 3}U(s)}_{=:V(s)}.$$

Then

$$\begin{aligned} &2\ddot{v} - \dot{v} + 3v = u \\ &y = \dot{v} - 2v. \end{aligned}$$

Taking $x = (v, \dot{v})$ we get

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{3}{2} & \frac{1}{2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} -2 & 1 \end{bmatrix}, \quad D = 0.$$

This technique extends to

$$y^{(n)} + \cdots + a_1 \dot{y} + a_0 y = b_{n-1} u^{(n-1)} + \cdots + b_0 u, \quad m < n.$$

The transfer function is

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}.$$

Then

$$G(s) = C(sI - A)^{-1}B,$$

where

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} b_0 & \cdots & b_{n-1} \end{bmatrix}.$$

This state model is called the **controllable (canonical) realization** of $G(s)$.

Another realization of $G(s)$ is

$$(A_o, B_o, C_o, D) = (A^T, C^T, B^T, D),$$

called the **observable realization**. You can verify that

$$C_o(sI - A_o)^{-1}B_o + D = B^T(sI - A^T)^{-1}C^T + D = G(s).$$

You can think about the case $n = m$. This gives $D \neq 0$. If $m > n$, there is no state model.

What if we have two inputs u_1, u_2, two outputs y_1, y_2, and coupled equations such as

$$\ddot{y}_1 - \dot{y}_1 + \dot{y}_2 + 3y_1 = u_1 + u_2$$

$$2\frac{d^3 y_2}{dt^3} - \dot{y}_1 + \dot{y}_2 + 4y_1 = u_2.$$

The natural state is

$$x = (y_1, \dot{y}_1, y_2, \dot{y}_2, \ddot{y}_2).$$

□

The most general case of this type would have m inputs, $u_1, \ldots, u_m$, p outputs, $y_1, \ldots, y_p$, and coupled ODEs. We could take Laplace transforms and collect terms to get the form

$$D(s)Y(s) = N(s)U(s),$$

where $D(s), N(s)$ are matrices whose elements are polynomials in s. The usual case is where $D(s)$ is square and invertible. This gives

$$Y(s) = \underbrace{D(s)^{-1}N(s)}_{G(s)} U(s).$$

Here $G(s)$ is a $p \times m$ matrix whose elements are rational functions is s (ratio of polynomials). So now the problem reduces to getting a state model from the transfer matrix $G(s)$.

Let's explore this further by getting the transfer matrix for the state model

$$\dot{x} = Ax + Bu, \quad y = Cx + Du.$$

Take Laplace transforms with zero initial conditions:

$$sX(s) = AX(s) + BU(s), \quad Y(s) = CX(s) + DU(s).$$

Eliminate $X(s)$:

$$(sI - A)X(s) = BU(s)$$

$$\Rightarrow X(s) = (sI - A)^{-1}BU(s)$$

$$\Rightarrow Y(s) = \underbrace{[C(sI - A)^{-1}B + D]}_{\text{transfer matrix}} U(s).$$

This leads to the **realization problem**: Given $G(s)$, find A, B, C, D such that

$$G(s) = C(sI - A)^{-1}B + D.$$

A solution exists iff $G(s)$ is rational and proper (every element of $G(s)$ has deg denom $\geq$ deg num). The solution is never unique.

1.3.2 Linearization

Let's recall some calculus. Consider the function

$$f : \mathbb{R}^2 \to \mathbb{R}^3 \ , \quad f(x) = y \ , \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 + 1 \\ x_1 x_2 \\ x_1^2 - x_2 \end{bmatrix}.$$

We shall linearize the equation $y = f(x)$ about a nominal point, say $x_0 = (1, -1)$. We have

$$y_0 = f(x_0) = \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}.$$

Introduce perturbations

$$x = x_0 + \delta x, \quad y = y_0 + \delta y.$$

Then using a Taylor series expansion

$$\begin{aligned}\delta y &= f(x) - f(x_0)\\ &= f(x_0 + \delta x) - f(x_0)\\ &= [f(x_0) + A\delta x + \text{higher order terms}] - f(x_0)\\ &\approx A\delta x,\end{aligned}$$

where A is the Jacobian of f at x_0:

$$A = \left.\begin{bmatrix} 1 & 0\\ x_2 & x_1\\ 2x_1 & -1\end{bmatrix}\right|_{x=x_0} = \begin{bmatrix} 1 & 0\\ -1 & 1\\ 2 & -1\end{bmatrix}.$$

So the linearized equation is $\delta y = A\delta x$.

In what sense does this approximate the equation $y = f(x)$ near $x = x_0$? The exact equation is

$$\delta y = f(x_0 + \delta x) - f(x_0)$$

and the linear approximation is

$$\delta y_a = A\delta x.$$

So the error is $\|\delta y - \delta y_a\|$. It can be proved that if f is continuously differentiable, then there exists $M > 0$ such that

$$\|\delta y - \delta y_a\| \leqslant M\|\delta x\|.$$

Thus the error $\|\delta y - \delta y_a\|$ is arbitrarily small if $\|\delta x\|$ is sufficiently small.

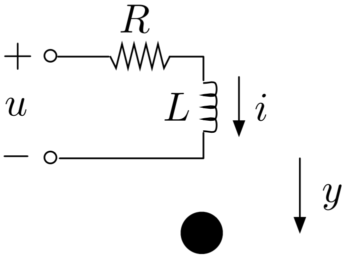
This idea extends to the nonlinear equation $\dot{x} = f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}^n$. Suppose φ is a known solution (it could be a constant solution), and let x be another solution near φ. Then the perturbation is $\delta x := x - \varphi$, and

$$\begin{aligned}\delta\dot{x} &= \dot{x} - \dot{\varphi}\\ &= f(x) - f(\varphi)\\ &= f(\varphi + \delta x) - f(\varphi)\\ &\approx A\delta x\ ,\end{aligned}$$

where A is the Jacobian of f at $x = \varphi$. We say $\delta\dot{x} = A\delta x$ is the linearization of $\dot{x} = f(x)$ at the nominal solution φ.

Finally, the idea extends to the nonlinear control equation $\dot{x} = f(x, u)$, as shown in the following example.

Magnetic levitation



Imagine an electromagnet suspending an iron ball. Let the input be the voltage u and the output the position y of ball below magnet; let i denote the current in circuit. Then

$$L\frac{di}{dt} + Ri = u.$$

Also, it can be derived that the magnetic force on the ball has the form Ki^2/y^2, K a constant. Thus

$$M\ddot{y} = Mg - K\frac{i^2}{y^2}.$$

Realistic numerical values are $M = 0.1$ Kg, $R = 15$ ohms, $L = 0.5$ H, $K = 0.0001$ Nm^2/A^2, $g = 9.8$ m/s^2. Substituting in these numbers gives the equations

$$0.5\frac{di}{dt} + 15i = u$$

$$0.1\frac{d^2y}{dt^2} = 0.98 - 0.0001\frac{i^2}{y^2}.$$

Define state variables $x = (x_1, x_2, x_3) = (i, y, \dot{y})$. Then the nonlinear state model is $\dot{x} = f(x, u)$, where

$$f(x, u) = (-30x_1 + 2u, x_3, 9.8 - 0.001x_1^2/x_2^2).$$

Suppose we want to stabilize the ball at $y = 1$ cm, or 0.01 m. We need a linear model valid in the neighbourhood of that value. Solve for the equilibrium point $(\bar{x}, \bar{u})$ where $\bar{x}_2 = 0.01$:

$$-30\bar{x}_1 + 2\bar{u} = 0, \quad \bar{x}_3 = 0, \quad 9.8 - 0.001\bar{x}_1^2/0.01^2 = 0.$$

Thus

$$\bar{x} = (0.99, 0.01, 0), \quad \bar{u} = 14.85.$$

The linearized model is

$$\dot{\delta x} = A\delta x + B\delta u, \quad \delta y = C\delta x,$$

where A equals the Jacobian of f with respect to x, evaluated at $(\bar{x}, \bar{u})$, and B equals the same except with respect to u:

$$A = \left[\begin{array}{ccc} -30 & 0 & 0 \\ 0 & 0 & 1 \\ -0.002x_1/x_2^2 & 0.002x_1^2/x_2^3 & 0 \end{array}\right]_{(\bar{x},\bar{u})} = \left[\begin{array}{ccc} -30 & 0 & 0 \\ 0 & 0 & 1 \\ -19.8 & 1940 & 0 \end{array}\right]$$

$$B = \left[\begin{array}{c} 2 \\ 0 \\ 0 \end{array}\right], \quad C = \left[\begin{array}{ccc} 0 & 1 & 0 \end{array}\right].$$

The eigenvalues of A are $-30, \pm 44.05$, the units being s^{-1}. The corresponding time constants are $1/30 = 0.033, 1/44.05 = 0.023$ s. The first is the time constant of the electric circuit; the second, the time constant of the magnetics. □

1.3.3 Solution of State Equation

Our goal is to solve

$$\dot{x} = Ax + Bu \ , \quad t \geq 0 \ , \quad x(0) = x_0. \tag{1.3}$$

The solution is the sum of two terms: one due to the initial state x_0, the other due to the input u.

For a square matrix M, the exponential e^M is defined as

$$\mathrm{e}^M := I + M + \frac{1}{2!}M^2 + \frac{1}{3!}M^3 + \cdots .$$

This series can be proved to converge. Facts:

1. e^M is invertible and $(\mathrm{e}^M)^{-1} = \mathrm{e}^{-M}$.
2. $\mathrm{e}^{M+N} = \mathrm{e}^M \mathrm{e}^N$ iff M and N commute, i.e., $MN = NM$.

The matrix function $t \longmapsto \mathrm{e}^{tA} : \mathbb{R} \to \mathbb{R}^{n \times n}$ is then defined. It has the properties

1. $\mathrm{e}^{tA}|_{t=0} = I$
2. $\dfrac{d}{dt}\mathrm{e}^{tA} = A\mathrm{e}^{tA} = \mathrm{e}^{tA}A$.

Now we solve (1.3) as follows:

$$\dot{x} - Ax = Bu.$$

Multiply by e^{-tA} :

$$\mathrm{e}^{-tA}\dot{x} - A\mathrm{e}^{-tA}x = \mathrm{e}^{-tA}Bu.$$

The left-hand side equals $\dfrac{d}{dt}[\mathrm{e}^{-tA}x(t)]$. Thus

$$\frac{d}{dt}[\mathrm{e}^{-tA}x(t)] = \mathrm{e}^{-tA}Bu(t).$$

Integrate from $t = 0$ to t :

$$\mathrm{e}^{-tA}x(t) - x(0) = \int_0^t \mathrm{e}^{-\tau A}Bu(\tau)d\tau.$$

Multiply by e^{tA}:

$$x(t) = \mathrm{e}^{tA}x(0) + \int_0^t \mathrm{e}^{(t-\tau)A}Bu(\tau)d\tau.$$

Let's summarize the main facts that follow from this. The solution of

$$\dot{x} = Ax + Bu,\ x(0) = x_0;\ y = Cx + Du$$

is

$$\begin{aligned} x(t) &= \mathrm{e}^{tA}x(0) + \int_0^t \mathrm{e}^{(t-\tau)A}Bu(\tau)d\tau \\ y(t) &= C\mathrm{e}^{tA}x(0) + \int_0^t C\mathrm{e}^{(t-\tau)A}Bu(\tau)d\tau + Du(t). \end{aligned}$$

In terms of Laplace transforms

$$\begin{aligned} X(s) &= (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s) \\ Y(s) &= C(sI - A)^{-1}x(0) + \underbrace{[C(sI - A)^{-1}B + D]}_{G(s)}\ U(s). \end{aligned}$$

If one can compute the Jordan form of A, then e^{At} can be written in closed form, as follows. The equation

$$AV = VA_{JF}$$

implies

$$A^2V = AVA_{JF} = VA_{JF}^2.$$

Continuing in this way gives

$$A^kV = VA_{JF}^k,$$

and then

$$\mathrm{e}^{At}V = V\mathrm{e}^{A_{JF}t},$$

so finally

$$\mathrm{e}^{At} = V\mathrm{e}^{A_{JF}t}V^{-1}.$$

The matrix exponential $\mathrm{e}^{A_{JF}t}$ is easy to write down. For example, suppose $A_{JF} = \lambda I + N$, N nilpotent, $n \times n$. Then

$$\begin{aligned}\mathrm{e}^{A_{JF}t} &= \mathrm{e}^{\lambda t}\mathrm{e}^{Nt} \\ &= \mathrm{e}^{\lambda t}\left(I + Nt + N^2\frac{t^2}{2!} + \cdots + N^{n-1}\frac{t^{n-1}}{(n-1)!}\right).\end{aligned}$$

1.4 Stability of State Models

Much of control design focuses on the behaviour of signals as time tends to infinity. This is closely related to the concept of stability. Consider the homogeneous linear system in state space form:

$$\dot{x} = Ax, \quad x(0) = x_0.$$

While there are more elaborate and formal definitions of stability for the above homogeneous system, we choose the following two : The system is **asymptotically stable** if $x(t) \longrightarrow 0$ as $t \longrightarrow \infty$ for all $x(0)$. The system is **stable** if $x(t)$ remains bounded as $t \longrightarrow \infty$ for all $x(0)$. Since $x(t) = \mathrm{e}^{At}x(0)$, the system is asymptotically stable if and only if (iff) every element of the matrix e^{At} converges to zero, and is stable iff every element of the matrix e^{At} remains bounded as $t \longrightarrow \infty$. Of course, asymptotic stability implies stability.

Asymptotic stability is relatively easy to characterize. Using the Jordan form, one can prove this very important result:

Lemma 1 *The system is asymptotically stable iff the eigenvalues of A all satisfy $\Re\lambda < 0$.*

Let's say the matrix A is **stable** if its eigenvalues satisfy $\Re\lambda < 0$. Then the system $\dot{x} = Ax$ is asymptotically stable iff A is stable.

Now we turn to the more subtle property of stability. We'll do some examples, and we may as well have A in Jordan form.

Consider the nilpotent matrix

$$A = N = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Obviously, $x(t) = x(0)$ for all t and so the system is stable. By contrast, consider

$$A = N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Then

$$\mathrm{e}^{Nt} = I + tN,$$

which is unbounded and so the system is not stable. This example extends to the $n \times n$ case: If A is nilpotent, the system is stable iff $A = 0$.

Here's the test for stability in general in terms of the Jordan form of A:

$$A_{JF} = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_p \end{bmatrix}.$$

Recall that each A_i has just one eigenvalue, λ_i, and that $A_i = \lambda_i I + N_i$, where N_i is a nilpotent matrix in Jordan form.

Lemma 2 *The system is stable iff the eigenvalues of A all satisfy $\Re\lambda \leq 0$ and for any eigenvalue with $\Re\lambda_i = 0$, the nilpotent matrix N_i is zero, i.e., A_i is diagonal.*

Here's an example with complex eigenvalues:

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad A_{JF} = \begin{bmatrix} j & 0 \\ 0 & -j \end{bmatrix}.$$

The system is stable since there are two 1 × 1 Jordan blocks. Now consider

$$A = \begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The eigenvalues are $j, j, -j, -j$ and so the Jordan form must look like

$$A_{JF} = \begin{bmatrix} j & \star & 0 & 0 \\ 0 & j & 0 & 0 \\ 0 & 0 & -j & \star \\ 0 & 0 & 0 & -j \end{bmatrix}.$$

Since the rank of $A - jI$ equals 3, the upper star is 1; since the rank of $A + jI$ equals 3, the lower star is 1. Thus

$$A_{JF} = \begin{bmatrix} j & 1 & 0 & 0 \\ 0 & j & 0 & 0 \\ 0 & 0 & -j & 1 \\ 0 & 0 & 0 & -j \end{bmatrix}.$$

Since the Jordan blocks are not diagonal, the system is not stable.

1.5 BIBO Stability

Bounded input, bounded output (BIBO) stability is a different stability concept. First, let's define precisely what boundedness of a signal means. Let $u(t)$ be a real-valued signal defined for $t \geq 0$. We say u is **bounded** if there exists a constant b such that, for all $t \geq 0$, $|u(t)| \leq b$. Familiar bounded signals are steps and sinusoids, but not ramps or blowing up exponentials. Then the least upper bound b is denoted $\|u\|_\infty$, the infinity-norm of u. Think of $\|u\|_\infty$ as $\max_t |u(t)|$—maximum 0 to peak.

Consider a linear time-invariant system with a single (i.e., one-dimensional) input and a single output. The system is **BIBO stable** if $y(t)$ is bounded for every bounded $u(t)$, that is, $\|u\|_\infty$ finite implies $\|y\|_\infty$ finite.

Now assume the system has a transfer function $G(s)$, rational and proper.

Lemma 3 *The system is BIBO stable iff all the poles of the transfer function $G(s)$ lie in $\Re s < 0$.*

Thus $\dfrac{1}{s+1}, \dfrac{s-1}{s+1}, \dfrac{1}{s^2+s+1}$ are BIBO stable, but not $\dfrac{1}{s}$ or $\dfrac{1}{s-1}$.

Finally, suppose the transfer function is from a state model:

$$G(s) = C(sI - A)^{-1}B + D.$$

Recall that the inverse of a matrix equals its adjoint divided by its determinant. Thus

$$(sI - A)^{-1} = \frac{1}{\det(sI - A)} \operatorname{adj}(sI - A).$$

The elements of the adjoint matrix are polynomials in s, and so any poles of any element of $(sI-A)^{-1}$ must be roots of the characteristic polynomial of A, that is, they must be eigenvalues of A. By this argument, we conclude that the poles of $G(s)$ are contained in $\sigma(A)$. Some cancellations may occur, so the containment may be proper—some eigenvalue of A may not be a pole. The conclusion is that stability of A implies BIBO stability of $G(s)$. Usually there are no cancellations, and the two stability concepts are equivalent.

1.6 Controllability

For historical reasons, controllability is defined initially as a reachability property. Reachability isn't an important control objective *per se*, but it does capture in a simple way the concept of control authority.

The system

$$\dot{x} = Ax + Bu, \quad x(0) = 0$$

is **controllable** [or the pair of matrices (A, B) is controllable] if for every target time $t_1 > 0$ and every target vector v, there is an open-loop control signal $u(t)$, $0 \leq t \leq t_1$, such that $x(t_1) = v$. Thus controllability says every state is reachable starting at the origin.

Controllability is a property just of the two matrices A and B. There is a simple algebraic test. Define the **controllability matrix**

$$W_c = \left[\begin{array}{ccccc} B & AB & A^2B & \cdots & A^{n-1}B \end{array}\right].$$

This matrix has n rows and nm columns, where n is the dimension of x and m the dimension of u. Thus the columns of W_c are vectors in state-space $\mathbb{R}^n$. It turns out that the span of these columns is exactly the set of vectors that are reachable. Thus, (A, B) is controllable iff the column span of W_c equals $\mathbb{R}^n$, and this is true iff the rank of W_c equals n. If $m = 1$, the single-input case, then (A, B) is controllable iff W_c is nonsingular.

It is convenient to say that an eigenvalue λ of A is **controllable** if

$$\operatorname{rank}\left[\begin{array}{cc} A - \lambda I & B \end{array}\right] = n.$$

This rank test is called the **PBH test**, named after Popov, Brunovsky, Hautus. Then another test for controllability is as follows: (A, B) is controllable iff each eigenvalue of A is controllable.

The most important fact about controllable systems is that their eigenvalues can be reassigned by state feedback. That is, consider applying the control signal

$$u = Fx + v, \quad v \text{ an external input}$$

to the given state model. Then (A, B) is transformed to $(A+BF, B)$. We have the celebrated **Pole Assignment Theorem**:

Theorem 2 (A, B) *is controllable iff for every set of desired eigenvalues, there exists a matrix F such that $A + BF$ has exactly that set of eigenvalues.*

Of course, the desired set of eigenvalues must have conjugate symmetry; that is, if λ is a desired eigenvalue, so too must $\bar{\lambda}$ be.

A procedure to assign eigenvalues is presented next.

A matrix of the form

$$\begin{bmatrix} 0 & 1 & & & \\ 0 & 0 & & & \\ & & \ddots & & \\ & & & 0 & 1 \\ -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \end{bmatrix}$$

is called a **companion matrix**. Its characteristic polynomial is

$$s^n + a_n s^{n-1} + \cdots + a_2 s + a_1.$$

Thus companion matrices arise naturally in going from a differential equation model to a state model. Example:

$$\frac{d^3y}{dt^3} + a_3\frac{d^2y}{dt^2} + a_2\frac{dy}{dt} + a_1 y = b_3\frac{d^2u}{dt^2} + b_2\frac{du}{dt} + b_1 u$$

$$\text{transfer function} \ = \frac{b_3 s^2 + b_2 s + b_1}{s^3 + a_3 s^2 + a_2 s + a_1}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}, \ D = 0$$

Notice that if A is a companion matrix, then there exists a vector B such that (A, B) is controllable, namely,

$$B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Example: The controllability matrix of

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

is

$$\begin{bmatrix} B & AB & A^2B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -a_3 \\ 1 & -a_3 & -a_2 + a_3^2 \end{bmatrix}.$$

The rank of the last matrix equals 3, so (A, B) is controllable.

Now we'll see that if (A, B) is controllable and B is $n \times 1$, then A is similar to a companion matrix.

Theorem 3 *Suppose (A, B) is controllable and B is $n \times 1$. Let the characteristic polynomial of A be*

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

Define

$$\tilde{A} = \begin{bmatrix} 0 & 1 & & & \\ 0 & 0 & & & \\ & & \ddots & & \\ & & & 0 & 1 \\ -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Then there exists a W such that

$$W^{-1}AW = \tilde{A}, \ W^{-1}B = \tilde{B}.$$

Proof Assume $n = 3$ to simplify the notation. The characteristic poly of A is

$$s^3 + a_3 s^2 + a_2 s + a_1.$$

By the Cayley-Hamilton theorem (a matrix satisfies its characteristic equation),

$$A^3 + a_3 A^2 + a_2 A + a_1 I = 0.$$

Multiply by B:

$$A^3 B + a_3 A^2 B + a_2 AB + a_1 B = 0.$$

Hence

$$A^3 B = -a_1 B - a_2 AB - a_3 A^2 B.$$

Using this equation, you can verify that

$$A\left[\begin{array}{ccc} B & AB & A^2B \end{array}\right] = \left[\begin{array}{ccc} B & AB & A^2B \end{array}\right] \begin{bmatrix} 0 & 0 & -a_1 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_3 \end{bmatrix}. \tag{1.4}$$

Define

$$W_c := \left[\begin{array}{ccc} B & AB & A^2B \end{array}\right], \quad M = \begin{bmatrix} 0 & 0 & -a_1 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_3 \end{bmatrix}.$$

Note that M^T is the companion matrix corresponding to the characteristic polynomial of A. From (1.4) we have

$$W_c^{-1}AW_c = M. \tag{1.5}$$

Regarding B, we have

$$B = \left[\begin{array}{ccc} B & AB & A^2B \end{array}\right] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

so

$$W_c^{-1}B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{1.6}$$

Now define

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

These are the matrices we want to transform A, B to. Define their controllability matrix $\tilde{W}_c$. Then, as in (1.5) and (1.6),

$$\tilde{W}_c^{-1}\,\tilde{A}\,\tilde{W}_c = M, \text{ (same } M!) \tag{1.7}$$

$$\tilde{W}_c^{-1}\,\tilde{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{1.8}$$

From (1.5), (1.7) and (1.6), (1.8),

$$W_c^{-1}\,AW_c = \tilde{W}_c^{-1}\,\tilde{A}\tilde{W}_c, \;\; W_c^{-1}\,B = \tilde{W}_c^{-1}\,\tilde{B}.$$

Define $W = W_c\,\tilde{W}_c^{-1}$. Then

$$W^{-1}AW = \tilde{A}, \quad W^{-1}B = \tilde{B}.$$

□

Let's summarize the steps to transform (A, B) to $(\tilde{A}, \tilde{B})$:

1. Find the characteristic poly of A:

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

2. Define W_c = controllability matrix of (A, B), $\tilde{W}_c$ = controllability matrix of $(\tilde{A}, \tilde{B})$, and $W = W_c \tilde{W}_c^{-1}$. Then $W^{-1}AW = \tilde{A},\ W^{-1}B = \tilde{B}$.

Example

$$A = \begin{bmatrix} 3 & -2 & 9 \\ -2 & 2 & -7 \\ -1 & 1 & -4 \end{bmatrix}, \quad B = \begin{bmatrix} -3 \\ 3 \\ 1 \end{bmatrix}$$

char poly $A = s^3 - s^2 - 2s + 1$

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 2 & 1 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$W_c = \begin{bmatrix} -3 & -6 & -10 \\ 3 & 5 & 8 \\ 1 & 2 & 3 \end{bmatrix}, \quad \tilde{W}_c = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix}$$

$$W = \begin{bmatrix} 2 & -3 & -3 \\ -3 & 2 & 3 \\ -1 & 1 & 1 \end{bmatrix}$$

□

Now we turn to the pole assignment problem when A is in companion form.

Example

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

Let us design $F = \begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix}$ to place the eigs of $A + BF$ at $\{-1, -2, -3\}$. We have

$$A + BF = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 + F_1 & -1 + F_2 & -1 + F_3 \end{bmatrix}.$$

Since $A + BF$ is a companion matrix, its characteristic polynomial is

$$s^3 + (1 - F_3)s^2 + (1 - F_2)s + (-1 - F_1).$$

But the desired char poly is

$$(s + 1)(s + 2)(s + 3) = s^3 + 6s^2 + 11s + 6.$$

Equating coefficients in these two equations, we get the unique F:

$$F = -\begin{bmatrix} 7 & 10 & 5 \end{bmatrix}.$$

□

The procedure to compute F to assign the eigs of $A + BF$ is therefore as follows:

1. Compute W so that

$$\tilde{A} := W^{-1}AW, \ \tilde{B} := W^{-1}B$$

have the form that $\tilde{A}$ is a companion matrix and

$$\tilde{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

2. Compute $\tilde{F}$ to assign the eigs of $\tilde{A} + \tilde{B}\tilde{F}$ to the desired locations.
3. Set $F = \tilde{F}W^{-1}$.

To see that $A + BF$ and $\tilde{A} + \tilde{B}\tilde{F}$ have the same spectra, simply note that

$$W^{-1}(A + BF)W = \tilde{A} + \tilde{B}\tilde{F}.$$

A related notion is that of stabilizability: (A, B) is **stabilizable** if there exists a matrix F such that $A + BF$ is stable, that is, all its eigenvalues are in the open left half-plane. Not surprisingly, (A, B) is stabilizable iff the (unstable) eigenvalues of A in $\Re s \geq 0$ are controllable.

Maglev example continued For the linearized model we had

$$A = \begin{bmatrix} -30 & 0 & 0 \\ 0 & 0 & 1 \\ -19.8 & 1940 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}.$$

Where to put the eigenvalues of $A + BF$? This is normally done by simulation/iteration because practical things like the ball hitting the magnet can be tested. But for an initial try, it's sensible to look at the eigenvalues of A, which are $-30, \pm 44$, the unstable one being $+44$. So our goal is to move this into the left half-plane. Now our plant is actually a toy, so there's no specified objective in terms of closed-loop speed of response. Let's say -150 is a reasonable pole location—five times that of the electric circuit. The unique F that places the three eigenvalues at -150 is

$$F = \begin{bmatrix} -210 & 107272 & 1753 \end{bmatrix}.$$

□

1.7 Observability

As we did in defining controllability as reachability, we define observability as the ideal concept of state reconstructibility. Then it will turn out to be equivalent to a more useful property.

The system

$$\dot{x} = Ax, \quad y = Cx$$

[or more commonly the pair of (C, A)] is **observable** if for every $x(0)$ and $t_1 > 0$, $x(0)$ can be computed from the data $\{y(t) : 0 \le t \le t_1\}$. Another way to say this is the initial state that produces a given output piece $\{y(t) : 0 \le t \le t_1\}$ is unique; that is, the mapping from $x(0)$ to $\{y(t) : 0 \le t \le t_1\}$ is one-to-one. The main result is that the following five conditions are equivalent:

1. (C, A) is observable.

2. (A^T, C^T) is controllable.

3. The observability matrix

$$W_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has rank n.

4. For each eigenvalue λ of A,

$$\text{rank} \begin{bmatrix} A - \lambda I \\ C \end{bmatrix} = n.$$

5. The eigenvalues of $A + LC$ can be arbitrarily assigned by suitable choice of L.

By analogy with controllable eigenvalues and in view of the fourth condition above, we say that an eigenvalue λ of A is **observable** if

$$\text{rank} \begin{bmatrix} A - \lambda I \\ C \end{bmatrix} = n.$$

If an L exists to make $A + LC$ stable, (C, A) is said to be **detectable**. The following three conditions are equivalent:

1. (C, A) is detectable.

2. (A^T, C^T) is stabilizable.

3. Every eigenvalue of A in $\Re s \ge 0$ is observable.

An **observer** is an asymptotic state estimator. Consider the system

$$\dot{x} = Ax + Bu, \quad y = Cx$$

and suppose we want to estimate x given u and y. We can do this iff (C, A) is detectable. Indeed, choose L so that $A + LC$ is stable and then define the observer to be the system

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

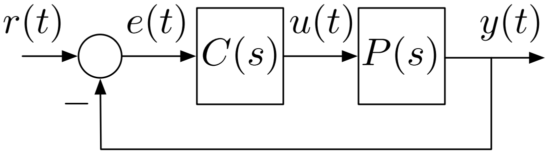
The state $\hat{x}$ is the estimate of the plant state, and the observer emulates the plant except with an additional term $C\hat{x} - y$ which is the output error. To see that the observer does work, derive the error equation

$$\frac{d}{dt}(\hat{x} - x) = (A + LC)(\hat{x} - x).$$

Thus for every $\hat{x}(0)$ and $x(0)$ and every input signal u, the error $\hat{x}(t) - x(t)$ converges to zero.

1.8 Feedback Stability

Now we turn to feedback systems. Consider this block diagram:



The transfer functions $C(s), P(s)$ represent the controller and plant respectively. They're assumed to be rational and strictly proper, though this could be relaxed. The signals are r, the reference or command input; u, the plant input; y, the plant output; and e, the tracking error. Feedback systems may have other configurations but this one is the most common.

The block diagram represents algebraic equations; for example,

$$E = R - Y, \quad Y = PU.$$

So, for example, the transfer function from r to e can be derived as follows:

$$E = R - PCE$$

$$(1 + PC)E = R$$

$$\frac{E}{R} = \frac{1}{1 + PC}.$$

In this way, the block diagram is a graphical representation of the functions $r \mapsto e$, $r \mapsto u$, and $r \mapsto y$; of the signals, r is the only "independent variable".

What does it mean for the feedback system to be stable? There are two ways to define it, and they're equivalent. The first is in terms of state models. Start with controllable, observable realizations of $P(s)$ and $C(s)$:

$$\dot{x}_p = A_p x_p + B_p u, \quad y = C_p x_p$$

$$\dot{x}_c = A_c x_c + B_c e, \quad u = C_c x_c.$$

The feedback system is said to be **internally asymptotically stable** if, when $r(t) = 0$, the states $x_p(t)$ and $x_c(t)$ converge to zero for every initial condition. To be more explicit, combine the two equations, with $r(t) = 0$:

$$\dot{x}_p = A_p x_p + B_p C_c x_c$$

$$\dot{x}_c = A_c x_c - B_c y = A_c x_c - B_c C_p x_p,$$

that is,

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A_p & B_p C_c \\ -B_c C_p & A_c \end{bmatrix} \begin{bmatrix} x_p \\ x_c \end{bmatrix}.$$

This has the form $\dot{x}_{cl} = A_{cl} x_{cl}$, the closed-loop state model with r set to 0. Thus the feedback system is internally asymptotically stable iff $A_{cl}$is stable.

The second feedback stability notion is a form of BIBO stability. Denote the numerator and denominator of P by N_p, D_p; likewise N_c, D_c for C. Then the transfer functions from r to y, e, u are

$$\frac{Y}{R} = \frac{PC}{1+PC} = \frac{\frac{N_p N_c}{D_p D_c}}{1 + \frac{N_p N_c}{D_p D_c}} = \frac{N_p N_c}{D_p D_c + N_p N_c}$$

$$\frac{E}{R} = \frac{D_p D_c}{D_p D_c + N_p N_c}$$

$$\frac{U}{R} = \frac{D_p N_c}{D_p D_c + N_p N_c}.$$

This implies that if all the roots of the polynomial $D_p D_c + N_p N_c$ are in $\Re s < 0$, then a bounded r produces bounded y, e, u—i.e., all signals in the loop. So we say the feedback system is **BIBO stable** if all the roots of the polynomial $D_p D_c + N_p N_c$ are in $\Re s < 0$, and we call this polynomial the **characteristic polynomial** of the feedback system.

It is a fact that the eigenvalues of A_{cl} are the same as the roots of the characteristic polynomial. Thus the two feedback stability definitions are equivalent, and we use the term **feedback stability** for either.

You can easily show that if upon forming the product PC an unstable pole is cancelled, then the feedback system is not stable. Example: $P(s) = \dfrac{s}{(s+1)^2}$, $C(s) = \dfrac{1}{s}$. The unstable pole of C is cancelled by the zero of P, so the feedback system is not stable.

There's a very nice way to stabilize an unstable plant using an observer. Start with the plant

$$\dot{x}_p = A_p x_p + B_p u, \quad y = C_p x_p.$$

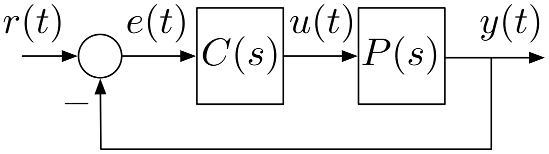
Choose F_p and L_p to stabilize $A_p + B_p F_p$ and $A_p + L_p C_p$. Write the observer equation, but denote the observer state by x_c:

$$\dot{x}_c = A_p x_c + B_p u + L_p(C x_c - y).$$

Now set $u = F_p x_c$:

$$\dot{x}_c = A_p x_c + B_p F_p x_c + L_p(C x_c - y) = (A_p + B_p F_p + L_p C_p) x_c - L_p y.$$

To fit this equation into the configuration of the block diagram



replace the input to the equation, $-y$, by the input to the controller in the diagram, $e = r - y$:

$$\dot{x}_c = (A_p + B_pF_p + L_pC_p)x_c + L_pe.$$

Thus the controller state matrices are

$$A_c = A_p + B_pF_p + L_pC_p, \quad B_c = L_p, \quad C_c = F_p. \tag{1.9}$$

To see that the feedback system is indeed stable, note that

$$A_{cl} = \left[\begin{array}{cc} A_p & B_pC_c \\ -B_cC_p & A_c \end{array}\right] = \left[\begin{array}{cc} A_p & B_pF_p \\ -L_pC_p & A_p + B_pF_p + L_pC_p \end{array}\right].$$

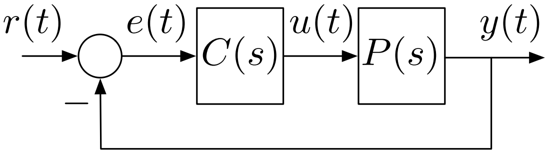
It turns out that $\sigma(A_{cl}) = \sigma(A_p + B_pF_p) \cup \sigma(A_p + L_pC_p)$. To prove this, do the following similarity transformation on A_{cl}:

$$T^{-1}A_{cl}T = \left[\begin{array}{cc} A_p + B_pF_p & B_pF_p \\ 0 & A_p + L_pC_p \end{array}\right], \quad T = \left[\begin{array}{cc} I & 0 \\ I & I \end{array}\right].$$

1.9 Tracking Steps

Besides feedback stability, another common control requirement is to be able to track a constant reference signal. A familiar example is cruise control. We set the desired car speed and then require the car to maintain that constant speed.

We continue with the block diagram



Suppose $r(t)$ is an arbitrary constant value and we require $y(t)$ to converge to that constant value. By linearity, it suffices to meet this requirement for just $r(t) = 1$. So the problem is: Given $r(t) = 1$, design $C(s)$ to achieve feedback stability and $\lim_{t\to\infty} y(t) = 1$, or equivalently, $\lim_{t\to\infty} e(t) = 0$.

Let's address this problem using transfer function methods. The Laplace transform of $r(t) = 1$ is $R(s) = 1/s$. Let $G(s)$ denote the closed-loop transfer function from r to the tracking error e. Then

$$E(s) = G(s)\frac{1}{s}.$$

By the final-value theorem, $\lim_{t\to\infty} e(t) = 0$ iff all the poles of $G(s)$ lie in $\Re s < 0$ and $G(0) = 0$. Having all the poles of $G(s)$ lie in $\Re s < 0$ will follow from the requirement of feedback stability. Since

$$G = \frac{1}{1+PC},$$

the condition $G(0) = 0$ is equivalent to the condition that $P(s)C(s)$ has a pole at $s = 0$. If P has a pole at $s = 0$, then C merely has to stabilize the feedback loop. If P does not have a pole at $s = 0$, then C must have one. If, furthermore, P has a zero at $s = 0$, then the problem isn't solvable; remember: there can be no unstable pole-zero cancellation. Finally, if P doesn't have a pole or zero at $s = 0$, we can solve the tracking problem by taking C of the form

$$C(s) = \frac{1}{s}C_1(s)$$

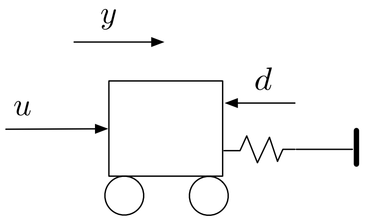
and designing C_1 to stabilize the feedback loop. For example we could design an observer-based controller $C_1(s)$ to stabilize $P(s)/s$. Thus integral control is the key to tracking constant references.

1.10 Tracking and Regulation

This section develops the state-space theory of tracking and regulation—the plant output should track a reference signal, such as a step, ramp, or sinusoid, and/or a plant disturbance should be rejected.

1.10.1 Introduction

Consider a cart/spring with control force u, disturbance force d, and position y:



We want the cart to follow a reference r in spite of the disturbance. To keep things simple, let's take the plant equation to be

$$M\ddot{y} = u - Ky - d.$$

This is an "$f = ma$ equation" where M is the mass and K the spring constant. To make things even simpler, let's take $M = K = 1$:

$$\ddot{y} = u - y - d.$$

Suppose we know that r is a constant (or a step) but we don't know its value; and we know that d is a sinusoid of frequency 10 rad/s but of unknown amplitude and phase. Then we know equations that can generate these signals, namely,

$$\dot{r} = 0, \quad \ddot{d} + 100d = 0.$$

We call these two equations together the **exomodel**, "exo" meaning "from outside the plant". The only unknowns are the initial conditions of these equations. Since we're not saying anything about the magnitudes of these signals, the most we could try to achieve is asymptotic regulation: $r(t) - y(t)$ tends to zero. We want to design a controller to achieve this. We restrict the controller to have input $r - y$ and output u.

We're going to develop a state-space theory for this problem, so it's convenient to make a state model of the plant and exomodel. The setup we want is a plant with state x_1 and an exomodel with state x_2 like this:

$$\begin{aligned} \dot{x}_1 &= A_1x_1 + A_3x_2 + B_1u \\ \dot{x}_2 &= A_2x_2 \\ e &= D_1x_1 + D_2x_2. \end{aligned}$$

The output e is the signal that we want to go to zero, typically, a tracking error. The exogenous signal x_2 also enters the plant via A_3x_2, a disturbance. It is natural to **assume** that all the eigenvalues of A_2 are unstable (but no assumption is made yet about A_1). For conciseness, the two states can be combined:

$$\dot{x} = Ax + Bu, \quad e = Dx, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}.$$

Let us set up the cart example like this. Taking the state variables

$$x_1 = (y, \dot{y}), \quad x_2 = (r, d, \dot{d})$$

and the output $e = r - y$, we have

$$A = \left[\begin{array}{rr|rrr} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -100 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \\ 0 \\ 0 \end{array}\right], \quad D = \left[\begin{array}{rr|rrr} -1 & 0 & 1 & 0 & 0 \end{array}\right].$$

The partition lines indicate the blocks

$$A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}.$$

We see that the eigenvalues of A_2 are $0, \pm 10j$, the frequencies of r and d. So A_2 is completely unstable—no stable eigenvalues.

The **regulator problem** is to design a controller, with input e and output u, such that the feedback loop is stable, meaning the plant state $x_1(t)$ and the controller state go to zero when $x_2(0) = 0$, and the output is regulated, meaning $e(t)$ goes to zero for all initial conditions.

1.10.2 Technical Preliminaries

In this section we develop the tools to solve the regulator problem. The notation is local to this section; for example, there will be an A_1 but it won't be the same as in other sections; however, A_2 will be the same.

Consider the system

$$\dot{x} = Ax, \; e = Dx, \; A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \; D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}$$

where A_1 is stable and A_2 has all its eigenvalues in the closed right half-plane. We're interested in when $e(t)$ goes to 0 for every $x(0)$. If $A_3 = 0$, that is, A is block diagonal, the question is easy: $e(t)$ goes to 0 for every $x(0)$ iff $D_2 = 0$. This follows from the equation

$$e(t) = D_1 \mathrm{e}^{A_1 t} x_1(0) + D_2 \mathrm{e}^{A_2 t} x_2(0).$$

To answer the question in general, it would be beneficial to transform A so that it becomes block diagonal. To this end, we need this result:

Lemma 4 *Assume A_1 is stable and A_2 has all its eigenvalues in the closed right half-plane. There exists a unique matrix X satisfying the equation*

$$A_1 X - X A_2 + A_3 = 0. \tag{1.10}$$

Proof If $A_2 = 0$, obviously X exists, namely, $X = -A_1^{-1} A_3$. Likewise, if $A_2 = cI$, with c a positive constant, then the equation is

$$(A_1 - cI)X + A_3 = 0.$$

This has a unique solution because c is not an eigenvalue of A_1. The proof in the general case is a bit involved and is therefore omitted. □

Using this X, we can block-diagonalize A by a similarity transformation:

$$T = \begin{bmatrix} I & X \\ 0 & I \end{bmatrix}, \; T^{-1} A T = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

Under the same transformation, D becomes

$$DT = \begin{bmatrix} D_1 & D_1 X + D_2 \end{bmatrix}.$$

Thus, $e(t)$ goes to 0 for every $x(0)$ iff $D_1 X + D_2 = 0$.

Let's summarize:

Lemma 5 *Suppose*

$$\dot{x} = Ax, \; e = Dx, \; A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \; D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}$$

where A_1 is stable and A_2 has all its eigenvalues in the closed right half-plane. Then $e(t)$ goes to 0 for every $x(0)$ iff $D_1 X + D_2 = 0$, where X is the unique solution of

$$A_1 X - X A_2 + A_3 = 0.$$

A special case is $A_2 = 0$. This will correspond to the case of constant references or disturbances. In this case $X = -A_1^{-1} A_3$. The result is this:

Corollary 4 *Suppose*

$$\dot{x} = Ax, \; e = Dx, \; A = \begin{bmatrix} A_1 & A_3 \\ 0 & 0 \end{bmatrix}, \; D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}$$

where A_1 is stable. Then $e(t)$ goes to 0 for every $x(0)$ iff $-D_1A_1^{-1}A_3 + D_2 = 0$.

This result is quite intuitive when one notices that $-D_1A_1^{-1}A_3 + D_2$ equals the DC gain matrix from x_2 to e for the system

$$\begin{aligned} \dot{x}_1 &= A_1x_1 + A_3x_2 \\ e &= D_1x_1 + D_2x_2. \end{aligned}$$

1.10.3 Regulator Problem Solution

To review, the setup is a plant with state x_1 and an exomodel with state x_2 like this:

$$\begin{aligned} \dot{x}_1 &= A_1x_1 + A_3x_2 + B_1u \\ \dot{x}_2 &= A_2x_2 \\ e &= D_1x_1 + D_2x_2. \end{aligned}$$

The two states can be combined:

$$\dot{x} = Ax + Bu, \;\; e = Dx, \;\; x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \;\; A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \;\; B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \;\; D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}.$$

We **assume** that all the eigenvalues of A_2 are unstable (but no assumption is made yet about A_1). It is also natural to **assume** that (D, A) is detectable because we're going to use an observer.

The solution is in two parts. We first look for a state feedback controller, $u = Fx$. Then we implement it via $u = F\hat{x}$ where $\hat{x}$ is from an observer with input e.

So let $u = Fx$. Then the controlled system is

$$\dot{x} = (A + BF)x, \;\; e = Dx,$$

where

$$A + BF = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \begin{bmatrix} F_1 & F_2 \end{bmatrix} = \begin{bmatrix} A_1 + B_1F_1 & A_3 + B_1F_2 \\ 0 & A_2 \end{bmatrix}.$$

Clearly feedback stability is equivalent to stability of $A_1 + B_1F_1$. Then asymptotic regulation is equivalent to the condition that $e(t)$ goes to 0 for every $x(0)$..

Theorem 5 *Assume A_2 has only unstable eigenvalues. Then the regulator problem is solvable by some $u = Fx$ iff (A_1, B_1) is stabilizable and there exist matrices X, U such that*

$$A_1X - XA_2 + A_3 + B_1U = 0, \;\; D_1X + D_2 = 0. \tag{1.11}$$

Proof Necessity. Assume $u = Fx$ solves the regulator problem. Certainly (A_1, B_1) is stabilizable. By Lemma 5, asymptotic regulation implies there exists a matrix X such that

$$(A_1 + B_1F_1)X - XA_2 + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0. \tag{1.12}$$

These can be written

$$A_1X - XA_2 + A_3 + B_1U = 0, \quad D_1X + D_2 = 0,$$

where $U = F_1X + F_2$.

Sufficiency. Choose F_1 so that $A_1 + B_1F_1$ is stable. Solve (1.11) for X, U and set $F_2 = U - F_1X$. Then (1.12) holds, so asymptotic regulation follows from Lemma 5. □

Now we turn to designing a controller with input e, not x. Assuming (D, A) is detectable, we can select L so that $A + LD$ is stable. The full-state observer is

$$\dot{\hat{x}} = A\hat{x} + Bu + L(D\hat{x} - e).$$

Setting $u = F\hat{x}$, we get the observer-based controller

$$\dot{\hat{x}} = (A + BF + LD)\hat{x} - Le, \quad u = F\hat{x}.$$

Theorem 6 *Assume (D, A) is detectable and A_2 has only unstable eigenvalues. Then the regulator problem is solved by the observer-based controller if $u = Fx$ is a state-feedback solution.*

Instead of the proof, let's do the cart example from start to finish.

Example We start with

$$A = \left[\begin{array}{rr|rrr} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -100 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \\ 0 \\ 0 \end{array}\right], \quad D = \left[\begin{array}{rr|rrr} -1 & 0 & 1 & 0 & 0 \end{array}\right].$$

We check that A_2 has no stable eigenvalues, that (D, A) is detectable, in fact observable, and that (A_1, B_1) is stabilizable, in fact controllable.

Next, we select F_1 to stabilize $A_1 + B_1F_1$. Arbitrarily selecting the eigenvalues to be -1, we get

$$F_1 = \left[\begin{array}{cc} 0 & -2 \end{array}\right].$$

Next, we have to check solvability of

$$(A_1 + B_1F_1)X - XA_2 + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0. \tag{1.13}$$

for X, F_2. The easiest way to do this is to try to solve them. So write

$$X = \left[\begin{array}{ccc} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{array}\right], \quad F_2 = \left[\begin{array}{ccc} f_{21} & f_{22} & f_{23} \end{array}\right]$$

and substitute them into (1.13). You will get 9 equations in the 9 unknowns. These indeed have a unique solution:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}.$$

Finally, assigning the eigenvalues of $A + LD$ at -1, we get

$$L = \begin{bmatrix} 5 & -91 & -0.01 & 490 & -9005 \end{bmatrix}^T.$$

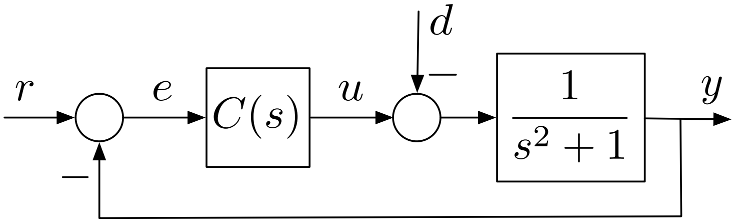
The resulting controller transfer function from e to u is obtained from the state equations

$$\dot{\hat{x}} = (A + BF + LD)\hat{x} - Le, \quad u = F\hat{x}.$$

We get

$$C(s) = \frac{-672s^4 + 8015s^3 - 679s^2 + 8007s + 1}{s(s^4 + 7s^3 + 20s^2 + 700s - 8000)}.$$

The controller has poles at $0, \pm 10j$, as it must to track the step and reject the disturbance. The structure has the familiar block diagram:



□

1.10.4 More Examples

Example Consider the cart/spring with no disturbance:

$$\ddot{y} = u - y.$$

This is a pure step-tracking problem with

$$A = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \end{array}\right], \quad D = \left[\begin{array}{cc|c} -1 & 0 & 1 \end{array}\right].$$

We check that A_2 has no stable eigenvalues—in fact $A_2 = 0$—that (D, A) is observable, and that (A_1, B_1) is controllable.

Next, we select F_1 to stabilize $A_1 + B_1F_1$. For the eigenvalues to be -1, we get the same F_1,

$$F_1 = \begin{bmatrix} 0 & -2 \end{bmatrix}.$$

Next, we have to check solvability of

$$(A_1 + B_1 F_1)X + A_3 + B_1 F_2 = 0, \quad D_1 X + D_2 = 0.$$

for X, F_2. We can write these as

$$\begin{bmatrix} A_1 + B_1 F_1 & B_1 \\ D_1 & 0 \end{bmatrix} \begin{bmatrix} X \\ F_2 \end{bmatrix} = -\begin{bmatrix} A_3 \\ D_2 \end{bmatrix}.$$

The matrix on the left is invertible, so we get

$$\begin{bmatrix} X \\ F_2 \end{bmatrix} = -\begin{bmatrix} A_1 + B_1 F_1 & B_1 \\ D_1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} A_3 \\ D_2 \end{bmatrix}.$$

This yields $F_2 = 1$.

Finally, again assigning the eigenvalues of $A + LD$ at -1, we get

$$L = \begin{bmatrix} 2 & 2 & -1 \end{bmatrix}^T.$$

The resulting controller is

$$C(s) = \frac{5s^2 - 4s + 1}{s(s^2 + 5s + 9)}.$$

□

Example Consider the cart with neither spring nor disturbance:

$$\ddot{y} = u$$

$$A = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \end{array}\right], \quad D = \left[\begin{array}{cc|c} -1 & 0 & 1 \end{array}\right].$$

We confront a problem, because (D, A) is not detectable and we won't be able to construct an observer. Since all the eigenvalues of A are unstable, being undetectable is the same as being unobservable. The reason (D, A) is unobservable is that the setup is redundant: Since the plant is a double integrator, **step-tracking will automatically follow from feedback stability**. So modelling r is unnecessary. In fact, we should not have modelled r.

Let's begin again. We have

$$\ddot{y} = u.$$

This is unstable and we merely need to stabilize it. With no r, the signal to be regulated is $e = -y$. Thus

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad D = \begin{bmatrix} -1 & 0 \end{bmatrix}.$$

Now (D, A) is observable. We select F to stabilize $A + BF$. For the eigenvalues to be -1, we get

$$F = \begin{bmatrix} -1 & -2 \end{bmatrix}.$$

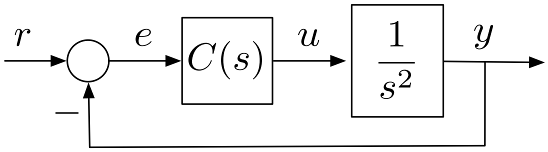
Finally, again assigning the eigenvalues of $A + LD$ at -1, we get

$$L = \left[\begin{array}{cc} 2 & 1 \end{array} \right]^T .$$

The resulting controller from e to u is

$$C(s) = \frac{4s+1}{s^2+4s+6}.$$

We can now put r back where it belongs in the block diagram:



□

Example Consider the cart without the spring but subject to the disturbance:

$$\ddot{y} = u - d.$$

This is similar to the previous example, except there will be an exomodel for d, just not for r. Again taking $e = -y$, we have

$$A = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & -100 & 0 \end{array} \right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \\ 0 \end{array} \right], \quad D = \left[\begin{array}{cc|cc} -1 & 0 & 0 & 0 \end{array} \right].$$

Proceeding as before we get

$$C(s) = \frac{-580s^3 + 8515s^2 + 6s + 1}{s^4 + 6s^3 + 15s^2 + 600s - 8500}.$$

□

Example Now, an example plant where step-tracking is not feasible. We just need the plant to have a zero at $s = 0$:

$$\ddot{y} = \dot{u} - y$$

$$A = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array} \right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \end{array} \right], \quad D = \left[\begin{array}{cc|c} 0 & -1 & 1 \end{array} \right].$$

Again, $A_2 = 0$, (D, A) is observable, and (A_1, B_1) is controllable. Again we take

$$F_1 = \left[\begin{array}{cc} 0 & -2 \end{array} \right].$$

Then we have to check solvability of

$$(A_1 + B_1F_1)X + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0.$$

for X, F_2 or equivalently

$$\begin{bmatrix} A_1 + B_1F_1 & B_1 \\ D_1 & 0 \end{bmatrix} \begin{bmatrix} X \\ F_2 \end{bmatrix} = -\begin{bmatrix} A_3 \\ D_2 \end{bmatrix}.$$

The numbers are

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & -2 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} X \\ F_2 \end{bmatrix} = -\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

This isn't solvable. □

The situation may arise where the reference value r is actually known. For example, for the maglev problem we had pre-specified that we wanted to regulate the ball exactly at 1 cm. Then the problem is really one of stabilization about a nonzero equilibrium point, as illustrated by an example.

Example The cart/spring system

$$\ddot{y} = u - y$$

where the desired position is $y = r = 1$, a fixed known value. Of course, we could use regulator theory, but we don't need to. The model is

$$\dot{x} = Ax + Bu, \quad y = Cx, \quad x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}] .$$

We want to stabilize the system at

$$x = (1, 0).$$

Consider the control law

$$u = Fx + \overline{u},$$

where $\overline{u}$ is a constant to be determined. Then

$$\dot{x} = (A + BF)x + B\overline{u}.$$

If $A + BF$ is stable, then $x(t)$ converges to $-(A + BF)^{-1}B\overline{u}$. Let's take

$$F = \begin{bmatrix} 0 & -2 \end{bmatrix}.$$

Then the solution of

$$-(A + BF)^{-1}B\overline{u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

is $\overline{u} = 1$. If only y, not the full state, is sensed, an observer-based controller can be used:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(C\hat{x} - y) \\ u &= F\hat{x} + \overline{u}. \end{aligned}$$

Again, $y(t)$ converges to the desired value. □

Chapter 2

Discrete-Time Linear Systems

Just as for continuous-time linear time-invariant systems, there are three common ways to describe discrete-time linear systems:

1. difference equation models
2. transfer function models
3. state space models.

We shall study how to use each of these models for analysis, and show how to go back and forth from one description to another.

Mostly, we shall study only single-input single-output systems; many of the results generalize to multivariable systems.

2.1 Difference Equations

The general form is

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m).$$

Here k is the discrete-time variable, an integer; $u(k)$ is the input sequence; $y(k)$ is the output sequence; and a_i, b_i are real, constant coefficients.

Example

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

Obviously, we can solve this equation for $\{y(k)\}_{k\geq 0}$ if we know

$$y(-1),\ \{u(k)\}_{k\geq -1}$$

by recursion:

$$y(k) = 2y(k-1) + 2u(k) + u(k-1),\ k \geq 0.$$

More generally, we can solve for $\{y(k)\}_{k>k_0}$ if we know

$$y(k_0),\ \{u(k)\}_{k\geq k_0}.$$

However, we may like to determine the analytical solution for y given u. □

As for linear differential equations with constant coefficients, the general solution of

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m)$$

can be written as $y = y_h + y_p$, where y_h is the general solution to the **homogeneous equation**

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = 0 \tag{2.1}$$

and y_p is a particular solution to the **nonhomogeneous equation**.

Homogeneous Equations

For ordinary differential equations with constant coefficients, the trial solution is a complex exponential, $\mathrm{e}^{\lambda t}$. This is because if you differentiate such a function, you get another function of the same form. Example:

$$\ddot{y} + 3\dot{y} - 2y = 0$$

Try $y(t) = \mathrm{e}^{\lambda t}$:

$$\lambda^2 \mathrm{e}^{\lambda t} + 3\lambda \mathrm{e}^{\lambda t} - 2\mathrm{e}^{\lambda t} = 0$$

$$\lambda^2 + 3\lambda - 2 = 0.$$

Distinct solutions λ_1, λ_2. Then the general solution is

$$y_h(t) = c_1 \mathrm{e}^{\lambda_1 t} + c_2 \mathrm{e}^{\lambda_2 t}.$$

For linear constant-coefficient difference equations, we take $y_h(k) = \lambda^k$ to be a trial solution to (2.1). This is because if you replace k by $k-1$ in such a function, you get a function of the same form. The complex variable λ must satisfy the characteristic equation

$$\lambda^n + a_1 \lambda^{n-1} + \cdots + a_n = 0. \tag{2.2}$$

Each distinct root of the characteristic equation gives rise to a distinct solution of the homogeneous equation. Suppose there are n distinct roots $\lambda_1, \ldots, \lambda_n$ to (2.2). The general solution to (2.1) is then the linear combination

$$y_h(k) = c_1 \lambda_1^k + \cdots + c_n \lambda_n^k.$$

Example

$$y(k) - y(k-1) - 4y(k-2) + 4y(k-3) = 0$$

Substitute $y(k) = \lambda^k$:

$$\lambda^k - \lambda^{k-1} - 4\lambda^{k-2} + 4\lambda^{k-3} = 0$$

$$\lambda^3 - \lambda^2 - 4\lambda + 4 = 0$$

$$\text{roots: } \lambda_{1,2,3} = \{1, 2, -2\}.$$

Three solutions

$$y_1(k) = \lambda_1^k, \quad y_2(k) = \lambda_2^k, \quad y_3(k) = \lambda_3^k.$$

They're linearly independent functions (review this concept from Linear Algebra). Thus the general solution of the equation is

$$y(k) = c_1 y_1(k) + c_2 y_2(k) + c_3 y_3(k), \quad c_i \in \mathbb{R}.$$

□

Example Find the general solution of

$$y(k) + 2y(k-1) - 3y(k-3) = 0.$$

Try $y(k) = \lambda^k$:

$$\lambda^k + 2\lambda^{k-1} - 3\lambda^{k-3} = 0$$

$$\lambda^3 + 2\lambda^2 - 3 = 0$$

$$\text{roots: } \lambda_{1,2,3} = \{1, -(3/2) \pm (\sqrt{3}/2)j\}$$

Three linearly independent solutions are

$$y_1(k) = \lambda_1^k, \quad y_2(k) = \lambda_2^k, \quad y_3(k) = \lambda_3^k.$$

But y_2, y_3 are complex-valued, whereas we want real-valued solutions. Notice that

$$\overline{\lambda_2} = \lambda_3$$

and

$$\overline{y_2(k)} = y_3(k).$$

Take the real and imaginary parts of $y_2(k)$: $w_2(k) = \Re y_2(k)$, $w_3(k) = \Im y_2(k)$. Then y_1, w_2, w_3 are three real-valued linearly-independent solutions. Thus the general solution is

$$y(k) = c_1 y_1(k) + c_2 w_2(k) + c_3 w_3(k), \quad c_i \in \mathbb{R}.$$

To get w_2, w_3 explicitly, write λ_2 in polar form: $\lambda_2 = r\mathrm{e}^{j\theta}$. Then

$$y_2(k) = \lambda_2^k = r^k \mathrm{e}^{j\theta k}$$

so from Euler's formula (look this up if you've forgotten)

$$w_2(k) = \Re\lambda_2^k = r^k \cos\theta k$$

$$w_3(k) = \Im\lambda_2^k = r^k \sin\theta k.$$

Alternatively, we can also express the general solution as

$$y(k) = d_1 y_1(k) + d_2 y_2(k) + d_3 y_3(k).$$

The fact that we have real solutions requires that d_2 and d_3 are complex numbers which are conjugates of each other. Setting $d_2 = \alpha + j\beta$ and $y_2(k) = r^k(\cos\theta k + j\sin\theta k)$, we find that

$$y(k) = d_1 y_1(k) + 2\alpha r^k \cos\theta k - 2\beta r^k \sin\theta k.$$

which is a linear combination of real linearly independent solutions, as before. □

For nonhomogeneous equations, we need to find a particular solution. If the nonhomogeneous part is a linear combination of terms of the form $\cos\theta k$, $\sin\theta k$, p^k, and polynomials of k, we can often guess what the form of the particular solution is and determine it using the method of undetermined coefficients. We illustrate this method with an example.

Example Find the solution to the initial value problem

$$y(k) - 2y(k-1) = k \qquad y(-1) = 1$$

We see easily that the homogeneous solution is $y_h(k) = 2^k\alpha$ so that the general solution is given by

$$y(k) = 2^k\alpha + y_p(k)$$

For the particular solution y_p, try

$$y_p(k) = Ak + B$$

This guess matches the form of the nonhomogeneous part, and $y_p(k-1)$ will also be a first order polynomial in k. We now seek equations for the undetermined coefficients A and B. Substituting into the difference equation, we obtain

$$Ak + B - 2[A(k-1) + B] = k$$

This gives, on matching coefficients,

$$2A - B = 0$$

and

$$-Ak = k$$

$$\Rightarrow \; A = -1 \qquad B = -2$$

The particular solution is therefore given by

$$y_p(k) = -(k+2)$$

The general solution is then

$$y(k) = 2^k \alpha - (k+2)$$

On putting $k = 0$, we get

$$y(0) = 2y(-1)$$

Substituting into the general solution, we find

$$\alpha = y(0) + 2 = 2(y_{-1} + 1)$$

giving the general solution in terms of the initial condition

$$y(k) = 2(y_{-1} + 1)2^k - (k+2)$$

The solution to the initial value is obtained by plugging in the value of y_{-1} to give

$$y(k) \quad = \quad 4 \times 2^k - (k+2)$$

□

The method of undetermined coefficients is only effective when the nonhomogeneous part is sufficiently simple. The better method for solving nonhomogeneous equations is via z-transforms, which we treat next.

2.2 Z-Transforms

Z-transforms are to discrete time as Laplace transforms are to continuous time. Let $x(k)$ be a real-valued discrete-time signal. Its z-transform is defined to be

$$X(z) = \sum_{k=0}^{\infty} x(k) z^{-k},$$

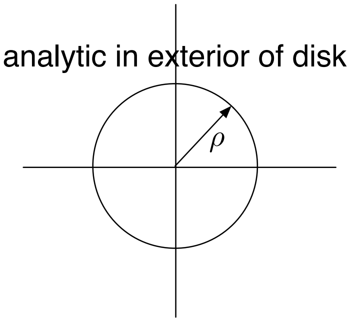
where z is a complex variable. (More precisely, this is the one-sided transform—there's a two-sided z-transform which you may have seen in a DSP course.) The sufficient condition for $x(k)$ to have a z-transform is that it satisfy a growth bound,

$$|x(k)| \leq M\rho^k, \;\; k \geq 0$$

where M, ρ are positive constants. Then setting $z = re^{j\theta}$ with $r > \rho$ we have

$$\begin{aligned}
|X(z)| &= \left|\sum_{k=0}^{\infty} x(k)(re^{j\theta})^{-k}\right| \\
&\le \sum_{k=0}^{\infty} \left|x(k)(re^{j\theta})^{-k}\right| \\
&= \sum_{k=0}^{\infty} |x(k)|r^{-k} \\
&\le M\sum_{k=0}^{\infty} \left(\frac{\rho}{r}\right)^k \\
&= M\frac{1}{1-\frac{\rho}{r}} \\
&< \infty \ \text{ if } r > \rho.
\end{aligned}$$

Thus the ROC includes the set $\{z : |z| > \rho\}$, which is the exterior of the disk $|z| \le \rho$. The ROC could be larger.



For convenience, we often use the symbol $\mathcal{Z}$ to denote the z-transform operator.

Example

$$x(k) = a^k, \quad k \ge 0$$

$$X(z) = \sum_{k=0}^{\infty}(az^{-1})^k = \frac{1}{1-az^{-1}} = \frac{z}{z-a}, \quad |z| > |a|$$

Special case: $a = 1$, unit step.

For notational convenience, we indicate a^k and $\frac{1}{1-az^{-1}}$ are z-transform pairs by writing

$$\mathcal{Z}(a^k) = \frac{1}{1-az^{-1}}$$

or

$$\mathcal{Z}^{-1}\left(\frac{1}{1-az^{-1}}\right) = a^k.$$

□

Example Sample $x(t) = \mathrm{e}^t \cos t$ every T seconds to get

$$y(k) = \mathrm{e}^{kT} \cos kT.$$

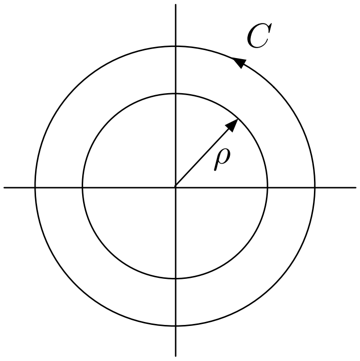
The ROC for $Y(z)$ is $|z| > \mathrm{e}^T$. □

Inverse Z-Transform

Let $x(k)$ be a discrete-time signal and $X(z)$ its z-transform. Suppose we have $X(z)$ and we want $x(k)$. The **inversion formula** is this:

$$x(k) = \frac{1}{2\pi j} \oint_C X(z) z^{k-1} dz, \quad k \geq 0.$$

This integral is a contour integral in the complex plane around a circle C inside the region of convergence, like this:



Here's an outline of the verification of this inversion formula:

$$\begin{aligned} \frac{1}{2\pi j} \oint_C X(z) z^{k-1} dz &= \frac{1}{2\pi j} \oint \Sigma_n x(n) z^{-n} z^{k-1} dz \\ &= \sum_n x(n) \left(\frac{1}{2\pi j} \oint z^{-(n-k+1)} dz \right). \end{aligned}$$

Now let's see what the value of $\frac{1}{2\pi j} \oint z^{-(n-k+1)} dz$ is. To simplify notation, let $m = n - k + 1$, so we're looking at $\frac{1}{2\pi j} \oint z^{-m} dz$. Taking $z = r\mathrm{e}^{j\theta}$ with r constant and θ going from 0 to 2π, we have $dz = jre^{j\theta} d\theta$, and

$$\frac{1}{2\pi j} \oint z^{-m} dz = \frac{1}{2\pi j} \int_0^{2\pi} jr^{1-m} \mathrm{e}^{j(1-m)\theta} d\theta.$$

The right-hand side equals 1 if $m = 1$ and zero otherwise, since $\int_0^{2\pi} e^{jn\theta} d\theta = 0,\ n \neq 0$. Thus we have

$$\frac{1}{2\pi j} \oint_C X(z) z^{k-1} dz = \sum_n x(n) \left(\frac{1}{2\pi j} \oint z^{-(n-k+1)} dz \right) = x(k).$$

This verifies the inversion formula.

Now we state, without proof, the **Residue Theorem**: The integral $\frac{1}{2\pi j}\oint_C F(z)dz$ equals the sum of the residues of $F(z)$ at its poles inside the contour C.

Thus from the inversion formula, we have

$$x(k) = \sum \text{ residues of } X(z)z^{k-1} \text{ at all its poles.}$$

Example

$$X(z) = \frac{z}{z-1}$$

Then for $k \geq 0$

$$x(k) = \sum \text{ residues of } \tfrac{z^k}{z-1} \text{ at all its poles.}$$

There's a simple pole at $z = 1$, the residue being $1^k = 1$. Thus $x(k) = 1$ for $k \geq 0$. □

Example

$$X(z) = \frac{1}{1 - az^{-1}}, \quad |z| > |a|$$

$$\begin{aligned} x(k) &= \sum \text{ residues of } \tfrac{z^k}{z-a} \text{ at all its poles.} \\ &= a^k, \quad k \geq 0 \end{aligned}$$

Alternatively, we can do an infinite series expansion to get

$$\frac{1}{1 - az^{-1}} = 1 + \frac{a}{z} + \frac{a^2}{z^2} + \cdots ,$$

from which we can recognize that a^k is the time signal. □

Since a discrete-time signal in computer control is usually defined for $k \geq 0$, it invariably gives rise to a z-transform with a region of convergence being the exterior of a disk with a sufficiently large radius. For this reason, the region of convergence for a transform $X(z)$ is often omitted with the understanding that it is the exterior of the smallest disk that includes all the poles of $X(z)$.

Using the method of residues, one can show that

$$\mathcal{Z}^{-1}\left[\frac{z}{(z-a)^{i+1}}\right] = \frac{k!}{i!(k-i)!}a^{k-i}, \quad i \geq 0. \tag{2.3}$$

This is a very useful formula which, as we shall see, will help us to invert many z-transforms quickly. Two cases of particular interest:

$$i = 0: \quad \mathcal{Z}^{-1}\left[\frac{z}{z-a}\right] = \mathcal{Z}^{-1}\left[\frac{1}{1-az^{-1}}\right] = a^k$$

$$i = 1: \quad \mathcal{Z}^{-1}\left[\frac{z}{(z-a)^2}\right] = ka^{k-1}$$

Properties of Z-Transforms

Convolution of (causal) signals

Convolution in the time domain corresponds to multiplication in the z-transform domain: If

$$w(k) = \sum_{l=0}^{k} x(l)y(k-l)$$

then

$$\begin{aligned} W(z) &= \sum_{k=0}^{\infty}\sum_{l=0}^{k} x(l)y(k-l)z^{-k} \\ &= \sum_{l=0}^{\infty}\sum_{k=l}^{\infty} x(l)y(k-l)z^{-(k-l)}z^{-l} \\ &= \sum_{l=0}^{\infty}\sum_{j=0}^{\infty} x(l)y(j)z^{-j}z^{-l} \\ &= X(z)Y(z). \end{aligned}$$

Multiplication of signals

Multiplcation in the time domain corresponds to convolution in the z-transform domain: Suppose

$$w(k) = x(k)y(k).$$

Suppose $X(z)$ has ROC $|z| > \rho_x$ and $Y(z)$ has ROC $|z| > \rho_y$. Then

$$\begin{aligned} W(z) &= \sum_k x(k)y(k)z^{-k} \\ &= \sum x(k)z^{-k}\frac{1}{2\pi j}\oint Y(\zeta)\zeta^{k-1}d\zeta \\ &= \frac{1}{2\pi j}\oint \sum x(k)\left(\frac{z}{\zeta}\right)^{-k} Y(\zeta)\frac{d\zeta}{\zeta} \\ &= \frac{1}{2\pi j}\oint X\left(\frac{z}{\zeta}\right) Y(\zeta)\frac{d\zeta}{\zeta} \end{aligned}$$

where the contour integral is over a circle $|\zeta| > \rho_y$. Since we require both $\left|\frac{z}{\zeta}\right| > \rho_x$ and $|\zeta| > \rho_y$, so $|z| > \rho_x\rho_y$ is the region of convergence.

Multiplication by a^k

$$\mathcal{Z}\{a^k x(k)\} = X\left(\frac{z}{a}\right)$$

since

$$\sum a^k x(k)z^{-k} = \sum x(k)\left(\frac{z}{a}\right)^{-k}$$

ROC:

$$|x(k)| \leq c\rho_0^k \Rightarrow |a^k x(k)| \leq c[|a|\rho_0]^k.$$

Hence $|z| > |a|\rho_0$ is the region of convergence for $X\left(\frac{z}{a}\right)$.

Backward shift

$$\begin{aligned}
&\mathcal{Z}[x(k-m)] \\
&= \sum_{k=0}^{\infty} x(k-m)z^{-k} \\
&= \sum_{k=0}^{m-1} x(k-m)z^{-k} + \sum_{k=m}^{\infty} x(k-m)z^{-(k-m)}z^{-m} \\
&= \sum_{k=0}^{m-1} x(k-m)z^{-k} + z^{-m}X(z) \\
&= z^{-m}X(z) + x(-m) + z^{-1}x(-m+1) + \cdots + x(-1)z^{-m+1}
\end{aligned}$$

Forward shift

$$\begin{aligned}
&\sum_{k=0}^{\infty} x(k+m)z^{-k} \\
&= \sum_{k=0}^{\infty} x(k+m)z^{-(k+m)}z^{m} \\
&= \sum_{l=m}^{\infty} x(l)z^{-l}z^{m} \\
&= z^{m}X(z) - \sum_{l=0}^{m-1} x(l)z^{-l}z^{m} \\
&= z^{m}X(z) - [z^{m}x(0) + z^{m-1}x(1) + \ldots + zx(m-1)]
\end{aligned}$$

Initial-Value Theorem

The initial value of a sequence $x(k)$ with z-transform $X(z)$ is given by

$$x(0) = \lim_{z\to\infty} X(z).$$

Final-Value Theorem

1. If $X(z)$ has no poles in $|z| \geq 1$, then $x(k)$ converges to 0 as $k \to \infty$.
2. If $X(z)$ has no poles in $|z| \geq 1$ except a simple pole at $z = 1$, then $x(k)$ converges as $k \to \infty$ and $\lim_{k\to\infty} x(k)$ equals the residue of $X(z)$ at the pole $z = 1$, i.e.,
$$\lim_{k\to\infty} x(k) = (z-1)\, X(z)\big|_{z=1}.$$
3. If $X(z)$ has a repeated pole at $z = 1$, then $x(k)$ doesn't converge as $k \to \infty$.
4. If $X(z)$ has a pole at $|z| \geq 1, z \neq 1$, then $x(k)$ doesn't converge as $k \to \infty$.

Transfer functions

A linear time-invariant system has a transfer function. Consider the equation

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m).$$

Take z-transforms assuming $u(k), y(k)$ equal zero for $k < 0$:

$$Y(z) + a_1 z^{-1} Y(z) + \cdots + a_n z^{-n} Y(z) = b_0 U(z) + b_1 z^{-1} U(z) + \cdots + b_m z^{-m} U(z).$$

Thus

$$Y(z) = G(z)U(z), \quad G(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}.$$

The function $G(z)$ is the *transfer function*; its inverse transform, $g(k)$, is the *impulse response function*; and in the time domain, $y(k)$ equals the convolution of $g(k)$ and $u(k)$.

Here's an application of the Final-Value Theorem: Consider a system with transfer function $G(z)$, input $u(k)$, and output $y(k)$. Suppose all the poles of $G(z)$ are in $|z| < 1$ and $u(k)$ is the unit step. Then $y(k)$ converges as $k \to \infty$ to $G(z)|_{z=1}$.

2.3 Solving Difference Equations by Z-Transforms

Let's begin with an example.

Example

$$y(k) - 2y(k-1) = u(k), \quad y(-1) = 1, \quad u(k) = k$$

Take z-transforms:

$$Y(z) - 2z^{-1}Y(z) - 2y(-1) = U(z),$$

i.e.,

$$Y(z) - 2z^{-1}Y(z) - 2 = U(z)$$

or

$$zY(z) - 2Y(z) = 2z + zU(z).$$

From the formula

$$\mathcal{Z}^{-1}\left[\frac{z}{(z-a)^2}\right] = ka^{k-1},$$

we get

$$U(z) = \frac{z}{(z-1)^2}.$$

Thus

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

Then $y(k)$ equals the sum of the residues of $Y(z)z^{k-1}$. This gives

$$y(k) = 4 \times 2^k - (k+2), \quad k \geq 0.$$

□

There's another possible form of the equation:

$$y(k+1) - 2y(k) = u(k), \quad k \geq 0;\ y(0) \text{ given.} \tag{2.4}$$

This can be solved by z-transforms:

$$\sum_{k=0}^{\infty} y(k+1)z^{-k} - 2\sum_{k=0}^{\infty} y(k)z^{-k} = \sum_{k=0}^{\infty} u(k)z^{-k}$$

and thus

$$z[Y(z) - y(0)] - 2Y(z) = U(z).$$

For the general n^{th} order linear difference equation with constant coefficients

$$y_k + a_1 y_{k-1} + \cdots + a_n y_{k-n} = b_0 u_k + \cdots + b_m u_{k-m} \text{ with } u_k = 0, k < 0.$$

Putting $a_0 = 1$, we can write the above equation as

$$\sum_{j=0}^{n} a_j y_{k-j} = \sum_{j=0}^{m} b_j u_{k-j}$$

Suppose $|u_k| \leq \beta r_u^k$ for some $\beta \geq 0$, $r_u > 0$. Almost all inputs in practice will satisfy some such geometric bound. Then the solution y_k will satisfy also a geometric bound and hence z-transformable. Taking z-transform of the left hand side gives

$$\begin{aligned}\sum_{k=0}^{\infty}\sum_{j=0}^{n} a_j y_{k-j} z^{-k} &= Y(z) + a_1 z^{-1} Y(z) + \ldots \\ &= A(z^{-1})Y(z) + \sum_{j=1}^{n}\sum_{k=0}^{j-1} a_j y_{k-j} z^{-k}\end{aligned}$$

where, for convenience, we have introduced

$$A(z^{-1}) = \sum_{j=0}^{n} a_j z^{-j}, \text{ with } a_0 = 1$$

Similarly, we set

$$B(z^{-1}) = \sum_{j=0}^{m} b_j z^{-j}$$

Then

$$Y(z) = \frac{I(z)}{A(z^{-1})} + \frac{B(z^{-1})}{A(z^{-1})}U(z) \tag{2.5}$$

where $I(z) = -\sum_{j=1}^{n}\sum_{k=0}^{j-1} a_j y_{k-j} z^{-k}$ is a polynomial depending on the initial condition. Let

$$Y_i(z) = \frac{I(z)}{A(z^{-1})}$$

and

$$Y_e(z) = \frac{B(z^{-1})}{A(z^{-1})}U(z)$$

In terms of the terminology of Section 2.1, $Y_i(z)$ is the transform of a homogeneous solution, and $Y_e(z)$ is the transform of a particular solution. The solution y_k can then be obtained by taking the inverse z-transform.

Now what if the initial time is, say, $k = 2$ (and a specific $u(k)$ is taken):

$$y(k+1) - y(k) = 2^k, \quad k \geq 2; \; y(2) = 1? \tag{2.6}$$

There exists a unique solution for $k \geq 2$. We're not given $y(k)$ for $k < 2$. Can we solve the equation by z-transforms? Yes—here are two ways.

Extrapolate backwards in time

We can use the equation in the form

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

to compute consistent values for $y(1), y(0)$ starting from $y(2) = 1$. This gives $y(1) = -1$, $y(0) = -2$. In this way we arrive at a consistent initial-value problem

$$y(k+1) - y(k) = 2^k, \quad k \geq 0; \; y(0) = -2 \tag{2.7}$$

that is in standard form (2.4). We get

$$z[Y(z) + 2] - Y(z) = \frac{z}{z-2}.$$

Solving for $Y(z)$ and inverse transforming gives

$$y(k) = -3 + 2^k, \quad k \geq 0.$$

Thus the solution to the initial-value problem (2.6) is

$$y(k) = -3 + 2^k, \quad k \geq 2.$$

Shift time

Since $y(2)$ is given and we're trying to determine $y(3), y(4)$ etc., it makes sense to define $x(k) = y(k+2)$, so that $x(0)$ is given and we're trying to determine $x(1), x(2)$ etc. From (2.6),

$$x(k-1) - x(k-2) = 2^k, \quad k \geq 2;\; x(0) = 1.$$

In this difference equation, define the new time variable $m = k - 2$. Then

$$x(m+1) - x(m) = 2^{m+2}, \quad m \geq 0;\; x(0) = 1$$

which is standard form (2.4) again.

Then

$$z[X(z) - 1] - X(z) = \frac{4z}{z-2},$$

which yields

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

and then

$$x(k) = -3 + 4 \times 2^k, \quad k \geq 0.$$

Thus

$$y(k) = x(k-2) = -3 + 4 \times 2^{k-2} = -3 + 2^k, \quad k \geq 2.$$

We have seen in the analysis above that for $m \geq 1$, z^{-m} is associated with shifting time backwards by m steps, while z^m is associated with shifting time forwards by m steps. Even though we have so far considered z only as a complex variable when it appears in the context of z-transforms, it is convenient to give it also a time-shift interpretation: for a discrete-time sequence $x(k)$, $z^{-m}x(k) = x(k-m)$, $m \geq 1$, and $z^m x(k) = x(k+m)$. Which interpretation we should use for z will always be clear from context.

2.4 State Space Analysis

The state equations are

$$x(k+1) = Ax(k) + Bu(k) \tag{2.8}$$

$$y(k) = Cx(k) + Du(k) \tag{2.9}$$

where A, B, C, D are matrices.

If $u(k) = 0$ and $x(0)$ is given, the state evolves according to

$$x(k) = A^k x(0).$$

The matrix A^k is called the **transition matrix**: It maps the state at time 0 to the state at time k. On the other hand, if $x(0)=0$ and $u(k)$ is given, the state evolves according to

$$x(k)=\sum_{j=0}^{k-1}A^{k-j-1}Bu(j).$$

This is a convolution summation. Finally, the response to both $x(0)$ and $u(k)$ simultaneously is the sum of the two responses:

$$x(k)=A^kx(0)+\sum_{j=0}^{k-1}A^{k-j-1}Bu(j).$$

Let's discuss computing the transition matrix by hand. We examine two methods: diagonalization and z-transform.

Diagonalization

Assume A can be diagonalized (i.e., it has n linearly independent eigenvectors; e.g., A has distinct eigenvalues or is symmetric). Then there exists a nonsingular T such that

$$V^{-1}AV=A_{JF},$$

where A_{JF} is the diagonal matrix of eigenvalues. Raising A_{JF} to the kth power gives

$$A_{JF}^k=V^{-1}A^kV=\left[\begin{array}{cccc}\lambda_1^k & 0 & \cdots & 0\\ 0 & \lambda_2^k & 0 & \vdots\\ \vdots & & \ddots & 0\\ 0 & \cdots & 0 & \lambda_n^k\end{array}\right]$$

so that

$$A^k=V\left[\begin{array}{cccc}\lambda_1^k & 0 & \cdots & 0\\ 0 & \lambda_2^k & 0 & \vdots\\ \vdots & & \ddots & 0\\ 0 & \cdots & 0 & \lambda_n^k\end{array}\right]V^{-1}.$$

Example

$$A=\left[\begin{array}{cc}0 & 1\\ -2 & -3\end{array}\right]$$

$$\det(zI-A)=\det\left[\begin{array}{cc}z & -1\\ 2 & z+3\end{array}\right]=z^2+3z+2=(z+2)(z+1)$$

Distinct eigenvalues; take V consisting of eigenvectors of A. The eigenvectors:

$$A-\lambda_1I=\left[\begin{array}{cc}2 & 1\\ -2 & -1\end{array}\right],\quad v_1=\left[\begin{array}{c}-1\\ 2\end{array}\right].$$

Similarly,

$$A - \lambda_2 I = \begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix}, \quad v_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

Defining

$$V = \begin{bmatrix} -1 & -1 \\ 1 & 2 \end{bmatrix}$$

we get

$$V^{-1}AV = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}.$$

Then

$$A^k = V \begin{bmatrix} (-1)^k & 0 \\ 0 & (-2)^k \end{bmatrix} V^{-1}.$$

□

Solution by z-transform

Taking z-transforms of both sides of

$$x(k+1) = Ax(k)$$

we obtain

$$zX(z) - zx(0) = AX(z)$$

$$X(z) = (zI - A)^{-1} zx(0) = (I - z^{-1}A)^{-1} x(0).$$

Comparing this with

$$x(k) = A^k x(0)$$

we see that

$$A^k = \mathcal{Z}^{-1}\left\{(I - z^{-1}A)^{-1}\right\}.$$

Example

Same as preceding example:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}.$$

$$\mathcal{Z}(A^k) = z(zI - A)^{-1} = \frac{z}{(z+1)(z+2)} \begin{bmatrix} z+3 & 1 \\ -2 & z \end{bmatrix}$$

Inversion via residues gives A^k.

□

From State Space to Transfer Function

We had

$$x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-j-1} B u(j)$$

$$y(k) = CA^k x(0) + \sum_{j=0}^{k-1} CA^{k-j-1} B u(j) + Du(k).$$

This $y(k)$ has the form

$$y(k) = CA^k x(0) + \sum_{j=0}^{k} h(k-j)u(j) = CA^k x(0) + (h \star u)(k),$$

where

$$h(k) = \begin{cases} CA^{k-1}B, & k > 0 \\ D, & k = 0 \\ 0, & k < 0 \end{cases}$$

This matrix-valued sequence $h(k)$ is called the **impulse response**. The z-transform of the output can similarly be expressed in terms of $x(0)$ and $U(z)$:

$$\begin{aligned} Y(z) &= CX(z) + DU(z) \\ &= C(zI - A)^{-1} z x(0) + [C(zI - A)^{-1}B + D]U(z). \end{aligned}$$

The transfer function from u to y is therefore

$$G(z) = C(zI - A)^{-1}B + D.$$

Recall that a **proper** rational function is a ratio of two polynomials with the degree of the numerator polynomial $\leq$ the degree of the denominator polynomial. A rational function is **strictly proper** if the degree of the numerator $<$ the degree of the denominator. In the SISO case, i.e., both u and y are scalar-valued, we can express

$$C(zI - A)^{-1}B = \frac{C\text{adj}(zI - A)B}{\det(zI - A)},$$

which is strictly proper, where adj$(\cdot)$ denotes adjoint. Hence the transfer function $G(z)$ is proper but not strictly proper if and only if $D \neq 0$.

From Transfer Function to State Space

The transfer function of the continuous-time state-space model is $G(s) = C(sI - A)^{-1}B + D$; the transfer function of the discrete-time state-space model is $G(z) = C(zI - A)^{-1}B + D$. Since these two functions are identical (only the arguments have different symbols), to go from a transfer function $G(z)$ to (A, B, C, D) in discrete time is the same process as in continuous time, which has been described in Chapter 1.

When the transfer functions is scalar-valued, we can, by padding with zeros, take the degrees of the numerator and denominator to be the same without loss of generality.

$$G(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}$$

The system is equivalently described by the difference equation

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k) + \cdots + b_n u(k-n)$$

We can, in this case, write down a state space representation of the difference equation, hence the transfer function $G(z)$, by expressing the state in terms of the inputs and outputs. Define the various components of the state vector $x(k)$ by:

$$x_{n-j}(k) = -\sum_{i=j+1}^{n} a_i z^{-(i-j)} y(k) + \sum_{i=j+1}^{n} b_i z^{-(i-j)} u(k) \tag{2.10}$$

Using the difference equation, it is readily seen that the output $y(k)$ is given by

$$\begin{aligned} y(k) &= x_n(k) + b_0 u(k) \\ &= [0 \cdots 0\ 1] x(k) + b_0 u(k) \end{aligned} \tag{2.11}$$

To see the state equation which this definition gives rise to, we note that

$$\begin{aligned} x_{n-j}(k+1) &= -\sum_{i=j+1}^{n} a_i z^{-(i-j-1)} y(k) + \sum_{i=j+1}^{n} b_i z^{-(i-j-1)} u(k) \\ &= -a_{j+1} y(k) + b_{j+1} u(k) \\ &\quad - \sum_{i=j+2}^{n} a_i z^{-(i-(j+1))} y(k) + \sum_{i=j+2}^{n} b_i z^{-(i-(j+1))} u(k) \\ &= x_{n-j-1}(k) - a_{j+1} y(k) + b_{j+1} u(k) \\ &= x_{n-j-1}(k) - a_{j+1}(x_n(k) + b_0 u(k)) + b_{j+1} u(k) \end{aligned}$$

Putting everything together, we finally get

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \cdots & \cdots & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} b_n - b_0 a_n \\ \vdots \\ b_1 - b_0 a_1 \end{bmatrix} u(k) \tag{2.12}$$

If $b_0 = 0$, the equation simplifies to

$$x(k+1) = \begin{bmatrix} 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \cdots & \cdots & 1 & -a_1 \end{bmatrix} x(k) + \begin{bmatrix} b_n \\ \vdots \\ b_1 \end{bmatrix} u(k) \tag{2.13}$$

$$y(k) = [0 \cdots 0\,1]x(k) \tag{2.14}$$

Since this is a single-input single-output system, the transfer function is a scalar rational function. Thus if we take the transpose of the transfer function, which does not change the transfer function, we see immediately that the following state equation

$$x(k+1) = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & & & 1 \\ -a_n & -a_{n-1} & & \cdots & -a_2 & -a_1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) \tag{2.15}$$

$$y(k) = [b_n \cdots b_1]x(k) \tag{2.16}$$

is also a realization of the difference equation. The state space realization, (2.13), (2.14) is referred to as being in observable canonical form, while the state space realization, (4.3), (2.16) is referred to as being in controllable canonical form. The reasons for these names will become clear when we study design of control systems based on state space methods.

2.5 Exercises

1. Solve the initial-value problem

$$\begin{aligned} y(k) - 0.5y(k-1) &= 0.2^k, \quad k \geq 1 \\ y(0) &= 1. \end{aligned}$$

2. (a) Consider

$$Ay = u, \quad A : 3 \times 7, \quad \text{rank} = 3.$$

Write the form of the general solution of the homogeneous equation and of the non-homogeneous equation.

(b) Write the form of the general solution of the homogeneous equation

$$\frac{d^3y}{dt^3} = 0.$$

Does this have a unique solution:

$$\frac{d^3y}{dt^3} = 0, \quad y(0) = 1, \; \ddot{y}(0) = -2.$$

(c) Write the form of the general solution of the homogeneous equation

$$y(k) - 2y(k-2) + y(k-4) = 0.$$

Write the form of the general solution of

$$y(k) - 2y(k-2) + y(k-4) = k^2 - 1, \quad k \geq 4.$$

3. Find two real-valued, linearly independent solutions to the homogeneous equation
$$y(k) + 4y(k-2) = 0.$$

4. The Laurent series expansion of
$$F(z) = \frac{z^4}{(z+1)^2(z-2)}$$
at $z = -1$ has the form
$$F(z) = \sum_{k=-\infty}^{\infty} c_k (z+1)^k.$$
Find the residue of $F(z)$ at the pole $z = -1$.

5. Consider $X(z) = \frac{z}{(z-2)(z+0.5)}$. What is the ROC? Does $x(k)$ converge to zero as k tends to infinity? Find $x(k)$.

6. Consider the averaging system
$$y(k) = \frac{1}{2}[u(k) + u(k-1)].$$
Find the transfer function, the ROC, and the impulse-response function.

7. Find the final value of $x(k)$, if it exists, where $X(z) = \dfrac{z^2}{(z+1)(z-0.2)}$. Repeat for $X(z) = \dfrac{z^2}{(z-1)(z-0.2)}$.

8. Solve the initial-value problem
$$\begin{aligned} y(k) - y(k-1) &= 4k, \quad k \geq 1 \\ y(0) &= 1 \end{aligned}$$
using z-transforms.

9. There is a nice way to compute $\mathrm{adj}(zI - A)$ recursively. This problem asks you to derive the formula for a 3×3 matrix. Let
$$\det(zI - A) = z^3 + a_1 z^2 + a_2 z + a_3.$$
Write
$$\mathrm{adj}(zI - A) = B_1 z^2 + B_2 z + B_3.$$
Using the equation
$$(zI - A)\mathrm{adj}(zI - A) = I\det(zI - A),$$
show that, by matching coefficients of powers of z, we can determine B_1, B_2, and B_3 recursively through the equations
$$B_1 = I, \quad B_2 = AB_1 + a_1 I, \quad B_3 = AB_2 + a_2 I.$$

10. Since inverting a 2×2 matrix is easy, it is generally fastest to determine A^k using z-transforms if A is 2×2. Let

$$A = \begin{bmatrix} 0.4 & -0.3 \\ 0.1 & 0.8 \end{bmatrix}.$$

Determine A^k using z-transforms.

11. For matrices larger than 2×2, determination of A^k is generally computationally quite tedious. In this problem, you will use Scilab or MATLAB to become familiar with routines useful in computing A^k. Let

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 3 & 0 & 2 \\ -12 & -7 & -6 \end{bmatrix}.$$

Determine A^k using

(a) Diagonalization. Use the function `spec` in Scilab or `eig` in MATLAB to determine the eigenvalues and eigenvectors. Scilab and MATLAB do not necessarily give “nice” values for the eigenvectors. You can often get nicer values by inspecting the eigenvectors generated by Scilab and MATLAB.

(b) z-transforms. You probably want to use the recursive formula for computing $\text{adj}(zI - A)$. You can also try out `pfss` (partial fraction expansion) in Scilab.

12. Find the 1000^{th} term in the famous Fibonnaci sequence

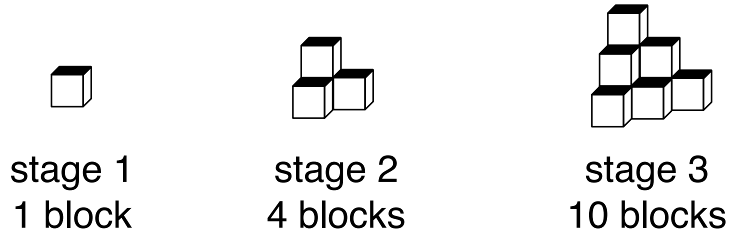
$$1, 1, 2, 3, 5, 8, 13, \ldots,$$

where, beginning with the third, each term is the sum of the previous two. Hint: The 1000^{th} term is huge—don’t bother to write it out. Instead, for example, give a formula for evaluating the term. (Leonardo Pisano Fibonacci: born 1170, died 1250.)

13. Develop a state model for the discrete-time system with transfer function

$$\frac{Y(z)}{U(z)} = \frac{1 + 0.2z^{-1} - 0.5z^{-2} + 0.4z^{-3}}{1 + 0.3z^{-1} + 0.8z^{-2} - 0.2z^{-3}}.$$

14. Stages 1, 2, and 3 of a pattern of cubes are shown. We would like to find how many blocks there will be at stage 9 of this pattern.



Solve the problem

(a) By z-transforms.

(b) By finding a state-space model.

15. (a) The left-half plane is to continuous time as what is to discrete time?

(b) $\dot{x} = Ax$ is to continuous time as what is to discrete time?

(c) e^{tA} is to continuous time as what is to discrete time?

(d) $1/s$ is to continuous time as what is to discrete time?

(e) $\delta(t)$ is to continuous time as what is to discrete time?

(f) $\mathrm{e}^{tA} \longleftrightarrow (sI - A)^{-1}$ is to continuous time as what is to discrete time?

(g) $C(sI - A)^{-1}B + D$ is to continuous time as what is to discrete time?

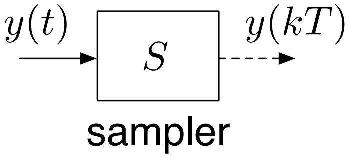
Chapter 3

Sampled-Data Systems

We study control loops where the plant is continuous-time and the controller digital. We ignore quantization.

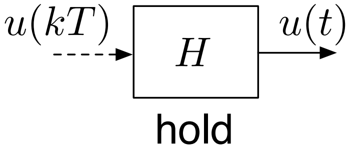
3.1 Introduction

We begin with the interface components between continuous time and discrete time. First the sampler:



Here T is the sampling period and the output $y(kT)$ is viewed as a function of k. We sometimes write $y_d(k)$ instead of $y(kT)$.

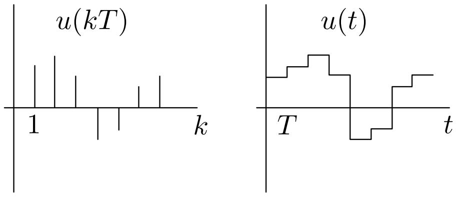
Next, the hold operator:



The definition of the output of H is this:

$$u(t) = u(kT), \quad kT \le t < (k+1)T.$$

Thus the action of H is this:



Let's briefly discuss S and H as mathematical functions. Is S linear? Yes. Linearity means two things:

$$Sy = y_d \Longrightarrow S(cy) = cy_d$$

$$\text{i.e., } S(cy) = cS(y)$$

$$Sy = y_d,\ Sv = v_d \Longrightarrow S(y+v) = y_d + v_d$$

$$\text{i.e., } S(y+v) = S(y) + S(v).$$

Or, equivalently, one thing:

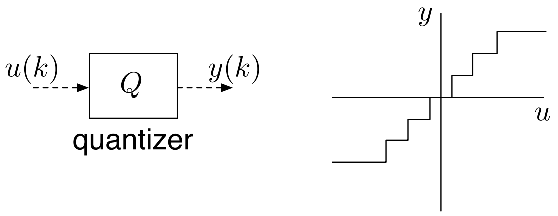
$$Sy = y_d,\ Sv = v_d \Longrightarrow S(\alpha y + \beta v) = \alpha y_d + \beta v_d$$

$$\text{i.e., } S(\alpha y + \beta v) = \alpha S(y) + \beta S(v).$$

Proof of the first property: Let $y(t)$ be given. Define $y_d = S(y)$, i.e., $y_d(k) = y(kT)$. Let c be a real constant. Define $v_d = S(cy)$, i.e., $v_d(k) = cy(kT)$. Then $v_d = cy_d$. Conclusion: $S(cy) = cS(y)$. You can prove the second property.

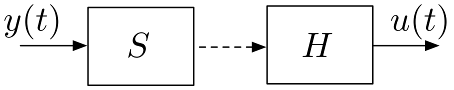
Likewise, H is linear. Thus SH and HS, compositions of linear functions, are linear.

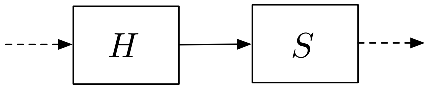
The thing that makes a real digital controller nonlinear is the quantizer. A quantizer Q is a function whose graph looks like this:



Check that Q is not a linear system.

Let's see what we get when we connect S and H in series:

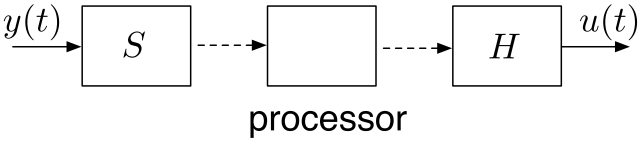




The first system is HS, because $u = H(Sy)$. Sketch $u(t)$ when $y(t) = t$, the ramp. You'll see that $u(t) = y(t)$ only at the sampling instants $t = kT$. Next, let $y(t)$ be the unit step at time $t = 0$; you'll see that $u(t)$ is the unit step too. Now let $y(t)$ be the unit step at time $t = T/2$; you'll see that $u(t)$ is the unit step at time $t = T$. Thus shifting the input by $T/2$ seconds resulted in the output being shifted by T seconds. Conclusion: HS **is linear but not time invariant.** So HS doesn't have a transfer function.

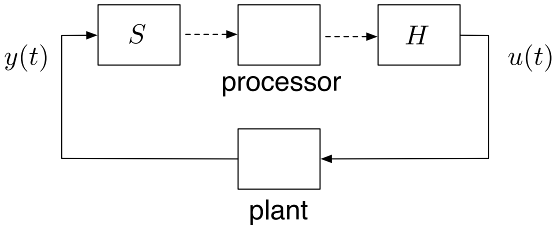
The second system is SH. Its output always equals its input. Thus SH is the identity system.

In terms of S and H, a digital controller has the form

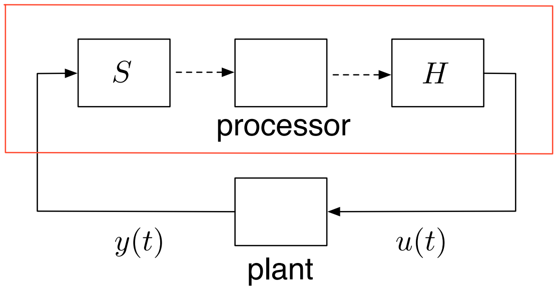


The input is sampled; the sampled values are processed in discrete time; then there's conversion back to continuous time. Even if the processor is doing time-invariant processing, the overall digital controller from $y(t)$ to $u(t)$ is not time invariant, so it doesn't have a transfer function.

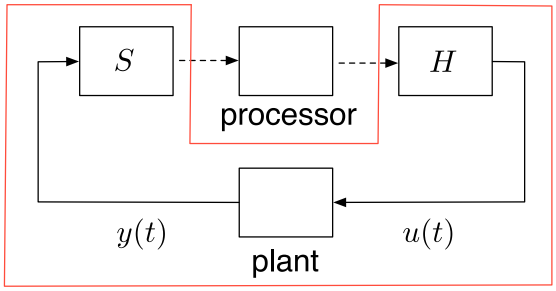
A digital controller in a loop might look like this:



The plant sees a linear, time-varying controller:



The processor sees a discretized plant:



In the next section we'll study the discretized plant.

In general there are limitations on the sampling frequency.

(1) Faster sampling requires more expensive hardware.

(2) Performing all the control calculations may not be feasible if the sampling is too fast.

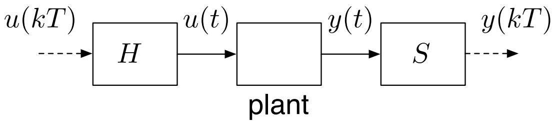
(3) A plant with slow dynamics together with fast sampling may yield samples with small differences, leading to problems with finite precision arithmetic.

(4) The sampling period is often dictated by other considerations unrelated to control, e.g., sensors and actuators may be connected to a bus with fixed rates.

Throughout the rest of the course, we assume that T is fixed and known unless otherwise stated.

3.2 State Space Analysis

We'll now derive the discrete-time system equations that arise as a result of discretizing continuous-time state equations.



Let the continuous-time system be described by the state equations

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

For any two times $\tau_1 \geq \tau_0$, we have

$$x(\tau_1) = \mathrm{e}^{A(\tau_1 - \tau_0)} x(\tau_0) \; + \int_{\tau_0}^{\tau_1} \mathrm{e}^{A(\tau_1 - \tau)} Bu(\tau) d\tau.$$

Here, e^{At} is the transition matrix.

Denote the sampling instants by $t_k = kT, k = 0, 1, \ldots$. Then letting t be any time in the interval $[t_k, t_{k+1})$, we have

$$x(t) = \mathrm{e}^{A(t-t_k)}x(t_k) + \int\limits_{t_k}^{t} \mathrm{e}^{A(t-\tau)}\,Bu(\tau)d\tau, \quad t_k \le t \le t_{k+1}.$$

Now $u(\tau)$, being the output of the hold, is constant $= u(t_k)$ over the interval $[t_k, t_{k+1})$. Thus

$$\begin{aligned} x(t) &= \mathrm{e}^{A(t-t_k)}x(t_k) + \int\limits_{t_k}^{t} \mathrm{e}^{A(t-\tau)}\,Bu(t_k)d\tau \\ &= \mathrm{e}^{A(t-t_k)}x(t_k) + \int\limits_{0}^{t-t_k} \mathrm{e}^{A\tau}d\tau Bu(t_k). \end{aligned}$$

Define the matrix-valued functions

$$\Phi(t) = \mathrm{e}^{At}, \quad \Gamma(t) = \int\limits_{0}^{t} \mathrm{e}^{A\tau}d\tau B.$$

Then, for t in the interval $[t_k, t_{k+1})$,

$$x(t) = \Phi(t - t_k)x(t_k) + \Gamma(t - t_k)u(t_k).$$

In particular, letting $t \to t_{k+1}$, we get

$$x(t_{k+1}) = \Phi(t_{k+1} - t_k)x(t_k) + \Gamma(t_{k+1} - t_k)u(t_k).$$

Since periodic sampling is used, $t_k = kT$ and

$$\Phi(t_{k+1} - t_k) = \mathrm{e}^{AT} = A_d$$

$$\Gamma(t_{k+1} - t_k) = \int\limits_{0}^{T} \mathrm{e}^{A\tau}d\tau B = B_d$$

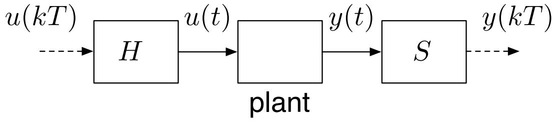
Thus the state equation at the sampling instants is

$$x[(k+1)T] = A_d x(kT) + B_d u(kT).$$

The output at sampling instants is simply

$$y(kT) = Cx(kT) + Du(kT).$$

Recap



If from $u(t)$ to $y(t)$ is

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

then from $u(kT)$ to $y(kT)$ is

$$\begin{aligned}x[(k+1)T] &= A_d x(kT) + B_d u(kT)\\ y(kT) &= Cx(kT) + Du(kT).\end{aligned}$$

Note that if A is nonsingular, we can explicitly integrate to get

$$\int\limits_0^T \mathrm{e}^{A\tau} d\tau = A^{-1}(\mathrm{e}^{AT} - I)$$

for use in the formula for B_d.

Proof

$$\int\limits_0^T A\mathrm{e}^{A\tau} d\tau = \int\limits_0^T \frac{d}{d\tau}\mathrm{e}^{A\tau} d\tau = \mathrm{e}^{A\tau}\Big|_0^T = \mathrm{e}^{AT} - I$$

□

Thus a continuous-time state model (A, B, C, D) is mapped to the discrete-time state model (A_d, B_d, C, D), where

$$A_d = \mathrm{e}^{AT}, \quad B_d = \int\limits_0^T \mathrm{e}^{A\tau} d\tau B.$$

We call this mapping c2d because that's the name of the MATLAB function: $[A_d, B_d] = c2d(A, B, T)$. In terms of transfer functions,

$$G(s) = C(sI - A)^{-1}B + D$$

is mapped to

$$G_d(z) = C(zI - A_d)^{-1}B_d + D.$$

The discrete-time transfer function $G_d(z)$ is called the pulse transfer function. Note that the discretized state model is exact, valid for all T—there's no approximation.

The determination of A_d and B_d involves the determination of e^{At}. The computation of e^{At} can be carried out using diagonalization or Laplace transforms.

Diagonalization

Suppose

$$V^{-1}AV = A_{JF} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n).$$

Then

$$\mathrm{e}^{At} = V\mathrm{e}^{A_{JF}t}V^{-1} \qquad = V \begin{bmatrix} \mathrm{e}^{\lambda_1 t} & & & \\ & \mathrm{e}^{\lambda_2 t} & & \\ & & \ddots & \\ & & & \mathrm{e}^{\lambda_n t} \end{bmatrix} V^{-1}.$$

Laplace transforms

Let $\mathcal{L}$ be the Laplace transform operator and $\mathcal{L}^{-1}$ its inverse. Then

$$\mathrm{e}^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}].$$

More details can be found in Chapter 1.

Example First-order system.

$$\dot{x} = \alpha x + \beta u$$

$$A_d = \mathrm{e}^{\alpha T}, \quad B_d = \int_0^T \mathrm{e}^{\alpha s} ds \beta = \frac{\beta}{\alpha}(\mathrm{e}^{\alpha T} - 1)$$

Discretized system is

$$x[(k+1)T] = \mathrm{e}^{\alpha T} x(kT) + \frac{\beta}{\alpha}(\mathrm{e}^{\alpha T} - 1)u(kT).$$

The case of $\alpha = 0$, $\beta = 1$, corresponding to an integrator, can be derived directly or by simply taking the limit of $\alpha \to 0$ to give

$$x[(k+1)T] = x(kT) + Tu(kT).$$

Thus, the discretization of the continuous-time transfer function $\frac{1}{s}$ gives rise to the discrete-time transfer function $\frac{T}{z-1}$. □

Example Double integrator.

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} x.$$

This is called the double integrator because

$$\ddot{y} = \ddot{x}_1 = \dot{x}_2 = u, \quad \text{i.e., } G(s) = \frac{1}{s^2}.$$

Since $A^2 = 0$ (A is nilpotent), so

$$\mathrm{e}^{At} = I + At = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}.$$

Alternatively, since

$$(sI - A)^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix},$$

so

$$\mathrm{e}^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}] = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}.$$

Hence

$$A_d = \mathrm{e}^{AT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

$$\begin{aligned} B_d &= \int_0^T \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} d\tau \\ &= \int_0^T \begin{bmatrix} \tau \\ 1 \end{bmatrix} d\tau \\ &= \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \end{aligned}$$

The discretized system is therefore

$$x(kT + T) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(kT) + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u(kT).$$

Observe that

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

$$A_d = \mathrm{e}^{AT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \text{eigs} = \{1, 1\}$$

The eigs of A are mapped to the eigs of A_d via

$$\lambda \mapsto \mathrm{e}^{\lambda T}.$$

This is an instance of the **spectral mapping theorem**, discussed below.

The discretized transfer function for the double integrator is

$$
\begin{aligned}
G_d(z) &= \begin{bmatrix} 1 & 0 \end{bmatrix} (zI - A_d)^{-1} B_d \\
&= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} z-1 & -T \\ 0 & z-1 \end{bmatrix}^{-1} B_d \\
&= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{z-1} & \frac{T}{(z-1)^2} \\ 0 & \frac{1}{z-1} \end{bmatrix} \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \\
&= \frac{T^2}{2}\frac{1}{z-1} + \frac{T^2}{(z-1)^2} = \frac{T^2}{2}\frac{z+1}{(z-1)^2}.
\end{aligned}
$$

□

Spectral Mapping Theorem

Let A be a square matrix and suppose λ is an eigenvalue:

$$Ax = \lambda x.$$

Consider the matrix

$$B = A^2 - 3I$$

which is a function of A. We have

$$Bx = A^2x - 3x = \lambda Ax - 3x = \lambda^2 x - 3x = (\lambda^2 - 3)x.$$

Thus $\lambda^2 - 3$ is an eigenvalue of B.

Look at this chart:

	A	λ_1, λ_2
$f(z) = z^2$	$f(A) = A^2$	λ_1^2, λ_2^2
$f(z) = z^2 - 3$	$f(A) = A^2 - 3I$	
$f(z) = \dfrac{1}{z+1}$	$f(A) = (A+I)^{-1}$	

The first line says A is a matrix with eigenvalues λ_1, λ_2. In the second line, $f(z) = z^2$ is a function of a variable z, and $f(A)$ is obtained by replacing z by A. Then the line says the eigenvalues of $f(A)$ are $f(\lambda_1), f(\lambda_2)$. In the third line, the eigenvalues of $f(A)$ are left out, but you can fill them in.

These are all instances of this theorem: Suppose A is square; suppose $f(z)$ is a function analytic at the eigenvalues of A; then the eigenvalues of $f(A)$ equal the numbers $f(\lambda)$, as λ ranges over the eigenvalues of A.

Back to our discretization: Given A, define $f(z) = \mathrm{e}^{zT}$. Then $A_d = \mathrm{e}^{AT} = f(A)$. So the eigenvalues of A_d are the numbers $\mathrm{e}^{\lambda T}$ as λ ranges over the eigenvalues of A.

So c2d discretizes the plant at the sampling instants. We can also determine the so-called **intersample behaviour**, i.e., the values of $x(t)$ and $y(t)$ for $kT < t < kT + T$.

We had

$$x(t) = \Phi(t - t_k)x(t_k) + \Gamma(t - t_k)u(t_k).$$

Setting $t_k = kT$, we get, for $kT < t < kT + T$,

$$x(t) = \Phi(t - kT)x(kT) + \Gamma(t - kT)u(kT) \tag{3.1}$$

The output $y(t)$, for $kT < t < kT + T$, is given by

$$y(t) = Cx(t) + Du(t) = Cx(t) + Du(kT)$$

Example Double integrator.

Replacing T by t in the discretization of the double integrator gives

$$\Phi(t) = e^{At} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

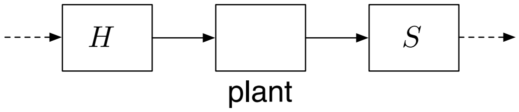
$$\Gamma(t) = \int_0^t e^{As} B ds = \begin{bmatrix} \frac{t^2}{2} \\ t \end{bmatrix}$$

Fix T. Suppose $u(t) = t$, so that $u(kT) = kT$, and $x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Let us determine $x(\frac{3T}{2})$. We have

$$x(T) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + T \\ 1 \end{bmatrix}$$

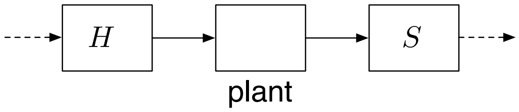
$$\begin{aligned} x(\frac{3T}{2}) &= \Phi(\frac{T}{2})x(T) + \Gamma(\frac{T}{2})u(T) \\ &= \begin{bmatrix} 1 & \frac{T}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 + T \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{(\frac{T}{2})^2}{2} \\ \frac{T}{2} \end{bmatrix} T \\ &= \begin{bmatrix} 1 + \frac{3T}{2} + \frac{T^3}{8} \\ 1 + \frac{T^2}{2} \end{bmatrix} \end{aligned}$$

So far, we have described how to determine $G_d(z)$ from a given continuous state-space description:



$$\begin{tikzcd}
\underline{\text{continuous time}} & \underline{\text{discrete time}} \\
G(s) & G_d(z) \\
(A,B,C,D) \arrow[r] & (A_d,B_d,C_d,D_d) \arrow[u]
\end{tikzcd}$$

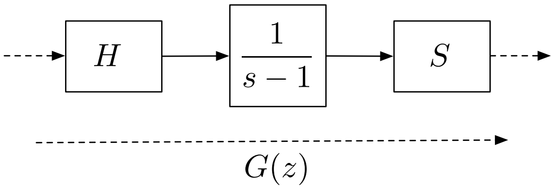
However, we often begin with a continuous-time $G(s)$. To determine the discretized transfer function $G_d(z)$, we can first realize $G(s)$ by its continuous time state space representation and then apply the above results. In other words, we want to do this:



$$\begin{tikzcd}
\underline{\text{continuous time}} & \underline{\text{discrete time}} \\
G(s) \arrow[d] & G_d(z) \\
(A,B,C,D) \arrow[r] & (A_d,B_d,C_d,D_d) \arrow[u]
\end{tikzcd}$$

We illustrate the steps with some examples.

Example



Let's find $G(z)$:

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

$\Longrightarrow$

$$\begin{aligned} x[(k+1)T] &= A_d x(kT) + B_d u(kT) \\ y(kT) &= Cx(kT) + Du(kT) \end{aligned}$$

$$\frac{Y}{U} = \frac{1}{s-1},$$

$$\dot{x} = x + u,\ \ y = x, \quad A = 1, B = 1, C = 1, D = 0$$

$$A_d = \mathrm{e}^{AT} = \mathrm{e}^{T}, \quad B_d = \int\limits_0^T \mathrm{e}^{A\tau} d\tau B = \int\limits_0^T \mathrm{e}^{\tau} d\tau = \mathrm{e}^{T} - 1$$

$$x[(k+1)T] = \mathrm{e}^{T} x(kT) + (\mathrm{e}^{T} - 1) u(kT)$$

$$zX(z) = \mathrm{e}^{T} X(z) + (\mathrm{e}^{T} - 1) U(z)$$

$$G(z) = \frac{\mathrm{e}^{T} - 1}{z - \mathrm{e}^{T}}$$

□

Example Given

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

find $G_d(z)$.

Step 1 Find a state space realization of $G(s)$: The controllable and observable realizations for a continuous-time transfer function can be written down in exactly the same way as their discrete-time counterparts described in Chapter 2. Usually the controllable realization is simpler to use because the B matrix in the controllable representation simplifies the determination of B_d. We have

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

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad D = 0.$$

Step 2 Determine e^{At}. Here A isn't diagonalizable, so we can use $\mathcal{L}^{-1}\{(sI - A)^{-1}\}$:

$$(sI - A)^{-1} = \begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 0 & 0 & s+1 \end{bmatrix}^{-1} = \frac{1}{s^2(s+1)} \begin{bmatrix} s(s+1) & (s+1) & 1 \\ 0 & s(s+1) & s \\ 0 & 0 & s^2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} & \frac{1}{s^2(s+1)} \\ 0 & \frac{1}{s} & \frac{1}{s(s+1)} \\ 0 & 0 & \frac{1}{s+1} \end{bmatrix}.$$

Inverting the Laplace transforms by the residue method, we obtain

$$\mathrm{e}^{At} = \begin{bmatrix} 1 & t & -1 + t + \mathrm{e}^{-t} \\ 0 & 1 & 1 - \mathrm{e}^{-t} \\ 0 & 0 & \mathrm{e}^{-t} \end{bmatrix}.$$

Step 3 Compute A_d and B_d:

$$A_d = \mathrm{e}^{AT} = \begin{bmatrix} 1 & T & -1+T+\mathrm{e}^{-T} \\ 0 & 1 & 1-\mathrm{e}^{-T} \\ 0 & 0 & \mathrm{e}^{-T} \end{bmatrix}$$

$$B_d = \int_0^T \mathrm{e}^{A\tau} d\tau B = \int_0^T \begin{bmatrix} -1+t+\mathrm{e}^{-t} \\ 1-\mathrm{e}^{-t} \\ \mathrm{e}^{-t} \end{bmatrix} dt = \begin{bmatrix} -T+\frac{T^2}{2}+1-\mathrm{e}^{-T} \\ T-1+\mathrm{e}^{-T} \\ 1-\mathrm{e}^{-T} \end{bmatrix}.$$

For example, for $T = 1$, we get $\mathrm{e}^{-T} = \mathrm{e}^{-1} = 0.3679$

$$A_d = \begin{bmatrix} 1 & 1 & 0.3679 \\ 0 & 1 & 0.6321 \\ 0 & 0 & 0.3679 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0.1321 \\ 0.3679 \\ 0.6321 \end{bmatrix}.$$

Step 4 Get $G_d(z)$:

$$G_d(z) = C(zI - A_d)^{-1}B_d$$

$$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z-1 & -1 & -0.3679 \\ 0 & z-1 & -0.6321 \\ 0 & 0 & z-0.3679 \end{bmatrix}^{-1} \begin{bmatrix} 0.1321 \\ 0.3679 \\ 0.6321 \end{bmatrix}.$$

—tedious calculation.

□

Note that the form of A_d lends itself to an easy computation of A_d^k, required in the solution of the sampled-data state equation:

$$A_d^k = [\mathrm{e}^{AT}]^k = \mathrm{e}^{AkT}.$$

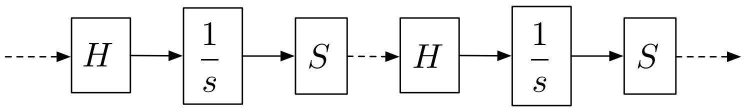
Since we would have determined e^{At} as part of the calculation for A_d, we simply replace t by kT to get A_d^k.

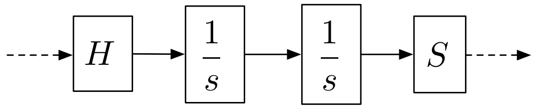
The Function c2d

To clarify what we've been doing, let's define the function c2d that maps $G(s)$ to $G_d(z)$: E.g.,

$$\mathsf{c2d} : \frac{1}{s} \mapsto \frac{T}{z-1}.$$

Look at this example:





$$\text{top}: \frac{T}{z-1}\frac{T}{z-1} = \frac{T^2}{(z-1)^2}$$

$$\text{bottom}: \frac{T^2}{2}\frac{z+1}{(z-1)^2}$$

Thus in general

$$\mathsf{c2d}(G_1G_2) \neq \mathsf{c2d}(G_1)\mathsf{c2d}(G_2).$$

But is c2d linear? For example, we know that

$$\frac{1}{s} \mapsto \frac{T}{z-1} \text{ and } \frac{1}{s-1} \mapsto \frac{\mathrm{e}^T - 1}{z - \mathrm{e}^T}.$$

Does it then follow that

$$\left\{\frac{1}{s} + \frac{1}{s-1}\right\} \mapsto \left\{\frac{T}{z-1} + \frac{\mathrm{e}^T - 1}{z - \mathrm{e}^T}\right\}?$$

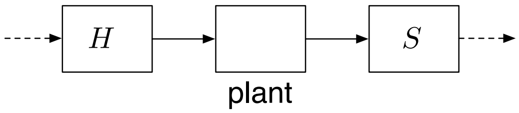
Yes. Try proving yourself or see the next section.

Finally, show that

$$\mathsf{c2d}: G(s) = 1 \;\mapsto\; G_d(z) = 1.$$

3.3 Transform Analysis

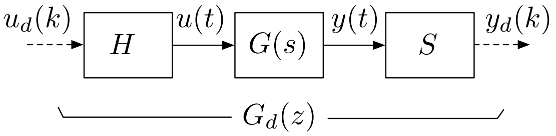
Is there a direct way to go from $G(s)$ to $G_d(z)$? We may assume $G(s)$ to be strictly proper since a constant term in $G(s)$ gives rise to the same constant term in $G_d(z)$.



continuous time discrete time

$$\begin{tikzcd}
G(s) \arrow[r] \arrow[d] & G_d(z) \\
(A,B,C,D) \arrow[r] & (A_d,B_d,C_d,D_d) \arrow[u]
\end{tikzcd}$$

Here's a procedure for getting $G_d(z)$ from $G(s)$ when $G(s)$ is strictly proper. In the following figure $u_d(k) = u(kT)$ and $y_d(k) = y(kT)$.



1. Let $u_d(k)$ be the unit impulse, so $U_d(z) = 1$. Now the corresponding output $y_d(k)$ is the impulse response of the discretized plant, hence its z-transform is the desired discretized transfer function.

2. Applying the hold operation, we get $u(t)$ is a unit-height pulse of width T, i.e., $u(t) = 1_+(t) - 1_+(t-T)$, so that $U(s) = \frac{1}{s}\left(1 - \mathrm{e}^{-sT}\right)$.

3. Then $Y(s) = G(s)U(s)$. Get $y(t)$. This can be done as follows:

   (a) Set $Y_1(s) = \frac{G(s)}{s}$. Determine $y_1(t)$ by taking the inverse Laplace transform of $Y_1(s)$.

   (b) Write $y(t) = y_1(t)1_+(t) - y_1(t-T)1_+(t-T)$ where $1_+(t)$ is the unit step.

4. Sample $y(t)$ to get $y_d(k) = y_1(kT) - y_1(kT-T) = y_{1d}(k) - y_{1d}(k-1)$. Then $G_d(z) = Y_d(z)$. As in step 3, we can find $G_d(z)$ by

   (a) Determine $Y_{1d}(z)$, the z-transform of $y_{1d}(k)$

   (b) Write $G_d(z) = (1 - z^{-1})Y_{1d}(z)$.

For a continuous-time signal $f(t)$ with Laplace transform $F(s)$ and sampling time T, it is convenient to introduce the notation

$$\mathcal{Z}(F(s)) = \sum_{k=0}^{\infty} f(kT)z^{-k}$$

The procedure for determining the pulse transfer function $G_d(z)$ from a strictly proper continuous-time transfer function $G(s)$ can then be summarized by the formula:

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

For signals $x(t)$ whose Laplace transform transform $X(s)$ has only simple poles, we can determine $\mathcal{Z}(X(s))$ easily. Let $X(s)$ be expanded into the partial fractions representation

$$X(s) = \sum_{j=1}^{n} \frac{A_j}{s - p_j}$$

Inverting the Laplace transform gives

$$\begin{aligned} x(t) &= \sum_k A_j e^{p_j t} \\ x(kT) &= \sum A_j e^{p_j kT} \\ &= \sum A_j (e^{p_j T})^k \end{aligned}$$

Taking z-transform yields

$$X(z) = \sum_j A_j \frac{1}{1 - e^{p_j T} z^{-1}}$$

We can write this as

$$\mathcal{Z}(\sum_{j=1}^{n}\frac{A_j}{s-p_j}) = \sum_j A_j \frac{1}{1-e^{p_jT}z^{-1}}$$

Example

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

Find $G_d(z)$.

(a) State Space Method

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$(sI-A)^{-1} = \begin{bmatrix} s & -1 \\ 0 & s+1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s(s+1)} \\ 0 & \frac{1}{s+1} \end{bmatrix}$$

$$\mathrm{e}^{At} = \begin{bmatrix} 1 & 1-\mathrm{e}^{-t} \\ 0 & \mathrm{e}^{-t} \end{bmatrix}$$

$$A_d = \mathrm{e}^{AT} = \begin{bmatrix} 1 & 1-\mathrm{e}^{-T} \\ 0 & \mathrm{e}^{-T} \end{bmatrix}$$

$$B_d = \int_0^T \mathrm{e}^{A\tau} d\tau B = \int_0^T \begin{bmatrix} 1-\mathrm{e}^{-\tau} \\ \mathrm{e}^{-\tau} \end{bmatrix} d\tau = \begin{bmatrix} T-(1-\mathrm{e}^{-T}) \\ 1-\mathrm{e}^{-T} \end{bmatrix}$$

$$\begin{aligned} & C(zI-A_d)^{-1}B_d \\ = \ & [1 \ \ 0]\begin{bmatrix} z-1 & \mathrm{e}^{-T}-1 \\ 0 & z-\mathrm{e}^{-T} \end{bmatrix}^{-1} \begin{bmatrix} T+\mathrm{e}^{-T}-1 \\ 1-\mathrm{e}^{-T} \end{bmatrix} \\ = \ & \cdots \\ = \ & \frac{(T-1+\mathrm{e}^{-T})z^{-1} + (1-\mathrm{e}^{-T}-T\mathrm{e}^{-T})z^{-2}}{(1-z^{-1})(1-\mathrm{e}^{-T}z^{-1})} \end{aligned}$$

(b) Transfer Function Method

$$U_d(z) = 1$$

$$U(s) = \frac{1}{s}\left(1-\mathrm{e}^{-sT}\right)$$

$$Y(s) = G(s)U(s) = \frac{1}{s^2(s+1)}\left(1-\mathrm{e}^{-sT}\right)$$

$$Y(s) = G(s)U(s) = \underbrace{\left(\frac{1}{s^2}-\frac{1}{s}+\frac{1}{s+1}\right)}_{Y_1(s)}\left(1-\mathrm{e}^{-sT}\right)$$

$$y_1(t) = (t - 1 + \mathrm{e}^{-t})1_+(t)$$

The z-transform of t sampled is

$$\sum_{k=0}^{\infty}(kT)z^{-k} = T\sum_{k=0}^{\infty} kz^{-k} = \frac{Tz^{-1}}{(1-z^{-1})^2}.$$

Thus the z-transform of $y_1(kT)$ is

$$Y_{1d}(z) = \frac{Tz^{-1}}{(1-z^{-1})^2} - \frac{1}{1-z^{-1}} + \frac{1}{1-\mathrm{e}^{-T}z^{-1}}.$$

Now

$$y(t) = y_1(t)1_+(t) - y_1(t-T)1_+(t-T)$$

so

$$y(kT) = y_1(kT)1_+(kT) - y_1((k-1)T)1_+(kT-T), \quad Y_d(z) = (1-z^{-1})Y_{1d}(z).$$

Finally

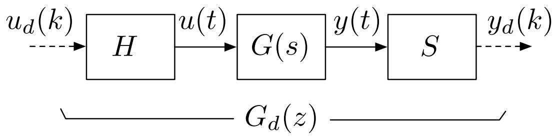
$$\begin{aligned} G_d(z) &= (1-z^{-1})\left\{\frac{Tz^{-1}}{(1-z^{-1})^2} - \frac{1}{1-z^{-1}} + \frac{1}{1-\mathrm{e}^{-T}z^{-1}}\right\} \\ &= \frac{(T-1+\mathrm{e}^{-T})z^{-1} + (1-\mathrm{e}^{-T} - T\mathrm{e}^{-T})z^{-2}}{(1-z^{-1})(1-\mathrm{e}^{-T}z^{-1})} \end{aligned}$$

—same as by the state space method.

□

3.4 Effect of Sampling

We have been studying the setup



What do the sample and hold operations have on $G(s)$ in producing $G_d(z)$?

Let ω_s denote the **sampling frequency** (rad/s), that is,

$$\omega_s := \frac{2\pi}{T}.$$

Example Take $G(s)$ to be an oscillator at frequency ω_s:

$$G(s) = \frac{\omega_s s}{s^2 + \omega_s^2}.$$

Apply the unit step at $u_d(k)$. Then $u(t)$ is the unit step in continuous time. Hence

$$y(t) = \mathcal{L}^{-1}\left(\frac{\omega_s}{s^2+\omega_s^2}\right) = \sin \omega_s t,$$

and so

$$y(kT) = 0.$$

It follows that $G_d(z) \equiv 0$. So here is a nonzero system $G(s)$ whose discretization is zero. But the discretization of $G = 0$ is also $G_d = 0$. This shows that the mapping $G(s) \mapsto G_d(z)$ is *not* one-to-one: Two different continuous-time systems can have the same discretization. Put another way, the linear mapping `c2d` has a nontrivial nullspace.

□

Frequency Response

If $G(j\omega)$ is the frequency response of a stable continuous-time system and if the sinusoid $\mathrm{e}^{j\omega t}$ is applied at the input, then the output will be the sinusoid $G(j\omega)\mathrm{e}^{j\omega t}$. (In this statement, either the time interval is $(-\infty, \infty)$ or else the output is taken in steady state.) Likewise, if $G_d\left(\mathrm{e}^{j\theta}\right)$ is the frequency response of a stable discrete-time system and if the sinusoid $\mathrm{e}^{j\theta k}$ is applied at the input, then the output will be the sinusoid $G_d\left(\mathrm{e}^{j\theta}\right)\mathrm{e}^{j\theta k}$. Both these results are easy to derive using the convolution equation.

Next, let us review sampling. Consider the sinusoid $x(t) = \mathrm{e}^{j\omega t}$ of frequency ω radians/second. Suppose it's sampled with sampling period T; this yields $x(kT) = \mathrm{e}^{j\omega Tk}$. We consider this as the discrete-time signal

$$x_d(k) = \mathrm{e}^{j\omega Tk} = \mathrm{e}^{j\theta k},$$

where $\theta = \omega T$ has units of radians. Now the ω axis can be divided into non-overlapping intervals of width $2\pi/T$:

$$\ldots, \left[-\frac{\pi}{T}, \frac{\pi}{T}\right), \left[\frac{\pi}{T}, \frac{3\pi}{T}\right), \ldots$$

Let us suppose ω is in the interval

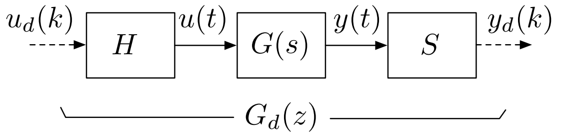
$$\left[\frac{\pi}{T}, \frac{3\pi}{T}\right).$$

Then $\omega = \frac{2\pi}{T} + \omega_0$, where ω_0 is in the "baseband" $\left[-\frac{\pi}{T}, \frac{\pi}{T}\right)$. Also,

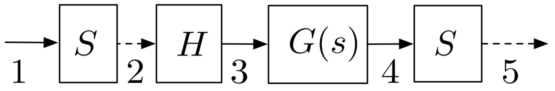
$$x_d(k) = \mathrm{e}^{j\theta_0 k},$$

where $\theta_0 = \omega_0 T$ is in the interval $[-\pi, \pi)$. In this way, the frequency range for signals in discrete time is $[-\pi, \pi)$.

Now we turn to systems. Let $G(s)$ be a transfer function and therefore $G(j\omega)$ a frequency response.



Then we have the transfer function $G_d(z)$ and the frequency response $G_d\left(\mathrm{e}^{j\theta}\right)$. What's the relationship between $G(j\omega)$ and $G_d\left(\mathrm{e}^{j\theta}\right)$? The exact relationship is derived in the Appendix. Here we look only at low frequency. Let ω_s denote the sampling frequency ($2\pi/T$) and ω_N the Nyquist frequency (π/T). Consider the setup



Let us take the signal labelled 1 to be $\mathrm{e}^{j\omega t}$ with $\omega << \omega_N$. Then signal 2 is $\mathrm{e}^{j\theta k}$ with $\theta = \omega T << \pi$. So signal 5 is $G_d(\mathrm{e}^{j\theta})\mathrm{e}^{j\theta k}$. On the other hand, signal 3 is very close to $\mathrm{e}^{j\omega t}$. So signal 4 is $G(j\omega)\mathrm{e}^{j\omega t}$, hence 5 is $G(j\omega)\mathrm{e}^{j\omega Tk}$. We conclude that

$$G_d(\mathrm{e}^{j\theta})\mathrm{e}^{j\theta k} \approx G(j\omega)\mathrm{e}^{j\omega Tk} = G\left(j\frac{\theta}{T}\right)\mathrm{e}^{j\theta k}.$$

Thus at low frequency

$$G_d(\mathrm{e}^{j\theta}) \approx G\left(j\frac{\theta}{T}\right) \text{ or } G_d(\mathrm{e}^{j\omega T}) \approx G(j\omega).$$

3.5 Exercises

1. The discretization (using S and H) of

$$\dot{x}(t) = Ax(t) + Bu(t)$$

is

$$x(kT+T) = A_d x(kT) + B_d u(kT),$$

where

$$A_d = \mathrm{e}^{AT}, \quad B_d = \int_0^T \mathrm{e}^{A\tau} d\tau B.$$

This problem develops a way for computing A_d and B_d; this is the method used in MATLAB's function c2d. (The Scilab function to discretize via S and H is dscr.)

Let

$$\Phi(t) = \mathrm{e}^{At}, \quad \Gamma(t) = \int_0^t \mathrm{e}^{A\tau} d\tau B.$$

(a) Show that

$$\frac{d}{dt}\begin{bmatrix} \Phi(t) & \Gamma(t) \\ 0 & I \end{bmatrix} = \begin{bmatrix} \Phi(t) & \Gamma(t) \\ 0 & I \end{bmatrix}\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}$$

(b) Verify that the solution to the matrix differential equation (M, R square)

$$\frac{d}{dt}M(t) = M(t)R$$

is $M(t) = M(0)\mathrm{e}^{Rt}$. Thus

$$\begin{bmatrix} \Phi(t) & \Gamma(t) \\ 0 & I \end{bmatrix} = \mathrm{e}^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}t}$$

(c) Finally,

$$\begin{bmatrix} A_d & B_d \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}\mathrm{e}^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}T}$$

2. Let $u_1(k) = 1$ and $u_2(k) = k$. Show that the zero-order hold H satisfies the superposition property

$$H(u_1 + u_2) = H(u_1) + H(u_2).$$

(In fact, H is a linear system.)

3. For this problem, redefine the zero-order hold H to act as follows: If the input is $u(k)$, then the output is $y(t)$, where

$$y(t) = u(k), \quad kT < t \leq (k+1)T.$$

(The $\leq$ sign has been moved.) Find the transfer function of the system SH (you don't have to prove it's time-invariant).

4. Consider the double integrator $G(s) = \dfrac{1}{s^2}$. Let $G_d(z)$ denote the discretization via c2d. Determine $G_d(z)$ by the following method:

(a) Write down the controllable state model $(A, B, C, D = 0)$ of $G(s)$.

(b) Determine $\mathrm{e}^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}T}$ and hence (A_d, B_d).

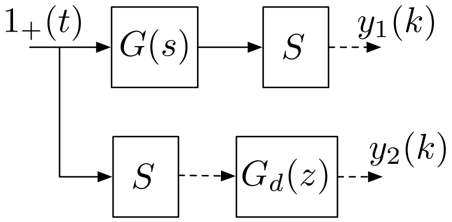
(c) Finally, get

$$G_d(z) = C(zI - A_d)^{-1}B_d.$$

5. Consider the double integrator $G(s) = \dfrac{1}{s^2}$. Let $G_d(z)$ denote the discretization via c2d. Determine $G_d(z)$ by the transfer function method.

6. The discretization c2d is sometimes called the *step-invariant transformation*. This problem explains why.

   Consider the block diagram

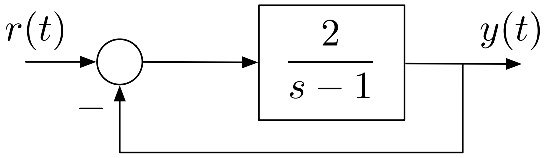


   Let $G(s)$ be the transfer function of a linear time-invariant system, let $G_d(z)$ be the discretization of $G(s)$ via c2d, and let $1_+(t)$ denote the unit step. Show that $y_1(k)$ and $y_2(k)$ are equal. Thus $G_d(z)$ exactly models the sampled output of $G(s)$ when the input is a step.

7. Consider the system $G(s) = \dfrac{s-1}{s(s-2)}$. Let $G_d(z)$ denote the discretization via c2d. Determine $G_d(z)$ by the transfer function method, using the fact that c2d is linear.

8. This problem gives you a preview of sampled-data feedback control systems.

   (a) Consider the following continuous-time system—unity feedback around a plant:



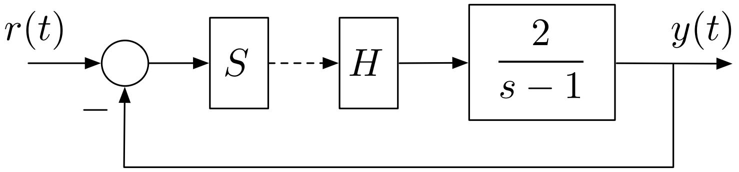
   Is the plant open-loop stable? Is the closed-loop system stable?

   (b) Suppose $r(t)$ is a finite-duration pulse, say,

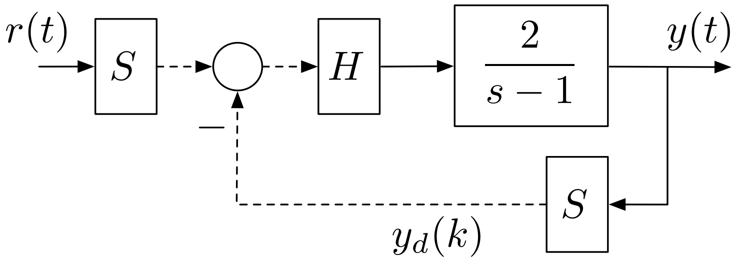
$$r(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0, & \text{else.} \end{cases}$$

   Sketch $y(t)$.

   (c) Suppose now that the plant input is obtained from the sampled tracking error like this:



What is the justification for moving S past the summing junction like this:



(d) Assume $T = \ln 4$ and $r(t)$ is the finite-duration pulse

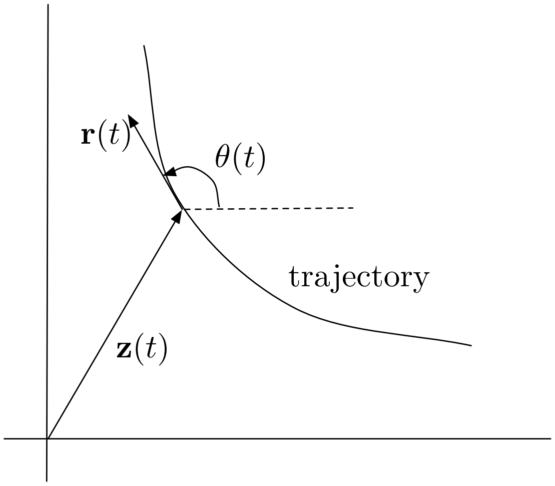
$$r(t) = \begin{cases} 1, & 0 \le t \le T/2 \\ 0, & \text{else.} \end{cases}$$

By hand, determine $y_d(k)$.

(e) Does $y_d(k) \to 0$?

(f) Is your expression for $y_d(k)$ exact or an approximation?

9. A robot rover has a complicated mathematical model. The main feature is that it has wheels and this makes control more difficult—think of parking a car. We're going to model the simplest wheeled rover, what's called a *kinematic unicycle* moving in the plane. At time t the unicycle has a position vector $\mathbf{z}(t)$ and a unit pointing vector $\mathbf{r}(t)$ (the robot is heading either in the direction $\mathbf{r}(t)$ or in the direction $-\mathbf{r}(t)$). Let $v(t)$ denote the speed, so that $v(t)\mathbf{r}(t)$ is the instantaneous velocity vector (v can be positive or negative). Next, let $\theta(t)$ denote the angle of $\mathbf{r}(t)$ with respect to the positive x-axis. Finally, let $\omega(t) = \dot{\theta}(t)$. The picture is this:



The velocity equation is

$$\dot{\mathbf{z}} = v\mathbf{r}.$$

The vector $\mathbf{r}$ equals $(\cos\theta, \sin\theta)$ and hence in terms of the components x, y of $\mathbf{z}$ we have

$$\begin{aligned}\dot{x} &= v\cos\theta\\ \dot{y} &= v\sin\theta\\ \dot{\theta} &= \omega.\end{aligned}$$

We regard the inputs as v, forward velocity, and ω, turning rate, and the state components as x, y, θ. Thus the unicycle model is nonlinear.

For this course, we need a linear model. Consider a nominal trajectory where the forward speed is a nonzero constant and the heading angle is constant:

$$x_0(t), y_0(t), \theta(t) = \theta_0, v_0(t) = v_0 \neq 0, \omega_0(t) = 0.$$

We have

$$\begin{aligned}\dot{x}_0 &= v_0\cos\theta_0\\ \dot{y}_0 &= v_0\sin\theta_0.\end{aligned}$$

So

$$\frac{dy_0}{dx_0} = \tan\theta_0,$$

and hence

$$y_0(t) = (\tan\theta_0)x_0(t) + c.$$

That is, the motion is confined to a straight line in the (x, y)-plane.

Fix such a solution and consider a variation about it:

$$\begin{aligned}x(t) &= x_0(t) + \delta x(t)\\ y(t) &= y_0(t) + \delta y(t)\\ \theta(t) &= \theta_0 + \delta\theta(t)\\ v(t) &= v_0 + \delta v(t)\\ \omega(t) &= \delta\omega(t).\end{aligned}$$

Then we have

$$\begin{aligned}\dot{x}_0 + \dot{\delta x} &= (v_0 + \delta v)\cos(\theta_0 + \delta\theta)\\ \dot{y}_0 + \dot{\delta y} &= (v_0 + \delta v)\sin(\theta_0 + \delta\theta)\\ \dot{\theta}_0 + \dot{\delta\theta} &= \omega.\end{aligned}$$

Expanding $\cos, \sin$ and retaining only first-order variations gives

$$\begin{aligned}\dot{\delta x} &= (\cos\theta_0)\delta v - (v_0\sin\theta_0)\delta\theta\\ \dot{\delta y} &= (\sin\theta_0)\delta v + (v_0\cos\theta_0)\delta\theta\\ \dot{\delta\theta} &= \omega.\end{aligned}$$

This linear model has matrices

$$A = \begin{bmatrix} 0 & 0 & -v_0 \sin\theta_0 \\ 0 & 0 & v_0 \cos\theta_0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \cos\theta_0 & 0 \\ \sin\theta_0 & 0 \\ 0 & 1 \end{bmatrix}.$$

and is therefore controllable for all $v_0 \neq 0$ and all θ_0 (PBH test). By this means, the unicycle can be locally stabilized about the trajectory. Let's take $v_0 = 1, \theta_0 = 0$. Then

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

Finally the exercise: Apply `c2d` to (A, B) to get (A_d, B_d).

10. Consider the continuous-time signal $y(t) = \cos 2t$. The poles of its Laplace transform are at $s = \pm 2j$. Suppose $y(t)$ is sampled with sampling period T to produce $y_d(k) = \cos 2Tk$. The poles of its z-transform can be determined to be $\mathrm{e}^{\pm j2T}$. Thus, if λ is a pole of $Y(s)$, then $\mathrm{e}^{\lambda T}$ is a pole of $Y_d(z)$. Is this an instance of the spectral mapping theorem? It can't be, because there's no matrix in sight, hence there are no eigenvalues to map.

    However, the spectral mapping can indeed be invoked as follows. Find a state model, forced only by initial conditions, whose output is $y(t) = \cos 2t$, that is, find $C, A, x(0)$ such that

    $$y(t) = C\mathrm{e}^{At}x(0).$$

    In terms of these state parameters, find $Y_d(z)$. Now invoke the spectral mapping theorem to show how the poles of $Y(s)$ are mapped to the poles of $Y_d(z)$.

    Finally, is this setup an instance of `c2d`?

Appendix 1: Relationship between $G(j\omega)$ and $G_d(\mathrm{e}^{j\theta})$

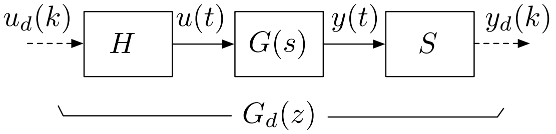
The first step is to derive the relationship between the continuous-time Fourier transform of $y(t)$ and the discrete-time Fourier transform of $y_d(k)$, that is, the frequency-domain action of S.

The **periodic extension** of a function $Y(j\omega)$ is

$$Y_e(j\omega) := \sum_{k=-\infty}^{\infty} Y(j\omega + jk\omega_s).$$

Note that Y_e is a periodic function of frequency, of period ω_s:

$$Y_e[j(\omega + \omega_s)] = Y_e(j\omega).$$



Lemma 6 *The Fourier transforms of $y(t)$ and $y_d(k)$ are related by the equation*

$$Y_d\left(\mathrm{e}^{j\omega T}\right) = \frac{1}{T} Y_e(j\omega). \tag{3.2}$$

Proof We prove this using the idea of **impulse-train modulation**. Define the continuous-time signal $v(t)$:

$$v(t) = y(t) \times \sum_k \delta(t - kT).$$

The impulse train $\sum_k \delta(t - kT)$ is a periodic function of time, of period T; its Fourier series is formally given by

$$\sum_k \delta(t - kT) = \frac{1}{T} \sum_k \mathrm{e}^{jk\omega_s t}.$$

Thus we have (formally)

$$\begin{aligned} v(t) &= y(t) \times \frac{1}{T} \sum_k \mathrm{e}^{jk\omega_s t} \\ &= \frac{1}{T} \sum_k y(t) \mathrm{e}^{jk\omega_s t}. \end{aligned}$$

Taking Fourier transforms, we get

$$\begin{aligned} V(j\omega) &= \frac{1}{T} \sum_k Y(j\omega + jk\omega_s). \\ &= \frac{1}{T} Y_e(j\omega). \end{aligned} \tag{3.3}$$

On the other hand,

$$\begin{aligned} v(t) &= y(t) \times \sum_k \delta(t - kT) \\ &= \sum_k y(kT) \delta(t - kT) \\ &= \sum_k y_d(k) \delta(t - kT). \end{aligned}$$

Taking Fourier transforms again, we get

$$\begin{aligned} V(j\omega) &= \int \left[\sum_k y_d(k) \delta(t - kT) \right] \mathrm{e}^{-j\omega t} dt \\ &= \sum_k y_d(k) \int \delta(t - kT) \mathrm{e}^{-j\omega t} dt \\ &= \sum_k y_d(k) \mathrm{e}^{-j\omega kT} \\ &= Y_d\left(\mathrm{e}^{j\omega T}\right). \end{aligned} \tag{3.4}$$

Comparing (3.3) and (3.4), we get the desired equation.

□

The second step is to derive the relationship between the discrete-time Fourier transform of $u_d(k)$ and the continuous-time Fourier transform of $u(t)$, that is, the frequency-domain action of H.

We need the system with impulse response

$$r(t) = \frac{1}{T}1_+(t) - \frac{1}{T}1_+(t-T),$$

where $1_+(t)$ is the unit step; that is,

$$r(t) = \begin{cases} 1/T, & 0 \le t < T \\ 0, & \text{elsewhere.} \end{cases} \tag{3.5}$$

The transfer function is therefore

$$R(s) = \frac{1-\mathrm{e}^{-sT}}{sT}. \tag{3.6}$$

The frequency-response function can be calculated like this:

$$\begin{aligned} R(j\omega) &= \frac{1-\mathrm{e}^{-j\omega T}}{j\omega T} \\ &= \mathrm{e}^{-j\omega\frac{T}{2}}\frac{\mathrm{e}^{j\omega\frac{T}{2}} - \mathrm{e}^{-j\omega\frac{T}{2}}}{j\omega T} \\ &= \mathrm{e}^{-j\omega\frac{T}{2}}\frac{\sin\omega\frac{T}{2}}{w\frac{T}{2}}. \end{aligned}$$

Observe that

$$R(s) \approx \mathrm{e}^{-s\frac{T}{2}} \qquad \text{at low frequency.}$$

So at low frequency R acts like a time delay of $\frac{T}{2}$.

Lemma 7 *The Fourier transforms of $u_d(k)$ and $u(t)$ are related by the equation*

$$U(j\omega) = TR(j\omega)U_d\left(\mathrm{e}^{j\omega T}\right). \tag{3.7}$$

Proof From the definitions of $r(t)$ and H, we can write

$$u(t) = T\sum_k u_d(k)r(t-kT).$$

Taking Fourier transforms, we get

$$\begin{aligned} U(j\omega) &= T\sum_k u_d(k)R(j\omega)\mathrm{e}^{-j\omega kT} \\ &= TR(j\omega)U_d\left(\mathrm{e}^{j\omega T}\right). \end{aligned}$$

□

Now we put the two preceding lemmas together:

Theorem 7 *The frequency responses $G(j\omega)$ and $G_d\left(\mathrm{e}^{j\theta}\right)$ are related by*

$$G_d\left(\mathrm{e}^{j\omega T}\right) = \sum_{k=-\infty}^{\infty} G(j\omega + jk\omega_s)R(j\omega + jk\omega_s).$$

Proof Let $u_d(k)$ be the unit impulse. Then

$$\begin{aligned}
U(j\omega) &= TR(j\omega), \quad \text{from Lemma 7} \\
Y(j\omega) &= G(j\omega)U(j\omega) \\
&= G(j\omega)TR(j\omega) \\
G_d\left(\mathrm{e}^{j\omega T}\right) &= Y_d\left(\mathrm{e}^{j\omega T}\right) \\
&= \frac{1}{T}\sum_{k=-\infty}^{\infty} Y(j\omega + jk\omega_s), \quad \text{from Lemma 6.}
\end{aligned}$$

□

Thus, at each frequency ω, $G_d\left(\mathrm{e}^{j\omega T}\right)$ depends not only on $G(j\omega)$, but also on all the values

$$G(j\omega + jk\omega_s), \quad k = \pm 1, \pm 2, \ldots.$$

In the ideal case that $G(j\omega)$ is bandlimited to the interval $(-\omega_N, \omega_N)$, then

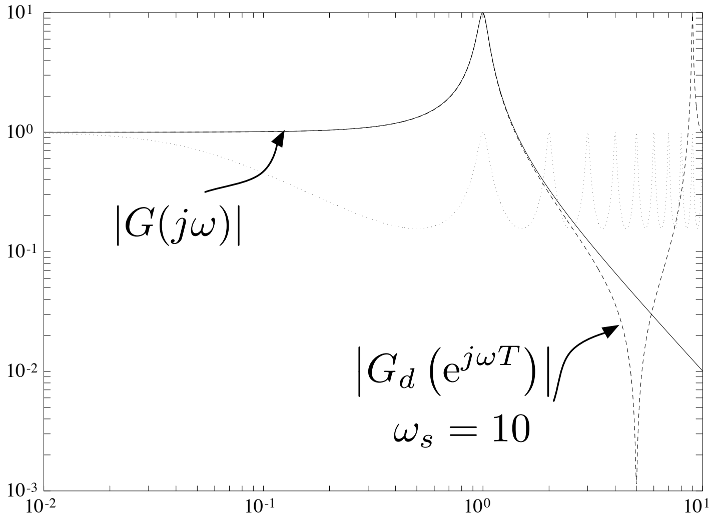
$$G_d\left(\mathrm{e}^{j\omega T}\right) = G(j\omega)R(j\omega), \quad -\omega_N < \omega < \omega_N.$$

In particular, at low frequencies

$$G_d\left(\mathrm{e}^{j\omega T}\right) \approx G(j\omega).$$

Example

$$G(s) = \frac{1}{s^2 + 0.1s + 1}$$



To graph $G_d\left(\mathrm{e}^{j\omega T}\right)$, you should use the formula

$$G_d\left(\mathrm{e}^{j\omega T}\right) = C\left(\mathrm{e}^{j\omega T} I - A_d\right)^{-1} B_d + D$$

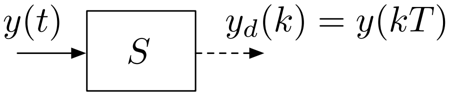
instead of Theorem 7.

□

Appendix 2: On the Sampling Theorem

In case you're taking or have taken the DSP course, it may be illuminating to make a connection between what we're doing in this chapter and the sampling theorem.

Consider a continuous-time signal $y(t)$, where t ranges over $-\infty < t < \infty$. Let it be sampled:



Assume $y(t)$ is bandlimited to β, where $\beta < \omega_N$, the Nyquist frequency. That is, $Y(j\omega) = 0$ for $\omega > \beta$. Let us bring in impulse-train modulation again as in the preceding appendix:

$$v(t) = y(t) \times \sum_k \delta(t - kT)$$

$$V(j\omega) = \frac{1}{T} \sum_k Y(j\omega + jk\omega_s).$$

Because of the bandlimited assumption, the graphs of the functions $Y(j\omega), Y(j\omega + jk\omega_s), k \neq 0$, don't overlap. Therefore, for $\omega < \omega_N$

$$V(j\omega) = \frac{1}{T} Y(j\omega).$$

Thus $y(t)$ can be recovered by ideally lowpass filtering $v(t)$ with a gain of T and cutoff frequency of ω_N. Such an ideal low-pass filter has frequency response

$$\begin{aligned} H(j\omega) &= T, \quad -\omega_N \leq \omega \leq \omega_N \\ &= 0, \quad \text{otherwise} \end{aligned}$$

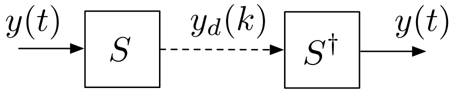
In the time domain, $y = h * v$, where $h(t)$ is the impulse response of this ideal filter:

$$h(t) = \frac{\sin \omega_N t}{\omega_N t}.$$

Therefore

$$\begin{aligned} y(t) &= h(t) * v(t) \\ &= h(t) * \sum y_d(k)\delta(t - kT) \\ &= \sum y_d(k) h(t - kT). \end{aligned}$$

This latter operation defines a linear mapping $S^\dagger$ (reconstructor) such that $y = S^\dagger y_d$. So the block diagram is



We see that $S^\dagger S y = y$, that is, $S^\dagger$ is a left-inverse of S, but only on the space of bandlimited signals. Furthermore, notice that the formula

$$y(t) = \sum_{k=-\infty}^{\infty} y_d(k) \frac{\sin \omega_N (t - kT)}{\omega_N (t - kT)}$$

shows that the reconstructor is noncausal: $y(t)$ is reconstructed from past and future sample values.

By contrast, our zero-order hold, H, is causal, because control systems are real-time systems. The equation $HSy = y$ does not hold for bandlimited y, but rather for signals that are already constant during sample periods.

Chapter 4

Control Design in Discrete Time

The method here is to discretize the plant and design the controller in the discrete-time domain. We'll develop state-space methods for the design of the digital controller.

The main topics are: stability of LTI systems, the stabilization problem, observers, and tracking and regulation.

4.1 Stability of LTI Systems

Notation

Just as we use a dot to denote derivative in continuous time, it's convenient to use a prime to denote time advance in discrete time. Thus $x'(k) = x(k+1)$. Then we can drop k and write, for example,

$$x' = Ax + Bu$$

in place of

$$x(k+1) = Ax(k) + Bu(k).$$

State-Space Models

Consider the homogeneous linear system in state space form:

$$x' = Ax, \quad x(0) = x_0.$$

The system is **asymptotically stable** if $x(k) \longrightarrow 0$ as $k \longrightarrow \infty$ for all $x(0)$. The system is **stable** if $x(k)$ remains bounded as $k \longrightarrow \infty$ for all $x(0)$. Since $x(k) = A^k x(0)$, the system is asymptotically stable iff every element of the matrix A^k converges to zero, and is stable iff every element of the matrix A^k remains bounded as $k \longrightarrow \infty$. Of course, asymptotic stability implies stability.

Asymptotic stability is relatively easy to characterize. Using the Jordan form, one can prove this very important result:

Lemma 8 *The system is asymptotically stable iff the eigenvalues of A all satisfy $|\lambda| < 1$.*

Let's say the matrix A is **stable** if its eigenvalues satisfy $|\lambda| < 1$. Then the system $x' = Ax$ is asymptotically stable iff A is stable.

Now we turn to stability. We'll do some examples, and we may as well have A in Jordan form.

Consider the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Obviously, $x(k) = x(0)$ for all k and so the system is stable. By contrast, consider

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = I + N.$$

Then

$$A^k = I + kN,$$

which is unbounded and so the system is not stable. This example extends to the $n \times n$ case: If $A = I + N$, N nilpotent, the system is stable iff $N = 0$.

Here's the test for stability in general in terms of the Jordan form of A:

$$A_{JF} = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_p \end{bmatrix},$$

Recall that each A_i has just one eigenvalue, λ_i, and that $A_i = \lambda_i I + N_i$, where N_i is a nilpotent matrix in Jordan form.

Lemma 9 *The system is stable iff the eigenvalues of A all satisfy $|\lambda| \leq 1$ and for any eigenvalue with $|\lambda_i| = 1$, the nilpotent matrix N_i is zero, i.e., A_i is diagonal.*

Here's an example with complex eigenvalues:

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad A_{JF} = \begin{bmatrix} j & 0 \\ 0 & -j \end{bmatrix}.$$

The system is stable. Now consider

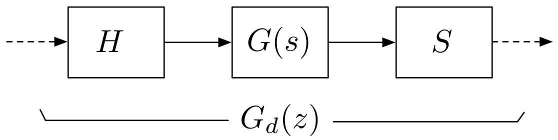
$$A = \begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The eigenvalues are $j, j, -j, -j$, all on the unit circle, and the Jordan form is

$$A_{JF} = \begin{bmatrix} j & 1 & 0 & 0 \\ 0 & j & 0 & 0 \\ 0 & 0 & -j & 1 \\ 0 & 0 & 0 & -j \end{bmatrix}.$$

Since the Jordan blocks are not diagonal, the system is not stable.

Example



Suppose $G(s)$ has the form

$$G(s) = \frac{\text{num}}{s^3 + 4s^2 + s + 1}.$$

Get a minimal realization

$$G(s) \to (A, B, C, D).$$

Then all the eigenvalues of A are in $\Re s < 0$ (the eigenvalues equal the poles of $G(s)$). Discretize via c2d:

$$(A, B, C, D) \to (A_d, B_d, C, D) \to G_d(z).$$

Then A_d is stable by the spectral mapping theorem. $\square$

Transfer-Function Models

Consider a linear system with input $u(k)$ and output $y(k)$. The natural concept of stability in an input-output setting is that of bounded-input, bounded-output stability. First, let's define precisely what boundedness of a signal means. Let $u(k)$ be a real-valued signal defined for $k \geq 0$. We say u is **bounded** if there exists a constant b such that, for all $t \geq 0$, $|u(k)| \leq b$. Familiar bounded signals are steps and sinusoids, but not ramps or blowing up signals like 2^k. Then the least upper bound b is denoted $\|u\|_\infty$, the infinity-norm of u. Think of $\|u\|_\infty$ as $\max_k |u(k)|$—maximum 0 to peak.

Consider a linear time-invariant system with a single (i.e., one-dimensional) input and a single output. The system is **BIBO stable** if $y(k)$ is bounded for every bounded $u(k)$, that is, $\|u\|_\infty$ finite implies $\|y\|_\infty$ finite.

Now assume the system has a transfer function $G(z)$, rational and proper.

Lemma 10 *The system is BIBO stable iff all the poles of the transfer function $G(z)$ lie in $|z| < 1$.*

Thus $\frac{1}{z}$, $\frac{1}{2z+1}$, $\frac{1}{z^2+0.1}$ are BIBO stable, but not $\frac{1}{z+1}$ or $\frac{1}{z-3}$.

Finally, suppose the transfer function is from a state model:

$$G(z) = C(zI - A)^{-1}B + D.$$

The poles of $G(z)$ are contained in $\sigma(A)$. Some cancellations may occur, so the containment may be proper—some eigenvalue of A may not be a pole. The conclusion is that stability of A implies BIBO stability of $G(z)$. Usually there are no cancellations, and the two stability concepts are equivalent.

Example

$$
\begin{aligned}
& y(k) = u(k) + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3) \\
& |y(k)| \le |u(k)| + |b_1||u(k-1)| + |b_2||u(k-2)| + |b_3||u(k-3)| \\
& |y(k)| \le \|u\|_\infty + |b_1|\|u\|_\infty + |b_2|\|u\|_\infty + |b_3|\|u\|_\infty \\
& \|y\|_\infty \le \|u\|_\infty + |b_1|\|u\|_\infty + |b_2|\|u\|_\infty + |b_3|\|u\|_\infty \\
& \|y\|_\infty \le (1 + |b_1| + |b_2| + |b_3|)\|u\|_\infty
\end{aligned}
$$

Every FIR filter is BIBO stable.

□

Example

$$y(k) = y(k-1) + u(k) + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3)$$

$$
\begin{aligned}
G(z) &= \frac{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 - z^{-1}} \\
&= \frac{z^3 + b_1 z^2 + b_2 z + b_3}{z^2(z-1)}
\end{aligned}
$$

Not BIBO stable, unless

$$z^3 + b_1 z^2 + b_2 z + b_3 = (z-1)(z^2 + b_4 z + b_5).$$

□

Example

$$y(k) = 0.5y(k-1) + u(k) + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3)$$

$$
\begin{aligned}
G(z) &= \frac{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 - 0.5z^{-1}} \\
&= \frac{z^3 + b_1 z^2 + b_2 z + b_3}{z^2(z-0.5)}
\end{aligned}
$$

BIBO stable.

□

Example

$$y(k) = 2y(k-1) + u(k) + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3)$$

$$
\begin{aligned}
G(z) &= \frac{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 - 2z^{-1}} \\
&= \frac{z^3 + b_1 z^2 + b_2 z + b_3}{z^2(z-2)}
\end{aligned}
$$

Not BIBO stable, unless

$$z^3 + b_1 z^2 + b_2 z + b_3 = (z - 2)(z^2 + b_4 z + b_5).$$

□

Note that if the discrete-time system is obtained via `c2d` applied to a continuous-time system with system matrix A, the discrete-time system matrix $A_d = \mathrm{e}^{AT}$ is stable iff the eigenvalues of A all lie in $\Re s < 0$, i.e., iff the continuous-time system matrix A is stable.

4.2 The Stabilization Problem

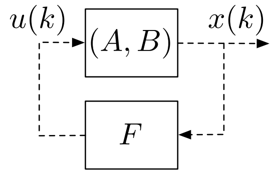
In this section, we begin our study of digital control design. We start with

$$x' = Ax + Bu.$$

We regard 0 as the equilibrium point for the state x. Assume that the state $x(k)$ is available from a sensor for all k. We search for a **state-feedback controller** $u(k) = Fx(k)$, F a real matrix. Then the controlled system is

$$x' = Ax + Bu = Ax + BFx = (A + BF)x.$$

The block diagram is



The **stabilization problem** is this: Given (A, B), find F so that $A + BF$ is stable. To solve this, we introduce the notion of controllability.

Controllability

The relevant system is

$$x' = Ax + Bu.$$

The results here are almost the same as in continuous time; the only difference is that we can't expect to hit a target vector in an arbitrarily small time because time is discrete. The results are summarized as follows:

The pair (A, B) is defined to be **controllable** if every target vector is reachable at some time by some input sequence, starting from $x(0) = 0$. Let's see where the state can get to at various times starting from the origin. First,

$$x(1) = Bu(0).$$

Thus at time $k = 1$ the state can reach any vector in the column span of B, and no other vector. Then

$$x(2) = Ax(1) + Bu(1) = ABu(0) + Bu(1).$$

Thus at time $k = 2$ the state can reach any vector in the column span of $\begin{bmatrix} B & AB \end{bmatrix}$, and no other vector. In n-steps, the state can reach any vector in the column span of the matrix

$$W = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}.$$

By the Cayley-Hamilton Theorem, A^n is expressible as a linear combination of $A^j,\ j = 0, 1, \cdots, n-1$. Hence we cannot reach states which do not lie in the column span of W. So, we get that (A, B) is controllable iff

$$\operatorname{rank} W = \operatorname{rank} \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} = n.$$

We call the matrix W the controllability matrix of the pair (A, B).

Next, the PBH test: (A, B) is controllable iff

$$\operatorname{rank} \begin{bmatrix} A - \lambda I & B \end{bmatrix} = n \quad \text{for each eigenvalue } \lambda \text{ of } A.$$

The definition of stabilizability is this: (A, B) is **stabilizable** if there exists a matrix F such that the eigenvalues of $A+BF$ are all inside the open unit disk. It's a fact that (A, B) is stabilizable iff

$$\operatorname{rank} \begin{bmatrix} A - \lambda I & B \end{bmatrix} = n \quad \text{for each eigenvalue } \lambda \text{ of } A \text{ with } |\lambda| \geq 1.$$

Example

$$A = \begin{bmatrix} 2 & 0 & 2 \\ 3 & 1 & 0 \\ 1 & 4 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & 2 & 6 \\ 0 & 0 & 6 \\ 1 & 1 & 3 \end{bmatrix}$$

rank $W = 3$. Thus (A, B) is controllable.

□

The stabilization problem is solved by appealing to a famous result in linear systems theory, referred to as the Pole Assignment Theorem. To state the theorem, we first make

Definition:
A set of complex numbers is called **symmetric** if for any complex number belonging to the set, its complex conjugate also belongs to the set.

Note that eigenvalues of real matrices form a symmetric set of complex numbers. Similarly, the roots of a polynomial with real coefficients form a symmetric set as well.

We can now state

The Pole Assignment Theorem:
There exists a matrix F such that the eigenvalues of the matrix $A + BF$ coincide with any given set of symmetric numbers if and only if (A, B) is controllable.

In this course, we shall apply the Pole Assignment Theorem exclusively in the case where the input u is scalar-valued (referred to as the single-input case).

In this case, the construction of the required pole placement matrix F, which is actually a row vector, can be carried out in 2 steps.

Step 1:

Given a set of symmetric numbers $p_j,\ j = 1, \cdots, n$ representing the closed loop poles, i.e. the eigenvalues of $A + BF$, form the corresponding desired characteristic polynomial

$$\begin{aligned} r(z) &= (z - p_1)(z - p_2) \cdots (z - p_n) && (4.1) \\ &= z^n + r_1 z^{n-1} + \cdots + r_n && (4.2) \end{aligned}$$

Assume that the single-input system has its system matrices (A, B) in the following special form

$$x' = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & 1 \\ -a_n & -a_{n-1} & \cdots & \cdots & -a_2 & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u = \bar{A}x + \bar{B}u \tag{4.3}$$

Then the desired pole placement feedback gain $\bar{F}$ which places poles of the closed loop system ()eignevalues of $\bar{A} + \bar{B}\bar{F}$) at the roots of $r(z)$ is given by

$$\bar{F} = [a_n - r_n \ \cdots \ a_1 - r_1]$$

The proof of this result is a direct consequence of the fact that the closed loop system is given by

$$x' = (\bar{A} + \bar{B}\bar{F})x = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \ddots & 1 \\ -r_n & -r_{n-1} & \cdots & -r_2 & -r_1 \end{bmatrix} x \tag{4.4}$$

It is easily verified that any (A, B) pair having this special structure is always controllable, regardless of the values of the a_j's. Such (A, B) pairs are said to be in controllable canonical form.

Step 2:

In general, (A, B) will not be in controllable canonical form. Let (A, B) be controllable with the characteristic polynomial of A given by

$$\alpha(z) = \det(zI - A) = z^n + a_1 z^{n-1} + \cdots + a_n$$

For any nonsingular matrix, if we set $\xi = P^{-1}x$, ξ satisfies the difference equation

$$\xi' = P^{-1}AP\xi + P^{-1}Bu$$

Note that such a transformation (corresponding to a change of basis on the state space) does not affect controllability. In fact the controllability matrix W_ξ for the ξ system and the controllability matrix W for the x system are related as follows:

$$W_\xi = P^{-1}[B \ \cdots \ A^{n-1}B] = P^{-1}W$$

In other words, the P matrix which transforms the x system to the ξ system is given by

$$P = WW_\xi^{-1}$$

Let us pick P so that $(P^{-1}AP, P^{-1}B) = (\bar{A}, \bar{B})$ is in controllable canonical form (4.3). Define the controllability matrix $\bar{W} = [\bar{B} \ \cdots \ \bar{A}^{n-1}\bar{B}]$. From the above discussion, the required P to take (A, B) to controllable canonical form is given by

$$P = W\bar{W}^{-1}$$

It can be shown that

$$\bar{W}^{-1} = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & \cdots & a_1 & 1 & 0 \\ \vdots & \iddots & 1 & 0 & 0 \\ a_1 & \iddots & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Thus the matrix

$$P \quad = \quad [B \ \cdots \ A^{n-1}B] \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & \cdots & a_1 & 1 & 0 \\ \vdots & \iddots & 1 & 0 & 0 \\ a_1 & \iddots & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{4.5}$$

will transform (A, B) into controllable canonical form, i.e., $(P^{-1}AP, P^{-1}B)$ is in controllable canonical form.

The required pole placement matrix F is then given by

$$F = [a_n - r_n \ \cdots \ a_1 - r_1]P^{-1} = \bar{F}P^{-1} \tag{4.6}$$

To see this, simply note that $r(z) = det(zI - \bar{A} - \bar{B}\bar{F}) = det(zI - P^{-1}AP - P^{-1}B\bar{F}) = det[P^{-1}(zI - A - B\bar{F}P^{-1})P] = det(zI - A - B\bar{F}P^{-1})$.

Ackermann's Formula

There's an alternative formula to assign eigenvalues. Let $\{p_1, \ldots, p_n\}$ be the desired eigenvalues. Let $r(z)$ be the desired characteristic polynomial:

$$r(z) = (z - p_1)\cdots(z - p_n) = z^n + r_1 z^{n-1} + \cdots + r_n.$$

Then

$$F = -\begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} W^{-1} r(A).$$

Here $r(A)$ refers to the matrix

$$r(A) = A^n + r_1 A^{n-1} + \cdots + r_n I.$$

Example: The double integrator

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

Realization:

$$\dot{x} = Ax + Bu$$

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

Discretization:

$$A_d = e^{AT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

$$\det[zI - A] = (z-1)^2 = z^2 - 2z + 1$$

$$B_d = \int_0^T e^{A\tau}\, d\tau\, B = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}$$

Desired characteristic polynomial:

$$r(z) = z^2 + r_1 z + r_2$$

Controllability matrix:

$$W = \begin{bmatrix} \frac{T^2}{2} & \frac{3T^2}{2} \\ T & T \end{bmatrix}$$

Ackermann's formula:

$$\begin{aligned} F &= -\begin{bmatrix} 0 & 1 \end{bmatrix} W^{-1} r(A_d) \\ &= -\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{T^2}{2} & \frac{3T^2}{2} \\ T & T \end{bmatrix}^{-1} (A_d^2 + r_1 A_d + r_2 I) \end{aligned}$$

where

$$\begin{aligned} A_d^2 + r_1 A_d + r_2 I &= (A_d + r_1 I) A_d + r_2 I \\ &= \begin{bmatrix} 1 + r_1 & T \\ 0 & 1 + r_1 \end{bmatrix} \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} r_2 & 0 \\ 0 & r_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 + r_1 + r_2 & r_1 T + 2T \\ 0 & 1 + r_1 + r_2 \end{bmatrix}. \end{aligned}$$

As a specific example, suppose $T = 0.1$ and desired closed loop poles are 0.2, 0.5. Then the desired characteristic polynomial is

$$r(z) = (z - 0.2)(z - 0.5) = z^2 - 0.7z + 0.1,$$

$$F = -\left[\begin{array}{cc} 100(0.4) & 5(2.2) \end{array}\right] = -\left[\begin{array}{cc} 40 & 11 \end{array}\right].$$

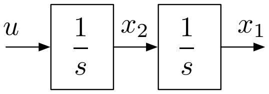
Check:

$$A_d + B_d F = \left[\begin{array}{cc} 0.8 & 0.045 \\ -4 & -0.1 \end{array}\right], \quad \text{eigs ok}$$

Now let's look at the block diagram to see how the controller is actually implemented. We have the continuous-time plant

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u. \end{aligned}$$

Thus



Recall the discretization:

$$\dot{x} = Ax + Bu, \quad u = Hu_d$$

$$\begin{aligned} x((k+1)T) &= \mathrm{e}^{AT}x(kT) + \int_{kT}^{'T} \mathrm{e}^{A((k+1)T-\tau)} d\tau B u_d(k) \\ &= A_d x(kT) + B_d u_d(k) \end{aligned}$$

$$x_d(k) := x(kT), \quad x_d = Sx$$

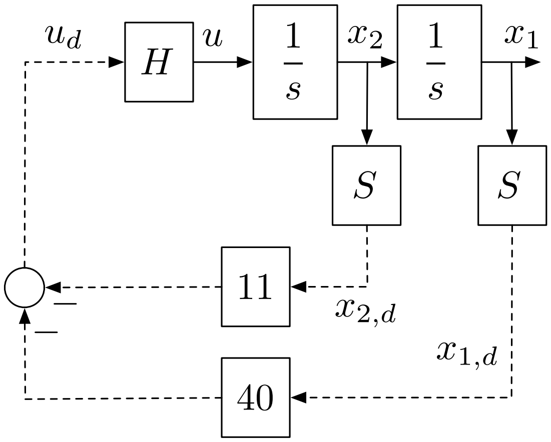
$$x_d' = A_d x_d + B_d u_d$$

Returning to the example, we have

$$u_d = F x_d, \quad F = -\left[\begin{array}{cc} 40 & 11 \end{array}\right]$$

$$u_d = -40 x_{1,d} - 11 x_{2,d}$$

Thus the block diagram with the digital controller is



□

Proof of Ackermann's formula

We will show that the feedback gain F given by Ackermann's formula is exactly the same as $\bar{F}P^{-1}$, the formula proved using transformation to controllable canonical form. First we find an alternative formula for $\bar{F}$.

Let $\alpha(z)$ denote the characteristic polynomial for $\bar{A}$. By Cayley-Hamilton theorem,

$$\alpha(\bar{A}) = \bar{A}^n + a_1\bar{A}^{n-1} + \cdots + a_{n-1}\bar{A} + a_n I = 0$$

If the desired closed loop characteristic polynomial is $r(z)$, then

$$\begin{aligned} r(\bar{A}) &= \bar{A}^n + r_1\bar{A}^{n-1} + \cdots + r_{n-1}\bar{A} + r_n I \\ &= r(\bar{A}) - \alpha(\bar{A}) = (r_1 - a_1)\bar{A}^{n-1} + \cdots + (r_{n-1} - a_{n-1})\bar{A} + (r_n - a_n)I \end{aligned}$$

Let e_j be the j^{th} Euclidean basis vector. Using the form of $\bar{A}$, it is easy to verify that $e_1^T\bar{A} = e_2^T$, $e_2^T\bar{A} = e_3^T$, etc., so that $e_1^T\bar{A}^k = e_{k+1}^T$, $0 \le k \le n-1$. Hence

$$\begin{aligned} -e_1^T r(\bar{A}) &= -e_1^T[(r_1 - a_1)\bar{A}^{n-1} + \cdots + (r_{n-1} - a_{n-1})\bar{A} + (r_n - a_n)I] \\ &= (a_1 - r_1)e_n^T + (a_2 - r_2)e_{n-1}^T + \cdots + (a_n - r_n)e_1^T \\ &= \bar{F} \end{aligned}$$

For a given controllable (A,B) pair, we have previously constructed the matrix P which transforms (A,B) to $(\bar{A},\bar{B})$, i.e. $\bar{A} = P^{-1}AP$, $\bar{B} = P^{-1}B$. Observe that

$$\bar{W} = \begin{bmatrix} \bar{B} & \bar{A}\bar{B} & \cdots & \bar{A}^{n-1}\bar{B} \end{bmatrix} = P^{-1}\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} = P^{-1}W$$

which shows that the transformation matrix P can also be written as

$$P = W\bar{W}^{-1}$$

Note also that $e_1^T\bar{W} = e_n^T$, by the special form of $\bar{W}$. Finally, we can assemble these results to write

$$\begin{aligned} F &= \bar{F}P^{-1} \\ &= -e_1^T r(\bar{A})P^{-1} \\ &= -e_1^T P^{-1} r(A) P P^{-1} = -e_1^T P^{-1} r(A) \\ &= -e_1^T \bar{W} W^{-1} r(A) \\ &= -e_n^T W^{-1} r(A) \end{aligned}$$

which is Ackermann's formula.

□

The above analytical procedures for computing the pole placement feedback matrix are not numerically stable for large n.

Deadbeat Control

Let's begin with

$$x' = Ax.$$

Then

$$x(k) \longrightarrow 0 \text{ in finite time } \forall x(0)$$

$$\Longleftrightarrow A^k \longrightarrow 0 \text{ in finite time}$$

$$\Longleftrightarrow A^n = 0$$

$$\Longleftrightarrow \text{all eigs of } A \text{ are } 0$$

$$\Longleftrightarrow A \text{ is nilpotent.}$$

Returning to state feedback, we see that the particular choice of $r(z) = z^n$ for the closed-loop characteristic polynomial leads to

$$x' = (A + BF)x$$

where $A + BF$ is nilpotent. Then $x(k) \longrightarrow 0$ in finite time ($k \leq n$). It is not possible to get this kind of behaviour from the continuous-time system $\dot{x} = (A + BF)x$, since the response will always be decaying exponentials and cannot go to zero in finite time.

On Intersample Ripple

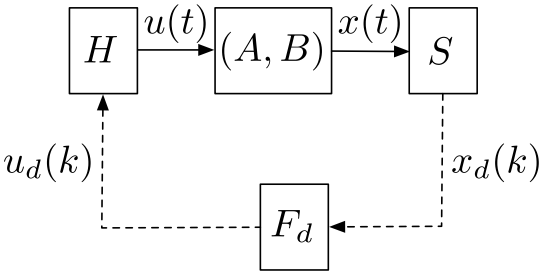
Consider the continuous-time state model

$$\dot{x} = Ax + Bu.$$

Discretize via `c2d`:

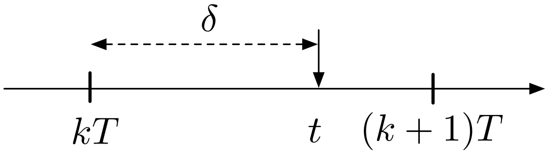
$$x_d' = A_d x_d + B_d u_d.$$

Thus the setup is



Suppose (A_d, B_d) is stabilizable and F_d is a feedback matrix such that $A_d + B_d F_d$ is stable—all eigenvalues satisfy $|\lambda| < 1$. Then $x(kT)$ converges to 0 as $k \longrightarrow \infty$ for every $x(0)$. But does $x(t)$ converge to zero too as $t \longrightarrow \infty$? Yes:

Proof Let's look at a general sample period $[kT, (k+1)T)$:



Let t be any time in that period and let $\delta = t - kT$, as shown. From basic theory, $x(t)$ depends on the state at the start of the sampling period together with the input over $[kT, t)$:

$$x(t) = \mathrm{e}^{A(t-kT)}x(kT) + \int_{kT}^{t} \mathrm{e}^{A(t-\tau)}Bu(\tau)d\tau.$$

Since u is constant during the sampling period,

$$x(t) = \mathrm{e}^{A(t-kT)}x(kT) + \int_{kT}^{t} \mathrm{e}^{A(t-\tau)}d\tau Bu(kT).$$

Change integration variable to $\sigma = t - \tau$:

$$x(t) = \mathrm{e}^{A(t-kT)}x(kT) + \int_{0}^{t-kT} \mathrm{e}^{A\sigma}d\sigma Bu(kT).$$

Replace $t - kT$ by δ:

$$x(t) = \mathrm{e}^{A\delta}x(kT) + \int_{0}^{\delta} \mathrm{e}^{A\sigma}d\sigma Bu(kT).$$

Substitute in $u(kT) = F_d x(kT)$:

$$x(t) = \left[\mathrm{e}^{A\delta} + \int_{0}^{\delta} \mathrm{e}^{A\sigma}d\sigma BF_d\right] x(kT).$$

Now $\mathrm{e}^{A\delta} + \int_0^\delta \mathrm{e}^{A\sigma}d\sigma BF_d$ is just some matrix, say M:

$$x(t) = Mx(kT).$$

Finally, replace t by $kT + \delta$:

$$x(kT + \delta) = Mx(kT).$$

Since $\lim_{k\to\infty} x(kT) = 0$, therefore $\lim_{k\to\infty} x(kT + \delta) = 0$. Since δ was arbitrary, we see that the continuous-time state converges to zero at all intersample times. □

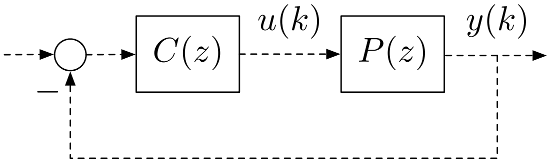
Here's a followup point: If $A_d + B_d F_d$ is nilpotent, then $x(kT)$ converges to 0 in finite time as $k \longrightarrow \infty$. Does $x(t)$ converge to 0 in finite time too? Yes—same proof as above. Does this contradict the last sentence before this subsection? No.

Output Feedback

We have solved the stabilization problem under the assumption that the state x is available directly, from a sensor, by using pole placement state feedback. We now relax the assumption that the full state x is available. Instead, we assume that the system is described by

$$\begin{aligned} x' &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

and only $y(k)$ is available as a controller input. Let $P(z)$ denote the transfer function from $u(k)$ to $y(k)$. The block diagram is now



What is a sensible notion of stability of this system? The plant has the model

$$\begin{aligned} x' &= Ax + Bu \\ y &= Cx + Du. \end{aligned}$$

Suppose the controller likewise has a state model:

$$\begin{aligned} x_c' &= A_c x_c - B_c y \\ u &= C_c x_c. \end{aligned}$$

(We assume the controller is strictly causal, for simplicity.) The closed-loop system then has the equations

$$\begin{aligned} x' &= Ax + BC_c x_c \\ x_c' &= A_c x_c - B_c[Cx + DC_c x_c]. \end{aligned}$$

Defining the closed-loop state $x_{cl} = \left[\begin{array}{c} x \\ x_c \end{array} \right]$, we have

$$x_{cl}' = A_{cl} x_{cl},$$

where

$$A_{cl} = \left[\begin{array}{cc} A & BC_c \\ -B_c C & A_c - B_c D C_c \end{array} \right].$$

Now we can define the **feedback system to be asymptotically stable** if A_{cl} is stable.

Example This example illustrates this definition of stability. Let the plant be

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

—unstable. Try PI control:

$$C(z) = K_1 + \frac{K_2}{z-1}.$$

This is not strictly causal, so the formula above for A_{cl} doesn't apply. Let's try to find K_1, K_2 for closed-loop stability. State models:

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

$$\begin{aligned} x' &= -2x + u \\ y &= x \end{aligned}$$

$$C(z) = K_1 + \frac{K_2}{z-1}$$

$$\begin{aligned} x_c' &= x_c - y \\ u &= K_2 x_c - K_1 y = -\frac{K_2}{z-1} y - K_1 y \end{aligned}$$

Combine:

$$\begin{aligned} x' &= -2x + u \\ y &= x \\ x_c' &= x_c - y \\ u &= K_2 x_c - K_1 y \end{aligned}$$

$$\begin{aligned} x' &= -2x + K_2 x_c - K_1 x \\ x_c' &= x_c - x \end{aligned}$$

Thus

$$A_{cl} = \begin{bmatrix} -2-K_1 & K_2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} -K_1 & K_2 \end{bmatrix}$$

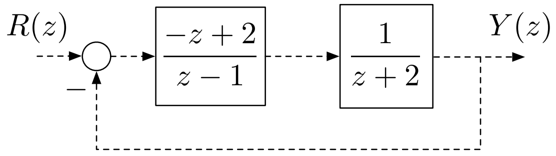
$$\begin{bmatrix} -2 & 0 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ controllable}$$

Therefore we can stabilize via $K_1 = -1, K_2 = 1$ and then

$$A_{cl} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Thus the controller is

$$C(z) = -1 + \frac{1}{z-1} = \frac{-z+2}{z-1}.$$



Notice that every closed-loop transfer function is stable. E.g.,

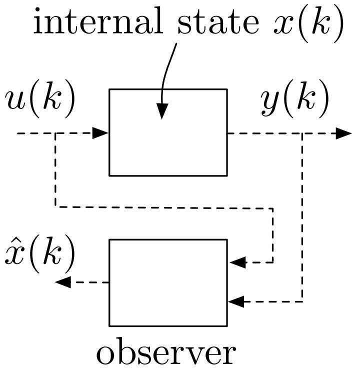
$$\frac{Y(s)}{R(s)} = \frac{\frac{-z+2}{z-1}\frac{1}{z+2}}{1+\frac{-z+2}{z-1}\frac{1}{z+2}} = \frac{-z+2}{z^2}.$$

□

The **output-feedback stabilization problem** is: Find a controller $C(z)$ such that A_{cl} is stable. To solve this problem, we introduce the concept of an observer.

4.3 Observers

An observer is a dynamic system that asymptotically reconstructs the state of a system using only input-output measurements:



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

$$\lim_{k\to\infty} e(k) = 0, \quad \forall u(\cdot), \text{ initial conditions}$$

Full-order Observer

Suppose we had $\hat{x}$. Then output estimate would be

$$\hat{y}(k) = C\hat{x}(k) + Du(k).$$

A promising observer might then be

$$\underbrace{\hat{x}' = A\hat{x} + Bu}_{\text{plant emulator}} + L[\ \underbrace{\hat{y} - y}_{\text{correction}}\ ].$$

When will this work? Let's see:

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

$$\begin{aligned} e' &= x' - \hat{x}' \\ &= \{Ax + Bu\} - \{A\hat{x} + Bu + L[\hat{y} - y]\} \\ &= Ax - A\hat{x} - L[\hat{y} - y] \\ &= Ax - A\hat{x} - L[C\hat{x} - Cx] \\ &= (A + LC)e \end{aligned}$$

Thus the observer works iff $A + LC$ is stable!

Observe that the eigenvalues of $A + LC$ are the same as those of $A^T + C^T L^T$. Hence we can arbitrarily place the eigenvalues of $A + LC$ iff (A^T, C^T) is controllable. And we can stabilize $A + LC$ iff (A^T, C^T) is stabilizable.

Observability

The relevant system is

$$x' = Ax, \quad y = Cx.$$

Again, the results parallel the continuous-time case, the major difference being that the observation interval cannot be arbitrarily small. The results are as follows:

1. The pair (C, A) is **observable** if the initial state $x(0)$ can be computed from the output sequence

   $$\{y(0), y(1), \ldots, y(k_1)\}$$

   for some sufficiently large k_1.

   Note that

   $$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} x(0)$$

   We can therefore conclude

2. (C, A) is observable iff

   $$\text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n.$$

The matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

is called the observability matrix of the pair (C, A).

3. PBH test for observability:

   (C, A) is observable iff

   $$\text{rank} \begin{bmatrix} A - \lambda I \\ C \end{bmatrix} = n \quad \text{for each eigenvalue } \lambda \text{ of } A.$$

4. (C, A) is defined to be **detectable** if there exists a matrix L such that all the eigenvalues of $A + LC$ are inside the open unit disk.

5. (C, A) is detectable iff

   $$\text{rank} \begin{bmatrix} A - \lambda I \\ C \end{bmatrix} = n \quad \text{for each eigenvalue } \lambda \text{ of } A \text{ with } |\lambda| \geq 1.$$

Example double integrator (cont'd)

Suppose there's a position sensor:

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

Then

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & T \end{bmatrix}$$

rank $\mathcal{O} = 2$ for all $T > 0$. Analogous to pole placement, we can place observer poles at 0 for deadbeat behaviour:

$$A^T = \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix}, \quad C^T = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad L^T = -\begin{bmatrix} 2 & \frac{1}{T} \end{bmatrix}$$

$$A^T + C^T L^T = \begin{bmatrix} -1 & -\frac{1}{T} \\ T & 1 \end{bmatrix}.$$

The deadbeat observer equation is

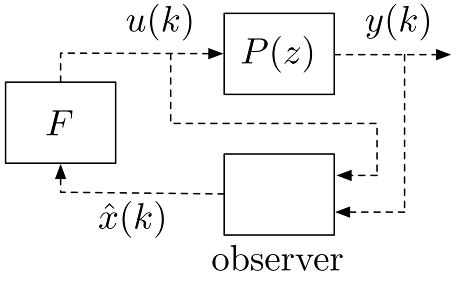
$$\hat{x}' \quad = \quad A\hat{x} + Bu + L[C\hat{x} - y].$$

□

We can now combine pole placement and observer design to solve the output-feedback stabilization problem.

Observer-Based Controller

The idea is, first, to design F to stabilize $A+BF$; then, to design an observer; and finally to connect them like this:



Will this work? Let's look at the equations. Plant:

$$\begin{aligned} x' &= Ax+Bu \\ y &= Cx+Du \end{aligned}$$

Observer:

$$\hat{x}' = A\hat{x}+Bu+L[C\hat{x}+Du-y]$$

Control:

$$u = F\hat{x}$$

Given $A+BF$ and $A+LC$ are stable, we want to prove that A_{cl} is too. Define the estimation error $e(k)=x(k)-\hat{x}(k)$. As we derived before

$$e' = (A+LC)e.$$

Also,

$$\begin{aligned} x' &= Ax+Bu \\ &= Ax+BF\hat{x} \\ &= Ax+BF[x-e] \\ &= (A+BF)x-BFe. \end{aligned}$$

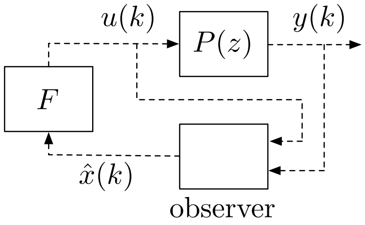
Thus

$$\begin{bmatrix} x' \\ e' \end{bmatrix} = \begin{bmatrix} A+BF & -BF \\ 0 & A+LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}.$$

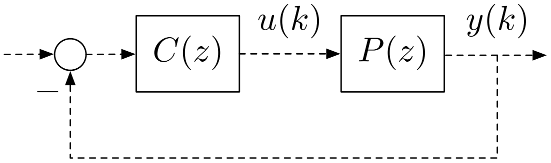
Owing to the block triangular structure of this system matrix, its eigenvalues are those of $A+BF$ and $A+LC$ together.

Thus $x(k)$ and $e(k)$ converge to 0 as $k \to \infty$ for every initial condition. Thus $x(k)$ and $\hat{x}(k)$ converge to 0 as $k \to \infty$ for every initial condition. Thus A_{cl} is stable. Thus we've solved the output-feedback stabilization problem.

Can we reconfigure



into the standard form

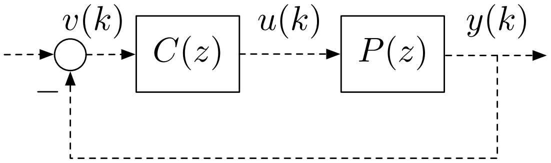


Let's find the controller, from $y(k)$ to $u(k)$:

$$\begin{aligned}\hat{x}' &= A\hat{x} + Bu + L[C\hat{x} + Du - y]\\ u &= F\hat{x}\end{aligned}$$

$$\begin{aligned}\hat{x}' &= A\hat{x} + BF\hat{x} + L[C\hat{x} + DF\hat{x} - y]\\ &= (A + BF + LC + LDF)\hat{x} - Ly\end{aligned}$$

Thus the controller in the block diagram



has the state model

$$\begin{aligned}\hat{x}' &= (A + BF + LC + LDF)\hat{x} + Lv\\ u &= F\hat{x}\end{aligned}$$

and hence the controller state-space matrices are

$$A_c = A + BF + LC + LDF, \quad B_c = L, \quad C_c = F.$$

Example double integrator (con'd)
We had

$$A_d = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Setting $T = 0.1$ (as before) gives

$$A_d = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The plant transfer function is

$$P(z) = C(zI - A_d)^{-1}B_d = 0.01\frac{2z-1}{(z-1)^2}.$$

For the desired eigenvalues 0.2, 0.5 we got

$$F = -\begin{bmatrix} 40 & 11 \end{bmatrix}.$$

And for a deadbeat observer we got

$$L = -\begin{bmatrix} 2 \\ \frac{1}{T} \end{bmatrix} = -\begin{bmatrix} 2 \\ 10 \end{bmatrix}.$$

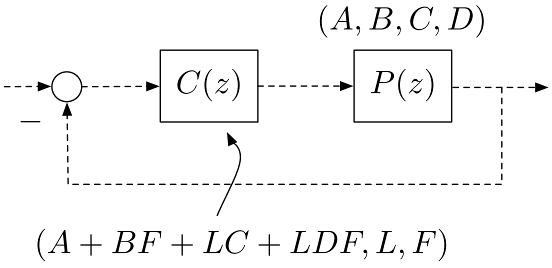
So the controller is given by

$$A_c = A_d + B_dF + LC = \begin{bmatrix} -1.2 & 0.045 \\ -14 & -0.1 \end{bmatrix}, \; B_c = L, \; C_c = F.$$

Then $C(z) = C_c(zI - A_c)^{-1}B_c$.

□

Recap of Observer-Based Controller



$A + BF, A + LC$ stable

Let's note a few points: The order of controller equals the order of plant; the controller itself, $C(z)$, may not stable.

4.4 Reduced-Order Observers

The observer we have described is called the full-order observer since the dimension of the observer is the same as the dimension of the plant. The full-order observer produces an estimate of every component of the state. However, since the output y provides direct measurement of some combination of the state variables, there should be no need to estimate certain components of the state. For example, it may be that $y = x_1$, in which case estimating x_1 would be redundant, indeed less accurate than using y directly! This suggests that we can use an observer which has a smaller dimension than the plant. We describe the construction, called the reduced-order observer.

Suppose the plant equation can be partitioned into the following form

$$\begin{aligned} \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= [I \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \end{aligned}$$

Thus $y = x_1$. Writing out

$$x_1' - A_{11}x_1 - B_1 u = A_{12}x_2$$

we see that the left-hand side is known from input-output measurements:

$$y' - A_{11}y - B_1 u = A_{12}x_2$$

We may therefore interpret the left-hand side as an observation of $A_{12}x_2$.

We now imitate the construction of the full-order observer:

$$\hat{x}_2' = A_{22}\hat{x}_2 + B_2 u + A_{21}y$$

$$+L_2 \left\{ A_{12}\hat{x}_2 - [y' - A_{11}y - B_1 u] \right\}.$$

Re-arrange:

$$\hat{x}_2' = (A_{22} + L_2 A_{12})\hat{x}_2 + (B_2 + L_2 B_1)\ u$$

$$+(A_{21} + L_2 A_{11})\ y - L_2\ y'.$$

This is the basic reduced-order observer equation. However, the right-hand side contains y', which makes the system noncausal. To fix this, rearrange the equation as

$$\hat{x}_2' + L_2\ y'$$

$$= (A_{22} + L_2 A_{12})\hat{x}_2 + (B_2 + L_2 B_1)\ u + (A_{21} + L_2 A_{11})\ y.$$

This suggests defining $v = \hat{x}_2 + L_2 y$. Then

$$v' = (A_{22} + L_2 A_{12})\hat{x}_2$$

$$+(B_2 + L_2 B_1)\ u + (A_{21} + L_2 A_{11})\ y.$$

Replace $\hat{x}_2$ by $v - L_2 y$ and collecting terms gives

$$v' = (A_{22} + L_2 A_{12})v + (B_2 + L_2 B_1)\ u$$

$$+(A_{21} + L_2 A_{11} - A_{22}L_2 - L_2 A_{12} L_2)\ y.$$

Recap The plant equation is (after transformation if necessary)

$$\begin{aligned}\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= [I \quad 0]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.\end{aligned}$$

Thus $y = x_1$. The reduced-order observer equation is

$$\begin{aligned} v' = (A_{22} + L_2A_{12})v + (B_2 + L_2B_1)\ u \\ +(A_{21} + L_2A_{11} - A_{22}L_2 - L_2A_{12}L_2)\ y \end{aligned}$$

and v is an estimate of $x_2 + L_2x_1$.

What about convergence? Define the estimation error

$$e_2 = [x_2 + L_2y] - v = x_2 - \hat{x}_2.$$

You can derive that

$$e_2' = (A_{22} + L_2A_{12})e_2.$$

Thus we want $A_{22} + L_2A_{12}$ to be stable. It can be shown that (A_{12}, A_{22}) is observable iff the original (C, A) is observable. This means that, if (C, A) is observable, we can find L_2 such that the eigenvalues of $A_{22} + L_2A_{12}$ can be arbitrarily assigned. This shows that we can solve the state estimation problem for x_2 alone. Coupled with the direct observation of x_1, we can therefore write the complete state estimate as

$$\hat{x} = \begin{bmatrix} y \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} y \\ v - L_2y \end{bmatrix}.$$

The double integrator example is left for you.

Finally, how to transform so that

$$y(k) = \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}.$$

Example Suppose

$$y(k) = Cx(k) = \begin{bmatrix} 2 & -2 & 1 \end{bmatrix} x(k).$$

Get $Q : 3 \times 3$, invertible, so that

$$\begin{bmatrix} 2 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} Q,$$

e.g.,

$$Q = \begin{bmatrix} 2 & -2 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

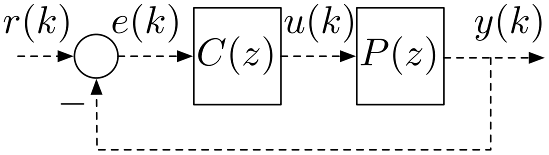
Set $\xi = Qx$. Then

$$y = Cx = CQ^{-1}\xi = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \xi.$$

□

4.5 Tracking Steps

Consider the block diagram



Suppose $r(k)$ is an arbitrary constant value and we require $y(k)$ to converge to that constant value. By linearity, it suffices to meet this requirement for just $r(k) = 1$. So the problem is: Given $r(k) = 1$, design $C(z)$ to achieve feedback stability and $\lim_{k\to\infty} y(k) = 1$, or equivalently, $\lim_{k\to\infty} e(k) = 0$.

The z-transform of $r(k) = 1$ is $R(z) = z/(z-1)$. Let $G(z)$ denote the closed-loop transfer function from r to the tracking error e. Then

$$E(z) = G(z)\frac{z}{z-1}.$$

By the final-value theorem, $\lim_{k\to\infty} e(k) = 0$ iff all the poles of $G(z)$ lie in $|z| < 1$ and $G(1) = 0$. Having all the poles of $G(z)$ lie in $|z| < 1$ will follow from the requirement of feedback stability. Since

$$G = \frac{1}{1+PC},$$

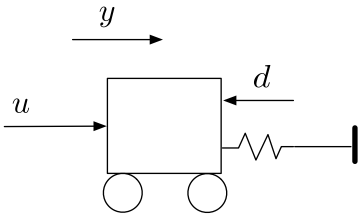
the condition $G(1) = 0$ is equivalent to the condition that $P(z)C(z)$ has a pole at $z = 1$. If P has a pole at $z = 1$, then C merely has to stabilize the feedback loop. If P does not have a pole at $z = 1$, then C must have one. If, furthermore, P has a zero at $z = 1$, then the problem isn't solvable; remember: there can be no unstable pole-zero cancellation. Finally, if P doesn't have a pole or zero at $z = 1$, we can solve the tracking problem by taking C of the form

$$C(z) = \frac{1}{z-1}C_1(z)$$

and designing C_1 to stabilize the feedback loop. For example we could design an observer-based controller $C_1(z)$ to stabilize $P(z)/(z-1)$. Thus integral control is the key to tracking constant references.

4.6 Tracking and Regulation

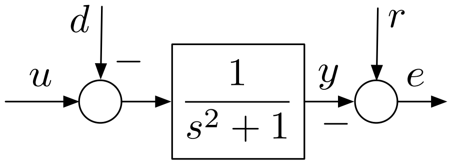
This section develops the state-space theory of tracking and regulation. As in Chapter 1, consider a cart/spring with control force u, disturbance force d, and position y:



We want the cart to follow a reference r in spite of the disturbance. Let us take the model

$$\ddot{y} = u - y - d, \quad e = r - y$$

and then the block diagram is



Suppose we know that r is a constant (or a step) but we don't know its value; and we know that d is a sinusoid of frequency 10 rad/s but of unknown amplitude and phase. Then we know equations that can generate these signals, namely,

$$\dot{r} = 0, \quad \ddot{d} + 100d = 0.$$

Taking the state variables

$$x_1 = (y, \dot{y}), \quad x_2 = (r, d, \dot{d}),$$

we have

$$A = \left[\begin{array}{rr|rrr} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -100 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} 0 \\ 1 \\ \hline 0 \\ 0 \\ 0 \end{array}\right], \quad D = \left[\begin{array}{rr|rrr} -1 & 0 & 1 & 0 & 0 \end{array}\right].$$

The equations are

$$\dot{x} = Ax + Bu, \; e = Dx, \; x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \; B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \; D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}.$$

Following the methodology of this chapter, we now select a sampling period T, say $T = 0.1$, and discretize this model via c2d:

$$x[(k+1)T] = A_d x(kT) + B_d u(kT), \quad e(kT) = Dx(kT).$$

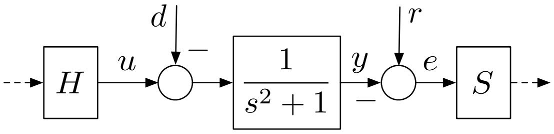
That is, the control signal $u(t)$ is the output of a zero-order hold, and we're going to impose the tracking requirement only at the sampling instants. In simplified notation

$$x_d' = A_d x_d + B_d u_d, \quad e_d = Dx_d.$$

The matrix A_d retains its block-triangular structure (here displayed to 3 significant figures):

$$A_d = \left[\begin{array}{cc|ccc} 0.995 & 0.100 & 0 & -459e-5 & -158e-6 \\ -998e-4 & 0.995 & 0 & -840e-4 & -459e-5 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.540 & 841e-4 \\ 0 & 0 & 0 & -841e-2 & 0.540 \end{array}\right], \quad B_d = \left[\begin{array}{c} 5e-3 \\ 998e-4 \\ \hline 0 \\ 0 \\ 0 \end{array}\right].$$

The exomodel has eigenvalues $1, 0.540 \pm 0.841j$. These are the continuous-time eigenvalues $0, \pm 10j$ mapped via $\lambda \mapsto \mathrm{e}^{\lambda T}$. The discrete-time reference signal is the sampled $r(t)$, and the discrete-time disturbance is the sampled $d(t)$. The block diagram is



To simplify notation further, we hereafter drop the subscript "d" and work only in discrete time. So the model is

$$x' = Ax + Bu, \quad e = Dx, \; x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \; B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \; D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}.$$

The **regulator problem** is to design a controller, with input e and output u, such that the feedback loop is stable, meaning the plant state $x_1(k)$ and the controller state go to zero when $x_2(0) = 0$, and the output is regulated, meaning $e(k)$ goes to zero for all initial conditions. We assume that all the eigenvalues of A_2 are unstable (but no assumption is made yet about A_1). It is also natural to assume that (D, A) is detectable because we will need to use an observer.

The solution is exactly the same as the continuous-time solution. We first look for a state feedback controller, $u = Fx$. Then we implement it via $u = F\hat{x}$ where $\hat{x}$ is from an observer with input e. For completeness, we briefly review the solution. First consider an uncontrolled system

$$x' = Ax = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix} x$$

$$e = Dx = \begin{bmatrix} D_1 & D_2 \end{bmatrix} x$$

with A_1 stable and A_2 unstable. Then it is known that there exists a unique X satisfying

$$A_1 X - X A_2 + A_3 = 0$$

Set

$$T = \begin{bmatrix} I & X \\ 0 & I \end{bmatrix}$$

resulting in

$$T^{-1} = \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix}$$

This gives

$$T^{-1} A T = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$$

Setting $\xi = T^{-1} x$ yields

$$\xi' = T^{-1} A T \xi = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \xi$$

$$e = DT\xi = \begin{bmatrix} D_1 & D_1X + D_2 \end{bmatrix} \xi$$

so that $e \to 0$ if and only if $D_1X + D_2 = 0$.

Now for the system under control

$$x' = Ax + Bu,$$

let $u = Fx$. Then the closed loop system is

$$x' = (A + BF)x, \quad e = Dx,$$

where

$$A + BF = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \begin{bmatrix} F_1 & F_2 \end{bmatrix} = \begin{bmatrix} A_1 + B_1F_1 & A_3 + B_1F_2 \\ 0 & A_2 \end{bmatrix}.$$

Based on the solvability condition for the uncontrolled system, it is straightforward to arrive at the following theorem.

Theorem 8 *Assume (D, A) is detectable and A_2 has only unstable eigenvalues. Then the regulator problem is solvable iff (A_1, B_1) is stabilizable and there exist matrices X, U such that*

$$A_1X - XA_2 + A_3 + B_1U = 0, \quad D_1X + D_2 = 0.$$

Design method

1. Choose F_1 so that $A_1 + B_1F_1$ is stable.

2. Solve

$$(A_1 + B_1F_1)X - XA_2 + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0 \tag{4.7}$$

   for X, F_2—we need only F_2.

3. Select L so that $A + LD$ is stable. The full-state observer is

$$\hat{x}' = A\hat{x} + Bu + L(D\hat{x} - e).$$

   Setting $u = F\hat{x}$, we get the observer-based controller

$$\hat{x}' = (A + BF + LD)\hat{x} - Le, \quad u = F\hat{x}.$$

Let us consider how to solve (4.7). First, in the case where the exogenous signals are constants, $A_2 = I$. Then the equations are

$$(A_1 + B_1F_1 - I)X + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0,$$

which can be written as

$$\begin{bmatrix} A_1 + B_1F_1 - I & B_1 \\ D_1 & 0 \end{bmatrix} \begin{bmatrix} X \\ F_2 \end{bmatrix} = - \begin{bmatrix} A_3 \\ D_2 \end{bmatrix}.$$

This is a linear matrix equation. It has a solution iff

$$\operatorname{rank}\begin{bmatrix} A_1+B_1F_1-I & B_1 \\ D_1 & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} A_1+B_1F_1-I & B_1 & A_3 \\ D_1 & 0 & D_2 \end{bmatrix}.$$

For example, if the matrix on the left is square and invertible,

$$\begin{bmatrix} X \\ F_2 \end{bmatrix} = -\begin{bmatrix} A_1+B_1F_1-I & B_1 \\ D_1 & 0 \end{bmatrix}^{-1}\begin{bmatrix} A_3 \\ D_2 \end{bmatrix}.$$

To solve (4.7) in general, let us introduce the following notation to keep track of the dimensions of various quantities. Let A_1 be an $n \times n$ matrix and A_2 be $p \times p$. Then A_3 is $n \times p$ and the state x is a $(n+p)$-vector. Let B_1 be $n \times m$. Then F_1 is $n \times m$ and F_2 is $m \times p$. Let D_1 be $q \times n$. Then D_2 is $q \times p$. Finally, if A is a $m \times p$ matrix and B is a $n \times q$ matrix, define the Kronecker product $A \otimes B$ as the $mn \times pq$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots \\ a_{21}B & a_{22}B & \cdots \\ \vdots & \vdots & \end{bmatrix}.$$

Now we turn to (4.7) for a general A_2. Since the equations *are* linear, they can be converted to a conventional vector equation. One way to do this is to take the columns of a matrix and stack them up to form a vector. For example, denoting the columns of X by $x_1, x_2, \ldots$, let v_X denote the column vector

$$v_X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}.$$

Since X is $n \times p$, v_X is a np-vector. If we do this vectorizing on the product matrix $(A_1+B_1F_1)X$, you can check that the result is

$$(I_p \otimes (A_1+B_1F_1))v_X,$$

where I_p is the $p \times p$ identity matrix, and

$$I \otimes A_1 = \text{block diag}(A_1, A_1, \ldots).$$

The vectorizing of a product of the form XD is a little trickier; it's $(D^T \otimes I)v_X$, so

$$D^T \otimes I = \begin{bmatrix} d_{11}I & d_{21}I & \cdots \\ d_{12}I & d_{22}I & \cdots \\ \vdots & \vdots & \end{bmatrix}.$$

In this way, equations (4.7) can be vectorized to

$$(I_p \otimes (A_1+B_1F_1) - A_2^T \otimes I_n)v_X + v_{A_3} + (I_p \otimes B_1)v_{F_2} = 0, \quad (I_p \otimes D_1)v_X + v_{D_2} = 0,$$

or, combining, to

$$\begin{bmatrix} I_p \otimes (A_1 + B_1F_1) - A_2^T \otimes I_n & I_p \otimes B_1 \\ I_p \otimes D_1 & 0_{pq,mp} \end{bmatrix} \begin{bmatrix} v_X \\ v_{F_2} \end{bmatrix} = - \begin{bmatrix} v_{A_3} \\ v_{D_2} \end{bmatrix}. \tag{4.8}$$

where $0_{j,k}$ denotes a $j \times k$ zero matrix. Thus solvability is equivalent to a rank test.

Example This is digital control of the cart/spring system with no disturbance, only a constant reference. Discretizing the model with $T = 0.1$ gives (with subscript "d" dropped)

$$A = \left[\begin{array}{cc|c} 0.995 & 0.100 & 0 \\ -998e-4 & 0.995 & 0 \\ \hline 0 & 0 & 1 \end{array}\right], \quad B = \left[\begin{array}{c} 5e-3 \\ 998e-4 \\ \hline 0 \end{array}\right]$$

$$D = \left[\begin{array}{cc|c} -1 & 0 & 1 \end{array}\right].$$

We select F_1 to stabilize $A_1 + B_1F_1$. Arbitrarily selecting the eigenvalues to be 0, we get

$$F_1 = \begin{bmatrix} -99.1 & -15.0 \end{bmatrix}.$$

Next, we have to solve

$$(A_1 + B_1F_1)X - XA_2 + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0.$$

Setting

$$\begin{bmatrix} X \\ F_2 \end{bmatrix} = - \begin{bmatrix} A_1 + B_1F_1 - I & B_1 \\ D_1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} A_3 \\ D_2 \end{bmatrix}.$$

gives $F_2 = 100$. Finally, assigning the eigenvalues of $A + LD$ at 0, we get

$$L = \begin{bmatrix} -97.1 & 24.8 & -100 \end{bmatrix}^T.$$

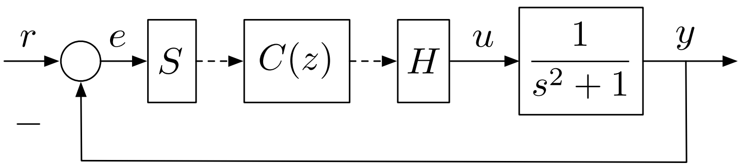
The resulting controller transfer function from e to u is obtained from the state equations

$$\dot{\hat{x}} = (A + BF + LD)\hat{x} - Le, \quad u = F\hat{x}.$$

We get

$$C(z) = \frac{768z^2 - 1090z + 423}{z^3 + 2z^2 - 0.875z - 2.12}.$$

The controller has a pole at $z = 1$, as it must. The structure has the block diagram:



□

Example This is digital control of the cart/spring system with the sinusoidal disturbance. We start with the discrete-time model (with subscript "d" dropped)

$$A = \left[\begin{array}{cc|ccc} 0.995 & 0.100 & 0 & -459e-5 & -158e-6 \\ -998e-4 & 0.995 & 0 & -840e-4 & -459e-5 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.540 & 841e-4 \\ 0 & 0 & 0 & -841e-2 & 0.540 \end{array}\right], \quad B = \left[\begin{array}{c} 5e-3 \\ 998e-4 \\ \hline 0 \\ 0 \\ 0 \end{array}\right]$$

$$D = \left[\begin{array}{cc|ccc} -1 & 0 & 1 & 0 & 0 \end{array}\right].$$

We select F_1 to stabilize $A_1 + B_1F_1$. Arbitrarily selecting the eigenvalues to be 0, we get

$$F_1 = \left[\begin{array}{cc} -99.1 & -15.0 \end{array}\right].$$

Next, we have to solve

$$(A_1 + B_1F_1)X - XA_2 + A_3 + B_1F_2 = 0, \quad D_1X + D_2 = 0.$$

Using the method in (4.8) gives

$$X = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & -926e-6 \end{array}\right], \quad F_2 = \left[\begin{array}{ccc} 100 & 0.919 & 364e-4 \end{array}\right].$$

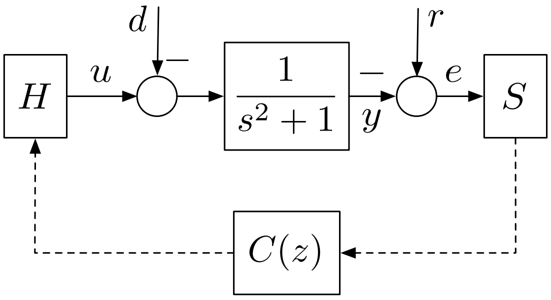
Finally, assigning the eigenvalues of $A + LD$ at 0, we get

$$L = \left[\begin{array}{ccccc} -105 & 42.6 & -109 & -126 & 473 \end{array}\right]^T.$$

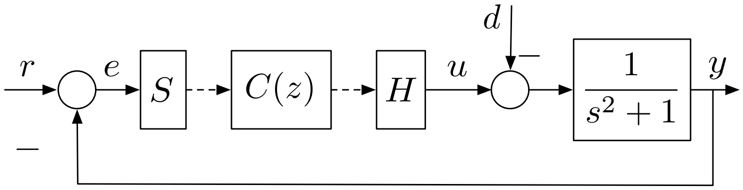
The resulting controller transfer function from e (i..e., $e_d(k)$) to u (i.e., $u_d(k)$) is

$$C(z) = \frac{1250z^4 - 3150z^3 + 3720z^2 - 2340z + 623}{z^5 + 1.99z^4 - 3.28z^3 + z^2 + 2.41z - 3.11}.$$

The structure has the block diagram:



This can be reconfigured into the more familiar form



□

Example What about digital control of the cart with the sinusoidal disturbance but with no spring? As in Chapter 1, if we model the constant reference, (D, A_d) will not be detectable. Designing the controller in this case is left as an exercise. □

4.7 Pathological Sampling

Let ω_s denote the sampling frequency (rad/s), that is,

$$\omega_s := \frac{2\pi}{T}. \tag{4.9}$$

Look at this example, an oscillator at frequency ω_s:

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_s^2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The discretization has

$$B_d = \int_0^T \mathrm{e}^{\tau A} d\tau B = A^{-1}(\mathrm{e}^{TA} - I)B = 0.$$

Thus (A, B) is controllable, but (A_d, B_d) is not. So sampling can destroy controllability if the sampling frequency is inappropriate for the system.

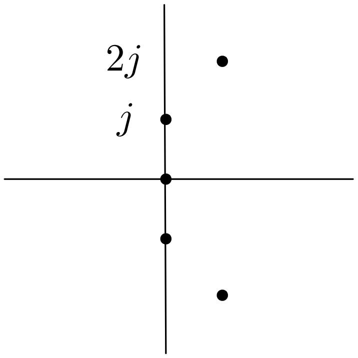
The sampling frequency ω_s is **pathological** (relative to A) if A has two different eigenvalues, $\lambda_1 \neq \lambda_2$, that become equal eigenvalues of A_d, i.e.,

$$\mathrm{e}^{\lambda_1 T} = \mathrm{e}^{\lambda_2 T}.$$

This happens iff $\lambda_1 = \lambda_2 + jk\omega_s$, for some integer k.

Example Suppose the eigenvalues of A (counting multiplicities) are

$$0, 0, \pm j, 1 \pm 2j.$$



The pathological sampling frequencies can be calculated as follows. The eigenvalues go through two vertical lines. For the line Re $s = 0$: The lowermost eigenvalue is $s = -j$. The distances from it to the other eigenvalues on this line are 1 and 2. Thus the sampling frequency is pathological if $k\omega_s = 1$ or $k\omega_s = 2$ for some positive integer k. Thus the following frequencies are pathological:

$$\left\{\frac{1}{k} : k \geq 1\right\} \cup \left\{\frac{2}{k} : k \geq 1\right\}.$$

For the line Re $s = 1$: The lowermost eigenvalue is $s = 1 - j$. The distance from it to the other eigenvalue on this line is 4. Thus the sampling frequency is pathological if $k\omega_s = 4$ for some positive integer k. Thus the following frequencies are pathological:

$$\left\{\frac{4}{k} : k \geq 1\right\}.$$

Since we have looked at all possible vertical lines, we have counted them all; thus the set of all pathological sampling frequencies is

$$\left\{\frac{1}{k} : k \geq 1\right\} \cup \left\{\frac{2}{k} : k \geq 1\right\} \cup \left\{\frac{4}{k} : k \geq 1\right\}.$$

Since 4 is divisible by 1 and 2, the set of all pathological sampling frequencies is

$$\left\{\frac{4}{k} : k \geq 1\right\}.$$

Notice that this set has an upper bound (4); therefore, ω_s will be non-pathological if it is large enough. □

It turns out that controllability and observability are preserved if the sampling frequency is non-pathological.

Theorem 9 *If the sampling frequency is non-pathological, then* (A, B) *controllable* $\Longrightarrow$ (A_d, B_d) *controllable, and* (C, A) *observable* $\Longrightarrow$ (C, A_d) *observable.*

Proof We'll prove just the second implication—the other is similar.

So assume the sampling frequency is non-pathological and (C, A) is observable. To prove that (C, A_d) is observable, we'll show all the eigenvalues of A_d are observable. Now each eigenvalue of A_d has the form $\mathrm{e}^{T\lambda}$, where λ is an eigenvalue of A. We must show that

$$\operatorname{rank}\begin{bmatrix} A_d - \mathrm{e}^{T\lambda} I \\ C \end{bmatrix} = n$$

given that

$$\operatorname{rank}\begin{bmatrix} A - \lambda I \\ C \end{bmatrix} = n.$$

Define the function

$$g(s) = \frac{\mathrm{e}^{Ts} - \mathrm{e}^{T\lambda}}{s - \lambda}. \tag{4.10}$$

This is analytic everywhere (the "pole" at $s = \lambda$ is cancelled by a "zero" there). Moreover,

$$\begin{aligned} \{\text{zeros of } g\} &= \{s : \mathrm{e}^{Ts} = \mathrm{e}^{T\lambda}, s \neq \lambda\} \\ &= \{s : Ts = T\lambda + j2\pi k, k = \pm 1, \pm 2, \ldots\} \\ &= \{s : s = \lambda + jk\omega_s, k = \pm 1, \pm 2, \ldots\}. \end{aligned}$$

By non-pathological sampling, the zeros of g are disjoint from the eigenvalues of A. Now the eigenvalues of the matrix $g(A)$ are precisely the values of g at the eigenvalues of A (spectral mapping theorem). Thus 0 is not an eigenvalue of $g(A)$, so $g(A)$ is invertible. Now

$$\mathrm{e}^{Ts} - \mathrm{e}^{T\lambda} = g(s)(s - \lambda). \tag{4.11}$$

Hence

$$\mathrm{e}^{TA} - \mathrm{e}^{T\lambda} I = g(A)(A - \lambda I), \tag{4.12}$$

that is,

$$A_d - \mathrm{e}^{T\lambda} I = g(A)(A - \lambda I). \tag{4.13}$$

Thus

$$\begin{bmatrix} A_d - \mathrm{e}^{T\lambda} I \\ C \end{bmatrix} = \begin{bmatrix} g(A) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A - \lambda I \\ C \end{bmatrix}.$$

Since $g(A)$ is invertible, so is

$$\begin{bmatrix} g(A) & 0 \\ 0 & I \end{bmatrix}.$$

Therefore

$$\operatorname{rank}\begin{bmatrix} A_d - \mathrm{e}^{T\lambda} I \\ C \end{bmatrix} = \operatorname{rank}\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}.$$

□

The above example shows we can lose controllability by sampling. What about the converse? Can we gain controllability by sampling? No.

Theorem 10 *If (A, B) is not controllable, neither is (A_d, B_d).*

Proof If (A, B) is not controllable, there exists an eigenvalue λ of A such that

$$\operatorname{rank} \begin{bmatrix} A - \lambda I & B \end{bmatrix} < n. \tag{4.14}$$

Then there exists a vector $x \neq 0$ orthogonal to all the columns of this matrix, that is,

$$x^* \begin{bmatrix} A - \lambda I & B \end{bmatrix} = 0. \tag{4.15}$$

This implies that

$$x^* A = \lambda x^*, \quad x^* B = 0. \tag{4.16}$$

Now we obtain in succession

$$\begin{aligned} x^* A^2 &= (x^* A) A \\ &= \lambda x^* A \\ &= \lambda^2 x^* \\ x^* A^3 &= \lambda^3 x^* \\ &\text{etc.} \end{aligned}$$

Since

$$\mathrm{e}^{TA} = I + TA + \frac{T^2}{2} A^2 + \ldots \tag{4.17}$$

we have

$$\begin{aligned} x^* \mathrm{e}^{TA} &= x^* \left(I + hA + \frac{T^2}{2} A^2 + \cdots \right) \\ &= x^* + h\lambda x^* + \frac{T^2}{2} \lambda^2 x^* + \cdots \\ &= \mathrm{e}^{T\lambda} x^*. \end{aligned}$$

Thus

$$\begin{aligned} x^* (A_d - \mathrm{e}^{T\lambda} I) &= x^* (\mathrm{e}^{TA} - \mathrm{e}^{T\lambda} I) \\ &= (\mathrm{e}^{T\lambda} - \mathrm{e}^{T\lambda}) x^* \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} x^* B_d &= x^* \int_0^T \mathrm{e}^{\tau A} d\tau B \\ &= \int_0^T \mathrm{e}^{\tau \lambda} d\tau \, \underbrace{x^* B}_{=0} \\ &= 0. \end{aligned}$$

Hence

$$x^* \left[\begin{array}{cc} A_d - \mathrm{e}^{T\lambda} I & B_d \end{array} \right] = 0 \tag{4.18}$$

and so

$$\operatorname{rank} \left[\begin{array}{cc} A_d - \mathrm{e}^{T\lambda} I & B_d \end{array} \right] < n. \tag{4.19}$$

This implies that $\mathrm{e}^{T\lambda}$ is an uncontrollable eigenvalue of A_d, proving that (A_d, B_d) is not controllable. □

Similarly, we can't gain observability by sampling.

The results for stabilizability and detectability are entirely analogous.

4.8 Conclusion

Some comments about selecting a sampling period T. First, the stabilization/tracking problem needs to be solvable for the discretized plant. Thus, T must not be pathological. Next, T needs to be small enough so that intersample response is good; note that the direct digital design method is for the discretized plant. This is best addressed by simulation. Finally, T should be otherwise as large as possible to keep the processor speed as low as possible for cost, and so that all computations can be done in real time.

4.9 Exercises

1. (by hand) Consider the discrete-time system

   $$x' = \begin{bmatrix} 0.3 & 0.4 \\ -0.5 & 1.6 \end{bmatrix} x + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u.$$

   We would like to find a feedback matrix F that places the closed loop eigenvalues of $A + BF$ at 0.55 and 0.54.

   (a) Determine F by transforming to controllable canonical form.

   (b) Repeat using Ackermann's formula.

2. (by Scilab or MATLAB) Consider the discrete-time system

   $$x' = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.16 & 0.84 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u.$$

   (a) Determine the state feedback matrix F such that the closed-loop system has all three eigenvalues at 0 (deadbeat response). Use transformation to controllable canonical form.

   (b) Repeat using Ackermann's formula.

3. (by Scilab or MATLAB where possible) Consider the continuous-time system

   $$\dot{x}(t) = \begin{bmatrix} 1 & 1 \\ 0 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

(a) For $T = 0.2$, discretize the system via `c2d` to yield a discrete-time state model.

(b) Assuming that the state $x(kT)$ is sensed, determine the matrix F to place the eigenvalues of the discrete-time closed-loop system at $0.8 \pm 0.5j$.

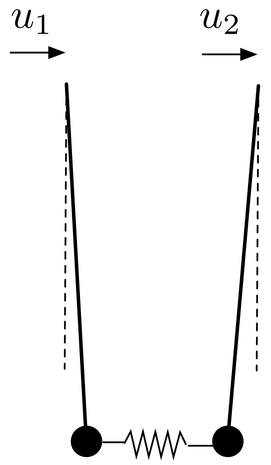
(c) Compute the intersample behaviour of $x(t)$ for $2T \leq t \leq 3T$ starting from $x(0) = (1, 0)$.

4. (by Scilab or MATLAB) Consider the continuous-time system

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ -25 & -6 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 3 & 1 \end{bmatrix} x(t).\end{aligned}$$

Check that the system is controllable and observable. Let the sampling period be $T = \pi/4$. Determine the discretized state model via `c2d`. Is it controllable? Observable?

5. Two pendula, of masses M_1 and M_2 and lengths L_1 and L_2, are coupled by a spring, of stiffness K:



The two inputs are the positions u_1 and u_2 of the pivots of the pendula, and the two outputs are their angles y_1 and y_2. The equations of motion are as follows:

$$\begin{aligned}M_1(\ddot{u}_1 - L_1\ddot{y}_1) &= M_1 g y_1 - K(u_1 - L_1 y_1) + K(u_2 - L_2 y_2) \\ M_2(\ddot{u}_2 - L_2\ddot{y}_2) &= M_2 g y_2 + K(u_1 - L_1 y_1) - K(u_2 - L_2 y_2).\end{aligned}$$

Take the following numerical values: $M_1 = 1$ kg, $M_2 = 10$ kg, $L_1 = L_2 = 1$ m, $K = 1$ N/m.

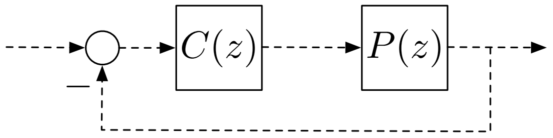
(a) By taking Laplace transforms, find the transfer function $G(s)$ from u_1 to y_2.

(b) From $G(s)$ find a continuous-time state model. What are the eigenvalues of A?

(c) Take $T = \pi/|\lambda|$, where λ is an eigenvalue of A. Using MATLAB, compute a state model of the discretization via `c2d`. Is (A_d, B_d) controllable?

6.

$$C(z) = \frac{z^2 - 1}{10z^2 + z + 2}, \quad P(z) = \frac{5}{(z-1)(z^3 + 2z)}$$



Find the closed-loop matrix A_{cl}. Is the feedback system stable?

7.

$$\begin{aligned} x' &= \begin{bmatrix} 1 & 0 & 2 \\ 3 & 1 & 1 \\ 4 & 0 & 0 \end{bmatrix} x \\ y &= \begin{bmatrix} 1 & 1 & 2 \end{bmatrix} x \end{aligned}$$

(a) Design a full-order observer such that the estimation error system dynamics have eigenvalues at $0.2, 0.5, 0.8$.

(b) Design a reduced-order observer with eigenvalues of the error dynamics at $0.5, 0.8$.

8. Design an observer-based controller to stabilize

$$P(z) = \frac{5z}{(z-1)(z+2)}.$$

9. Five mobile robots move around a lab floor without human supervision. Let's model them in a very idealized way as points moving around in the complex plane. Suppose the robots are numbered $1, \dots, 5$ and the position of robot i is denoted z_i. Suppose the velocity of robot i is directly controllable, so the kinematic equations are

$$\dot{z}_i = u_i, \quad i = 1, \dots, 5.$$

Each robot has onboard a camera and a computer. Each robot is assumed to see the relative positions of its two immediate neighbours in the ordering $1, 2, \dots, 5$, with wrapping; that is,

1	sees	$5, 2$
2	sees	$1, 3$
3	sees	$2, 4$
4	sees	$3, 5$
5	sees	$4, 1$

The strategy of robot 1 is to head towards the centroid of the two robots it sees. Thus, the camera of robot 1 sees the relative positions $z_5 - z_1, z_2 - z_1$, these are sampled, then averaged, and then output through a zero-order hold to become u_1. The other four robots do likewise.

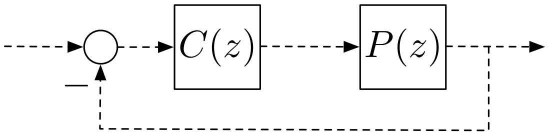
(a) Draw the complete block diagram. There will be integrators, samplers, holds, and summing junctions.

(b) Find a discrete-time state model of the overall system of robots that is an exact model at the sampling instants.

(c) Set $T = 0.1$. Using simulation (or analysis if you can), find how the formation of robots evolves.

(d) Is there some sampling period T for which the formation is unstable?

10. Consider the setup



Let both $P(z)$ and $C(z)$ be proper but not strictly proper:

$$P(z) = C(zI - A)^{-1}B + D, \quad C(z) = C_c(zI - A_c)^{-1}B_c + D_c.$$

Derive the closed-loop matrix A_{cl}. Hint: There's an assumption required for A_{cl} to exist.

11. (a) Consider a plant with input u, output y, and transfer function

$$P(z) = \frac{5z}{(z-2)(2z+1)}.$$

The goal is for y to asymptotically track a unit step r. Design a stabilizing controller with input $r - y$ and output u.

(b) Repeat for

$$P(z) = \frac{5(z-1)}{(z-2)(2z+1)}.$$

(c) Repeat for

$$P(z) = \frac{2z+1}{z^2+1},$$

guaranteeing that the tracking error e converges to 0 in finite time.

12. (a) Consider a plant with input $u + d$, where d is a disturbance, output y, and transfer function

$$P(z) = \frac{5z}{(z-2)(2z+1)}.$$

The disturbance $d(k)$ is a sinusoid with frequency $\pi/4$. The goal is for y to asymptotically track a unit step r. Design a stabilizing controller with input $r - y$ and output u.

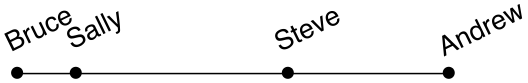
(b) Repeat but with d the step and r a sinusoid with frequency $\pi/4$.

13. Given (A, B) controllable, $A: 2 \times 2$, $B: 2 \times 1$. Suppose the desired characteristic polynomial is $r(z) = z^2$. Then $r(A) = A^2$. Ackermann's formula is

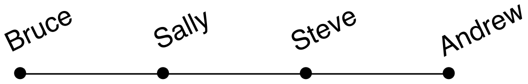
$$F = -\left[\begin{array}{cc} 0 & 1 \end{array}\right] W^{-1} A^2.$$

Prove that $A + BF$ is nilpotent, that is, $(A + BF)^2 = 0$.

14. (How soldiers line up) Bruce, Sally, Steve, and Andrew play the following game: They stand in a line in that order:



Bruce and Andrew don't move throughout the game. Sally starts by moving to the midpoint between Bruce and Steve; then Steve moves to the midpoint between Sally and Andrew; then Sally repeats her strategy; then Steve his; and so on. Prove that Bruce, Sally, Steve, and Andrew end up being equi-spaced on the line:

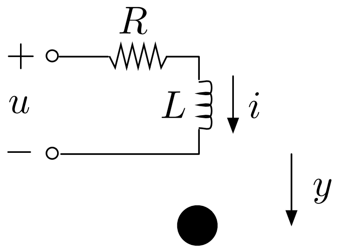


15. In the coupled-pendulum problem, the A-matrix is

$$A = \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -106.82 & 0 & -20.7 & 0 \end{array}\right].$$

What are the pathological sampling frequencies?

16. Consider the Maglev system



$$L\frac{di}{dt} + Ri = u$$

$$M\frac{d^2y}{dt^2} = Mg - K\frac{i^2}{y^2}$$

Realistic numerical values are $M = 0.1$ Kg, $R = 15$ ohms, $L = 0.5$ H, $K = 0.0001$ Nm^2/A^2, $g = 9.8$ m/s^2.

(a) Linearize the system about $y = 0.01$ m and find a continuous-time state model. Assume there's a sensor for y only.

(b) Using tracking/redulator theory, design a digital controller to suspend the ball at $y = 0.01$. Use the sampling frequency of 2 kHz.

17. Consider the plant model

$$\dot{x} = u - d.$$

Suppose the disturbance is a sinusoid of frequency 2 rad/s; the exomodel is

$$\ddot{d} + 4d = 0.$$

Suppose u is the output of the hold H with $T = 1$. At the sampling inputs, the plant model can be written

$$x_d' = x_d + u_d - w,$$

where $x_d(k) = x(kT)$ and $u_d(k) = u(kT)$. What is the exomodel for the discrete-time disturbance, w?

18. Design a digital controller for the cart example with the sinusoidal disturbance but with no spring.

Chapter 5

Introduction to Real-Time Scheduling

5.1 Examples of Task Scheduling: TrueTime

In the course so far, we've made the following simplifying idealizations (among others) about the digital controller:

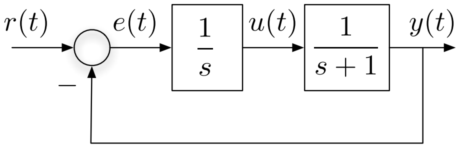
1. It takes no time to sample a signal.

2. It takes no time to compute the new state variables and control values.

3. It takes no time to output the control values.

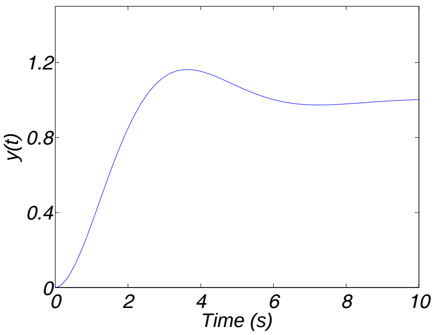
In this chapter, we'll be more realistic and model the non-zero times required to complete these instructions. In the examples to follow, we will be using a MATLAB simulation tool called TrueTime[1] that simulates and schedules real-time systems.

Example 1 Let us begin with a simple continuous-time feedback loop with integral control:

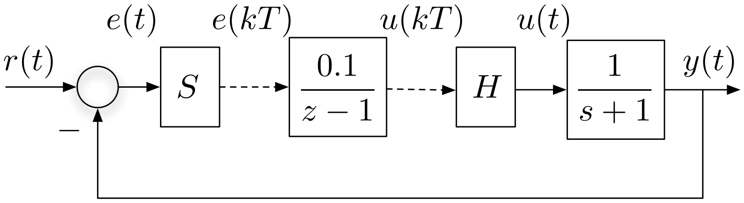


This system's continuous-time step response is

[1] Developed by Anton Cervin of Lund University in his PhD thesis, Integrated Control and Real-Time Scheduling, http://www.control.lth.se/Publication/cer03dis.html.



Suppose we've selected the sampling frequency of 10 Hz ($T = 0.1$) and we've discretized the integral controller via $C2D$:



The digital controller equation is therefore

$$u[(k+1)T] = u(kT) + 0.1e(kT).$$

In more detail, the discrete sample times are

$$0, 0.1, 0.2, 0.3, \ldots$$

During each intersample period, the digital controller should perform the following sequence of instructions:

Instruction 1: read e from the input port

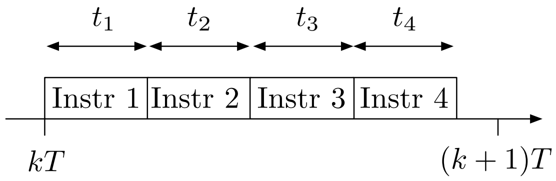
Instruction 2: read u from memory

Instruction 3: compute $u + 0.1e$ and assign this value to u

Instruction 4: send u to the output port

These instructions taken together make up a **task**, τ.

Let us suppose each of these operations requires a period t_i for its execution. Then they should be executed in sequence as follows:

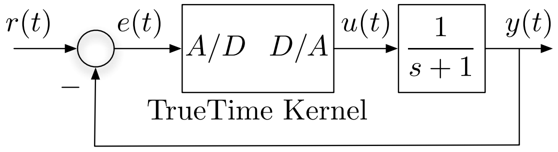


We're neglecting the time required to switch from one instruction to the next. The total time to complete the task is

$$C = t_1 + t_2 + t_3 + t_4.$$

Notice that the output signal is sent at the end of t_4, not at the next sample instant, $(k+1)T$. So the hold block is now time-variable in this sense: Whenever the output is computed and sent to the output port, it is held until a new output is sent. Consequently, we are therefore no longer dealing with a time-invariant discrete-time model. The TrueTime simulations will be consistent with this new time-varying model.

Using TrueTime gives the following block diagram:



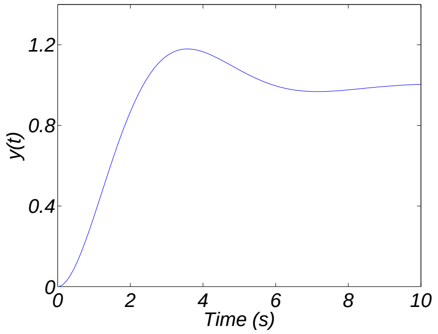
Here, the kernel discretizes $e(t)$, computes the control signal, and converts the signal back to analog.

Feasibility of execution obviously requires $C \leq T$, or

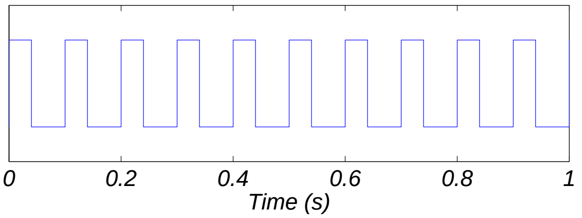
$$U = \frac{C}{T} \leq 1,$$

where U is defined as the **utilization factor** (or simply utilization). The utilization is the fraction of processor time spent on executing a task. If $U \leq 1$ for this simple example, then the task has enough time to complete before the next period. Of course, the period T must be chosen sufficiently small so that the system's performance doesn't degrade considerably.

Assume that $C = 40ms$ and the period is still $T = 100ms$. In this case, $U = 0.4$. This system's step response was simulated in TrueTime:

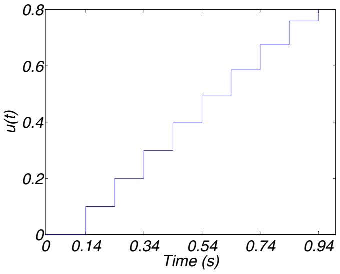


This is very similar to the continuous-time step response. Here's a plot of the task schedule, from 0 to 1 seconds:



The tasks are released at the beginning of each period, and the schedule is *high* during the time it takes to execute these tasks. In the first period, the schedule is high for $40ms$, then returns to *low* until the next period. When studying the task schedule, we observe that during each sample period, the task completes before its deadline (next period).

There's a slight anomaly. Here's the control signal $u(t)$ from 0 to 1 seconds:



As you can see, the signal $u(t)$ doesn't start proper action until the second period. The reason for this is that TrueTime can have problems reading analog inputs at time zero because of block execution problems in Simulink. The first control value is sent at $140ms$, and is updated every $100ms$ thereafter. Notice that, after the first period, the instants the control values are updated are correct.

What would happen if $C > T$, that is, the task hasn't been completed before a new sample value must be read in? This would depend on how the processor was programmed. We shall assume that the current task is aborted.

In this example there is no issue of scheduling: There is only one task to perform repeatedly, and the instructions must be executed in order. □

The problem becomes much more interesting when several plants have to be controlled by one shared processor. The tasks must compete with one another for CPU time. There is therefore a need to schedule the tasks such that all tasks will meet their deadlines. The set of rules that say when the processor should execute the tasks makes up a **scheduling algorithm**. A task set is defined to be **schedulable** under a given scheduling algorithm if all tasks meet their deadlines.

One way for the scheduler to decide when the processor should execute each task is first to assign priorities to the tasks. Under the **fixed-priority** (FP) scheduling protocol, each task is periodic and can be modeled as having

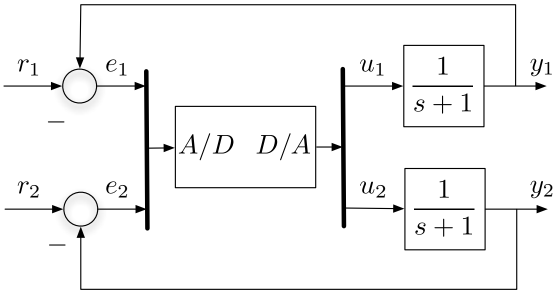
(a) an execution time, C_i

(b) a period, T_i

(c) a unique priority, P_i.

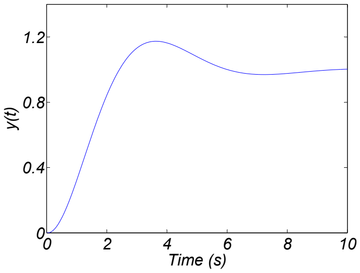
In general, C_i is determined by the number of instructions that make up the task and hardware-specific parameters such as clock speed. The execution times are determined by actual testing. The task is ready to be executed (released) at the beginning of each period T_i. The period is a design parameter that must be assigned to each task, with the utilization factor kept in mind. As we'll see later, there are limits to how high U can reach to guarantee that all tasks will meet their deadlines. During run-time, if two tasks are released at the same time, the task with the higher priority executes. When this task is completed, the next task may begin its execution. In addition, scheduling protocols for real-time systems incorporates **preemption**. If more than one task are available for execution, the one that has the highest priority will be executed, thereby preempting a lower priority task which may currently be executing.

Example 2 Suppose two plants are to be controlled and must share the same control processor. Both plants have the transfer function $\frac{1}{s+1}$. They are to be controlled by a discretized integral controller, similar to Example 1.

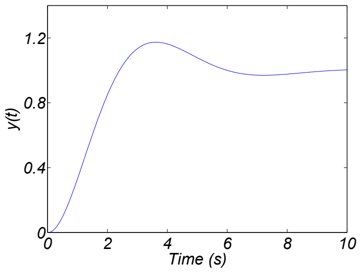


Let's denote the two control tasks by τ_1 and τ_2.

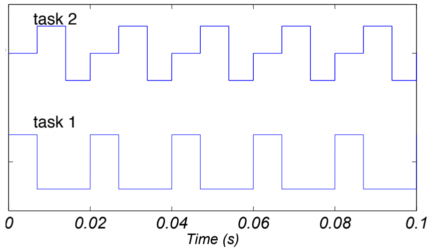
Suppose we've found that each task takes $7ms$ to complete: $C_1 = C_2 = 7$. Suppose the periods are chosen to be $20ms$ long for both tasks: $T_1 = T_2 = 20$. Let's assign a higher priority to τ_1: P_1=1 and $P_2 = 2$. The systems' step responses were simulated under the fixed-priority scheduling protocol. The results are as follows: Step response of System 1:



Step response of System 2:



The task schedules are as follows:



At $t = 0$, the task with the higher priority, τ_1, begins executing and continues until $t = 7ms$. Task 2 waits until Task 1 is finished (as indicated by a medium height in the task schedule), at which time it begins its execution. At $t = 14ms$, no task is ready, so the CPU becomes idle. At the beginning of the next period, the sequence repeats itself. This process continues until the end of the simulation. It is obvious that both tasks meet their deadlines. Therefore, this task set is schedulable under FP scheduling.

Since C_1/T_1 is the fraction of time spent running τ_1, and similarly for τ_2, the total utilization factor is defined to be

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} = 0.7.$$

The utilization is small enough to schedule the task set (as we will prove). □

The task schedule shown above is generated by TrueTime. A similar graphical tool that can be hand-drawn easily is the timing diagram. The following example illustrates its use.

Example 3: Scheduling Periodic Tasks with Completion Deadlines

2 Processes A and B

Deadline for A, which is the same as its period, is 20 ms.

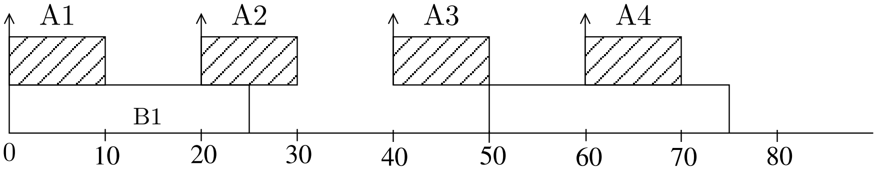
Deadline for B, which is the same as its period, is 50 ms.

It takes 10 ms to execute process A.

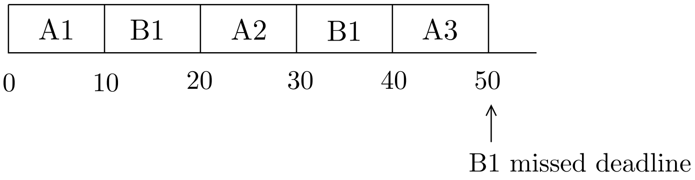
It takes 25 ms to execute process B.

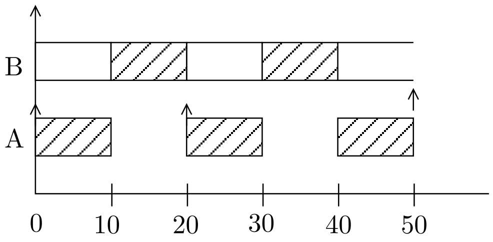
Process	Arrival Time	Exec. Time	Deadline
A(1)	0	10	20
A(2)	20	10	40
A(3)			
A(4)			
A(5)			
⋮			
B(1)	0	25	50
B(2)	50	25	100
⋮			

Timing Table for both processes

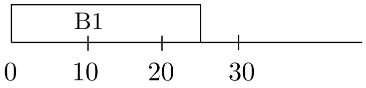


If the priority is fixed and A has higher priority, sequence will be





Note that preemption of B by A occurs at 20 ms and 40 ms.
If B has higher priority, the sequence will be

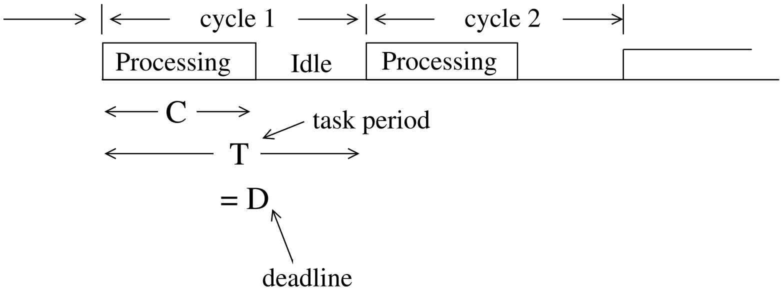


A1 misses deadline

The above example shows that depending on the scheduling algorithm, processes may meet or miss their deadlines.

For the rest of the chapter, we shall focus on the simplest situation: scheduling of real-time periodic processes. This is sufficient to demonstrate the central role played by scheduling in the design of real-time systems.

We introduce some terminology for a periodic real-time process.



Here C denotes the worst-case processing or execution time, and T is the task period, which will be for simplicity assumed to be the same as D, the deadline.

5.1.1 Schedulability

A key concept in the scheduling of processes is that of schedulability.

Definition:

A set of tasks or processes is **schedulable** if there exists a schedule so that all processes meet their deadlines.

The schedule that enables all processes to meet their deadlines is called a feasible schedule.

Intuitively, if the total demand for resources is too high, the processes will not be schedulable. We have the following

Necessary Condition for Schedulability

For a set of periodic processes to be schedulable, with deadlines equal to the periods, it is necessary that

$$\sum \frac{C_i}{T_i} \leq 1$$

Think of $\frac{C_i}{T_i}$ as the percentage that the processor has to devote to process i to complete its execution. These percentages must sum to ≤ 1 in order for the processor to complete all tasks successfully. We call $U = \sum \frac{C_i}{T_i}$ the utilization.

A more formal proof of the necessity of $U \leq 1$ is now given. For simplicity, we assume the T_is are integer-valued. Extending it to the non-integer-valued case is not difficult. We give a proof by contradiction. Suppose $\sum_1^n \frac{C_i}{T_i} > 1$, where n is the total number of processes. We need to show that some task cannot be completed within the deadline. Let $\hat{T}$ be the least common multiple of $T_1, T_2, \cdots, T_n$, and set $L_i = \frac{\hat{T}}{T_i}$. The quantity L_i corresponds to the number of times task i is released. Over the interval $[k\hat{T}, (k+1)\hat{T}]$, requests of $\sum L_iC_i$ units of work will be made to the processor. Since by assumption, $\sum L_iC_i = \hat{T} \sum \frac{C_i}{T_i} > \hat{T}$, the total amount of computation requested cannot be completed within the time period so that some task will miss its deadline.

As an illustration, suppose we have 3 tasks with periods $T_1 = 2$, $T_2 = 6$, $T_3 = 9$ and corresponding execution times $C_1 = 1$, $C_2 = 2$, and $C_3 = 2$. The utilization $U = \frac{1}{2} + \frac{1}{3} + \frac{2}{9} > 1$. In terms of work units needed in an interval of length $\hat{T}$, we have $\hat{T} = 18$ so that $L_1 = 9$, $L_2 = 3$, and $L_3 = 2$ giving $\sum_i L_i C_i = 9 + 6 + 4 = 19 > \hat{T}$.

We have now a condition that tells us when it is impossible to find a feasible schedule. A more important question is: what conditions guarantee the existence of a feasible schedule, and how do we find such feasible schedules. We focus on priority-based scheduling with preemption.

5.1.2 Rate Monotonic Scheduling (RMS)

We now introduce a commonly used fixed priority scheduling with preemption policy, the **Rate-Monotonic** scheduling protocol. The rate-monotonic scheduler assigns priority in the following manner: **The highest priority task is the one with the shortest period**. As a shorter period correponds to a higher rate, hence the name rate-monotonic. The rate-monotonic scheduler is an optimal algorithm in the class of fixed priority schedulers in the following sense: there is no other fixed priority scheduler which can schedule a task set that cannot be scheduled by RMS.

The following idealized ssumptions are made in the analysis of RMS:

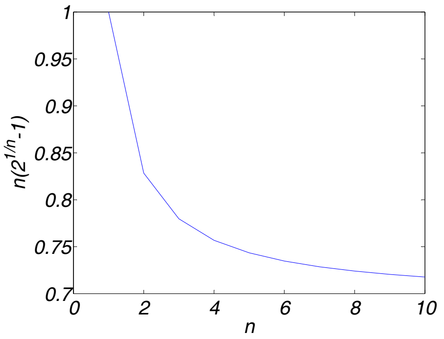
- All tasks to be scheduled are periodic
- $D_i = T_i$
- C_i known
- no interprocess interaction and communication, i.e. all processes are independent.
- tasks may not suspend themselves
- priorities are unique
- overheads, context switch times, and interrupt times are assumed to be 0.

Schedulability condition using RMS is: All tasks will meet their deadlines if

$$U = \sum_{i}^{n} \frac{C_i}{T_i} \leq n\left(2^{\frac{1}{n}} - 1\right) \tag{5.1}$$

where n is the number of tasks. (5.1) is only a **sufficient** condition, but not necessary in general. For $n = 1, 2, 3$, $n\left(2^{\frac{1}{n}} - 1\right) = 1, 0.8284, 0.7798$, respectively. We shall examine the RM schedulability condition in greater detail in the next section.

The upper bound $UB(n) = n(2^{1/n} - 1)$ is plotted from $n = 1$ to $n = 10$ here:



From the graph, we see $UB(n)$ is monotonically decreasing. We can also check this analytically by letting

$$f(p) = p(e^{\frac{1}{p}} - 1), \quad p \geq 1$$

We have

$$\frac{d}{dp} f(p) = f'(p) = (1 - \frac{1}{p})e^{\frac{1}{p}} - 1$$

Note that $f'(p) = -1$ at $p = 1$ and converges to 0 as $p \to \infty$. Furthermore,

$$f''(p) = -\frac{1}{p^2}e^{\frac{1}{p}}(1 - \frac{1}{p}) + \frac{1}{p^2}e^{\frac{1}{p}} = \frac{1}{p^3}e^{\frac{1}{p}} > 0, \quad \text{for } p \geq 1$$

This shows $f'(p)$ monotonically increases from -1 to 0, corresponding to the behaviour shown in the graph. The limiting value of $UB(n)$ is given by

$$\lim_{n\to\infty} UB(n) = \ln 2 \approx 0.69.$$

Proof: Let $x = 1/n$. Then $UB(x) = (2^x - 1)/x$ and

$$\begin{aligned} \lim_{x\to 0} UB(x) &= \lim_{x\to 0} \frac{2^x - 1}{x} \\ &= \left.\frac{d}{dx}2^x\right|_{x=0} \\ &= \left.2^x \ln 2\right|_{x=0} \\ &= \ln 2. \end{aligned}$$

□

We now give some examples on how RMS schedules tasks.

Example:

I.

	T	D	C	P	U
A	80	80	32	3	0.4
B	40	40	5	2	0.125
C	16	16	4	1	0.25

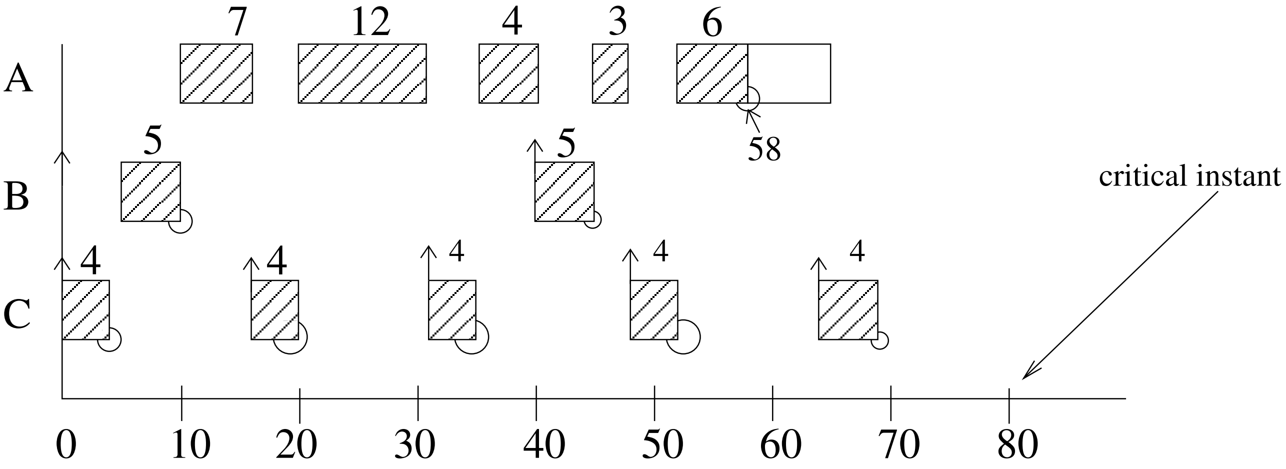
$U = 0.775$

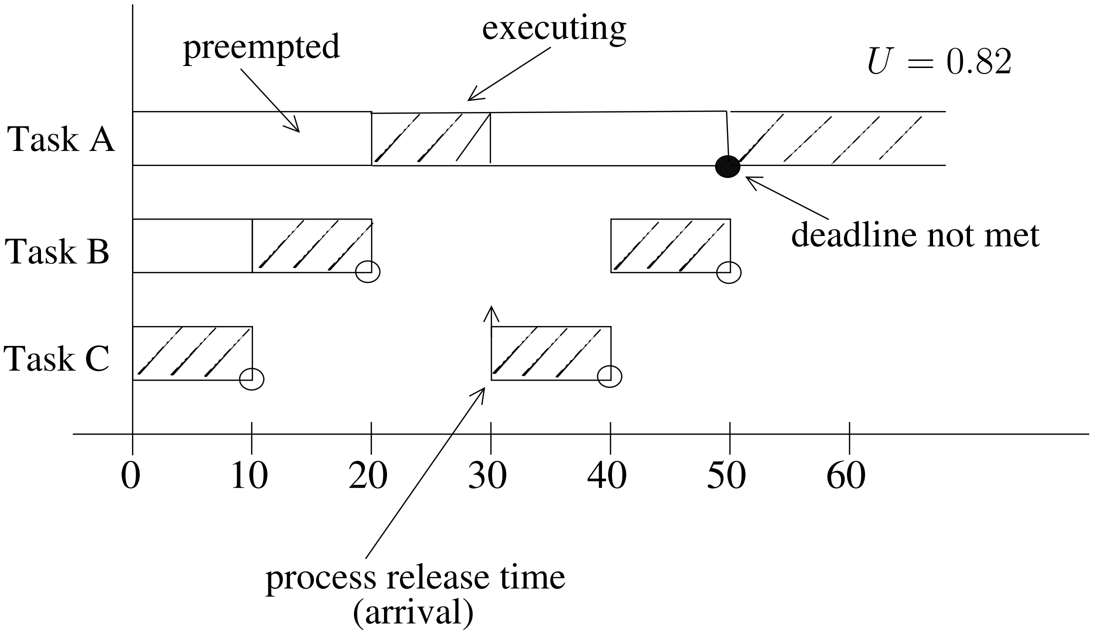
$n(2^{\frac{1}{n}} - 1)$ for $n = 3 \simeq 0.78$

Under RMS, the above processes should be schedulable. We draw the timing diagram for the processes.

II.

Task	T	D	C	P	U
A	50	50	12	3	0.24
B	40	40	10	2	0.25
C	30	30	10	1	0.33


7
12
4
3
6
A
5
5
58
B
critical instant
4
4
4
4
4
C
0
10
20
30
40
50
60
70
80


preempted
executing
$U = 0.82$
Task A
Task B
deadline not met
Task C
0
10
20
30
40
50
60
process release time
(arrival)

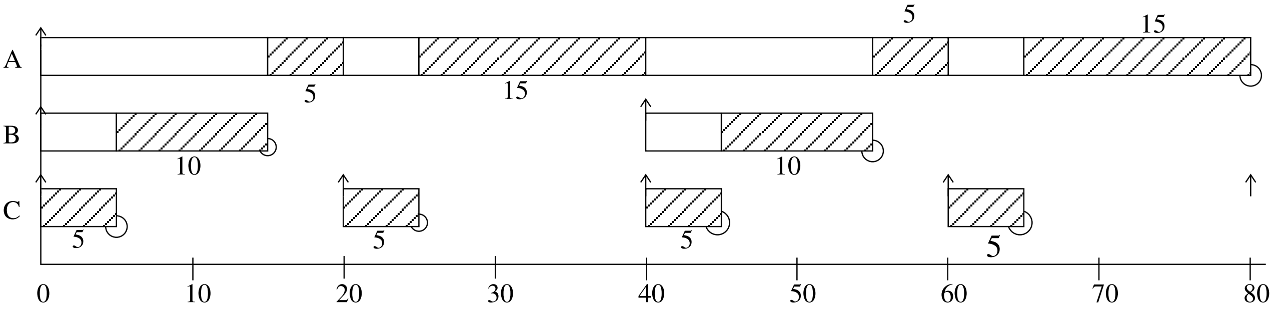
In this case, the utilization is higher than the bound given in (5.1). So there is no guarantee that RMS can schedule the 3 processes. Indeed the timing diagram is given by

Process A does not meet its deadline.

III.

	T	D	C	P	U
A	80	80	40	3	0.5
B	40	40	10	2	0.25
C	20	20	5	1	0.25

Utilization = 100% but deadlines are met, as shown by the following timing diagram.



5.1.3 Response Time Analysis

The schedulability condition for the rate monotonic scheduler is sufficient but not necessary. A more accurate analysis for schedulability with a fixed priority scheduler is based on worst case response time analysis. Let the worst case response time for task i be R_i. Then a more accurate condition for schedulability is $R_i \leq D_i$ for all i. R_i can be estimated using the equation

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \tag{5.2}$$

where $hp(i)$ = set of tasks having higher priority than i, and $\lceil x \rceil$ is the ceiling of x, i.e. the smallest integer $\geq x$. The reason for this equation is that for the highest priority process, the worst case response time will equal its own response time. Lower priority tasks experience $R_i = C_i + I_i$ where I_i is the interference time due to processes with higher priority than i. The interference is worst when all tasks with higher priority than i is released at the same time as i. Such an instant is called a critical instant for task i. If the critical instant for process i occurs at 0, physically all processes with priority $\geq i$ are released, so that the response time will satisfy

$$R_i \geq C_i + \sum_{j \in hp(i)} C_j$$

This will be the end of the P_i-busy period (period during which process i and all processes of higher priority than i are being executed) unless some higher priority process is released a second time. In general then, if j has higher priority than i, within the interval $[0, R_i)$, it will be released a number of times given by

$$\# \text{ of releases} = \left\lceil \frac{R_i}{T_j} \right\rceil$$

Each release generates an interference of C_j so that maximum interference from j is $\left\lceil \frac{R_i}{T_j} \right\rceil C_j$. Hence if task i has a critical instant at 0, then the maximum interference it experiences is given by the second term on the right hand side of (5.2).

Equation (5.2) is a nonlinear equation. If there are multiple solutions, R_i is given by the smallest solution. Equation (5.2) is usually solved by iteration

$$R_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^n}{T_j} \right\rceil C_j$$

$$R_i^0 = C_i$$

Since all terms are nonnegative, the sequence R_i is nondecreasing. If the iteration converges and the result is less than or equal to D_i, then the set of tasks is schedulable. If R_i^n keeps increasing as n increases until it is greater than D_i, then the set of tasks is not schedulable using the given fixed priority scheduler.

Response Time Analysis Example (as opposed to utilization based analysis)

	T	D	C	P	U
Task A	52	52	12	3	0.23
B	40	40	10	2	0.25
C	30	30	10	1	0.33

$$\begin{aligned} R_A^0 &= 12 \\ R_A^1 &= 12 + \left\lceil \frac{12}{T_B} \right\rceil C_B + \left\lceil \frac{12}{T_C} \right\rceil C_C \\ &= 12 + 10 + 10 = 32 \end{aligned}$$

$$R_A^2 = 12 + \left\lceil \frac{32}{40} \right\rceil 10 + \left\lceil \frac{32}{30} \right\rceil 10$$

$$= 12 + 10 + 20 = 42$$

$$R_A^3 = 12 + \left\lceil \frac{42}{40} \right\rceil 10 + \left\lceil \frac{42}{30} \right\rceil 10 = 52$$

$$\begin{aligned} R_A^4 &= 12 + \left\lceil \frac{52}{40} \right\rceil 10 + \left\lceil \frac{52}{30} \right\rceil 10 = 52 \\ \text{so that } R_A &= 52 \end{aligned}$$

.

$$\text{Similarly} \quad R_B^1 = R_B^2 = R_B = 20$$
$$R_C^0 = R_C = 10$$

Based on the response time analysis, the 3 tasks are schedulable. On the other hand, utilization analysis will give

$$U = \sum \frac{C_i}{T_i} = \frac{12}{52} + \frac{1}{4} + \frac{1}{3} = 0.814$$
$$\geq 3(2^{1/3} - 1) = 0.7798$$

so that the rate monotonic bound is not satisfied.

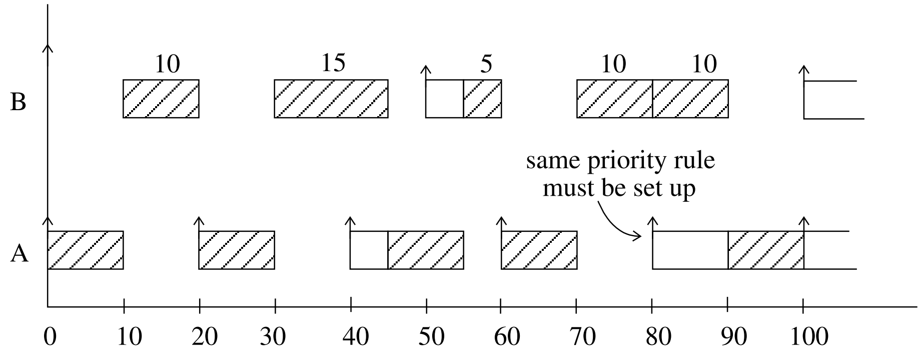
There exists a necessary and sufficient condition to schedule a task set under fixed priority scheduling for any priority assignment, and can be found in the reference

> M. Joseph and P. Pandya, "Finding response times in a real-time system," *The Computer Journal*, 29(5) pp.390-395, 1986.

5.1.4 Dynamic Scheduling: EDF scheduling

While no fixed priority scheduler can schedule the 2 tasks in Example 3, they can be scheduled by a dynamic priority scheduler. The optimal dynamic priority scheduler is the **Earliest Deadline First (EDF)** scheduler. The EDF scheduler is optimal in the sense that if it cannot schedule the tasks, then no other scheduler can. This scheduling algorithm checks, at task release points, **which one among the released tasks has the earliest deadline and schedules that task to be executed next**. The necessary and sufficient condition for schedulability under EDF scheduling is simply $U = \sum_i \frac{C_i}{T_i} \leq 1$.

As an illustration, for Example 3, the timing diagram is:



The pattern then repeats. Note that here the utilization $U = 1$, yet all tasks are schedulable under EDF even though there is no feasible FP schedule.

5.1.5 Process Interaction and Blocking

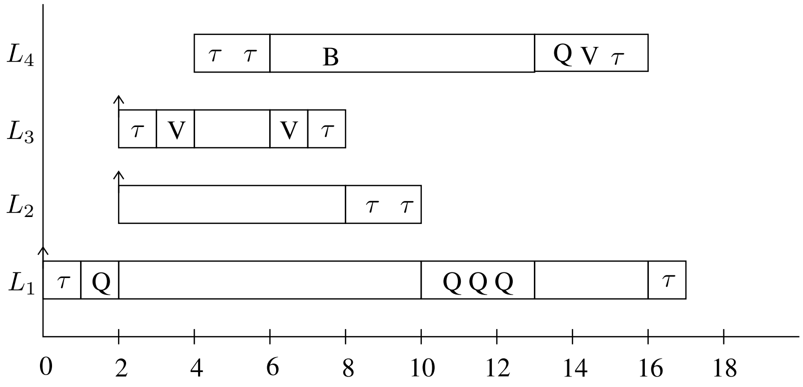
Processes interact. They can do so safely by using some form of protected shared data (such as semaphores). A process can therefore be suspended until some necessary future event has occurred.

This leads to the possibility of priority inversion, i.e. a higher priority process having to wait for a lower priority process (because the lower priority processes holds a resource that the higher priority one needs to proceed). In this case the higher priority process is blocked. It would be desirable to make blocking small and it must be bounded and measurable.

Example: 4 periodic processes (period not important here). L_1 and L_4 share the critical resource Q and V. Q and V also refer to one unit of execution time using resource Q and V, respectively. τ refers to one unit of regular execution time, not requiring resource Q or V. B denotes the period during which a process is blocked.

	P	Execution sequence	Release time
L_4	4	$\tau\tau$ Q V τ	4
L_3	3	τ V V τ	2
L_2	2	$\tau\tau$	2
L_1	1	τ Q Q Q Q τ	0

Time line is:



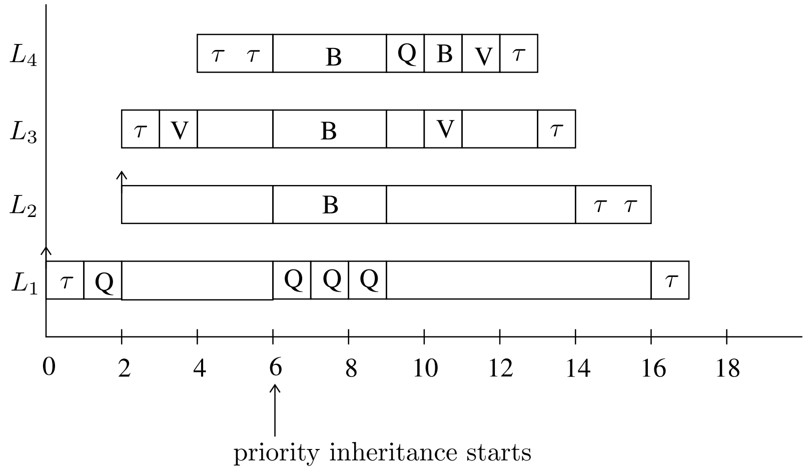
Response time of L_4 is 16 - 4 = 12

$$L_3 = 8 - 2 = 6$$
$$L_2 = 10 - 2 = 8$$
$$L_1 = 17$$

L_4, the highest priority process, gets blocked by L_1, L_2 and L_3.

One method of handling such a problem is to use **priority inheritance**: If a process p is suspended waiting for process q to execute, then the priority q becomes equal to p even if q initially has lower priority.

In example, L_1 will be given the priority of L_4. Time line becomes



Response time

$$\begin{aligned}
L_4 &= 13 - 4 = 9 \\
L_3 &= 14 - 2 = 12 \\
L_2 &= 16 - 2 = 14 \\
L_1 &= 17
\end{aligned}$$

Using response time analysis to perform scheduling is not easy to implement since it is not straightforward to identify the process which should have priority inheritance. Response time calculation now needs to be modified to

$$R_i = C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

where B_i is the maximum blocking time that process i can suffer due to priority inversion.

5.2 Scheduling in Real-Time Control Systems

We have only scratched the surface of the scheduling problem for real-time systems. The literature is huge. There are many more results on scheduling which deal with more complicated processor and computation models. For example, there are results which deal with sporadic processes as well as periodic processes, allow precedence relations between processes, and handle arbitrary deadlines. In this section, we briefly describe how one can use scheduling results to design real-time systems. The description is meant only to highlight the role of scheduling analysis. It does not imply that this is the standard procedure in designing real-time systems.

Conceptually, one can develop an appropriate real-time computer control design as follows.

I. Determine the timing specifications for all real-time processes. This involves a study of the requirements of the physical system and determining which tasks have critical timing requirements.

II. Decide on the hardware platform for the control system. This is often dictated by other constraints such as cost, physical size and weight, etc.

III. Develop real-time software which implements the control system tasks on the hardware platform chosen. This involves either using a programming language such as Ada which can handle real-time requirements, or using a real-time operating system.

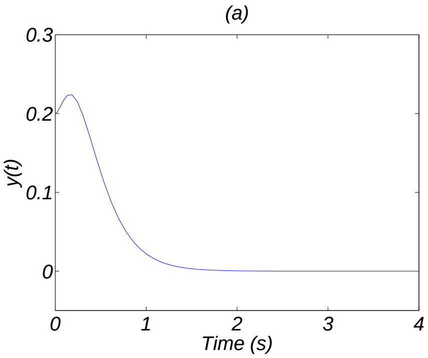
IV. Use timing analysis to estimate the worst case execution time for the various real-time processes. This involves estimating hardware-related times such as interrupt latencies and context switch times, as well as the execution time of the software implementation.

V. Apply scheduling analysis to see if all processes will meet their deadlines. If they do, a tentative design is achieved. Extensive testing and simulation is then done, sometimes supported by formal validation methods, to ensure that all the timing specifications are indeed met. If scheduling shows that not all deadlines are met, re-examine the hardware and software implementations. Iterate on this design process until satisfactory performance is reached.

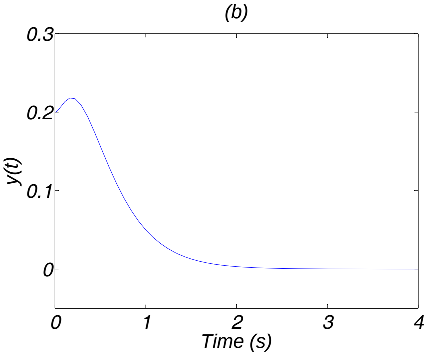
With some basic understanding of real-time scheduling in place, we can return to using TrueTime to examine how real-time scheduling affects digital controllers when the computer needs to schedule multiple tasks.

Example 4a In this example, there are three plants with state models: Think of three cart-pendulum systems. Assume state-feedback controllers have already been designed. Independent continuous-time step responses, all starting from $y(0) = 0.2$, are as follows:

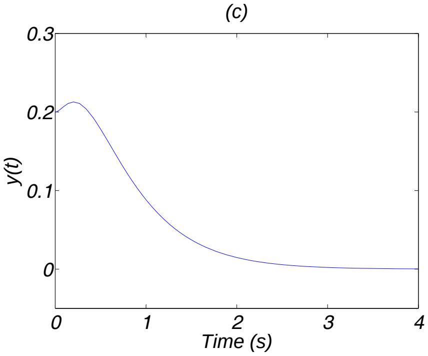
System 1



System 2

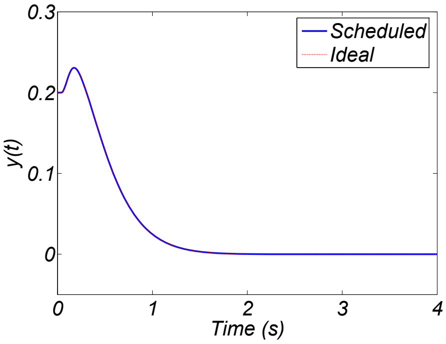


System 3

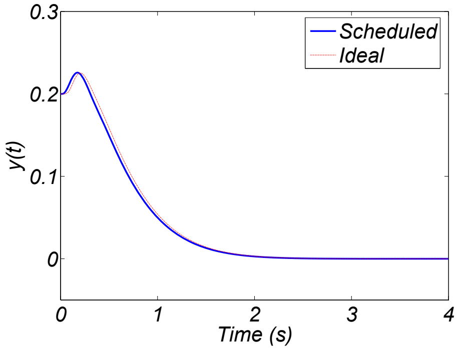


Suppose testing has shown that the control tasks τ_1, τ_2 and τ_3 require $C_1 = 7ms$, $C_2 = 11ms$, and $C_3 = 12ms$ to complete. To guarantee a certain level of performance, suppose we have selected the tasks' periods to be $T_1 = 32ms$, $T_2 = 41ms$ and $T_3 = 56ms$. This corresponds to $U \approx 0.7$. We use the RM scheduling policy. Therefore, $P_1 = 1$, $P_2 = 2$ and $P_3 = 3$. (This is the TrueTime convention: smaller number indicates higher priority. Other texts may do the exact opposite!)

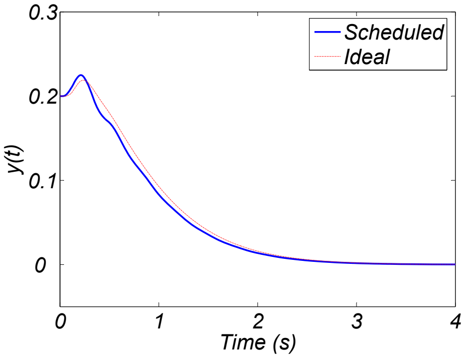
The TrueTime simulation results are shown in the following figures. The systems' ideal (independent) responses are plotted on the same graphs for comparison.
Systems 1:



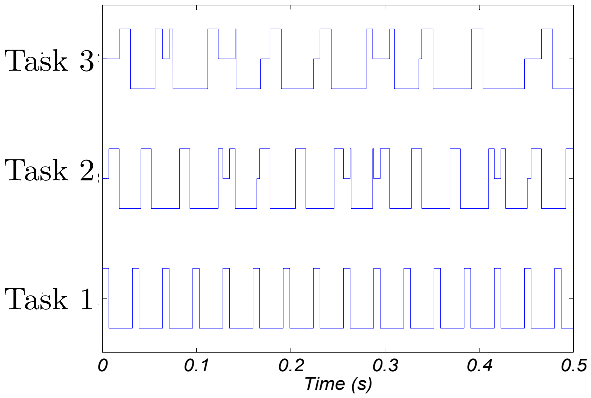
System 2:



System 3:

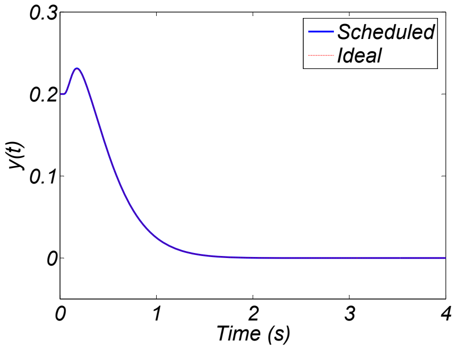


The task schedule is

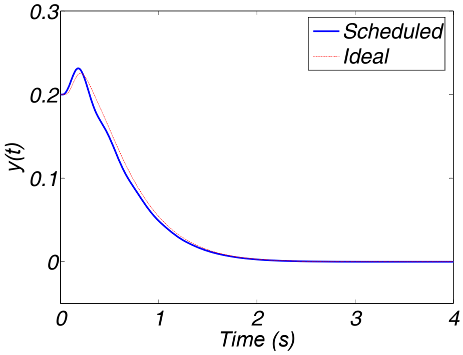


Notice that τ_1 has a periodic scheduling pattern because it has the highest priority—it is never preempted. Task 3 is preempted potentially by two other tasks at one time, giving it the most irregular scheduling pattern. □

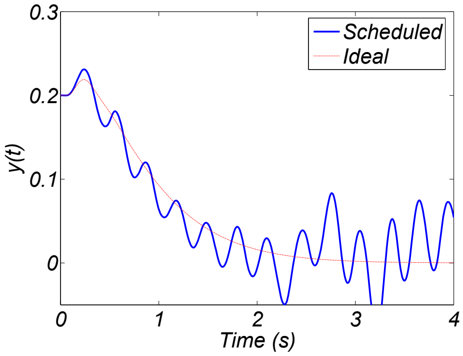
Example 4b Now suppose we have underestimated the execution times of the tasks; the times to execute τ_1, τ_2 and τ_3 are really $8ms, 15ms$ and $14ms$. The value of U is really 0.866. The simulations are now as follows.
System 1:



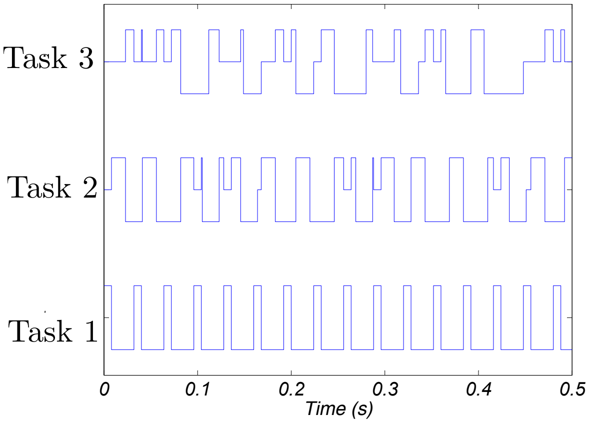
System 2:



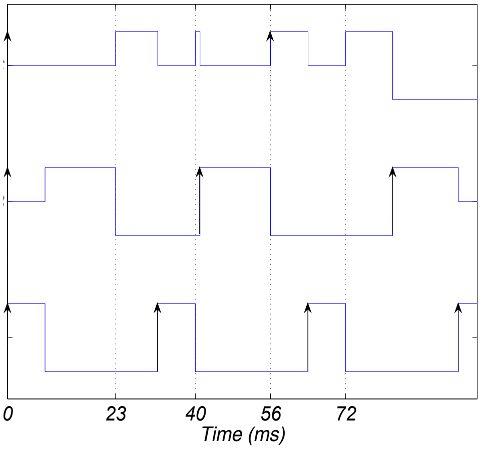
System 3:



Task schedule:



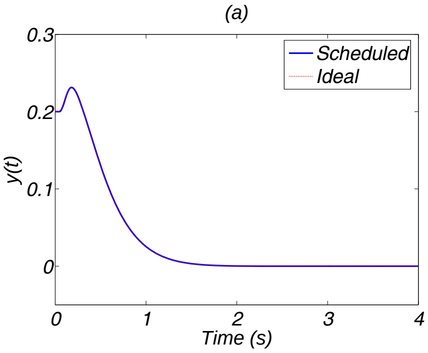
The response of System 3 is unacceptable. Let's take a closer look at the schedule:



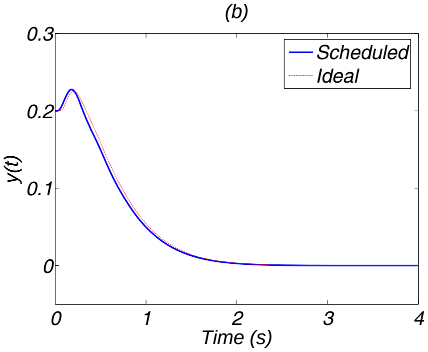
A release is indicated by an upwards arrow. In its first period, τ_3 has the opportunity to run for $9ms$ at $t = 23$, and then for $1ms$ at $t = 40$, before it is interrupted. It has no other chance to execute before its next period. Since the execution time of τ_3 is $14ms$, it misses its deadline at $t = 56ms$. We assume the task is therefore aborted.

□

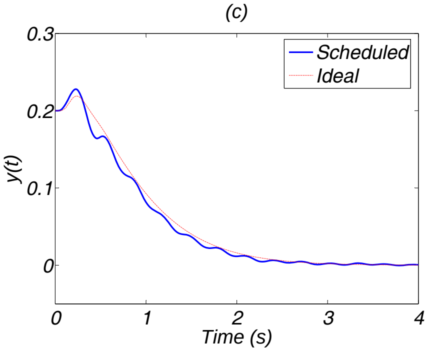
Example 4c The examples so far have used fixed-priority RM scheduling. Now we look at the system response under the dynamic scheduler **Earliest-Deadline-First** (EDF). Here, the priorities are dynamic: They change depending on their deadlines. The task with the shortest time to its next period is chosen for execution. The same systems as Example 3b were simulated under EDF scheduling.
System 1:



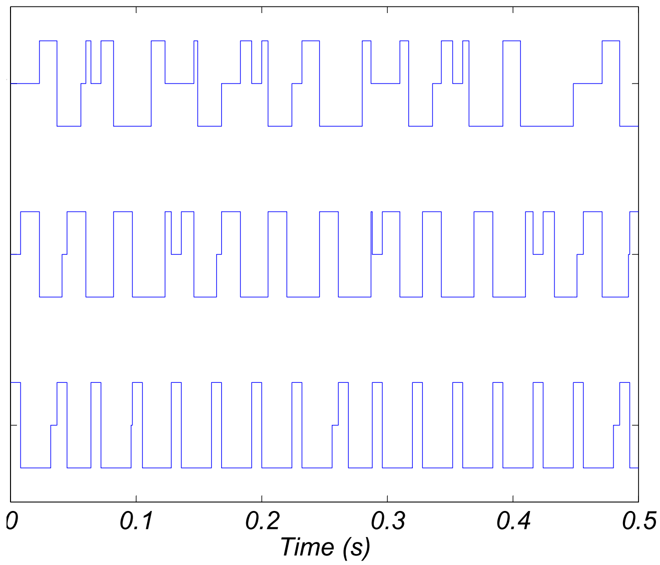
System 2:



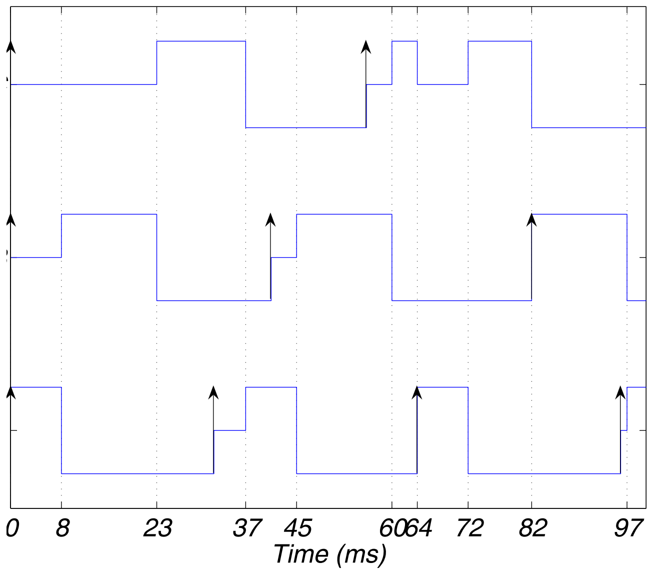
System 3:



Task schedule:



The response of System 3 is a lot better. Taking a closer look at the task schedule, we see the following:



At $t = 0$, all the tasks are released. Task 1 is executed because it has the shortest period, and hence the earliest deadline. Once τ_1 is completed, τ_2 has the chance to start because its period is next in line. At $t = 23ms$, τ_3 can start running and continues until finished. At $t = 32ms$, τ_3 is not preempted by τ_1, because its deadline is nearer. The task schedule for τ_1 is now slightly irregular because it may be preempted by the other tasks. This situation, of course, does not happen under FP scheduling (assuming rate-monotonic priority assignment). The response of System 3 has improved significantly because the processor, under EDF, can be utilized to a greater extent than under FP, as we will see. □

Recap of Example 4

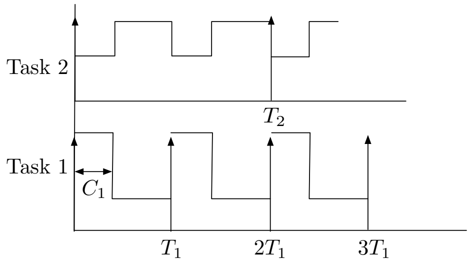
We simulated three systems under the rate-monotonic priority assignment, with what we believed were the correct execution times of the three control tasks, τ_1, τ_2 and τ_3. We chose the periods to yield a utilization factor of about 0.7, and the simulations of the three systems were satisfactory. However, it turned out that the actual execution times were slightly longer than expected. This small difference meant the utilization rose to about 0.87 and τ_3 missed its deadline. Indeed, in this case, the RM upper bound on the utilization is not satisfied. Under EDF scheduling, the performance of System 3 greatly improved, as all tasks are schedulable.

5.3 Analysis of Rate-Monotonic Scheduling

In this section we give an introduction to the theory of RM scheduling. Without further mention, all results are for that scheduling scheme.[2]

We look in detail at the analysis of two illustrative examples where there are just two tasks. The general case, even when there are just two tasks, is quite difficult and is examined in the Appendix to Chapter 5.

Example Let us fix T_1 and study the special case $T_2 = 2T_1$. The task schedule diagram looks like this:



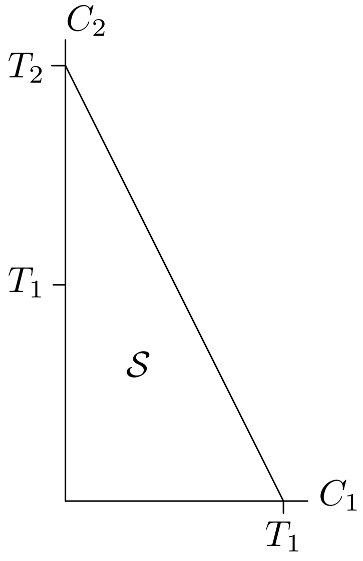
Let's accept that the worst case is when the two tasks are released simultaneously. The situation is very simple: Starting at $t = 0$, Task 2 must wait for Task 1 to finish, and then it is pre-empted once, at $t = T_1$. In an interval of length T_1, the amount of time available for Task 2 is $T_1 - C_1$.

The utilization in this case depends on C_1 and C_2 and we can write it as a function, $U(C_1, C_2)$. So

$$U(C_1, C_2) = \frac{C_1}{T_1} + \frac{C_2}{T_2} = \frac{C_1}{T_1} + \frac{C_2}{2T_1}.$$

Now let $\mathcal{S}$ denote the set of values (C_1, C_2) for which the tasks are schedulable. What does the set $\mathcal{S}$ look like in the (C_1, C_2)-plane? Clearly $(C_1, C_2) \in \mathcal{S}$ if and only if $C_2 \leq 2(T_1 - C_1)$, since $2(T_1 - C_1)$ is the amount of time available for Task 2 in an interval of length $T_2 = 2T_1$. Thus $\mathcal{S}$ is the set that lies on or below the line $C_2 = 2(T_1 - C_1)$:

[2] Reference for this section: C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," Journal of the Association for Computing Machinery, January 1973, pp. 46-61.



Now we would like to find an upper bound UB such that

$$(C_1, C_2) \in \mathcal{S} \iff U(C_1, C_2) \leq UB.$$

From the formula

$$U(C_1, C_2) = \frac{C_1}{T_1} + \frac{C_2}{2T_1},$$

it follows that $U(C_1, C_2) \leq UB$ iff

$$\frac{C_1}{T_1} + \frac{C_2}{2T_1} \leq UB.$$

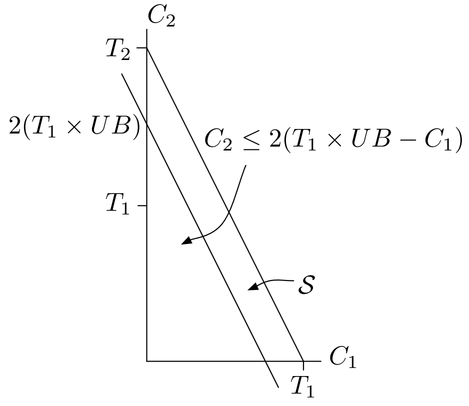
Equivalently,

$$C_2 \leq 2(T_1 \times UB - C_1).$$

The inequality

$$C_2 \leq 2(T_1 \times UB - C_1)$$

defines the region shown here:



Thus UB should be chosen so that the two regions coincide: $UB = 1$.

In conclusion, when $T_2 = 2T_1$, the task set is schedulable iff $U \leq 1$. □

In general it can be shown that if T_2 is an integer multiple of T_1, then $U \leq 1$ is the necessary and sufficient condition for schedulability. In fact, Example III in Section 5.1.2 is an illustration of this for 3 tasks.

The next example is a little more complicated.

Example Let us fix T_1 and study the special case $T_2 = 1.5T_1$. Let us first see that $U \leq 1$ is not the right test for schedulability. A counterexample is

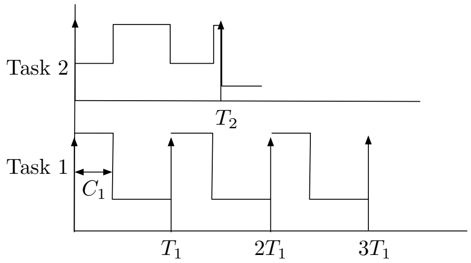
$$T_1 = 1,\ C_1 = 0.5,\ T_2 = 1.5,\ C_2 = 0.75.$$

Then $U = 1$ but the tasks are not schedulable; draw the task schedule diagram.

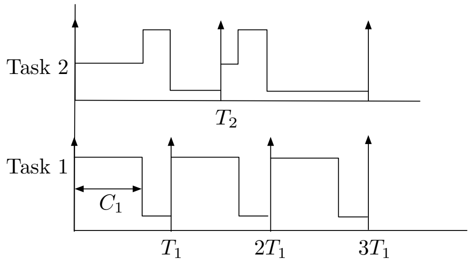
To find a valid upper bound, again we can write

$$U(C_1, C_2) = \frac{C_1}{T_1} + \frac{C_2}{T_2} = \frac{C_1}{T_1} + \frac{C_2}{1.5T_1}.$$

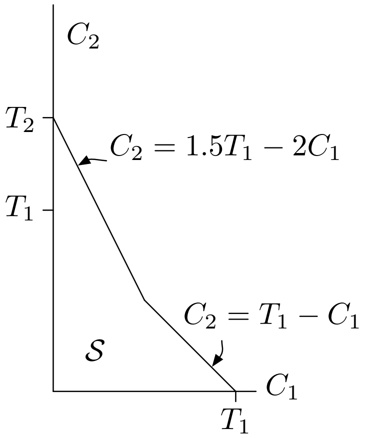
There are two situations. First, if $C_1 \leq T_2 - T_1 = 0.5T_1$, the schedule diagram is this:



The timing diagram shows that at T_1, another Task 1 is released and then completed before $T_2 = 1.5T_1$. In an interval of length $1.5T_1$, $2C_1 \leq T_1$ is consumed by Task 1. So $(C_1, C_2) \in \mathcal{S}$ iff $C_2 \leq 1.5T_1 - 2C_1$. On the other hand, if $C_1 > 0.5T_1$, the situation is this:



Here, $T_1 + C_1 > 1.5T_1 = T_2$, so that $T_1 - C_1$ is the amount of time available for completing Task 2 before the next release of Task 1. So $(C_1, C_2) \in \mathcal{S}$ iff $C_2 \leq T_1 - C_1$. Thus $\mathcal{S}$ is this set:



Now we would like to find an upper bound UB such that

$$(C_1, C_2) \in \mathcal{S} \iff U(C_1, C_2) \leq UB.$$

From the formula

$$U(C_1, C_2) = \frac{C_1}{T_1} + \frac{C_2}{1.5T_1},$$

it follows that $U(C_1, C_2) \leq UB$ iff

$$\frac{C_1}{T_1} + \frac{C_2}{1.5T_1} \leq UB.$$

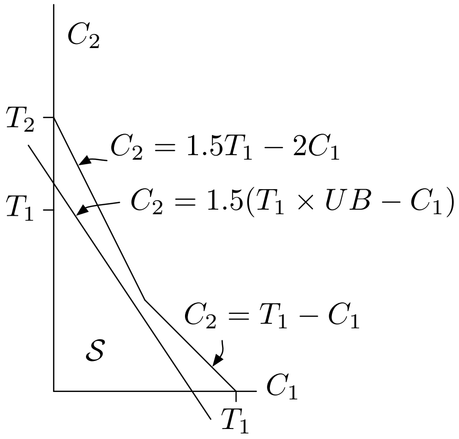
Equivalently,

$$C_2 \leq 1.5(T_1 \times UB - C_1).$$

The inequality

$$C_2 \leq 1.5(T_1 \times UB - C_1)$$

defines the following region:



The 2 lines intersect at the point such that

$$T_1 - C_1 = 1.5T_1 - 2C_1$$

This shows that the point of intersection is at $(C_1, C_2) = (0.5T_1, 0.5T_1)$. So we see there is no necessary and sufficient UB, only a sufficient UB, when the line meets the corner. This UB can be solved from

$$\frac{1}{2}T_1 = 1.5\left(T_1 \times UB - \frac{1}{2}T_1\right).$$

The solution is $UB = 5/6 \approx 0.83333$.

In conclusion,when $T_2 = 1.5T_1$, the task set is schedulable if $U \leq 5/6$. □

The analysis so far uses the explicit relationship between T_2 and T_1. The general case where T_2 can be any number $> T_1$ is substantially more difficult and involves a different proof technique. The details are provided in the Appendix to Chapter 5.

In Example 2, we saw that the tasks met their deadlines. We can verify this using Theorem 1: The task set consisted of two tasks, thus

$$\begin{aligned} UB &= 2(2^{1/2} - 1) \\ &\approx 0.83 \end{aligned}$$

The utilization was only 0.7, confirming that the set was schedulable.

Whether or not all tasks will meet their deadlines under EDF scheduling is determined fully by the utilization. A task set is schedulable under EDF if and only if $U \leq 1$. This EDF schedulability test is much simpler than the FP test. Also notice that, under EDF, the CPU can be utilized 100% of the time, as shown in Example 3 when EDF scheduling is used.

5.4 Exercises

1. Consider a real-time scheduling problem with task-set data

task τ	execution time C	period T
1	1	4
2	2	6
3	1	10

(a) Is the task set schedulable by rate-monotonic fixed-priority scheduling?

(b) Draw the timeline plot.

2. Ali and Chuan have to move 16 small boxes and 10 large boxes. The table indicates the time that each person takes to move each type of box. The goal is to schedule the box moving so that all the boxes are moved in minimum time.

	Ali	Chuan
small box	2 min.	3 min.
large box	6 min.	5 min.

The optimal schedule is obvious: Ali should move all the small boxes; meanwhile, Chuan should move big boxes; when Ali has finished the small boxes, he should help Chuan. Study this problem as follows:

(a) Let m and n denote respectively the numbers of small and large boxes moved by Ali. Thus

$$0 \leq m \leq 16, \quad 0 \leq n \leq 10.$$

Draw this region in the (n, m) plane.

(b) Let T denote the time it takes to move all 26 boxes. Then Ali takes less than or equal to time T:

$$2m + 6n \leq T.$$

Draw this region in the (n, m) plane.

(c) Likewise, Chuan takes less than or equal to time T:

$$3(16 - m) + 5(10 - n) \leq T.$$

Draw this region in the (n, m) plane.

(d) Thus we arrive at the problem: minimize T such that there exists a point (n, m) (i.e., a feasible point) satisfying the constraints

$$0 \leq m \leq 16, \quad 0 \leq n \leq 10, \quad 2m + 6n \leq T, \quad 3(16 - m) + 5(10 - n) \leq T.$$

By studying the shape of the feasible set, conclude that the optimal m is 16.

(e) Why is this problem different from the one in Chapter 5?

3. Consider the problem of scheduling two tasks according to the rate-monotonic fixed priority schedule. Take $T_1 = 1,\ T_2 = 1.6,\ C_1 = 0.6$.

   (a) What is the maximum C_2 for which the tasks are schedulable?

   (b) Draw the task schedule plot from $t = 0$ to $t = 4$ for the maximum C_2 and assuming the two tasks are both released at $t = 0$.

4. Now take $T_1 = 1$, $T_2 = 1.8$, C_1 not fixed. Derive an upper bound UB such that the tasks are schedulable if the utilization satisfies $U \leq UB$. (UB should be larger than $2(\sqrt{2} - 1)$, which is the upper bound when T_2 is not fixed.)

5.5 Appendix: Rate-Monotonic Scheduling of 2 Tasks - The General Case

Theorem 11 *Under the fixed-priority scheduling algorithm with the rate-monotonic priority assignment, if $U \leq 2(2^{1/2}-1)$ then a task set consisting of two tasks is schedulable.*

Proof It is convenient to fix T_1 and to consider only the case $T_2 > T_1$, so that $P_1 > P_2$. Then U is a function of T_2, C_1, C_2 and we write

$$U(T_2, C_1, C_2).$$

We consider only T_2, C_1, C_2 for which the tasks are schedulable. Then for each T_2, C_1, there is a maximum C_2 such that the task set is schedulable. Denote by $U^*(T_2, C_1)$ the utilization for this C_2, i.e.,

$$U^*(T_2, C_1) = \max_{C_2} U(T_2, C_1, C_2).$$

Also define

$$UB = \min_{T_2, C_1} U^*(T_2, C_1).$$

We will first show that any task set with $U \leq UB$ is schedulable. For a proof by contradiction, consider a task set with parameters T_2', C_1', C_2' (remember that T_1 is fixed), and suppose $U(T_2', C_1', C_2') \leq UB$ but the task set is not schedulable. We can certainly reduce C_2' just enough, to some C_2'', so that the tasks are schedulable. That is, C_2'' is the maximum feasible execution time for Task 2 for the parameters T_2', C_1'. Then

$$U(T_2', C_1', C_2'') < U(T_2', C_1', C_2') \leq UB.$$

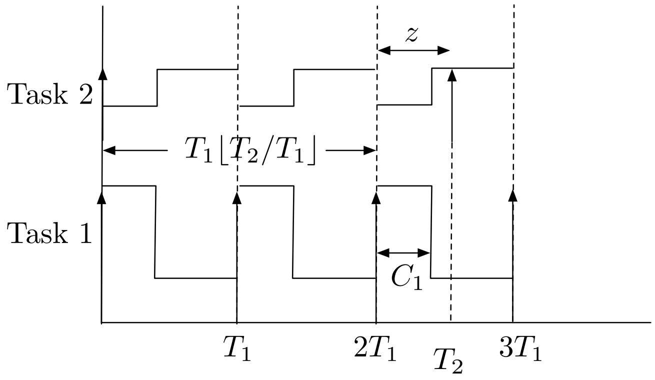
But $UB \leq U^*(T_2', C_1') = U(T_2', C_1', C_2'')$, a contradiction.

We now proceed to find the minimum possible $U^*(T_2, C_1)$ over all T_2, C_1. We will minimize first over C_1, then over T_2.

Fix T_2 and define $z := T_2 - T_1\lfloor T_2/T_1 \rfloor$. There are two sub-cases to consider:

Case 1: $C_1 \leq z$

For example, if Task 1 is released three times during T_2, the picture is



Since C_2 is maximum,

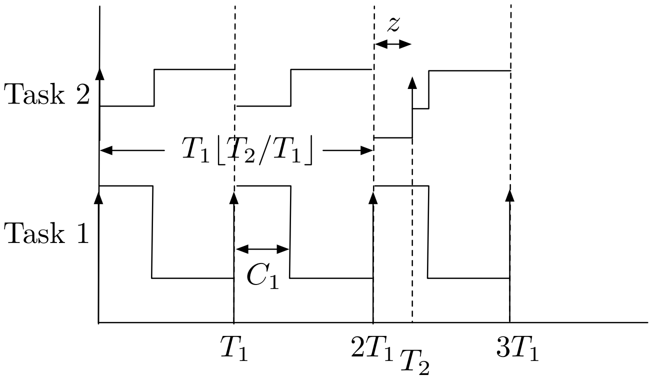
$$C_2 = T_2 - C_1\lceil T_2/T_1\rceil.$$

Thus

$$\begin{aligned}
U^*(T_2, C_1) &= \frac{C_1}{T_1} + \frac{C_2}{T_2} \\
&= \frac{C_1}{T_1} + \frac{T_2 - C_1\lceil T_2/T_1\rceil}{T_2} \\
&= 1 + C_1\left(\frac{1}{T_1} - \frac{\lfloor T_2/T_1\rfloor + 1}{T_2}\right) \\
&= 1 - C_1\left(\frac{T_1 - z}{T_1T_2}\right).
\end{aligned}$$

Thus as C_1 increases from 0 to z, $U^*(T_2, C_1)$ decreases monotonically.

Case 2: $C_1 > z$
The picture is like this:



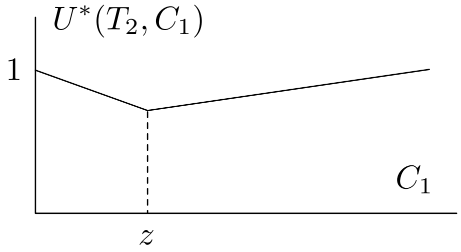
In this case

$$C_2 = (T_1 - C_1)\lfloor T_2/T_1\rfloor$$

and

$$\begin{aligned}
U^*(T_2, C_1) &= \frac{C_1}{T_1} + \frac{C_2}{T_2} \\
&= \frac{C_1}{T_1} + \frac{(T_1 - C_1)\lfloor T_2/T_1\rfloor}{T_2} \\
&= \frac{T_1}{T_2}\lfloor T_2/T_1\rfloor + C_1\left(\frac{1}{T_1} - \frac{\lfloor T_2/T_1\rfloor}{T_2}\right) \\
&= \frac{T_1}{T_2}\lfloor T_2/T_1\rfloor + C_1\left(\frac{z}{T_1T_2}\right).
\end{aligned}$$

Now $U^*(T_2, C_1)$ increases monotonically as C_1 increases from z to T_1:



The minimum of $U^*(T_2, C_1)$ over C_1 occurs when the two formulas are equal:

$$1 - C_1\left(\frac{T_1 - z}{T_1T_2}\right) = \frac{T_1}{T_2}\lfloor T_2/T_1\rfloor + C_1\left(\frac{z}{T_1T_2}\right),$$

i.e.,

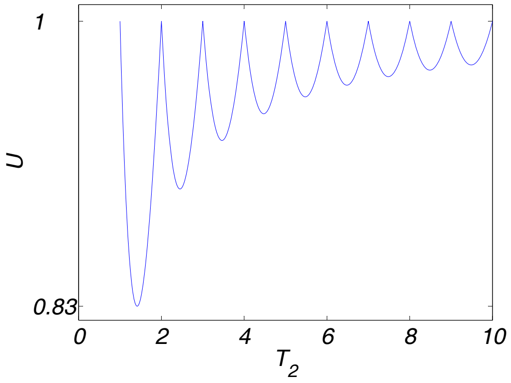
$$C_1 = z = T_2 - T_1\lfloor T_2/T_1\rfloor.$$

Then

$$\begin{aligned} U^*(T_2, C_1) &= 1 - z\left(\frac{T_1 - z}{T_1T_2}\right) \\ &= 1 - \frac{(T_2 - T_1\lfloor T_2/T_1\rfloor)(T_1 - T_2 + T_1\lfloor T_2/T_1\rfloor)}{T_1T_2} \\ &= 1 - \frac{(T_2 - T_1\lfloor T_2/T_1\rfloor)(T_1\lceil T_2/T_1\rceil - T_2)}{T_1T_2}. \end{aligned}$$

It remains to minimize this expression with respect to T_2.

The following is a plot of U^* versus T_2 with $T_1 = 1$:



For example, take $T_1 = 1$ and $T_2 = x$. Then

$$U^*(x) = 1 - \frac{(x - \lfloor x\rfloor)(\lceil x\rceil - x)}{x}.$$

From the graph, U^* is minimum over $1 < x < 2$, in which case

$$U^*(x) = 1 - \frac{(x-1)(2-x)}{x} = x - 2 + \frac{2}{x}.$$

The minimizing x is $\sqrt{2}$ and then

$$UB = 2(2^{1/2} - 1) \approx 0.83.$$

In general, a set of n tasks is schedulable under Rate-Monotonic scheduling if

$$U \leq n(2^{1/n} - 1).$$

□

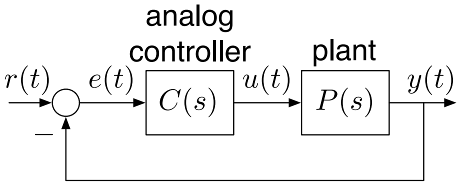
Chapter 6

Discretization of Continuous-Time Controllers

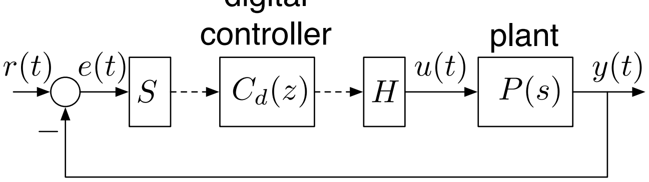
In Chapter 3, we studied in detail the design of digital controllers using state space methods in discrete time. Now we look at the other main approach: A continuous-time controller is already in place that satisfies the performance specifications. A reasonable approach in this case is to use a discrete approximation of the continuous-time controller, expecting that this should lead to a good digital controller.

6.1 Introduction

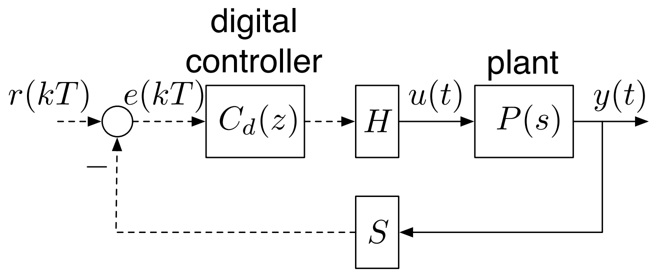
Suppose we have designed an analog controller for the unity-feedback loop



From the analog controller $C(s)$ we want now to select a sampling period T and a digital controller $C_d(z)$:



Or the configuration might be



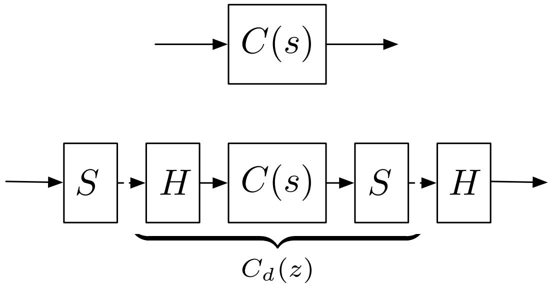
6.2 Discrete Approximations

Since the implicit assumption is that the continuous-time controller performs satisfactorily, the discretization will be constructed to preserve certain properties of the continuous-time controller. There are two common types of discretization.

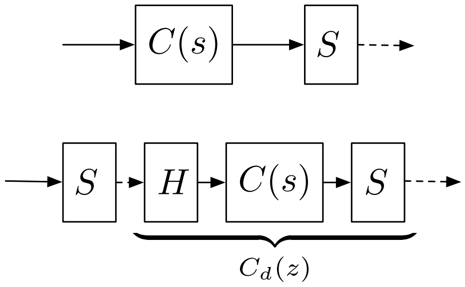
1. Matching the response characteristics of the continuous controller. These include the impulse-invariance method and the **step-invariance method** (same as a zero-order hold).
2. Approximating the controller transfer function using the **bilinear transformation**.

Step-Invariance Method

The simplest way to approximate $C(s)$ is like this:



That is, C_d equals `c2d` applied to C. In effect, we simply hold the sampled signal, put in the continuous-time controller, and sample the output from the controller. Why is this called the "step-invariance method?" These two systems have the same step response:



(Because HS is the identity system on steps.)

As we saw in Chapter 2, when we compute $C_d(z)$ from $C(s)$ by the step-invariance method, the poles of $C(s)$ are mapped to the poles of $C_d(z)$ by the mapping $z = e^{sT}$. Unfortunately, there is no simple relationship between the zeros of $C(s)$ and those of $C_d(z)$. Unstable zeros of $C(s)$ can be mapped to stable zeros of $C_d(z)$. In fact, there may even be no zeros present in $C(s)$, yet $C_d(z)$ may have zeros in $|z| \geq 1$.

Example Suppose

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

For $T = 0.1$, we find, using the MATLAB `c2d` command, that $C_d(z)$ has 3 poles, at 0.9048 and $0.9477 \pm 0.0823j$, but also 2 zeros, at -3.549 and -0.255, when the continuous-time controller has none.

□

The above results mean that there is no simple way to see quantitatively what discretization does to the properties of the continuous-time transfer function. Therefore, if we have a continuous-time controller $C(s)$ which works satisfactorily, we should just try to use a discretization method which preserves reasonably well the qualitative properties of $C(s)$, and hope that the resulting digital controller will also perform satisfactorily. Unlike direct discrete-time design, its performance is not guaranteed, so we always need to verify the properties of the closed loop system using simulation.

Bilinear Transformation

This method is motivated by considering the trapezoidal approximation of an integrator. Consider an integrator—transfer function $1/s$, input $u(t)$, and output $y(t)$. The trapezoidal approximation of

$$y(kT+T) = y(kT) + \int_{kT}^{kT+T} u(\tau)d\tau$$

is

$$y(kT+T) = y(kT) + \frac{T}{2}[u(kT+T) + u(kT)].$$

The transfer function of the latter equation is

$$\frac{T}{2}\frac{z+1}{z-1}.$$

This motivates the bilinear transformation

$$\frac{1}{s} = \frac{T}{2}\frac{z+1}{z-1},$$

that is,

$$s = \frac{2}{T}\frac{z-1}{z+1}.$$

So a continuous-time transfer function $C(s)$ is mapped into $C_d(z)$ where

$$C_d(z) = C\left(\frac{2}{T}\frac{z-1}{z+1}\right).$$

Example

$$C(s) = \frac{a}{s+a}$$

$$\begin{aligned} C_d(z) = \frac{a}{\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} + a} &= \frac{aT(1+z^{-1})}{2 - 2z^{-1} + aT + aTz^{-1}} \\ &= \frac{aT(1+z^{-1})}{2 + aT + (aT-2)z^{-1}} \end{aligned}$$

We can see how the bilinear transformation maps poles and zeros of $C(s)$ into those of $C_d(z)$: The mapping from s to z is given by

$$z = \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s}.$$

Let $s = \sigma_a + j\omega_a$. In terms of σ_a and ω_a, we get

$$z = \frac{1 + \frac{T}{2}\sigma_a + \frac{T}{2}j\omega_a}{1 - \frac{T}{2}\sigma_a - \frac{T}{2}j\omega_a}$$

Hence the absolute value of z is given by

$$|z| = \frac{(1 + \frac{T}{2}\sigma_a)^2 + (\frac{T}{2}\omega_a)}{(1 - \frac{T}{2}\sigma_a)^2 + (\frac{T}{2}\omega_a)}$$

We see that

$$\begin{aligned} \sigma_a < 0 &\iff |z| < 1 \\ \sigma_a = 0 &\iff |z| = 1 \end{aligned}$$

This shows that the left half s-plane is mapped onto the interior of the unit circle in the z-plane, the right half s-plane is mapped onto the exterior of the unit circle in the z-plane, and the imaginary axis is mapped onto the unit circle. In particular, if $C(s)$ has no poles or zeros in the right half-plane, then $C_d(z)$ has none outside the unit disk. To study the mapping of the imaginary axis a bit further, we represent the unit circle in the z-plane in the form of $e^{j\omega T}$, where ω is interpreted as discrete frequency. The mapping of the imaginary axis can then be interpreted as mapping of the analogue frequency ω_a to the discrete frequency ω. The precise relationship between the frequencies is given by

$$\begin{aligned} j\omega_a &= \frac{2}{T}\frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} = \frac{2}{T}\frac{e^{\frac{j\omega T}{2}} - e^{\frac{j\omega T}{2}}}{e^{\frac{j\omega T}{2}} + e^{\frac{j\omega T}{2}}} \\ &= \frac{2j}{T}\tan\frac{\omega T}{2} \end{aligned}$$

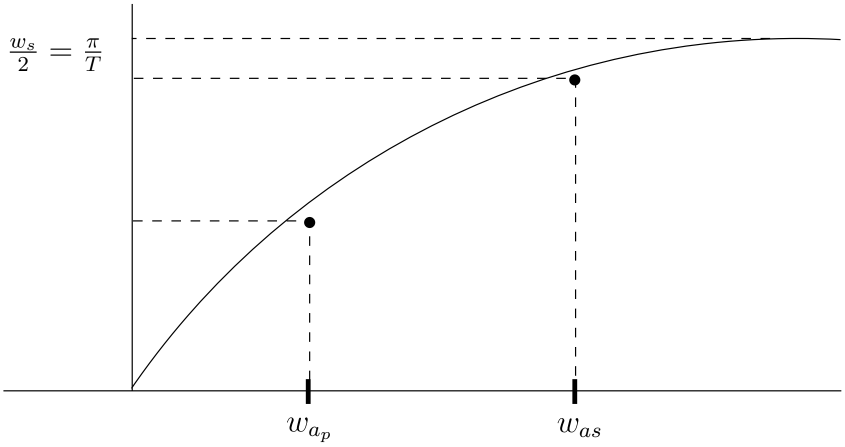
i.e.,

$$\omega_a = \frac{2}{T}\tan\frac{\omega T}{2}$$

Equivalently, we can also write

$$\omega = \frac{2}{T}\tan^{-1}\frac{\omega_a T}{2}$$

Note that there is no aliasing of frequencies as each ω_a is mapped into a different value of ω. For small values of ω_a, $\tan^{-1}\frac{\omega_a T}{2} \approx \frac{\omega_a T}{2}$ so that $\omega_a \approx \omega$. However, as $\omega_a \to \infty$, the discrete frequency $\omega \to \frac{\omega_s}{2} = \frac{\pi}{T}$ so that high frequencies are compressed near $\frac{\omega_s}{2}$. This property of the bilinear transformation is often referred to as frequency warping. The compression phenomenon is illustrated in the following figure.



Compression of High Frequencies under the Biliear Transformation

For analogue frequencies up to ω_{ap}, the frequency mapping is almost linear. For frequencies higher than ω_{as}, the frequencies are heavily compressed. Under the bilinear transformation, the frequency response of $C_d(z)$ is given by

$$\begin{aligned} C_d(e^{j\omega T}) &= C(\frac{2}{T}\frac{e^{j\omega T}-1}{e^{j\omega T}+1}) \\ &= C(j\frac{2}{T}tan\frac{\omega T}{2}) \end{aligned}$$

It can be derived that if (A, B, C, D) is a continuous-time state model for $C(s)$, then a discrete-time state model for $C_d(z)$, obtained by the bilinear transformation, is

$$A_d = \left(I - \frac{T}{2}A\right)^{-1}\left(I + \frac{T}{2}A\right), \quad B_d = \frac{T}{2}\left(I - \frac{T}{2}A\right)^{-1}B$$

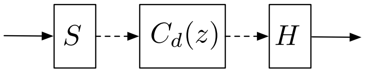
$$C_d = C(I + A_d), \quad D_d = D + CB_d.$$

This state formula is valid provided the indicated inverse exists, that is, $2/T$ is not an eigenvalue of A.

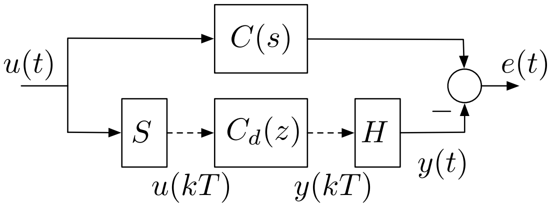
Discretization Error

Now we have two ways to discretize a continuous-time system, step-invariance transformation and bilinear transformation. Which is better? Indeed, how can we judge how good a discretization is? We now study a way to answer this.

Suppose $C(s)$ is a transfer function of an analog system and $C_d(z)$ is a discretization. We cannot talk about the error between $C(s)$ and $C_d(z)$ just as we cannot compare apples and oranges. We have to put the systems in the same domain. To compare them in the continuous-time domain, it makes sense to compare $C(s)$ and this continuous-time system:



In this way we are led to the **error system**:



We are going to study the error system in the frequency domain. For this, we need to know the relationship between the Fourier transforms $E(j\omega)$ and $U(j\omega)$.

Caution The error system is time-varying, so it has no transfer function!

Lemmas 1 and 2 in Chapter 3 allow us to write the frequency-domain relationships as follows:

$$U_d\left(\mathrm{e}^{j\omega T}\right) = \frac{1}{T} U_e(j\omega),$$

where $U_e(j\omega)$ is the periodic extension of $U(j\omega)$;

$$Y_d\left(\mathrm{e}^{j\omega T}\right) = C_d\left(\mathrm{e}^{j\omega T}\right) U_d\left(\mathrm{e}^{j\omega T}\right);$$

$$Y(j\omega) = TR(j\omega) Y_d\left(\mathrm{e}^{j\omega T}\right),$$

where $R(s) = (1 - \mathrm{e}^{-sT})/(sT)$ is the transfer function associated with the hold. Thus

$$Y(j\omega) = R(j\omega) C_d\left(\mathrm{e}^{j\omega T}\right) U_e(j\omega).$$

Thus in the error system

$$E(j\omega) = C(j\omega)U(j\omega) - R(j\omega) C_d\left(\mathrm{e}^{j\omega T}\right) U_e(j\omega).$$

Now assume $U(j\omega)$ is bandlimited to frequencies less than the Nyquist frequency ω_N, that is,

$$U(j\omega) = 0 \text{ for } \omega \geq \omega_N.$$

Then there's no aliasing:

$$U_e(j\omega) = U(j\omega) \text{ for } \omega < \omega_N.$$

So for $\omega < \omega_N$

$$E(j\omega) = \left[C(j\omega) - R(j\omega)C_d\left(\mathrm{e}^{j\omega T}\right)\right]U(j\omega).$$

This motivates the definition of the **error function**,

$$error(\omega) := \left|C(j\omega) - R(j\omega)C_d\left(\mathrm{e}^{j\omega T}\right)\right|,$$

and the **maximum error**,

$$error_{max} := \max_{\omega<\omega_N} error(\omega).$$

Recap

$C(s)$ is a given continuous-time system and $C_d(z)$ is a discretization of $C(s)$ obtained in some way; for inputs that are bandlimited to frequencies less than ω_N, $error_{max}$ is a measure of how closely $C_d(z)$ approximates $C(s)$ in the continuous-time domain.

It is natural to use this measure to compare two different discretizations to see which is "better."

Error for the Bilinear Transformation

Recall

$$C_d\left(\mathrm{e}^{j\omega T}\right) \quad = \quad C\left(j\frac{2}{T}\tan\frac{\omega T}{2}\right).$$

Thus

$$error(\omega) := \left|C(j\omega) - R(j\omega)C\left(j\frac{2}{T}\tan\frac{\omega T}{2}\right)\right|.$$

The error is therefore due to two factors: the presence of the hold function

$$R(j\omega) = \frac{1-\mathrm{e}^{-j\omega T}}{j\omega T};$$

the "frequency warping"

$$\omega \mapsto \frac{2}{T}\tan\frac{\omega T}{2}.$$

Note that $error(0) = 0$, that is, there is no error at DC.

Because of the frequency distortion, sometimes the so-called prewarping is used to reduce distortion over a frequency range of interest. For this, we use the following form for the bilinear transformation

$$s = c\frac{z-1}{z+1},$$

or, in terms of frequency response,

$$\omega_a = c \tan \frac{\omega T}{2}$$

where the constant c is chosen to map an analog frequency point to a digital frequency point. To illustrate, consider the following example.

Example 6.2:

$$C(s) = \frac{a}{s+a}$$

If we want to preserve bandwidth after applying the bilinear transformation, we can choose c so that the analog frequency point $\omega_a = a$ is mapped to the digital frequency point $\omega = a$ also. This gives

$$c = \frac{a}{\tan \frac{aT}{2}}$$

After a little bit of algebra, we get

$$C_d(z) = \frac{\tan \frac{aT}{2}}{\frac{z-1}{z+1} + \tan \frac{aT}{2}}$$

$$\begin{array}{rrcl} \text{Note} & |C(ja)| & = & \dfrac{a}{\sqrt{a^2+a^2}} = \dfrac{1}{\sqrt{2}} = 0.707 = -3\,db \\ & |C_d(e^{j\omega T})| & = & \dfrac{\tan \frac{aT}{2}}{j \tan \frac{\omega T}{2} + \tan \frac{aT}{2}} \\ \text{At} & \omega = a, & & |C_d(e^{jaT})| = \left| \dfrac{1}{1+j} \right| \\ & & = & \dfrac{1}{\sqrt{2}} = |C(ja)| \end{array}$$

Once again, if the frequencies are small, prewarping is not necessary.

We can now summarize the reasons why the bilinear transformation is so commonly used to discretize a continuous controller:

(a) It preserves stability properties of $C(s)$.

(b) It preserves the DC-gain of $C(s)$ ($s = 0$ is mapped to $z = 1$).

(c) For small values of frequency, the frequency response is approximately preserved.

(d) It is easy to compute $C_d(z)$ from $C(s)$.

(e) There is no aliasing although high frequencies are warped. Since most control systems operate at relatively low frequencies, frequency warping is usually not a problem.

6.3 Design Based on Approximating Continuous-Time Controllers

Now we turn to the following digital control design procedure:

1. Design a continuous-time controller $C(s)$, if one does not already exist, to achieve the required design specifications.

2. Approximate $C(s)$ using one of the discretization procedures to give $C_d(z)$. We consider only the bilinear transformation in these notes.

3. Simulate the closed-loop sampled-data system.

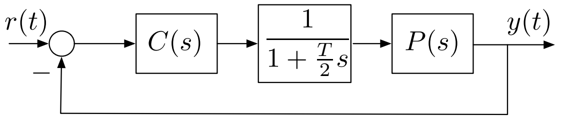
It is often useful to add a refinement in the design of the continuous-time controller. We know that the digital controller will contain sample and hold operators. In fact, the discretization error analysis shows that the controller approximation error is given by $|C(j\omega) - R(j\omega)C_d(e^{j\omega T})|$. We can partially account for their effects by including as part of the plant for the continuous-time controller design an approximation of the sample/hold transfer function. We then only need to approximate $C(s)$ well by $C_d(z)$, for example, using the bilinear transformation. A suitable approximation is based on the first-order Padé approximation for e^{-sT}, which is given by

$$\mathrm{e}^{-sT} \approx \frac{1 - \frac{T}{2}s}{1 + \frac{T}{2}s}.$$

Based on this approximation, the sample/hold operator transfer function would be approximated by

$$R(s) = \frac{1 - e^{-sT}}{sT} \approx \frac{1}{1 + \frac{T}{2}s}.$$

To recap, we shall design the continuous-time controller $C(s)$ to satisfy the design specifications for the augmented plant



Note that this is purely part of the procedure to get a digital controller through approximating a continuous-time controller. Once the digital controller has been determined, the sample/hold approximation plays no further role in the design. Including the approximation of the hold operator for designing the continuous time controller just usually results in a digital controller that performs better.

Example

Let the plant be $P(s) = \frac{1}{s(s+2)}$ and suppose the specifications are

1. feedback stability

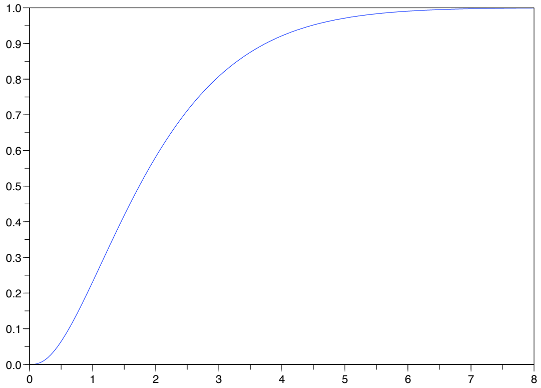
2. step tracking

3. percentage overshoot $\leq 5\%$
4. 2% settling time $T_s \leq 3$ seconds.

Suppose we happen to want $T = 0.2$ for hardware reasons. The augmented plant is

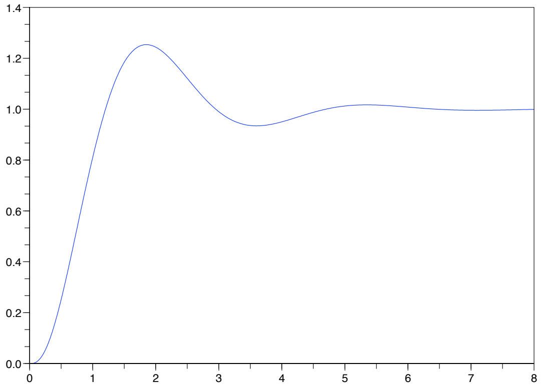
$$P_a(s) = \frac{10}{s(s+10)(s+2)}.$$

We can choose any design method we wish. We shall use classical lead-lag compensation. Let's first try just $C(s) = 1$. The closed-loop step response is



Clearly the settling time spec is not satisfied.

Let's try some more gain to speed up the response: $C(s) = 4$. The closed-loop step response now is



Now there's too much overshoot. To reduce the overshoot, we can increase the phase margin by lead compensation. A lead compensator has the form

$$\frac{\alpha\tau s + 1}{\tau s + 1}, \quad \tau > 0, \ \alpha > 1.$$

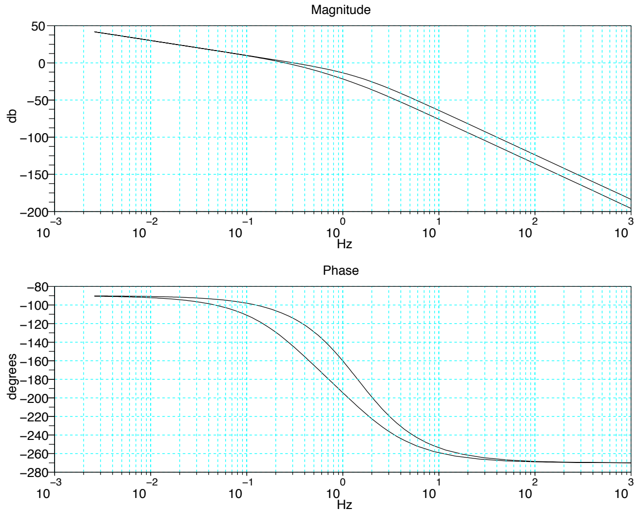
There's a methodical procedure for designing τ, α—see the ECE311 or ECE356 course notes, for example. To raise the phase margin from 43° to 66° yields

$$\alpha = 4, \quad \tau = 1/8.$$

The Bode plots of $4P_a(s)$ and

$$4\frac{\alpha\tau s + 1}{\tau s + 1}P_a(s)$$

are here:



In this way the controller is

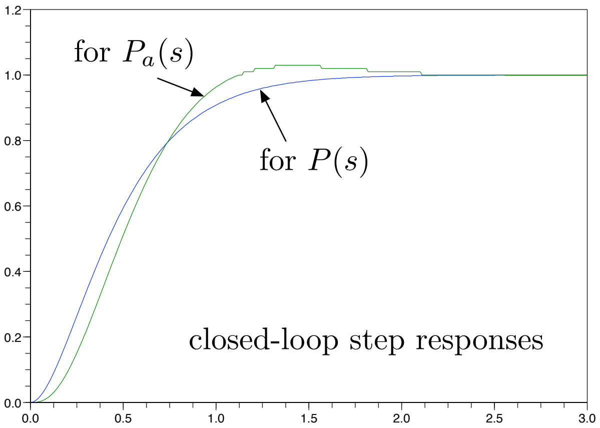
$$C(s) = 16\frac{s+2}{s+8}.$$

Note that there is a stable pole-zero cancellation. If you don't want to do this,

$$C(s) = 16\frac{s+2.1}{s+8.6}.$$

works well too.

Shown next are closed-loop step-response plots with this controller, for both $P(s)$ and the modified plant $P_a(s)$:



The performance with the modified plant definitely deteriorates, as expected.

To get the corresponding digital controller, we can apply the bilinear transformation or `c2d`. The resulting controllers are, respectively,

$$C_{bt}(z) = 10.67\frac{z-0.667}{z-0.111}, \quad C_{c2d}(z) = 16\frac{z-0.8}{z-0.2}.$$

For example, under the bilinear transformation

$$\begin{aligned} C_{bt}(z) &= 16\frac{\frac{2}{0.2}\frac{z-1}{z+1}+2}{\frac{2}{0.2}\frac{z-1}{z+1}+8} \\ &= 16\frac{10(z-1)+2(z+1)}{10(z-1)+8(z+1)} \\ &= 16\frac{12z-8}{18z-2} \\ &= \frac{32}{3}\frac{z-\frac{2}{3}}{z-\frac{1}{9}} \end{aligned}$$

The explicit control computation for the $C_{bt}(z)$ controller is given by

$$u(kT) = 0.1111u(kT-T) + 10.67e(kT) - 7.1167e(kT-T)$$

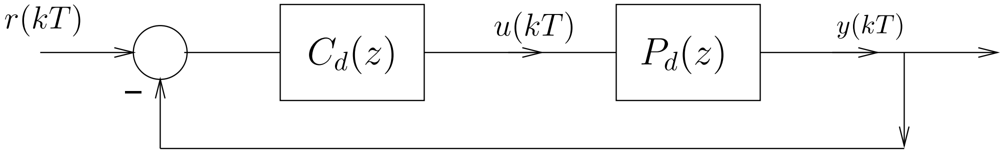
To evaluate the performance of the digital controller at sampling instants, we first determine the pulse transfer function of the plant. For this, we can use the results of Chapter 3.

The pulse transfer function of the plant is given by

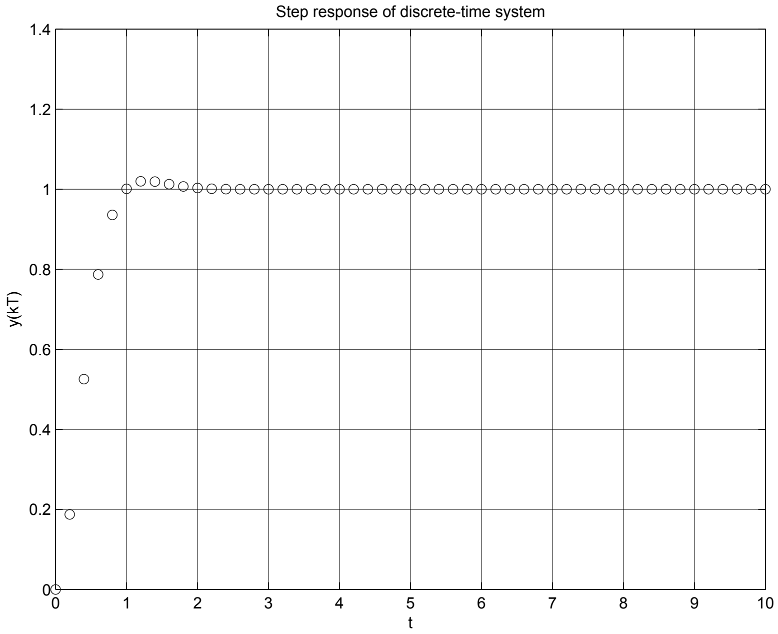
$$\begin{aligned}
G_d(z) &= (1-z^{-1})\mathcal{Z}\frac{1}{s^2(s+2)} \\
&= (1-z^{-1})\mathcal{Z}\{\frac{0.5}{s^2}-\frac{0.25}{s}+\frac{0.25}{s+2}\} \\
&= \frac{z-1}{z}[0.5\frac{0.2z}{(z-1)^2}-\frac{0.25z}{z-1}+\frac{0.25z}{z-e^{-0.4}}] \\
&= 0.01758\frac{z+0.876}{(z-1)(z-0.6703)}
\end{aligned}$$

Note that the true transfer function for the hold operator is now used to compute $G_d(z)$, as in Chapter 3. The Pade approximation for the hold operator is *n*ot part of $G_d(z)$.

The discrete-time system is described by

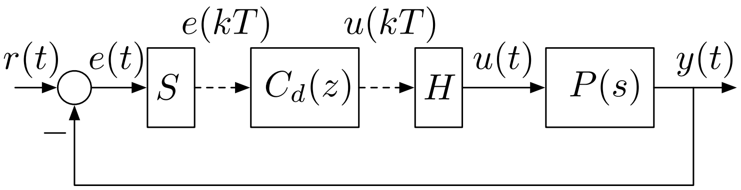


Assume that the controller $C_{bt}(z)$ is used. Substituting in the transfer functions $G_d(z)$ and $C_bt(z)$, we readily find that for the closed loop poles are located at 0.6144, and $0.4890 \pm 0.2411j$, all of which are inside the unit circle. The step response of the discrete-time system is given in the following MATLAB plot.



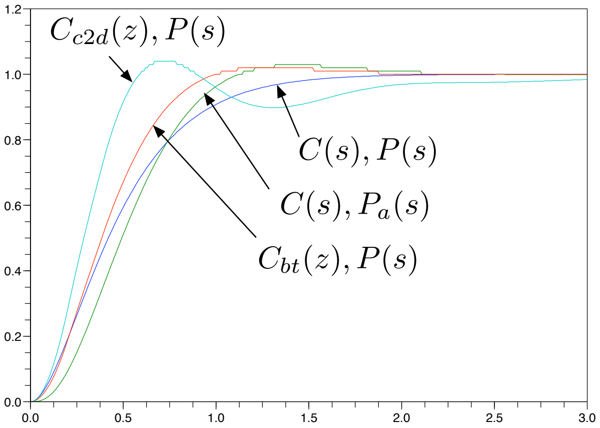
The response at sampling instants shows that the design specs are satisfied.

We now examine the performance of the digital controller in controlling the original continuous-time plant. With either of these control laws, the digital control system is given by the following configuration:



Again, note that the response of the original plant $P(s)$, not the augmented plant $P_a(s)$, is evaluat

We can now compare the performances of the digital controllers with that of the continuous-time controller:



The plot labeled "$C(s), P(s)$" is for the analog controller applied to the original plant, $P(s)$; and so on. We can see the following things:

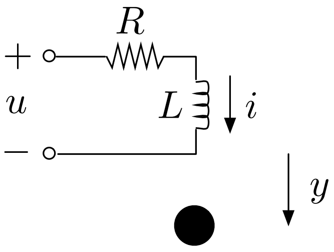
1. The performance of $C_{bt}(z)$ applied to $P(s)$ is comparable to the performance of $C(s)$ applied to the modified plant $P_a(s)$. This justifies this technique of "worsening" the plant model by adding the lag $\frac{1}{1+(T/2)s}$.

2. The performance of $C_{c2d}(z)$ is inferior to that of $C_{bt}(z)$, and not very predictable.

The success of this digital control design method is dependent on having a continuous-time controller which satisfies the design specs. We have used lead-lag compensation as an illustration, but of course other design methods can be used. □

Example Magnetic levitation.

This example illustrates some different issues. The schematic diagram and equations are as follows:



$$L\frac{di}{dt} + Ri = u$$

$$M\frac{d^2y}{dt^2} = Mg - K\frac{i^2}{y^2}$$

Realistic numerical values are $M = 0.1$ Kg, $R = 15$ ohms, $L = 0.5$ H, $K = 0.0001$ Nm^2/A^2, $g = 9.8$ m/s^2. Assume there's a sensor for y only. The problem is to design a digital controller that will stabilize the ball at $y = 0.01$ m. Assume the sampling frequency is 2 kHz.

Substitute in the numbers to get the nonlinear model:

$$0.5\frac{di}{dt} + 15i = u$$

$$0.1\frac{d^2y}{dt^2} = 0.98 - 0.0001\frac{i^2}{y^2}.$$

Define state variables $x = (x_1, x_2, x_3) = (i, y, \dot{y})$. Then the nonlinear state model is

$$\dot{x} = f(x, u),$$

where

$$f(x, u) = (-30x_1 + 2u, x_3, 9.8 - 0.001x_1^2/x_2^2).$$

Solve for the equilibrium point $(\bar{x}, \bar{u})$ where $\bar{x}_2 = 0.01$:

$$-30\bar{x}_1 + 2\bar{u} = 0, \quad \bar{x}_3 = 0, \quad 9.8 - 0.001\bar{x}_1^2/0.01^2 = 0.$$

Thus

$$\bar{x} = (0.99, 0.01, 0), \quad \bar{u} = 14.85.$$

The linearized model is

$$\dot{\Delta x} = A\Delta x + B\Delta u, \quad \Delta y = C\Delta x,$$

where Δ denotes displacement away from equilibrium, and

$$\begin{aligned} A &= \frac{\partial f}{\partial x}(\bar{x},\bar{u}) \\ &= \left[\begin{array}{ccc} -30 & 0 & 0 \\ 0 & 0 & 1 \\ -0.002x_1/x_2^2 & 0.002x_1^2/x_2^3 & 0 \end{array}\right]_{(\bar{x},\bar{u})} \\ &= \left[\begin{array}{ccc} -30 & 0 & 0 \\ 0 & 0 & 1 \\ -19.8 & 1940 & 0 \end{array}\right] \end{aligned}$$

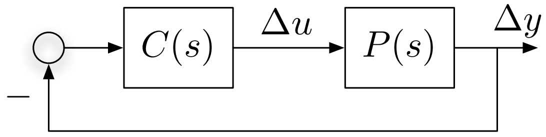
$$B = \frac{\partial f}{\partial u}(\bar{x},\bar{u}) = \left[\begin{array}{c} 2 \\ 0 \\ 0 \end{array}\right], \quad C = \left[\begin{array}{ccc} 0 & 1 & 0 \end{array}\right].$$

The eigenvalues of A are $-30, \pm 44.05$, the units being s^{-1}. The corresponding time constants are $1/30 = 0.033, 1/44.05 = 0.023$ s. The first is the time constant of the electric circuit; the second, the time constant of the magnetics.

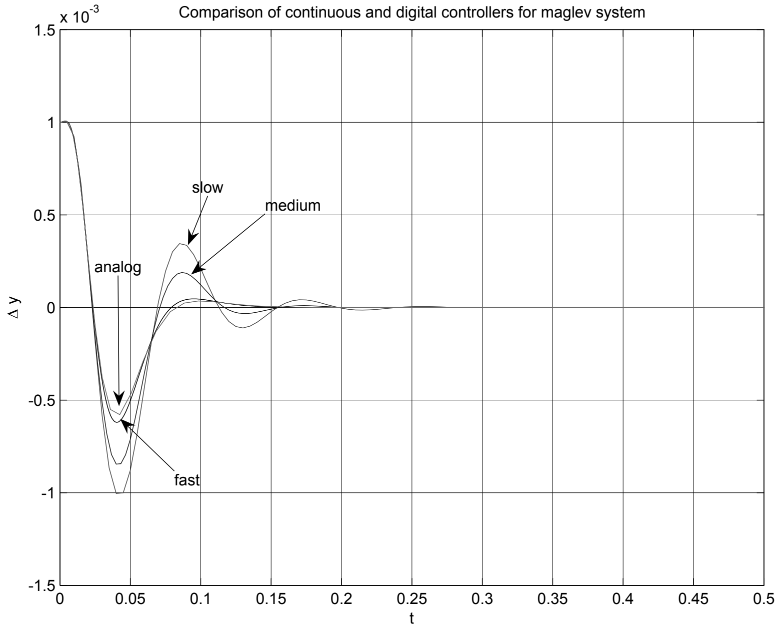
We can stabilize by an observer-based controller. Where to put the eigenvalues? This is normally done by iteration/simulation because practical things like the ball hitting the magnet can be tested. But for an initial try, it's sensible to look at the open-loop plant eigenvalues, $-30, \pm 44$, the unstable one being $+44$. So our goal is to move this into the left half-plane. Now, our plant is actually a toy, so there's no specified objective in terms of closed-loop speed of response. Let's say -100 is a reasonable pole location—about three times that of the electric circuit. In this way, we design (unique) F and L to place the eigenvalues of $A+BF$ and $A+LC$ all at -100. The analog controller for the linearized plant then has the state matrices

$$A_c = A + BF + LC, \quad B_c = L, \quad C_c = F.$$

Since this is a stabilization problem, the block diagram with the analog controller is this:

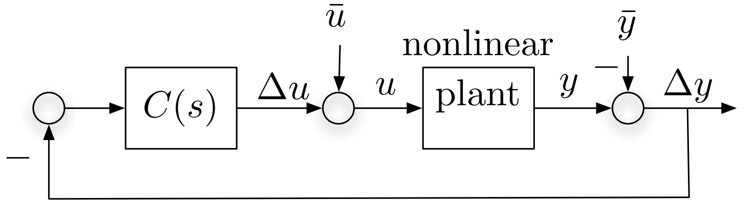


The following graph shows the response of $\Delta y(t)$ to an initial error of $\Delta y(0) = 0.001$, i.e., 1 mm:



Let's discretize the controller via the bilinear transformation with $T = 1/2000$ (fast), $T = 1/300$ (medium), and $T = 1/200$ (slow). The response of the sampled-data system with fast sampling is very similar to that of the analog controller. The response worsens noticeably for discretization with $T = 1/300$, and deteriorates further with $T = 1/200$, although the system remains stable. By contrast, if we use c2d with $T = 1/200$, the closed-loop system is actually unstable. This further demonstrates that discretizing controllers using c2d is not a good idea.

Finally, how would the analog controller be connected to the actual nonlinear plant? Like this:



The digital controller is connected similarly. □

6.4 Exercises

1. Another method of discretizing a continuous-time controller $C(s)$ is called *pole-zero matching.* It is based on the observation that under c2d, a continuous-time system pole is mapped into a discrete-time system pole by the mapping $z = \mathrm{e}^{sT}$. We can apply the same mapping to the continuous-time finite zeros. If the relative degree d (number of poles minus number of zeros) of $C(s)$ is ≥ 1, it gives rise to d infinite zeros. As in the bilinear transformation, each infinite zero produces a zero at -1 for the sampled-data system. With these considerations, the pole-zero matching method of determining $C_d(z)$, the discrete-time approximation of $C(s)$, proceeds as follows:

   (a) Factor $C(s)$ into the form
   $$C(s) = K\frac{(s+b_1)(s+b_2)\cdots(s+b_m)}{(s+a_1)(s+a_2)\cdots(s+a_n)} \quad \text{with } n \geq m.$$

   (b) Each factor $(s+b_i)$ gives a factor $z - e^{-b_iT}$ in the numerator of $C_d(z)$, and each factor $(s+a_i)$ gives a factor $z - e^{-a_iT}$ in the denominator of $C_d(z)$.

   (c) If $n > m$, so that the relative degree $d = n-m > 0$, add a factor $(z+1)^d$ to the numerator of $C_d(z)$ (MATLAB changes this slightly by putting a factor $(z+1)^{d-1}$ instead).

   (d) Finally, set
   $$C_d(z) = K_d\frac{(z+1)^d(z-e^{-b_1T})(z-e^{-b_2T})\cdots(z-e^{-b_mT})}{(z-e^{-a_1T})(z-e^{-a_2T})\cdots(z-e^{-a_nT})}.$$

   Choose K_d so that $C_d(1) = C(0)$. This matches the DC-gain.

   Apply the pole-zero matching approximation to the following continuous-time controller transfer functions

   (i) $C(s) = \frac{1}{s+1}$

   (ii) $C(s) = \frac{s+1}{s+2}$

   Compare the resulting $C_d(z)$ to those obtained using the bilinear transformation in terms of the poles and zeros.

2. Consider the plant $P(s) = \dfrac{1}{s^2}$ and controller $C(s) = 64\dfrac{s+1}{s+16}$.

   (a) Discretize the controller using the bilinear transformation and $T = 0.1$. Determine the poles of the closed-loop discretized feedback system.

   (b) Repeat using c2d to discretize $C(s)$.

   (c) Repeat using the pole-zero matching method of discretization.

   (d) Simulate the closed-loop responses for the three digital controllers when the reference input is a unit step.

6.5 Appendix: Lead-Lag Compensator Design

There are 2 simple ways of increasing the phase margin. One is to add more phase at the gain crossover frequency. The other is to shift by attenuation the gain crossover frequency to a lower value, where the phase margin is larger. The first may be achieved by a lead compensator, the latter by a lag compensator.

Lead Compensation

A lead compensator has the transfer function

$$C(s) = \frac{1 + a\tau s}{1 + \tau s}, \quad a > 1$$

The phase added by the lead compensator ϕ is given by

$$\phi = \tan^{-1} a\tau\omega - \tan^{-1} \tau\omega$$

so that

$$\tan\phi = \frac{a\tau\omega - \tau\omega}{1 + a\tau^2\omega^2}$$

The maximum phase occurs at $\omega_m = \frac{1}{\sqrt{a}}\frac{1}{\tau}$, as may be shown by differentiation. Since

$$\tan\phi_m = \tan\phi(\omega_m) = \frac{(a-1)\frac{1}{\sqrt{a}}}{1+1} = \frac{a-1}{2\sqrt{a}}$$

so that

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

or

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

Note that the effect of the lead compensator is to shift the gain plot to the right and to add phase at the frequency ω_m. The first effect means that the gain crossover frequency will be moved to the right, resulting in a decrease in the phase margin. In order to increase sufficiently the phase margin of the overall system, the phase lead introduced must compensate for this decrease as well as add additional phase lead. This suggests the following procedure.

The following comments apply to lead compensation.

(a) The main mechanism is phase advance at the gain crossover frequency. If the rate of change of the phase of the original system near ω_g is large, the lead compensator will not be effective.

(b) Increases bandwidth: generally faster response, but may have poor noise rejection properties.

1. Compute the required controller gain from the steady state specs.

2. Plot the Bode plot of the plant with the adjusted gain. Determine the gain and phase margins.

3. If the phase margin needs to be increased and a lead compensator is considered, determine the phase lead ψ that needs to be added. Typically the lead compensator shifts the gain crossover frequency and reduces the phase margin to a value less than expected. So we should choose the maximum phase lead ϕ_m of the compensator to be, say,

$$\phi_m = \psi + \theta$$

   for some θ. Compute a.

4. Determine the frequency ω_m at which the original plot has gain$=-10\log a$. This will be the new gain crossover frequency with the lead compensator in place. Determine the phase margin of the original plant at ω_m. If the reduction in the phase margin is larger than θ, the lead compensator will add phase lead that is less than ψ. Repeat steps 3 and 4 until the reduction is approximately θ.

5. Compute τ. Check that the design specifications are met.

Lag Compensation

A lag compensator has the transfer function

$$C(s) = \frac{1+b\tau s}{1+\tau s}, \quad b < 1$$

The phase lag introduced by the lag compensator ϕ is given by

$$\phi = \tan^{-1} b\tau\omega - \tan^{-1}\tau\omega$$

so that

$$\tan\phi = \frac{b\tau\omega - \tau\omega}{1+b\tau^2\omega^2}$$

The maximum phase lag occurs at $\omega_m = \frac{1}{\sqrt{b}}\frac{1}{\tau}$, as may be shown by differentiation. Since

$$\tan\phi_m = \tan\phi(\omega_m) = \frac{(b-1)\frac{1}{\sqrt{b}}}{1+1} = \frac{b-1}{2\sqrt{b}}$$

so that

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

or

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

The main idea of lag compensation is to produce attenuation at the gain crossover frequency so that the new ω_g will be shifted to a lower value where there is more phase margin. The steps are:

1. Compute the required gain from steady state specs.
2. Plot the Bode plot of $KG(i\omega)$. Determine the gain and phase margins.
3. Determine the frequency at which the uncompensated plant has the required phase margin + 6°. This will be the new gain crossover frequency ω_g'.
4. Compute the required attenuation α to shift ω_g to ω_g'. Set $-20\log b = \alpha$, since $-20\log b$ is the asymptotic attenuation introduced by the lag compensator.
5. Set $\frac{1}{b\tau} = \frac{\omega_g'}{10}$. This results in the phase lag introduced by the lag compensator at the new gain crossover frequency ω_g' to satisfy

$$\begin{aligned}\phi &= \tan^{-1} b\tau\omega_g' - \tan^{-1}\tau\omega_g' \\ &= \tan^{-1} 10 - \tan^{-1}\tau\omega_g' \\ &\leq 84.2894^\circ - 90^\circ \approx 5.7^\circ\end{aligned}$$

   The reduction in phase margin has been accounted for by the addition of 6°. Of course, one can introduce an even larger safety margin.

6. Verify that the system satisfies all specs.