Transfer functions

Systems theory is all about input and output. One common way to relate them to each other, is by using transfer functions. In this chapter, we'll go more into depth on what these functions really are.

1 Transfer function basics

1.1 The Laplace transform

When we are dealing with systems in the normal way, we say we're working with the system in the time domain. There is an alternative: the **Laplace domain**. To use this domain, we have to make use of the **Laplace transform**. Let's examine a function $f(t)$ in the time domain. Its Laplace transform $F(s)$ is given by

$$F(s) = \mathscr{L}(f(t)) = \int_0^\infty f(t)e^{-st}dt. \tag{1.1}$$

Now let's examine a system with impulse response matrix $G(t-\tau)$. If we assume that the input $\mathbf{u}(\tau)$ is zero for $\tau < 0$, then we have

$$\mathbf{y}(t) = \int_{-\infty}^t G(t-\tau)\mathbf{u}(\tau)d\tau = \int_0^t G(t-\tau)\mathbf{u}(\tau)d\tau. \tag{1.2}$$

The **convolution theorem** now states that this equation is equivalent to $Y(s) = H(s)U(s)$. Here, $U(s)$ and $Y(s)$ are the Laplace transforms of $\mathbf{u}(t)$ and $\mathbf{y}(t)$. (Transforming vectors and matrices is simply done element-wise.) Also, $H(s)$ is the **transfer matrix** of the system.

So, to find $H(s)$, we can simply use $H(s) = \mathscr{L}(G(t)) = \mathscr{L}(Ce^{At}B)$. But there is also another way to find $H(s)$. For that, we have to use the property that, for any $\mathbf{x}(t)$, we have $\mathscr{L}(\dot{x}) = sX(s) - \mathbf{x_0}$. Applying this to the state space equation of a linear time-invariant system gives

$$sX(s) - \mathbf{x_0} = AX(s) + BU(s) \qquad \Rightarrow \qquad X(s) = (sI-A)^{-1}\mathbf{x_0} + (sI-A)^{-1}BU(s). \tag{1.3}$$

If we assume that $\mathbf{x_0} = \mathbf{0}$ and $D = 0$, then

$$H(s) = \mathscr{L}(Ce^{At}B) = C(sI-A)^{-1}B. \tag{1.4}$$

The matrix function $(sI-A)^{-1} = \mathscr{L}(e^{At})$ is known as the **resolvente** of the matrix A.

1.2 Connecting series

Let's suppose that we want to connect two systems. When we use state space representations, this will give us a lot of work. This is where the Laplace transform saves time. Because connecting systems is easy with the Laplace transform.

Let's suppose that we want to connect two systems $H_1(s)$ and $H_2(s)$. What will the resulting transfer matrix $H(s)$ be? If we connect the systems in **series** (figure 1, left), then we will have $H(s) = H_2(s)H_1(s)$. (Note that the order of multiplication is important, since we are using matrices.) If we connect them in **parallel** (figure 1, middle), then we have $H(s) = H_1(s) + H_2(s)$. And if we connect them in a **feedback connection** pattern (figure 1, right), we will get $H(s) = (I + H_1(s)H_2(s))^{-1}H_1(s)$.

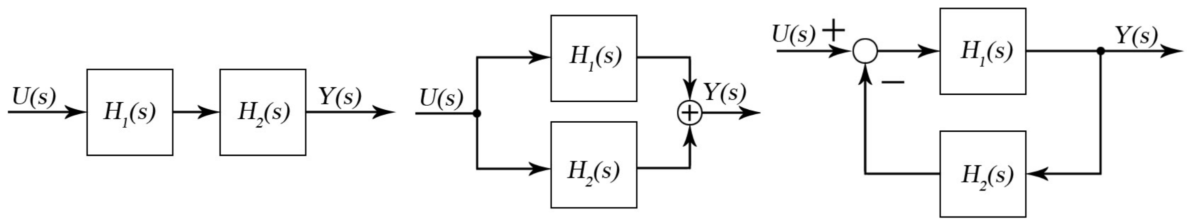

Figure 1: Three ways to connect two systems.

1.3 Transfer function properties

We can assign several properties to transfer functions. We start with the property of rationality. We say that a function $H(s)$ is a **rational function** if it can be written as

$$H(s) = \frac{q(s)}{p(s)} = \frac{q_k s^k + q_{k-1} s^{k-1} + \ldots + q_1 s + q_0}{s^n + p_{n-1} s^{n-1} + \ldots + p_1 s + p_0}. \tag{1.5}$$

In other words, $H(s)$ can be written as the ratio of two polynomials. Let's take a closer look at the numerator $q(s)$ and the denominator $p(s)$. The roots of $q(s)$ are called the **zeroes** of the function, whereas the roots of $p(s)$ are called the **poles**. We also define the **degree** of a polynomial as the highest power of the variable s. Thus, in the above equation we have $\deg(q) = k$ and $\deg(p) = n$.

We say that a transfer function $H(s)$ is **proper** if it has $\deg(q) \leq \deg(p)$. (This implies that $\lim_{s\to\infty} H(s) =$ constant.) And a function is **strictly proper** if $\deg(q) < \deg(p)$. (We now have $\lim_{s\to\infty} H(s) = 0$.) We can always transform a linear time-invariant state space system to a transfer function. If we do this, we always get a proper transfer function $H(s)$. And, if $D = 0$, we even get a strictly proper transfer function.

2 State space realizations

2.1 The controller form

Let's examine a proper transfer function $H(s)$. We can always transform this transfer function to a state space representation. We will illustrate this process for a single-input single-output system $h(s)$. (Later on, we examine multi-dimensional systems.) First, let's assume that $h(s)$ is strictly proper, and thus $k < n$. This implies that $D = 0$. We can now find the so-called **controller form** (also known as the **standard controllable realization**). It is given by

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -p_0 & -p_1 & \cdots & -p_{n-2} & -p_{n-1} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} q_0 & q_1 & \cdots & q_{n-2} & q_{n-1} \end{bmatrix}. \tag{2.1}$$

(The reason why they call this the controllable realization is because it is easy to see that the system is controllable.) You may still wonder, what if $k = n$? In this case, we first have to rewrite $h(s)$ as

$$h(s) = q_n + \frac{\bar{q}(s)}{p(s)} = q_n + \frac{(q_{k-1} - q_k p_{k-1}) s^{k-1} + \ldots + (q_1 - q_k p_1) s + (q_0 - q_n p_0)}{s^n + p_{n-1} s^{n-1} + \ldots + p_1 s + p_0}. \tag{2.2}$$

We now take $D = q_n$ and use the strictly proper function $\bar{q}(s)/p(s)$ to find A, B and C in the normal way. Do note though, that a system can always be represented by infinitely many different state space representations. We have just found one of them.

2.2 Other forms

Next to the standard controllable realization, we also have a **standard observable realization**. To find it, we have exactly the same process as previously. But now, we have the matrices

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & -p_0 \\ 1 & 0 & \cdots & 0 & -p_1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & -p_{n-2} \\ 0 & 0 & \cdots & 1 & -p_{n-1} \end{bmatrix}, \qquad B = \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-2} \\ q_{n-1} \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{2.3}$$

The last representation that we will discuss is the **diagonal realization**. To find it, we first have to write

$$h(s) = D + \frac{q(s)}{p(s)} = D + \frac{\gamma_1}{s-a_1} + \frac{\gamma_2}{s-a_2} + \ldots + \frac{\gamma_n}{s-a_n}. \tag{2.4}$$

This is possible as long as $h(s)$ has n different poles. If this is the case, then we will find that

$$A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_n \end{bmatrix}. \tag{2.5}$$

You may wonder, what if some factor $(s - a_i)$ occurs more than once in $p(s)$? In this case, A won't be a diagonal matrix. Instead, ones will appear in A and disappear in B. For example, let's examine the function

$$h(s) = \frac{\gamma}{s-1} + \frac{\delta}{(s-a)^2} + \frac{\alpha}{(s-a)^3}. \tag{2.6}$$

This will give the state space matrices

$$A = \begin{bmatrix} a & 1 & 0 \\ 0 & a & 1 \\ 0 & 0 & a \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} \alpha & \delta & \gamma \end{bmatrix}. \tag{2.7}$$

2.3 Multiple inputs/outputs

We can also turn systems with multiple inputs/outputs into a state space representation. Let's examine the $p \times m$ transfer matrix $H(s)$. We can write it as

$$H(s) = D + \frac{1}{p(s)} Q(s) = D + \frac{1}{p(s)} \left(Q_{n-1} s^{n-1} + Q_{n-2} s^{n-2} + \ldots + Q_1 s + Q_0 \right). \tag{2.8}$$

In this equation, $Q(s)$ denotes a $p \times m$ transfer matrix and Q_i (with $i = 0, 1, \ldots, n-1$) denotes a constant $p \times m$ matrix. The matrices A, B and C are now given by

$$A = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & I \\ -p_0 I & -p_1 I & \cdots & -p_{n-2} I & -p_{n-1} I \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} Q_0 & Q_1 & \cdots & Q_{k-2} & Q_{k-1} \end{bmatrix}. \tag{2.9}$$

A now is a matrix of size $mn \times mn$. Note that the above matrices are in some sort of controller form. In a similar way, we can put a multiple input/output system in an observable realization.

2.4 The McMillan degree

Now we know how to write a multiple input/output system in state space form. But we did need nm state variables for that. We could ask ourselves, can't we use less state variables? Well, that actually depends on the transfer function $H(s)$. Let's call the minimum amount of state variables needed z. This number z is also known as the **McMillan degree** of $H(s)$. To find it, we can examine

$$H(s) = L_0 + L_1 s^{-1} + L_2 s^{-2} + \ldots. \tag{2.10}$$

The McMillan degree z of $H(s)$ is now given by the rank of the matrix L (so $z = \text{rank}(L)$), where

$$L = \begin{bmatrix} L_1 & L_2 & \cdots & L_r \\ L_2 & L_3 & \cdots & L_{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ L_r & L_{r+1} & \cdots & L_{2r-1} \end{bmatrix}. \tag{2.11}$$

The parameter r is the degree of the least common multiple of all denominators of $H(s)$.

When we use the minimum amount of state variables to create a state space representation, then we say that we have a **minimum realization**. If we have a minimum realization system, then it can be shown that the system is always both controllable and observable. Equivalently, if the system is either not controllable or not observable (or both), then it is not a minimum realization system.