Chapter Two

Transfer Function Approach

In the previous chapter it has been indicated that modeling, analysis, and design of control systems can be performed in two domains, namely in the time and frequency domains. In this chapter we will consider the frequency (complex) domain technique, also known as the transfer function method. Our main goal is to present methods for finding the system transfer function. This is particularly important for systems composed of many blocks, where each block represents an internal transfer function. In Chapter 9, the frequency domain approach will be used to design controllers for linear time invariant systems.

Modern control theory has its foundation in the state space approach; classical control theory is based on the transfer function approach. The state space method is widely used in modern control theory and practice due to the extensive support from modern packages for computer-aided control system analysis and design. The state space method will be considered in detail in Chapter 3. The transfer function approach is based on the Laplace and $\mathcal{Z}$-transforms and their time derivative properties, which convert differential/difference equations into algebraic equations with complex coefficients. The algebraic equations obtained are frequency domain representations of the considered dynamic systems. The basics of the Laplace and $\mathcal{Z}$-transforms are reviewed in Appendices A and B.

In classical control theory, it is desirable to have a tool that permits analysis and design of control systems, especially when instead of knowing the internal state of the system, we just need to know the relationship between the system inputs and outputs. This can be facilitated if the system model is transformed from the time domain into the frequency domain. The transfer function—the main concept of the frequency domain technique—is considered for both continuous- and discrete-time systems in Section 2.1. This book emphasizes continuous-time systems since many discrete-time results are easily derived by analogy from the corresponding continuous-time results.

A conventional way of representing linear time invariant systems is via block diagrams. This provides a pictorial view of a control system. Block diagrams are considered in Section 2.2. Block diagram algebra is introduced in Section 2.3 as a suitable tool for obtaining transfer functions of systems whose block diagrams are known. The use of block diagram algebra to find the system transfer function is advisable for simple systems, but for complex systems it gets quite involved.

The signal flow graph technique is employed in Section 2.4 as an alternative to the block diagram system representation. Mason's gain rule, the main result of the signal flow graph technique, is an elegant way of finding transfer functions, especially for complex and high-dimensional systems. Several examples are given in order to demonstrate the power of Mason's rule.

In Section 2.5 we present specialized methods for finding transfer functions of sampled data control systems obtained by sampling continuous-time systems.

At the end of this chapter, in Section 2.6, a laboratory MATLAB experiment on the system transfer function is designed.

Chapter Study Guide and Objectives

Students not completely familiar with the Laplace and $\mathcal{Z}$-transforms should first read Appendices A and B. Instructors not interested in teaching transfer functions of sampled data control systems may skip Section 2.5 without loss of continuity. Sections 2.1–2.4 represent the core of the chapter. The main objective of this chapter is that students master a technique for finding transfer functions of any time invariant linear control system by using either the block diagram algebra or Mason's rule.

2.1 Frequency Domain Representations

Real dynamic systems operate in real continuous time so that it is natural to describe and study their dynamical behavior and evolution in continuous time. This is done by using differential equations to model them. Some artificial dynamic systems operate in discrete time so that their models are represented by difference equations. In addition, discrete-time systems can be obtained by discretizing continuous-time systems so that the obtained sampled data systems are also described by difference equations. The study of dynamic systems in both continuous and discrete time will be presented in detail in Chapter 3.

Another way of studying continuous- and discrete-time systems is the frequency domain approach. This approach is performed in the space of complex numbers: by using the Laplace and $\mathcal{Z}$-transforms, the differential/difference equations are transformed into linear algebraic equations with complex coefficients. In general, it is easier to solve linear algebraic equations than linear differential/difference equations, and hence the frequency domain approach seems to be very attractive. The frequency domain approach is often called the complex domain approach. Since all calculations have to be performed in the complex domain, and since the complex numbers $s = \sigma + j\omega$ and $z = e^{-sT}$ (where T stands for a sampling period) are also known in engineering as complex frequencies, the common name for these methods is the frequency domain methods. The importance of such a representation of a system is especially emphasized in classical control system theory.

We would like to point out that the frequency domain very often gives a better understanding of the actual control system phenomena than the time domain, but from the computational point of view the frequency domain is inferior to the time domain state space approach, especially for high-order dimensional systems.

2.1.1 System Transfer Functions

The system transfer function relates to the frequency domain system outputs and inputs. In other words, the system transfer function gives what is in between the system inputs and outputs, i.e. it indicates what kind of dynamic elements input signals have to face before they appear on the system outputs. This pictorial definition can be put in rigorous mathematical form by using the Laplace and $\mathcal{Z}$-transforms.

In the first part of this section we present the transfer functions for single-input single-output systems, and in the second part we study the general case of multi-input multi-output systems.

Definition 2.1: The transfer function of a *continuous-time single-input single-output system* is defined as the ratio of the *Laplace transform* of the system output over the *Laplace transform* of the system input, when *all initial conditions are zero*.

Definition 2.2: The transfer function of a *discrete-time single-input single-output system* is defined as the ratio of the $\mathcal{Z}$*-transform* of the system output over the $\mathcal{Z}$*-transform* of the system input, when *all initial conditions are zero*.

Note that there are several other ways to introduce the definition of the system transfer function (Franklin *et al.*, 1990; Kuo, 1995), but all of them are basically the same.

Some preliminary results on system transfer functions have been presented in Section 1.3. Since the presentation of the discrete-time transfer function parallels that for continuous time, we will mostly present the results for continuous-time transfer functions and give only the final results for discrete-time transfer functions. In Section 2.5 we will pay special attention to the transfer functions of sampled data systems.

Consider a single-input single-output control system represented by an n-order differential equation, that is

$$\begin{aligned}\frac{d^n y(t)}{dt^n} + a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}} + \cdots + a_1\frac{dy(t)}{dt} + a_0 y(t)\\ = b_m\frac{d^m u(t)}{dt^m} + b_{m-1}\frac{d^{m-1}u(t)}{dt^{m-1}} + \cdots + b_1\frac{du(t)}{dt} + b_0 u(t)\end{aligned} \tag{2.1}$$

with $n \geq m$. If the initial conditions are zero, its complex counterpart is obtained simply by a substitution of d^i/dt^i by s^i and $y(t) \to Y(s)$, $u(t) \to U(s)$ (see (a.4) in Appendix A), to give

$$\begin{aligned}\left(s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0\right)Y(s)\\ = \left(b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0\right)U(s)\end{aligned} \tag{2.2}$$

Hence, the transfer function of this system is

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} = \frac{N_m(s)}{D_n(s)} \tag{2.3}$$

Similarly, for discrete-time systems the transfer function is obtained by applying the $\mathcal{Z}$-transform to the difference equation describing system dynamics

$$\begin{aligned}y(k+n) + a_{n-1}y(k+n-1) + \cdots + a_1 y(k+1) + a_0 y(k)\\ = b_m u(k+m) + b_{m-1}u(k+m-1) + \cdots + b_1 u(k+1) + b_0 u(k)\end{aligned} \tag{2.4}$$

This yields the discrete-time transfer function of the form

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0} = \frac{N_m(z)}{D_n(z)} \tag{2.5}$$

Polynomials N_m and D_n (with the s and z arguments dropped) have real coefficients, and for the so-called proper systems (real physical systems or causal systems) it must be satisfied that $m \leq n$. The meaning of a proper system is that the system cannot respond before an input to the system is applied (system causality, see for example, Kamen, 1990).

The polynomial in the denominator of a single-input single-output system transfer function, D_n, is called the *characteristic polynomial*, and its roots are known as the *system poles*. At any of these n roots the denominator polynomial D_n is zero, so that the overall transfer system function becomes infinite. The roots of the numerator polynomial N_m are called the *system zeros* since at these m values both the numerator polynomial and the system transfer function are zero. If the system poles and zeros are known, the transfer function $G(s)$ can be recorded in *pole-zero* form as

$$G(s) = K \frac{(s + z_1)(s + z_2) \cdots (s + z_m)}{(s + p_1)(s + p_2) \cdots (s + p_n)} \tag{2.6}$$

or in *time constant* form

$$G(s) = K_\tau \frac{(\tau_{b_1} s + 1)(\tau_{b_2} s + 1) \cdots (\tau_{b_m} s + 1)}{(\tau_{a_1} s + 1)(\tau_{a_2} s + 1) \cdots (\tau_{a_n} s + 1)} \tag{2.7}$$

From (2.3) and (2.6)–(2.7) we have

$$K = b_m, \qquad K_\tau = K \frac{z_1 z_2 \cdots z_m}{p_1 p_2 \cdots p_n}$$

and

$$\tau_{a_i} = \frac{1}{p_i},\ i = 1, 2, ..., n; \qquad \tau_{b_i} = \frac{1}{z_i},\ i = 1, 2, ..., m$$

Discrete-time transfer functions can be represented by the forms identical to (2.6) and (2.7) with the complex frequency s replaced by the complex frequency z.

Example 2.1: The discrete transfer function of the following system

$$y(k + 4) + 3y(k + 2) - y(k + 1) + 5y(k) = u(k + 1) + 2u(k)$$

is obtained by assuming that all initial conditions are equal to zero and by applying the derivative (left shift in time) property of the $\mathcal{Z}$-transform, which leads to

$$\left(z^4 + 3z^2 - z + 5\right)Y(z) = (z+2)U(z)$$

so that

$$G(z) = \frac{Y(z)}{U(z)} = \frac{z+2}{z^4 + 3z^2 - z + 5}$$

$\diamond$

Example 2.2: For the transfer function

$$G(s) = \frac{s^3 + 0.4s^2 - 0.95s - 0.45}{s^5 + 8.3s^4 + 23.1s^3 + 26.2s^2 + 10.4s}$$

we find the pole-zero form by using MATLAB function `tf2zp`. This produces

$$G(s) = \frac{(s-1)(s+0.5)(s+0.9)}{s(s+1)(s+1.3)(s+2)(s+4)}$$

$\diamond$

Example 2.3: The transfer function for the linearized system from Example 1.1 is easily obtained by setting all initial conditions to zero and taking the Laplace transform, which leads to

$$G(s) = \frac{\Theta(s)}{U(s)} = \frac{-1}{s^2 - 1}$$

$\diamond$

It is important to point out that the system transfer function carries information about the *system impulse response*. In general, we have

$$\begin{aligned} Y(s) &= G(s)U(s) \\ Y(z) &= G(z)U(z) \end{aligned} \tag{2.8}$$

Using impulse delta functions as inputs ($U(s) = 1, U(z) = 1$), we get

$$Y_{impulse}(s) = G(s), \quad Y_{impulse}(z) = G(z) \tag{2.9}$$

so that, in the time domain

$$\begin{aligned} g(t) &= \mathcal{L}^{-1}(Y_{impulse}(s)) = \mathcal{L}^{-1}\{G(s)\} \\ g(k) &= \mathcal{Z}^{-1}\{Y_{impulse}(z)\} = \mathcal{Z}^{-1}\{G(z)\} \end{aligned} \tag{2.10}$$

The impulse response is obtained simply by finding the inverse transformation (from the frequency domain to the time domain) of the corresponding system transfer function.

For multi-input multi-output (multivariable) systems the definition of the transfer function is more general since vectors and matrices are involved in the system transfer function description.

Definition 2.3: Transfer functions of *multivariable systems* relate the frequency representation of system vector inputs and system vector outputs assuming that all initial conditions are equal to zero, that is

$$\begin{aligned} \mathbf{Y}(s) &= \mathbf{G}(s)\mathbf{U}(s) \\ \mathbf{Y}(z) &= \mathbf{G}(z)\mathbf{U}(z) \end{aligned} \tag{2.11}$$

Note that $\mathbf{G}(s)$ and $\mathbf{G}(z)$ are matrices of dimensions $p \times r$ since we have assumed that the number of inputs is r and the number of outputs is p, so that vectors $\mathbf{U}(s), \mathbf{U}(z)$ are of dimensions $r \times 1$ and vectors $\mathbf{Y}(s), \mathbf{Y}(z)$ have dimensions $p \times 1$. Due to this "matrix" nature, one has to be careful while relating inputs and outputs for multivariable systems. For single-input single-output systems, one can write

$$\begin{aligned} Y(s) &= G(s)U(s) = U(s)G(s) \\ Y(z) &= G(z)U(z) = U(z)G(z) \end{aligned} \tag{2.12}$$

However, for multivariable systems this commutativity does not hold since in that case we are dealing with vectors and matrices.

We have seen in Section 1.3 that the transfer matrix of a system with r inputs and p outputs has the form

$$\mathbf{G}^{p\times r}(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1r}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2r}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{p1}(s) & G_{p2}(s) & \dots & G_{pr}(s) \end{bmatrix}^{p\times r} \tag{2.13}$$

where the coefficients G_{ij} denote the transfer functions between the jth input and the ith output, when all other inputs are zero, that is

$$G_{ij}(s) = \frac{Y_i(s)}{U_j(s)}\Big|_{all\ inputs\ except\ jth\ are\ set\ to\ zero} \tag{2.14}$$

The following example illustrates the procedure for finding the transfer function for an inverted pendulum, which can be viewed as a multivariable system with two outputs and one input.

Example 2.4: Recall a linearized model of the inverted pendulum, given by (1.77), that is

$$(m_1 + m_2)\frac{d^2x(t)}{dt^2} + m_1 l \frac{d^2\theta(t)}{dt^2} = F(t)$$

$$\frac{d^2x(t)}{dt^2} + l\frac{d^2\theta(t)}{dt^2} = g\theta(t)$$

Let us first find the transfer function $\Theta(s)/F(s)$. The above system of two equations can be simplified by eliminating d^2x/dt^2, which produces

$$m_2 l \frac{d^2\theta(t)}{dt^2} - (m_1 + m_2)g\theta(t) = -F(t)$$

After taking the Laplace transform of this equation, the transfer function is obtained as

$$G_{11}(s) = \frac{\Theta(s)}{F(s)} = \frac{-\frac{1}{m_2 l}}{s^2 - \left(1 + \frac{m_1}{m_2}\right)\frac{g}{l}}$$

The second transfer function, $X(s)/F(s)$, is obtained by taking the Laplace transform of the second equation, that is

$$s^2X(s) + ls^2\Theta(s) = g\Theta(s)$$

so that

$$X(s) = -\frac{ls^2 - g}{s^2}\Theta(s) = -\frac{ls^2 - g}{s^2}G_{11}(s)F(s)$$

which implies

$$G_{21}(s) = -\frac{ls^2 - g}{s^2}G_{11}(s)$$

The same result could have been obtained by taking simultaneously the Laplace transform of both the equations

$$\begin{bmatrix}(m_1+m_2)s^2 & m_1 l s^2\\ s^2 & ls^2-g\end{bmatrix}\begin{bmatrix}X(s)\\ \Omega(s)\end{bmatrix}=\begin{bmatrix}F(s)\\ 0\end{bmatrix}$$

and then solving this system of algebraic equations with respect to $X(s)$ and $\Theta(s)$, that is

$$\begin{bmatrix}X(s)\\ \Omega(s)\end{bmatrix}=\begin{bmatrix}(m_1+m_2)s^2 & m_1 l s^2\\ s^2 & ls^2-g\end{bmatrix}^{-1}\begin{bmatrix}F(s)\\ 0\end{bmatrix}=\begin{bmatrix}G_{11}(s)\\ G_{21}(s)\end{bmatrix}F(s)$$

◇

Sometimes systems are so complex that playing with algebraic equations in the complex domain in order to obtain the system transfer function(s) is mathematically very involved. A graphical system representation in terms of either block diagrams or signal flow graphs will help us to develop systematic methods for finding the system transfer function(s).

2.2 Block Diagrams

A pictorial description is a very convenient way of representing dynamic systems. It gives a clear picture of all components of the control system and the flow of information (signals) in the system. Such a representation is called the *system block diagram*. In the following, we show how to use block diagrams in order to obtain information about input and output variables, the relationships between these variables, and how to get the transfer function(s). In some cases the block diagram is used just to represent the composition and interconnections of a system.

The simplest possible block diagram of a single-input single-output system is represented in Figure 2.1, where $U(s)$ and $Y(s)$ are, respectively, the Laplace transforms of the input and output signals, and $G(s)$ is the block transfer function. This is consistent with the transfer function definition given in (2.3). The rule that connects (2.3) and the block diagram in Figure 2.1, and which is also valid for all block diagrams, can be formulated as follows.

Main Block Diagram Rule: *The output signal is a product of the transfer function of the given block and block's input signal.*

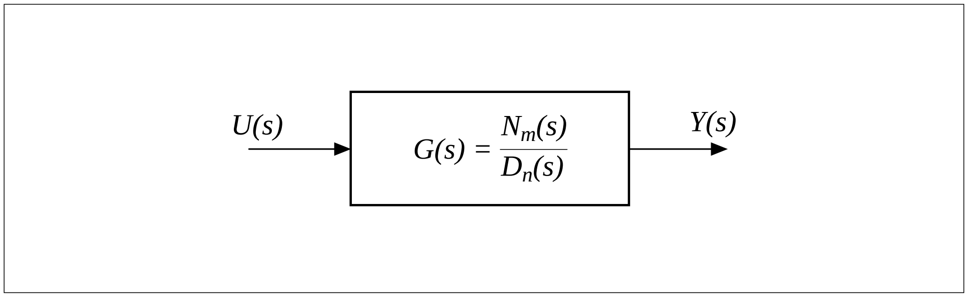

Figure 2.1: Block diagram of a general linear system defined in (2.3)

The basic structure of a single-input single-output *closed-loop system* with a non-unit feedback is presented in Figure 2.2a, and with a unit feedback is given in Figure 2.2b. Applying the main block diagram rule and taking into account the directions of the flow of signals in the system as indicated by arrows, we have

$$Y(s) = G(s)E(s)$$
$$E(s) = U(s) - H(s)Y(s)$$

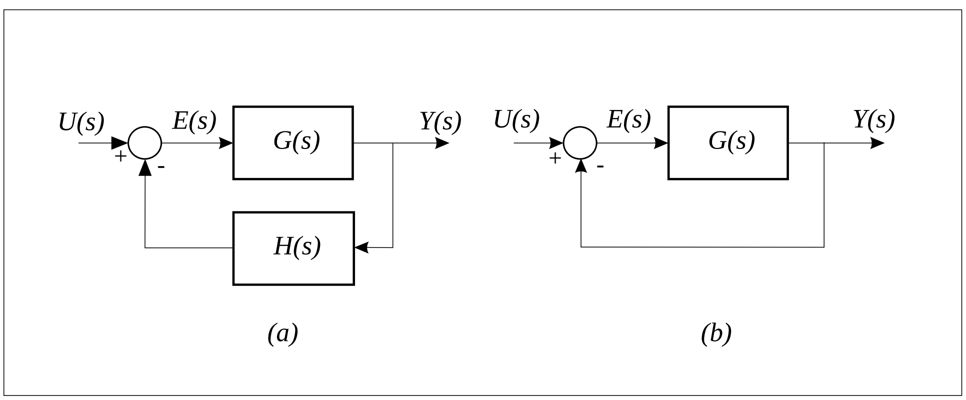

Figure 2.2: Simple feedback structures

Eliminating $E(s)$ from the last equation, it follows

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

so that for a single-input single-output control system the *closed-loop transfer function*, denoted by $M(s)$, is given by

$$M(s) = \frac{Y(s)}{U(s)} = \frac{G(s)}{1 + G(s)H(s)} \tag{2.15}$$

Note that if the system loop is open, which is the case for $H(s) = 0$, we have the *open-loop system transfer function*

$$Y(s) = G(s)E(s) = G(s)U(s) \;\Rightarrow\; \frac{Y(s)}{U(s)} = G(s) \tag{2.16}$$

A more general block diagram of a closed-loop control system is given in Figure 2.3.

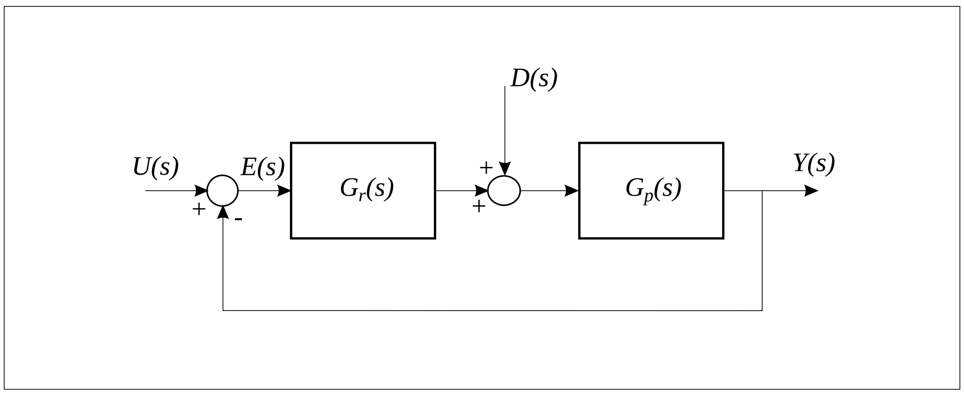

Figure 2.3: Basic structure of a control system

The two principal components (blocks) in this diagram are the *plant* and the *controller*. The plant has its own dynamics represented by the transfer function $G_p(s)$. The purpose of the controller (often called the regulator) $G_r(s)$ is to "reshape" the dynamics of the plant, such that the overall system transfer function has the desired form. In Figure 2.3, the signal $D(s)$ represents the potential system disturbance. The main roles of the controllers (regulators) are system stabilization, improvement of the system transient response, reduction of steady state errors, disturbance rejection, etc. These issues will be addressed in the subsequent chapters. Here, we study only the problem of obtaining the system transfer function from the block diagrams.

In this block diagram we have two inputs, $U(s)$ and $D(s)$, and one output, $Y(s)$. Setting $D(s) = 0$, we get

$$Y(s) = G_p(s)G_r(s)E(s)$$
$$E(s) = U(s) - Y(s)$$

that is

$$Y(s) = G_p(s)G_r(s)U(s) - G_p(s)G_r(s)Y(s)$$

so that the system closed-loop transfer function is given by

$$M(s) = \frac{Y(s)}{U(s)} = \frac{G_p(s)G_r(s)}{1 + G_p(s)G_r(s)} \tag{2.17}$$

It is interesting to find the transfer function from the system disturbance $D(s)$ to the system output $Y(s)$. By setting $U(s) = 0$, we get

$$E(s) = -Y(s)$$
$$Y(s) = G_p(s)[D(s) + G_r(s)E(s)] = G_p(s)D(s) - G_p(s)G_r(s)Y(s)$$

so that

$$\frac{Y(s)}{D(s)} = \frac{G_p(s)}{1 + G_p(s)G_r(s)} \tag{2.18}$$

Since it is not desirable for the disturbance to affect the system output, the magnitude of the corresponding transfer function

$$\left|\frac{G_p(s)}{1 + G_p(s)G_r(s)}\right| \tag{2.19}$$

should be minimized as much as possible by a proper choice of the controller $G_r(s)$.

In the case of *multivariable systems*, the closed-loop transfer function can be found using the same technique, bearing in mind that the corresponding quantities are vectors and matrices. The closed-loop block diagram of a multivariable system is given in Figure 2.4.

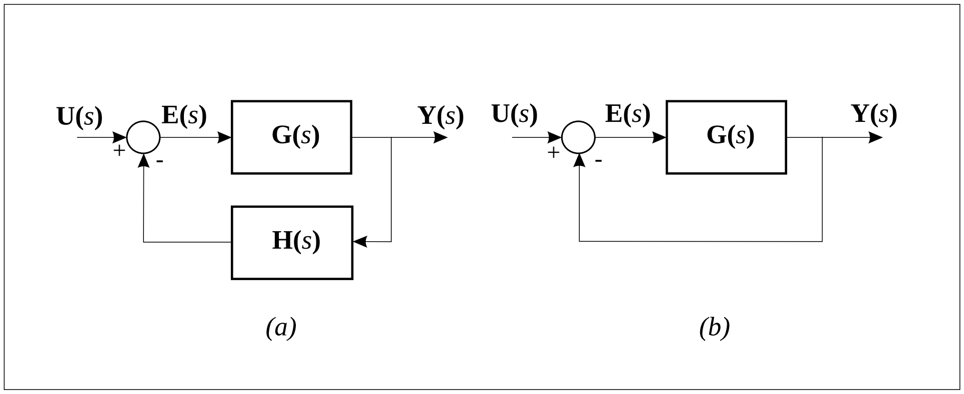

Figure 2.4: Multivariable feedback system

From this block diagram by using the main block diagram rule, *to be applied strictly in the order: output = transfer function* $\times$ *input*, it follows

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{E}(s)$$
$$\mathbf{E}(s) = \mathbf{U}(s) - \mathbf{H}(s)\mathbf{Y}(s)$$

so that

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s) - \mathbf{G}(s)\mathbf{H}(s)\mathbf{Y}(s)$$

which implies

$$\mathbf{Y}(s) = [\mathbf{I} + \mathbf{G}(s)\mathbf{H}(s)]^{-1}\mathbf{G}(s)\mathbf{U}(s) \tag{2.20}$$

The *closed-loop multivariable control system transfer function* relates the frequency representations of the system vector input and vector output. From (2.20) it is given by

$$\mathbf{M}(s) = [\mathbf{I} + \mathbf{G}(s)\mathbf{H}(s)]^{-1}\mathbf{G}(s) \tag{2.21}$$

For $\mathbf{H}(s) = 0$, we have the open-loop transfer function of a multivariable control system obtained from (2.20) as $\mathbf{G}(s)$.

In the next subsection we show how to perform modeling and construct a block diagram from mathematical equations describing system dynamics. We consider a model of a DC motor, which is frequently used in control systems.

2.2.1 Modeling and Block Diagrams of a DC Motor

A DC motor is an electromechanical energy converter which converts electrical energy into mechanical energy. It is often used as an actuator in control systems. Figure 2.5 illustrates such a motor schematically. In this section we present the modeling of a DC motor and draw two block diagrams corresponding to two different working conditions.

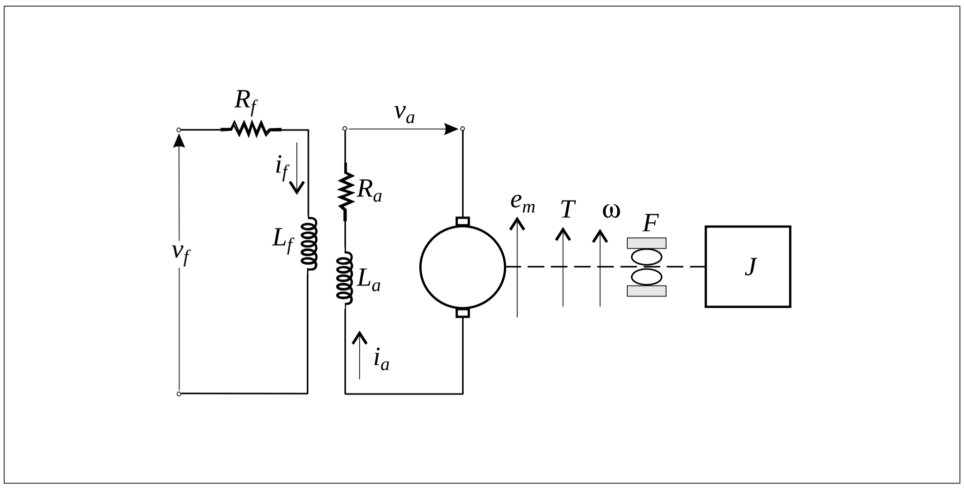

Figure 2.5: Schematic of a DC motor

The basic equations of a DC motor (electric part) are obtained from Maxwell's electromagnetic theory. The *magnetic flux* $\phi(t)$ is proportional to the field current $i_f(t)$

$$\phi(t) = k_1 i_f(t) \tag{2.22}$$

The *torque* produced by a motor is proportional to the product of the magnetic flux $\phi(t)$ and the armature current $i_a(t)$

$$T(t) = k_2 \phi(t) i_a(t) = k_1 k_2 i_f(t) i_a(t) \tag{2.23}$$

The motor's electromotive force (induced voltage), denoted by $e_m(t)$, is proportional to the product of the magnetic flux and the rotor shaft rotational speed $\omega(t)$

$$e_m(t) = k_3 \phi(t) \omega(t) = k_1 k_3 i_f(t) \omega(t) \tag{2.24}$$

The last three equations are valid if the values of $\phi(t)$, $T(t)$, and $e_m(t)$ are close to their nominal values.

On the mechanical side, the torque $T(t)$, developed by the motor, is balanced by the load and disturbance torques. $T(t)$ is also related to the rotational speed $\omega(t)$ by the differential equation

$$T(t) = T_l(t) + T_d(t) = J\frac{d\omega(t)}{dt} + F\omega(t) + T_d(t) \tag{2.25}$$

where J is the combined load and armature mass moment, and F is the viscous friction coefficient. $T_l(t)$ represents the load torque and $T_d(t)$ is a disturbance torque (T_d is frequently negligible).

Balancing the voltages in the field and armature windings, we obtain

$$v_f(t) = L_f\frac{di_f(t)}{dt} + R_f i_f(t) \tag{2.26}$$

$$\begin{aligned} v_a(t) &= L_a\frac{di_a(t)}{dt} + R_a i_a(t) + e_m(t) \\ &= L_a\frac{di_a(t)}{dt} + R_a i_a(t) + k_1 k_3 i_f(t)\omega(t) \end{aligned} \tag{2.27}$$

The above set of equations is nonlinear due to the presence of the products $i_f(t)i_a(t)$ and $i_f(t)\omega(t)$. However, usually one of the currents is kept constant. For constant $i_f(t)$, we have the so-called *armature-controlled* DC motor; if $i_a(t)$ is constant, then the motor is said to be *field-controlled*. Mathematical models for these two regimes are different. They are presented below.

Armature-Controlled DC Motor

In this case $i_f(t) = I_{fo} = const$, so that $v_f(t) = R_f I_{fo}$. Balancing the voltages and torques, we obtain from (2.23), (2.25), and (2.27)

$$v_a(t) = L_a\frac{di_a(t)}{dt} + R_a i_a(t) + k_4\omega(t) \tag{2.28}$$

$$k_5 i_a(t) = J\frac{d\omega(t)}{dt} + F\omega(t) + T_d(t) \tag{2.29}$$

where

$$k_4 = k_1 k_3 I_{fo}, \qquad k_5 = k_1 k_2 I_{fo}$$

Quantities $\tau_a = L_a/R_a$ and $\tau_m = J/F$ are usually called the system time constants. The Laplace transform of the above system of equations produces

$$I_a(s) = \frac{1}{(L_a s + R_a)}[V_a(s) - k_4\Omega(s)] \tag{2.30}$$

$$\Omega(s) = \frac{1}{(Js + F)}[k_5 I_a(s) - T_d(s)] \tag{2.31}$$

The block diagram for this system is easily drawn by looking at equations (2.30) and (2.31) and using the *main block diagram rule*. This is shown in Figure 2.6.

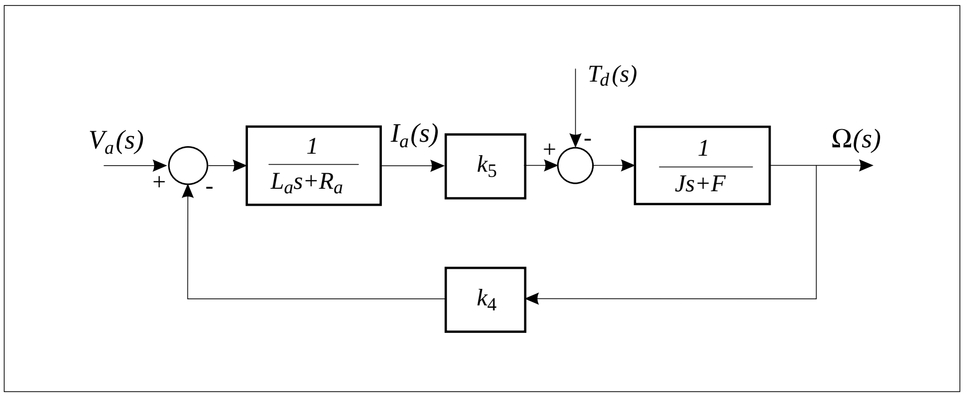

Figure 2.6: Block diagram for an armature-controlled DC motor

Field-Controlled DC Motor

In this case $i_a = I_{ao} = const$, the following differential equation is obtained from (2.23) and (2.25)

$$k_6 i_f(t) = J\frac{d\omega(t)}{dt} + F\omega(t) + T_d(t), \quad k_6 = k_1 k_2 I_{ao} \tag{2.32}$$

Taking the Laplace transforms of (2.26) and (2.32), we get

$$\begin{gathered} V_f(s) = R_f(\tau_f s + 1)I_f(s), \qquad \tau_f = \frac{L_f}{R_f} \\ k_6 I_f(s) = F(\tau_m s + 1)\Omega(s) + T_d(s) \end{gathered} \tag{2.33}$$

where τ_f is a time constant. Rearranging these equations in the form

$$\begin{gathered} I_f(s) = \frac{1}{R_f(\tau_f s + 1)}V_f(s) \\ \Omega(s) = \frac{1}{F(\tau_m s + 1)}[k_6 I_f(s) - T_d(s)] \end{gathered} \tag{2.34}$$

and using the *main block diagram rule*, the block diagram is easily drawn and is represented in Figure 2.7.

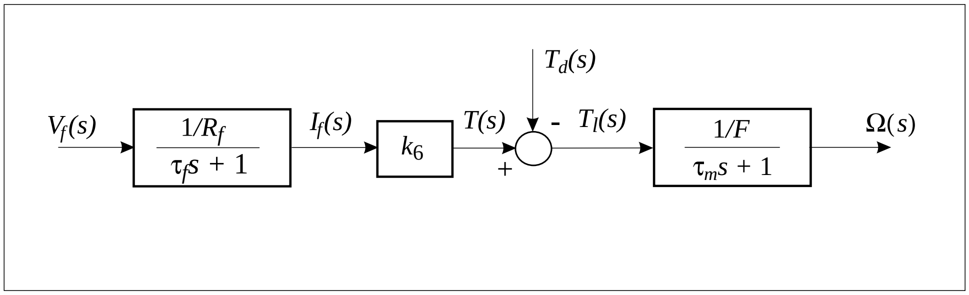

Figure 2.7: Block diagram for a field-controlled DC motor

Real control systems can have complex structures including several local feedback loops, and many inputs and outputs. Local feedback loops can arise for two reasons: they are either the result of the physical nature of the specific element in the system or the system as an entity, or they can be intentionally built in, with the aim of achieving a desirable performance for the system. No matter how complex the block diagram of a system is, it can be reduced to one of the basic structures given in Figures 2.1–2.3 by using the block diagram algebra rules. In the following we present the main results of the block diagram algebra.

2.3 Block Diagram Algebra

Block diagram algebra is a set of rules that facilitates modification and simplification of block diagrams. The rules of block diagram algebra for continuous-time systems are quite simple. They are based on simple principles of algebra that are used for writing input–output relations for the specific blocks in the block diagram.

Cascade (serial) connection: The transfer function equivalent to a serial connection of n blocks with transfer functions $G_1(s), G_2(s), ..., G_n(s)$, represented in Figure 2.8, is given by

$$G(s) = G_1(s)G_2(s)\cdots G_n(s) = \prod_{i=1}^{n} G_i(s) \tag{2.35}$$

This rule is obtained by generalizing the main block diagram rule.

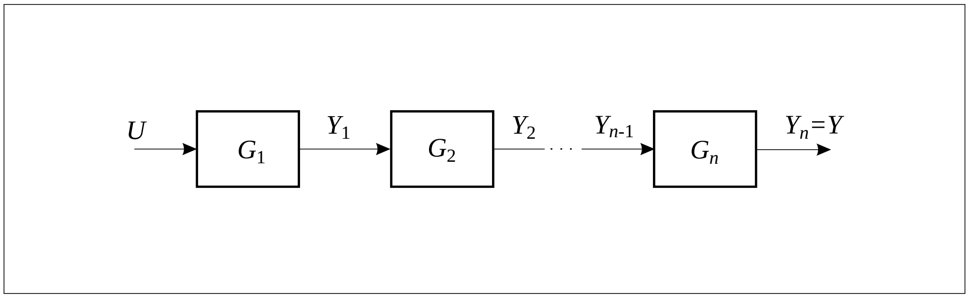

Figure 2.8 A serial connection of n blocks

Parallel (tandem) connection: The equivalent transfer function for such a connection representing a summation of signals, given in Figure 2.9, is obtained as

$$G(s) = G_1(s) + G_2(s) + \cdots + G_n(s) = \sum_{i=1}^{n} G_i(s) \tag{2.36}$$

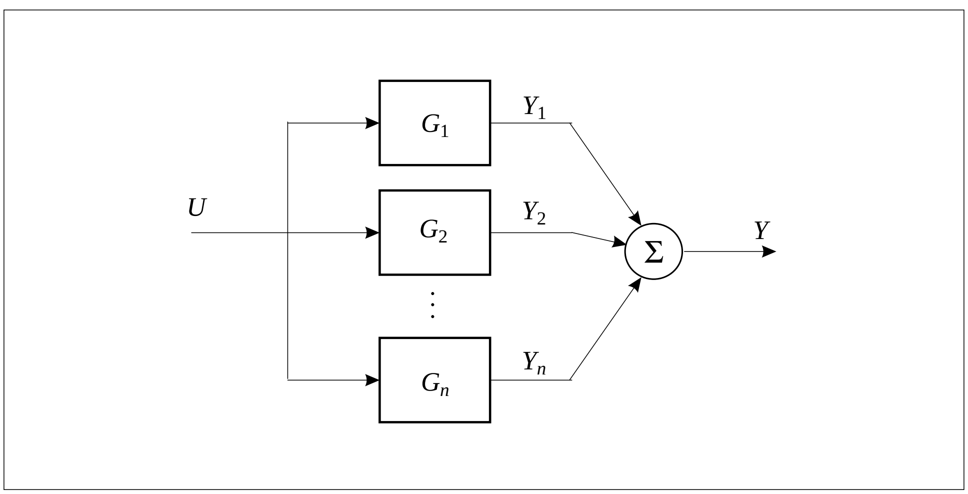

Figure 2.9: Parallel connection of n blocks

Feedback connection: The simplest form of a feedback control system is given in Figure 2.2. For such a system connection the transfer function is given by (2.15).

Example 2.5: In this example we demonstrate the procedure for obtaining the transfer functions for cascade, parallel, and feedback connections by using MATLAB. Consider the transfer functions

$$G_1(s) = \frac{5}{s(s+1)(s+2)}, \quad G_2(s) = \frac{s+4}{s+5}$$

We obtain the *cascade* (series) transfer function by the following sequence of MATLAB operators

```
% define G1(s)
z1=[inf;inf;inf];
% all three zeros of G1(s) are at infinity
p1=[0;-1;–2];
k1=5;
[n1,d1]=zp2tf(z1,p1,k1);
% zp2tf maps zero-pole transfer function into
        numerator-denominator transfer function
% define G2(s)
n2=[1 4];
d2=[1 5];
% find the series connection
[ns,ds]=series(n1,d1,n2,d2);
% print
printsys(ns,ds,’s’)
```

Execution of these operators produces the following result

$$G_1(s)G_2(s) = \frac{5s+20}{s^4+8s^3+17s^2+10s}$$

The transfer function for *parallel* connection is obtained by using

```
[np,dp]=parallel(n1,d1,n2,d2);
printsys(np,dp,’s’)
```

This produces

$$G_1(s)+G_2(s) = \frac{s^4+7s^3+14s^2+13s+25}{s^4+8s^3+17s^2+10s}$$

The *feedback* connection is executed by

```
[nf,df]=feedback(n1,d1,n2,d2,-1);
% –1 indicates negative feedback
printsys(nf,df,'s')
```

which leads to

$$\frac{G_1(s)}{1+G_1(s)G_2(s)} = \frac{5s+25}{s^4+8s^3+17s^2+15s+20}$$

◇

In addition to algebraic formulas (2.35) and (2.36), block diagram algebra is complemented by several "geometric" rules. Two of them are given below.

Moving Pick-Off Point: In some cases it is desirable to move a pick-off point in front or behind a block in a block diagram, such that the terminal signals do not change their values. Figure 2.10 shows the equivalent block diagrams for the cases before and after replacement of the pick-off points.

Moving Summing Point: An adder or subtracter may be moved from one side of a block to another as Figure 2.11 illustrates. It is easy to show that the diagrams on the left-hand and right-hand sides are equivalent. We leave the proofs to the reader.

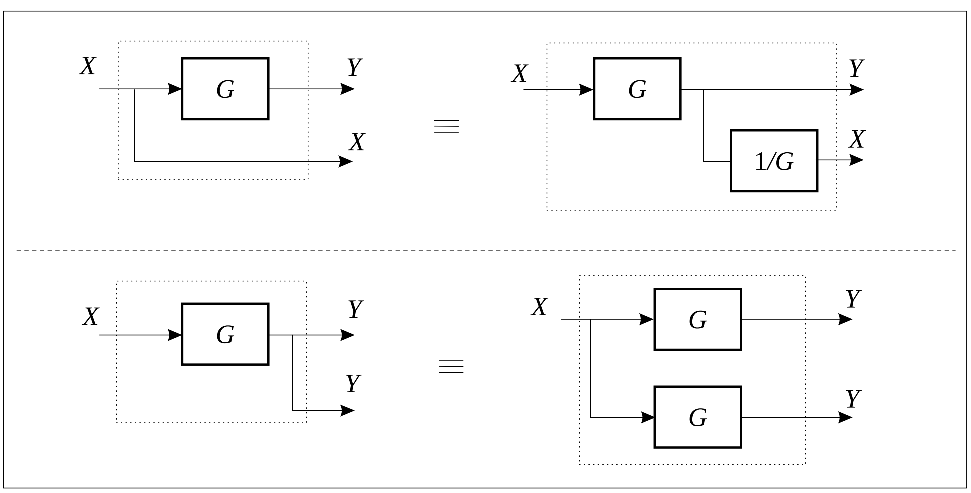

Figure 2.10: Moving pick-off point transformation

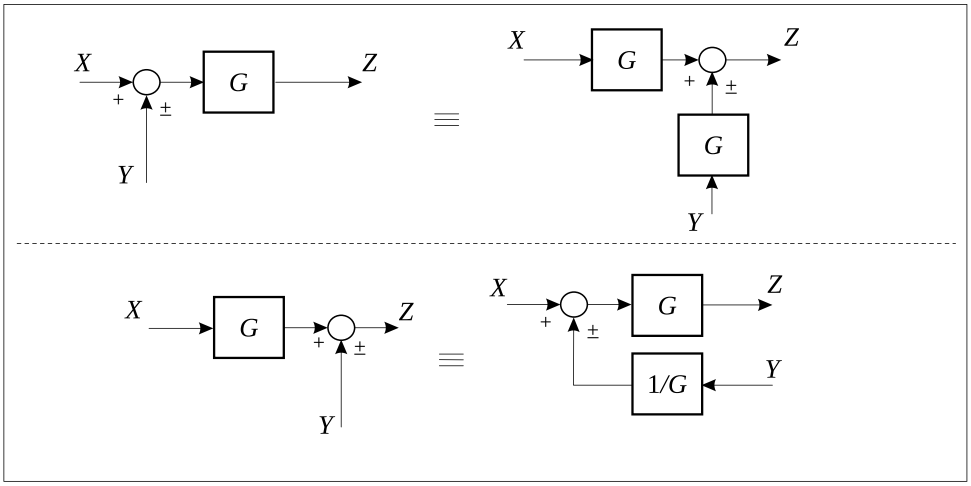

Figure 2.11: Moving adder/subtracter transformation

Using formulas (2.35) and (2.36) and moving pick-off and summing point rules, additional block diagram algebra rules can be derived. A summary of some additional useful block diagram algebra "geometric" rules is given in Table 2.1.

Next we solve two simple examples in order to demonstrate the procedure for finding the system transfer function from block diagrams.

Example 2.6: The original block diagram, presented in Figure 2.12a, is first simplified by moving the adder in front of the block G_1 (Figure 2.12b), then by interchanging adder and adder/subtracter (Figure 2.12c), and finally by finding the corresponding closed-loop transfer functions (Figure 2.12d).

◇

Example 2.7: The block diagram from Figure 2.13a is redrawn in Figure 2.13b in order to explicitly indicate block connections and signal flows. In the next step, presented in Figure 2.13c, the closed-loop transfer function of blocks G_1 and H_2 is found and the pick-off point is moved in front of block G_2. Finally, two closed-loop transfer functions are found (Figure 2.13d) and their cascaded connection is evaluated (Figure 2.13e).

◇

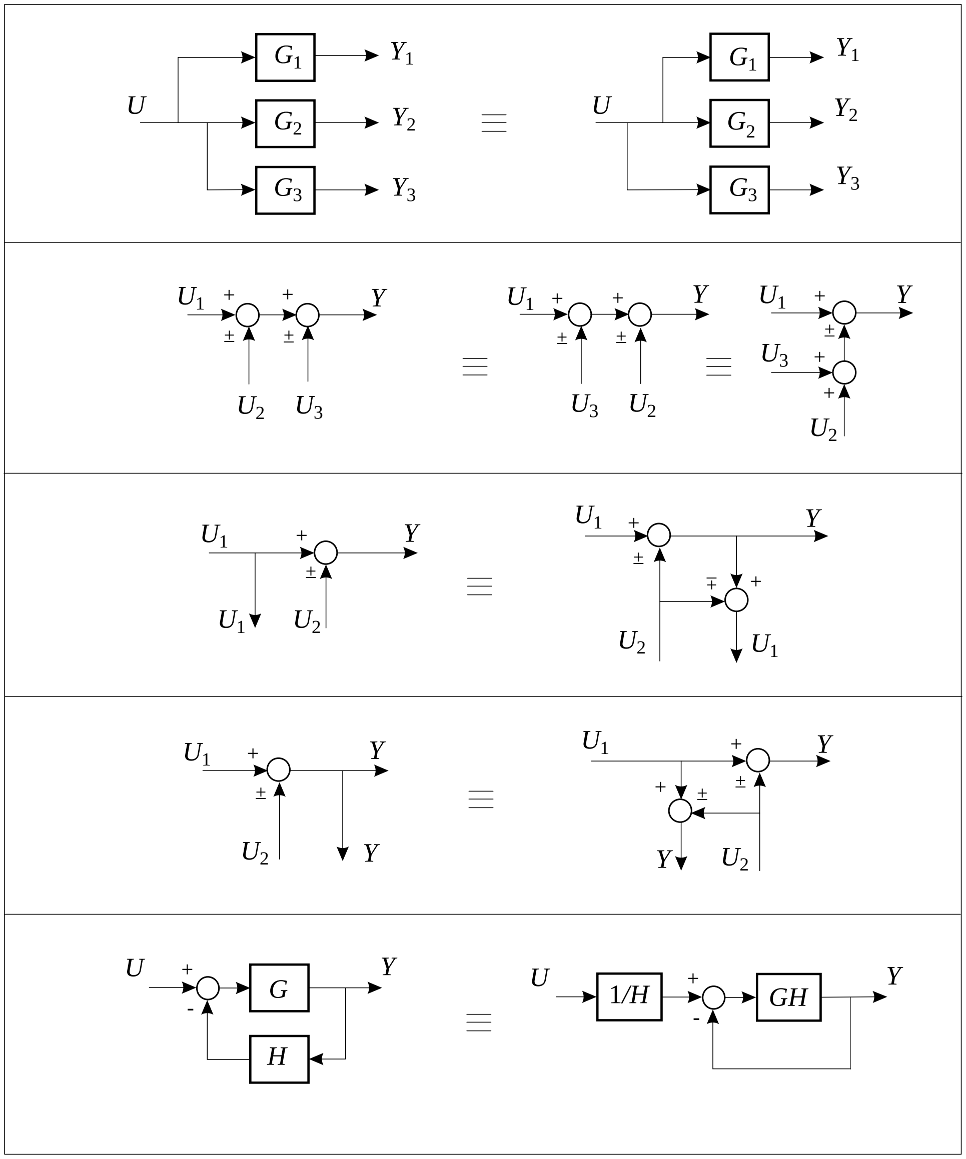

Table 2.1: Block diagram algebra rules

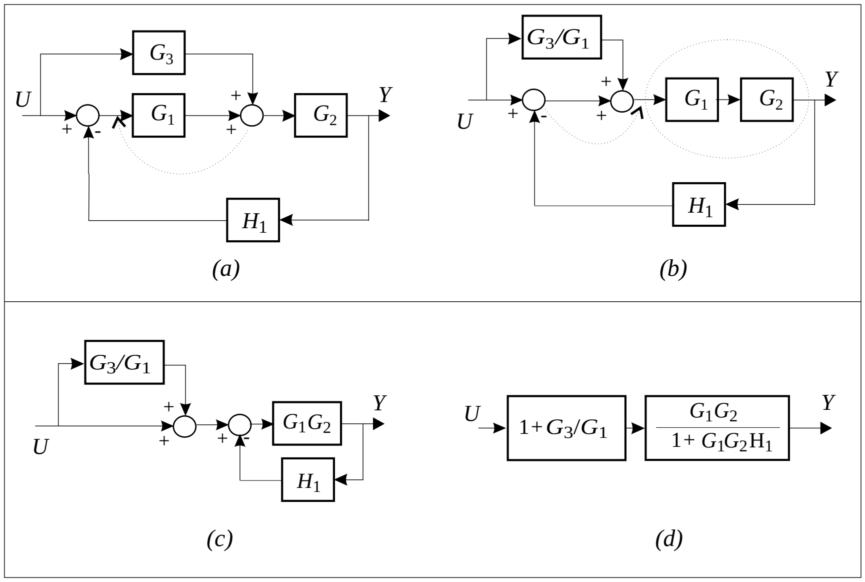

Figure 2.12: Simplification of the block diagram in Example 2.6

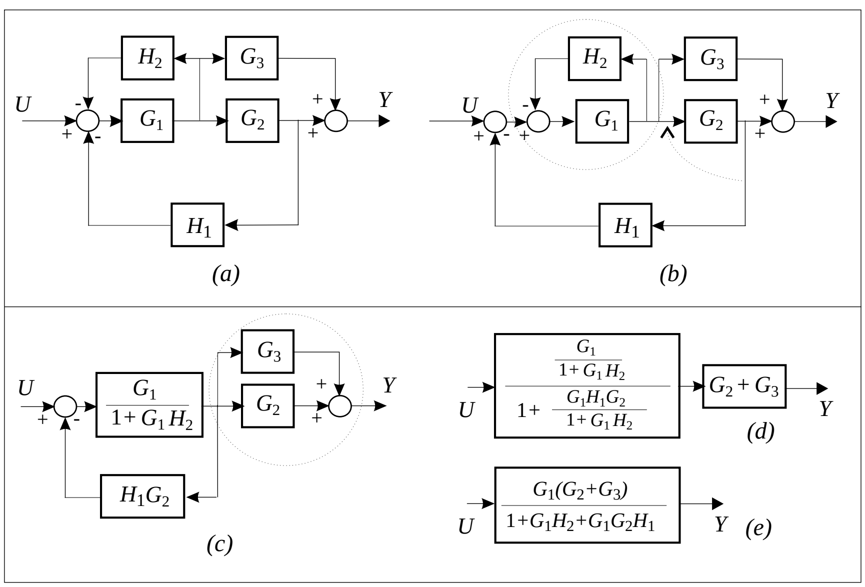

Figure 2.13: Simplification of the block diagram in Example 2.7

Using the rules for simplifying the block diagrams presented in formulas (2.35) and (2.36), Figures 2.10 and 2.11, and Table 2.1, and finding the corresponding transfer functions is relatively easy for simple systems. However, for complex systems the procedure can be quite involved since it requires drawing of many intermediate block diagrams before the final (simple feedback form) is reached. Example 2.8 demonstrates the required procedure for a complex feedback system.

Example 2.8: The reduction of a complex block diagram for a system shown in Figure 2.14a is illustrated in Figures 2.14b–i. The process of reduction is pretty much self-explanatory from the corresponding Figures 2.14b–i. The above simplification is primarily done by using the established rules, but in addition, one has to use common sense, as was done in going from Figure 2.14b to Figure 2.14c and from Figure 2.14c to Figure 2.14d (see Problems 2.12 and 2.13).

The final expression for the transfer function is given by

$$\frac{Y(s)}{U(s)} = \frac{G_1G_3G_4(G_2+G_5)}{1+G_2(H_1G_1+H_2G_3)+H_3G_1G_3G_4(G_2+G_5)-H_1H_2G_1G_2G_3G_5}$$

It can be seen from this particular example that for complex systems the block diagram algebra produces the required answer after many redrawings of the original block diagram.

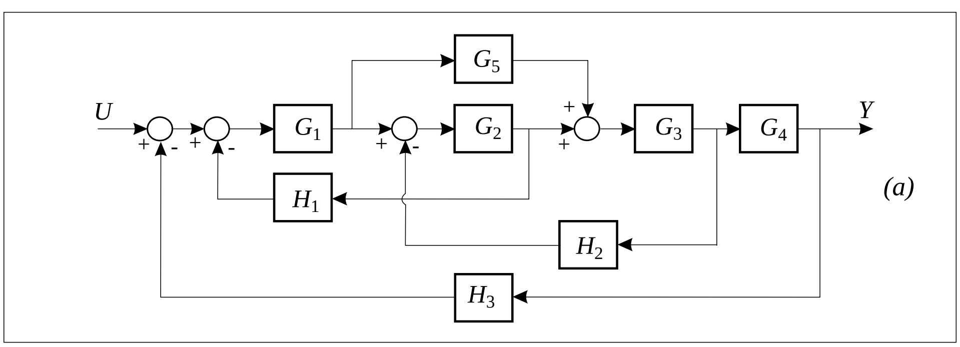

Figure 2.14a: Block diagram of the control system for Example 2.8

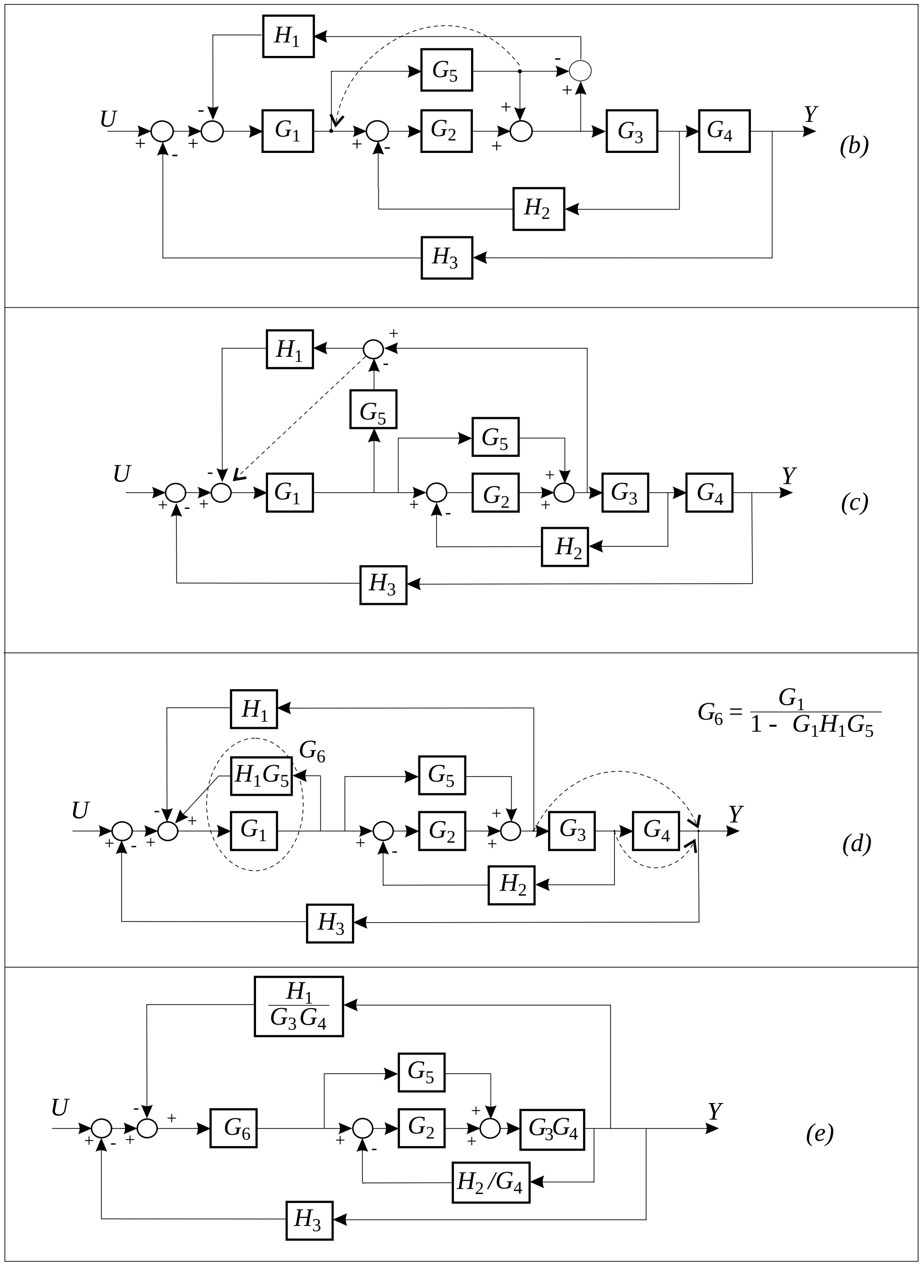

Figure 2.14b–e: Simplification of the block diagram from Figure 2.14a

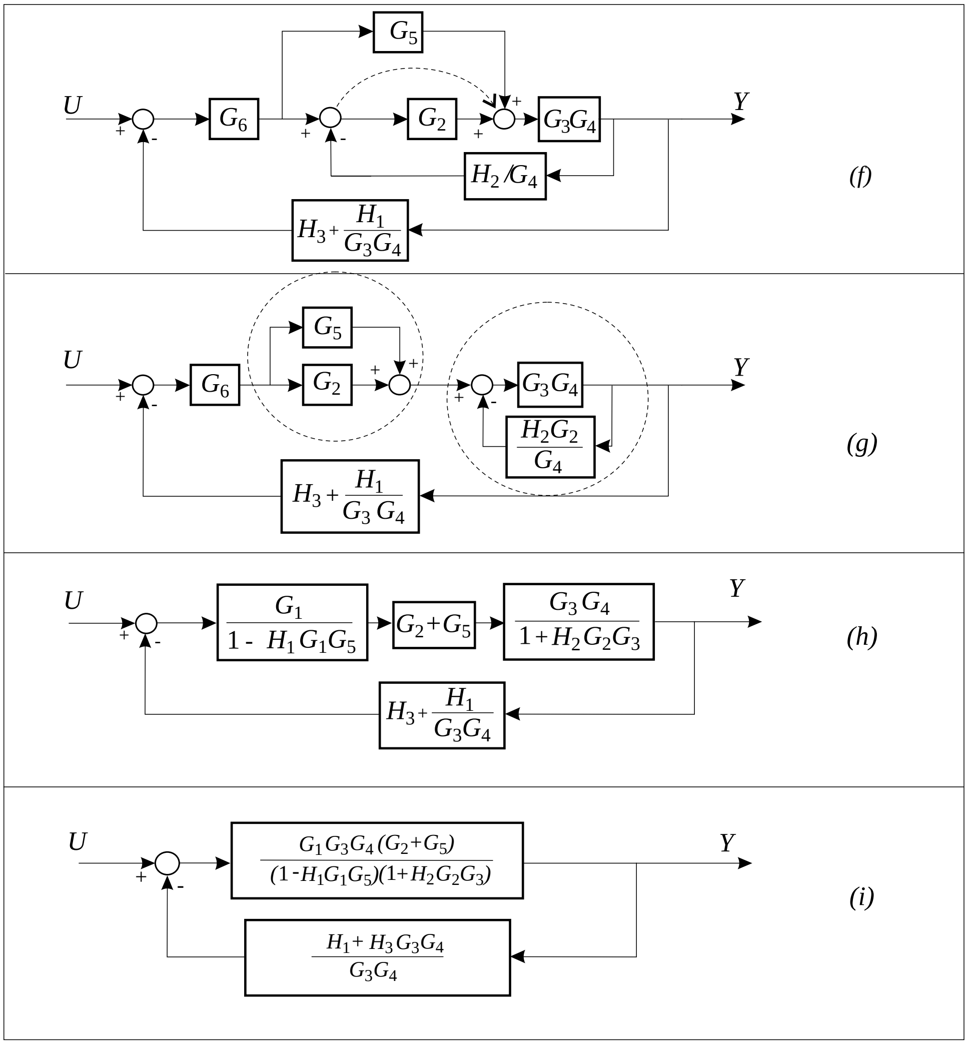

Figure 2.14f–i: Simplification of the block diagram from Figure 2.14a

◇

In the next section we present another method, based on signal flow graph theory, known as Mason's rule, which is particularly efficient for high-dimensional and complex systems.

2.4 Signal Flow Graphs and Mason's Rule

Another way of finding transfer functions of linear time invariant systems represented by their block diagrams is the so-called Mason's rule, which is based on signal flow graph theory (Mason, 1953, 1956). Signal flow graph techniques are also used in several other areas of engineering and sciences (Robichaud *et al.*, 1962; Rao and Koshy, 1991).

There is an analogy between block diagrams and signal flow graphs. The main elements of the signal flow graph technique are *nodes* and *branches*, with the nodes being connected by branches. *A branch is equivalent to a block in the block diagram and represents the transfer function between the nodes.* A branch consists of input node, output node, and an arrow showing the signal flow direction. A transfer function is associated with each branch. A branch in a signal flow graph and its transfer function counterpart are represented in Figure 2.15.

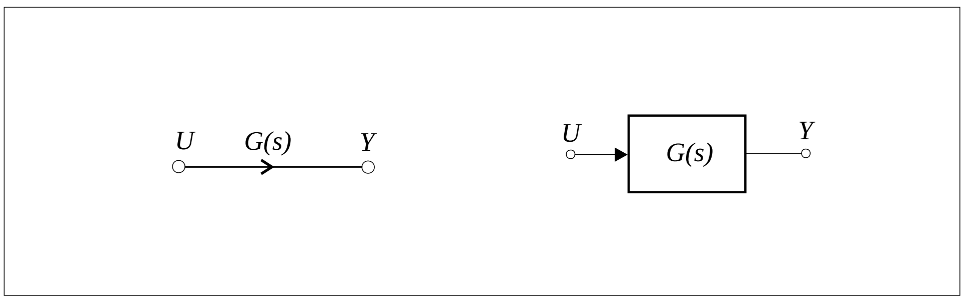

Figure 2.15: Equivalent elements in a block diagram and a signal flow graph

A node represents a signal. The basic rule for nodes is that *a signal at a node is equal to the sum of signals coming into the node from branches*. Note that signals leaving a node do not count. It is only important to pay attention to the signals coming into a node. *A signal entering a node from a branch is equal to the signal from the input node of that branch multiplied by the branch transfer function.*

In Figure 2.16 a simple feedback block diagram and its signal flow graph are given. It can be seen from this figure that the expressions for two signals at two nodes are given by

$$\begin{aligned} E &= 1 \times U - HY \\ Y &= GE \end{aligned} \tag{2.37}$$

which produces our familiar closed-loop result

$$Y = GU - GHY \;\Rightarrow\; Y = \frac{G}{1+GH}U \tag{2.38}$$

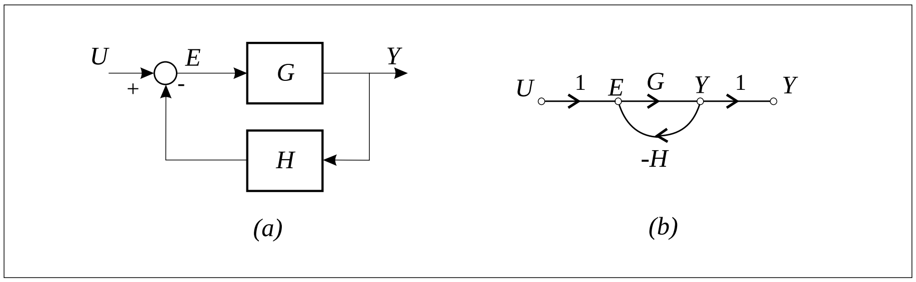

Figure 2.16: Equivalence between closed-loop systems

In order to be able to present the method for finding transfer functions by the signal flow graph technique, we need to introduce the following terminology.

Source node — a node at which signals flow *only away* from the node. Input signals are represented by such nodes.

Sink node — a node at which signals flow *only towards* the node. These nodes represent output signals. It is customary to extract the inputs and outputs out of a signal flow graph by using additional branches whose transfer functions are equal to 1 (see Figure 2.16b).

Path — a succession of branches from a source node (input) to a sink node (output) with all arrowheads in the same direction *which does not pass any node more than once*. *The path gain is the product of all transfer functions in the path*.

Loop — a closed path of branches with all arrowheads in the same direction in which *no node is encountered more than once*. A source node cannot be a part of a loop since each node in the loop must have at least one branch into the node and at least one branch out of it. *The loop gain is the product of transfer functions of the branches comprising the loop*.

Nontouching loops — two loops are nontouching if they have no common node.

The above notions are demonstrated on a signal flow graph presented in Figure 2.17.

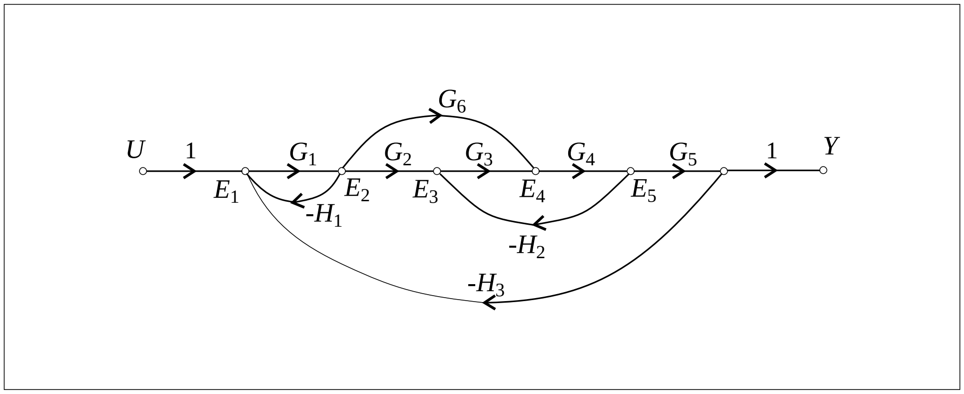

Figure 2.17: Example of a signal flow graph

In this figure the input node is U and the output node is Y. There are two paths connecting the input U and the output Y. One of them has a path gain equal to $G_1G_2G_3G_4G_5$, and the other has a path gain of $G_1G_6G_4G_5$. There are four loops in this signal flow graph with gains $-G_1H_1$, $-G_3G_4H_2$, $-G_1G_2G_3G_4G_5H_3$, and $-G_1G_6G_4G_5H_3$. Two loops, $-G_1H_1$ and $-G_3G_4H_2$, do not touch each other, i.e. they represent nontouching loops.

Note that the signal flow graph contains fewer elements than the corresponding block diagram. Signal flow graphs can be simplified by employing similar rules to those we have been using in block diagram algebra in order to determine system transfer functions. However, we have seen from the examples in the previous section that the simplification procedure based on block diagram algebra is quite lengthy.

In Mason (1953, 1956) an elegant and powerful formula for finding the transfer function between input and output nodes was derived. That formula is known as Mason's gain formula and is given by

$$G(s) = \frac{1}{\Delta}\sum_{k=1}^{N} P_k\Delta_k = \frac{1}{\Delta}(P_1\Delta_1 + P_2\Delta_2 + \cdots + P_N\Delta_N) \tag{2.39}$$

where P_k is the path gain for the kth path, N stands for the number of paths, Δ is the *determinant* of the signal flow graph, and Δ_k is the cofactor of path k.

Δ and Δ_k are computed as follows:

Δ = *1 – (sum of all loop gains) + (sum of products of the loop gains of all possible combinations of nontouching loops taken two at a time) – (sum of products of the loop gains of all possible combinations of nontouching loops taken three at a time) +* $\cdots$.

Δ_k = *value of* Δ *for the part of the flow graph not touching the* k*th forward path*.

The application of formula (2.39) is demonstrated on the signal flow graph presented in Figure 2.17. We have already found that there are two paths with the corresponding gains

$$P_1 = G_1G_2G_3G_4G_5, \qquad P_2 = G_1G_4G_5G_6$$

and four loops whose loop gains are

$$L_1 = -G_1H_1, \quad L_2 = -G_3G_4H_2$$
$$L_3 = -G_1G_2G_3G_4G_5H_3, \quad L_4 = -G_1G_4G_5G_6H_3$$

There are also two nontouching loops with gains L_1 and L_2. Then

$$\Delta = 1 - (L_1 + L_2 + L_3 + L_4) + L_1L_2$$

Apparently, if we eliminate the path $G_1G_2G_3G_4G_5$, the remaining signal flow graph will have no loops left, so that $\Delta_1 = 1$. The same conclusion is obtained if we eliminate the path $G_1G_6G_4G_5$ leading to $\Delta_2 = 1$. Thus, the transfer function of the considered signal flow graph, according to formula (2.39), is given by

$$\frac{Y(s)}{U(s)} = G(s) = \frac{P_1 \times 1 + P_2 \times 1}{1 - (L_1 + L_2 + L_3 + L_4) + L_1L_2} =$$

$$\frac{G_1G_2G_3G_4G_5 + G_1G_4G_5G_6}{1 + G_1H_1 + G_3G_4H_2 + G_1G_2G_3G_4G_5H_3 + G_1G_4G_5G_6H_3 + G_1G_3G_4H_1H_2}$$

In the above expression $G_i, i = 1, ..., 6$, and $H_j, j = 1, 2, 3$, are either constants or functions of the complex frequency s.

Note that all signals in a signal flow graph are mutually related by linear algebraic equations. For example, in the case of the signal flow graph given in

Figure 2.17 we have

$$\begin{aligned} E_1 &= U - H_1 E_2 - H_3 Y \\ E_2 &= G_1 E_1 \\ E_3 &= G_2 E_2 - H_2 E_5 \\ E_4 &= G_6 E_2 + G_3 E_3 \\ E_5 &= G_4 E_4 \\ Y &= G_5 E_5 \end{aligned}$$

where $G_i, i = 1, ..., 6$, and $H_j, j = 1, 2, 3$, are coefficients. By playing simple algebra with the above system of linear equations one is able to obtain the required relationship between Y and U, i.e. the required transfer function. However, that approach is not systematic. Mason's formula (2.39) is derived by using Kramer's determinant method for solving systems of linear algebraic equations, which are obtained by relating signals in a signal flow graph. The complete proof of formula (2.39) is beyond the scope of this textbook; it can be found in Mason (1956) and Younger (1963).

Next, we give an example to find the system transfer function by using both block diagram algebra and Mason's rule.

Example 2.9: Consider the block diagram given in Figure 2.18.

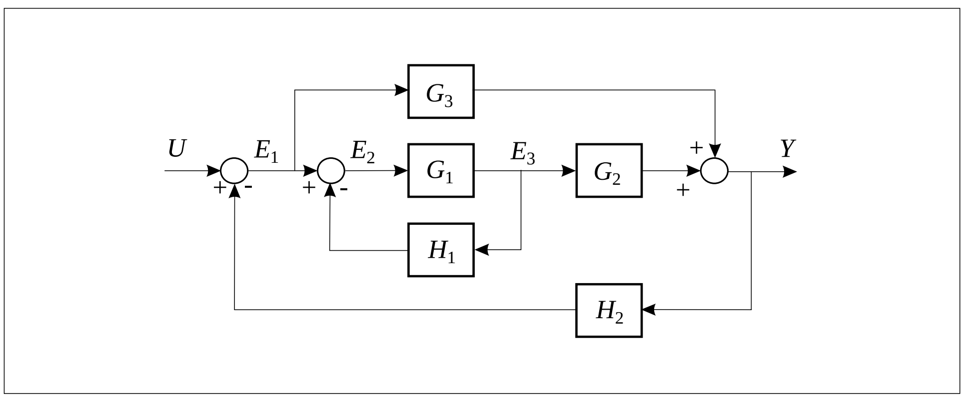

Figure 2.18: Block diagram of a feedback control system

The block transfer functions are given by

$$G_1(s) = \frac{5}{s(s+1)}, \quad G_2(s) = \frac{2}{s}, \quad G_3(s) = 2$$

$$H_1(s) = \frac{s}{s+4}, \quad H_2(s) = \frac{5s}{s+2}$$

(a) Using block diagram algebra rules, this block diagram is simplified as shown in Figure 2.19. The required transfer function is given in Figure 2.19d.

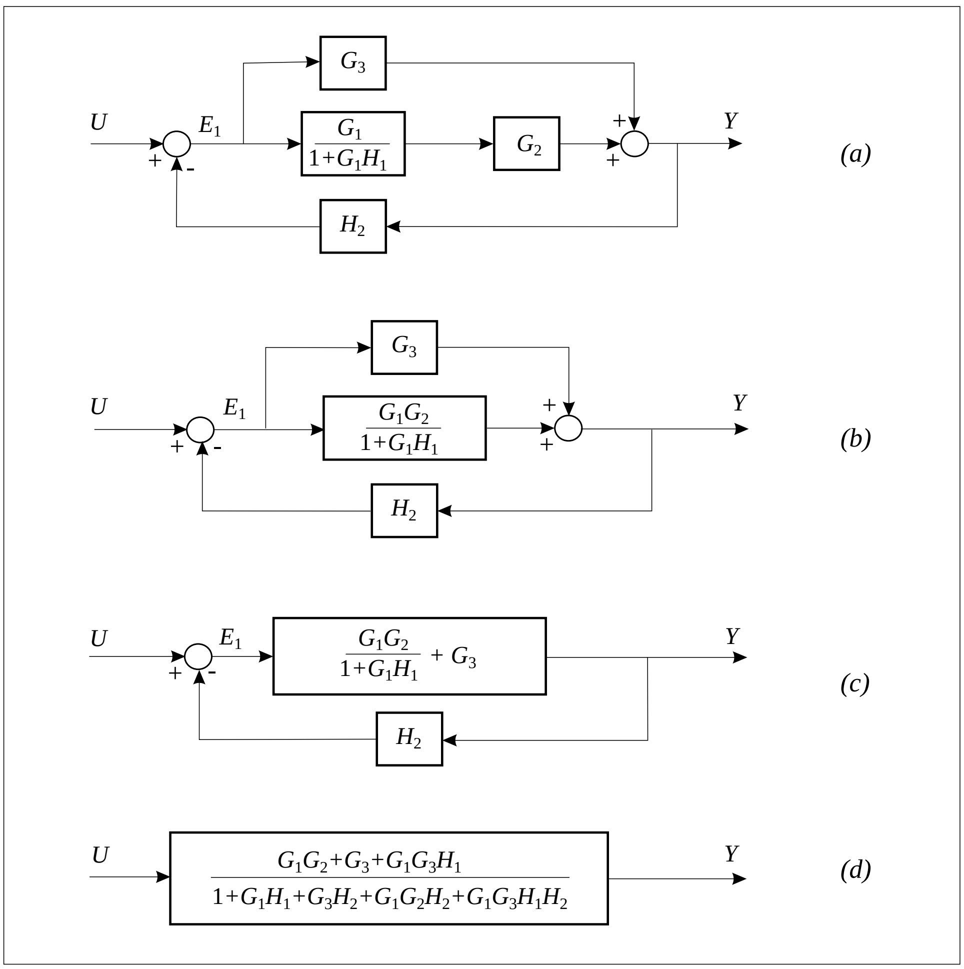

Figure 2.19: Block diagram simplification using block diagram algebra rules

(b) The signal flow graph for this example is represented in Figure 2.20. It contains two paths and three loops. In addition the loops G_1H_1 and G_3H_2 are nontouching ones. Expressions for path and loop gains are

$$P_1 = G_1G_2, \quad P_2 = G_3, \quad L_1 = -G_1H_1, \quad L_2 = -G_3H_2, \quad L_3 = -G_1G_2H_2$$

The graph's determinant and path's cofactors are obtained as

$$\Delta = 1 - (L_1 + L_2 + L_3) + L_1L_2, \quad \Delta_1 = 1, \quad \Delta_2 = 1 - L_1$$

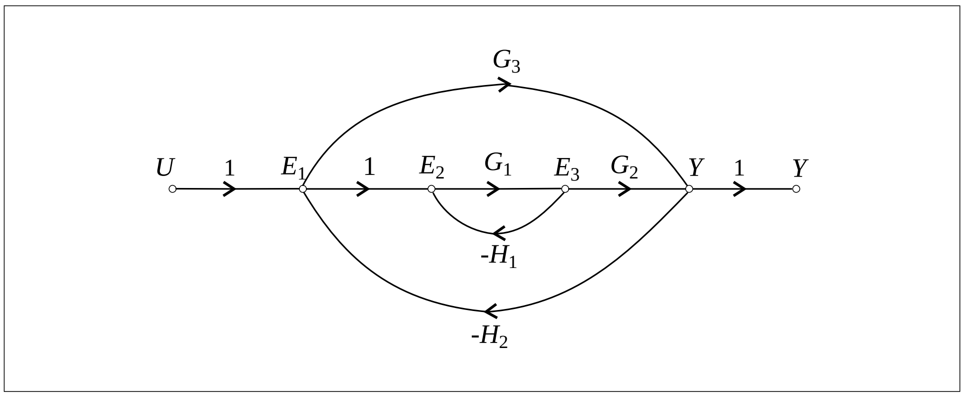

Figure 2.20: Signal flow graph for the system given in Figure 2.18

The required closed-loop transfer function, according to Mason's formula (2.39), is given by

$$\begin{aligned}\frac{Y(s)}{U(s)} &= \frac{P_1 + P_2(1 - L_1)}{1 - (L_1 + L_2 + L_3) + L_1L_2} \\ &= \frac{G_1G_2 + G_3 + G_1G_3H_1}{1 + G_1H_1 + G_3H_2 + G_1G_2H_2 + G_1G_3H_1H_2}\end{aligned}$$

After substitution of the given values for $G_i(s), i = 1, 2, 3$, and $H_j(s), j = 1, 2$, the transfer function is obtained as a ratio of two polynomials with respect to the complex frequency s. The final result for the transfer function is given in Part (c) of this example.

(c) The same problem can be solved by MATLAB using its functions for feedback, parallel, and series connections. Note that the procedure given below cannot be applied to any signal flow graph. It can be applied only to those with explicitly distinguished feedback loops, series, and parallel connections. However, using the SIMULINK package one is able to obtain the transfer function for any block diagram and the corresponding signal flow graph. The transfer function of the feedback control system given in Figure 2.18 is found by using the following sequence of MATLAB instructions

```
% feedback configuration of G1 and H1
[n,d]=feedback([0 0 5],[1 1 0],[1 0],[1 4],-1);
% cascade connection to G2
[n,d]=series(n,d,[0 2],[1 0]);
% parallel connection to G3
[n,d]=parallel(n,d,[2],[1]);
% feedback connection with H2
[n,d]=feedback(n,d,[5 0],[1 2],-1);
printsys(n,d,'s')
```

This MATLAB program produces the following result

$$\frac{Y(s)}{U(s)} = \frac{2s^5 + 14s^4 + 38s^3 + 46s^2 + 60s + 80}{11s^5 + 57s^4 + 109s^3 + 68s^2 + 200s}$$

◇

Example 2.10: Consider the block diagram given in Figure 2.21.

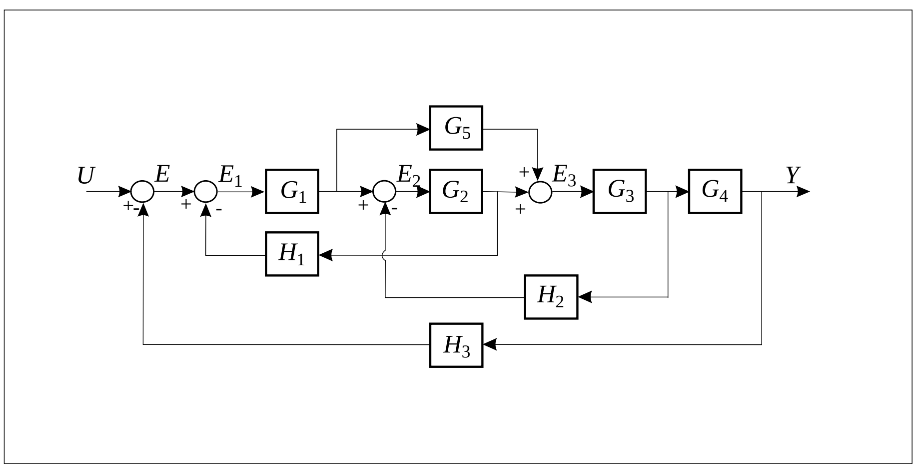

Figure 2.21: Block diagram of a feedback control system

The corresponding signal flow graph is presented in Figure 2.22. In this example we have two paths and five loops. There are no nontouching loops.

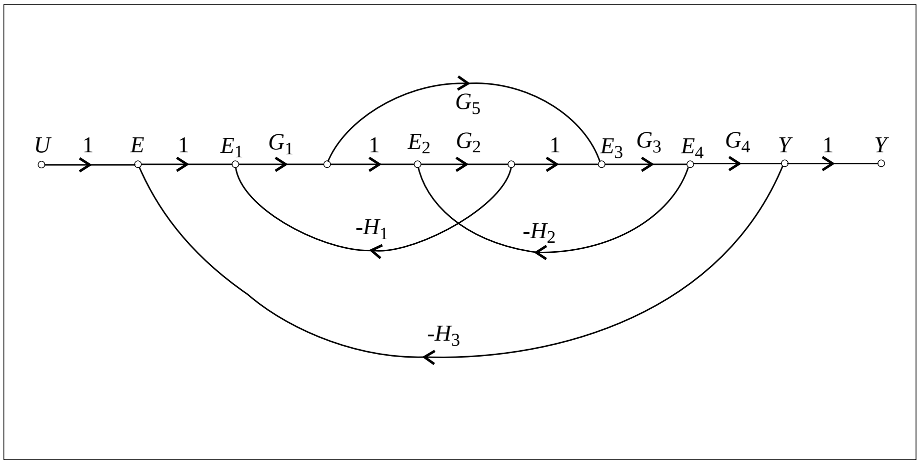

Figure 2.22: Signal flow graph for the system given in Figure 2.21

The corresponding path gains $P_i, i = 1, 2$, loop gains $L_j, j = 1, 2, ..., 5$, signal flow graph determinant, and cofactors are given by

$$P_1 = G_1G_2G_3G_4, \quad P_2 = G_1G_5G_3G_4$$

$$L_1 = -G_1G_2H_1, \quad L_2 = -G_2G_3H_2, \quad L_3 = G_1G_5G_3H_2G_2H_1$$

$$L_4 = -G_1G_5G_3G_4H_3, \quad L_5 = -G_1G_2G_3G_4H_3$$

$$\Delta_1 = 1, \quad \Delta_2 = 1$$

$$\Delta = 1+G_1G_2H_1+G_2G_3H_2+G_1G_2G_3G_4H_3+G_1G_5G_4H_3-G_1G_5G_3H_2G_2H_1$$

so that the closed-loop transfer function for this system is obtained as

$$\frac{Y(s)}{U(s)} =$$

$$\frac{G_1G_2G_3G_4 + G_1G_3G_4G_5}{1 + G_1G_2H_1 + G_2G_3H_2 + G_1G_2G_3G_4H_3 + G_1G_3G_4G_5H_3 - G_1G_2G_3G_5H_1H_2}$$

◇

Note that the same problem is studied in Example 2.8 using block diagram algebra. Comparing the required calculations done in Examples 2.8 and 2.10, it is obvious that for complex systems Mason's rule is much more efficient than the block diagram algebra approach.

It should be pointed out that Mason's rule is also applicable to the signal flow graphs corresponding to multi-input multi-output linear systems (see Problem 2.17d). Mason's rule can be also used for finding transfer functions from any two nodes in the signal flow graph (see Problem 2.14).

Having obtained the system transfer functions, we will be able to design controllers in the frequency domain such that feedback systems satisfy certain specifications, like desired transient and steady state responses. A frequency domain controller design technique based on Bode diagrams will be presented in Chapter 9. Note that Bode diagrams, in fact, represent the frequency plots, for $s = j\omega$, of the magnitude and phase of the system transfer function.

For discrete-time systems that are inherently discrete, duality can be employed and the same rules for finding discrete transfer functions as for continuous-time systems are valid. However, there are some differences in the case of discrete-time systems obtained through sampling (sampled data systems). Transfer functions of sampled data systems are considered in the next section.

2.5 Sampled Data Control Systems[1]

In determining discrete transfer functions of sampled data systems the procedure is a little bit more complex. In some cases, the corresponding transfer function even does not exist since it is impossible to find a linear relationship in the frequency domain between output and input signals.

In Sections 2.5.1 and 2.5.2 we present procedures for finding the basic transfer functions of open-loop and closed-loop sampled data control systems. Section 2.5.3 studies the closed-loop transfer function for a special class of sampled data control systems known as digital computer controlled systems. Here, we present only the basics. For more information about the sampled data control systems, the reader is referred to specialized books (e.g. Astrom and Wittenmark, 1990; Ogata, 1987; Franklin *et al.*, 1990; Kuo, 1992; Phillips and Nagle, 1995).

[1] This section may be skipped without loss of continuity.

2.5.1 Open-Loop Transfer Functions

Depending on where the samplers are positioned, even for very simple block diagrams, several interesting cases may arise. These are illustrated in Figure 2.23, where T stands for the sampling period.

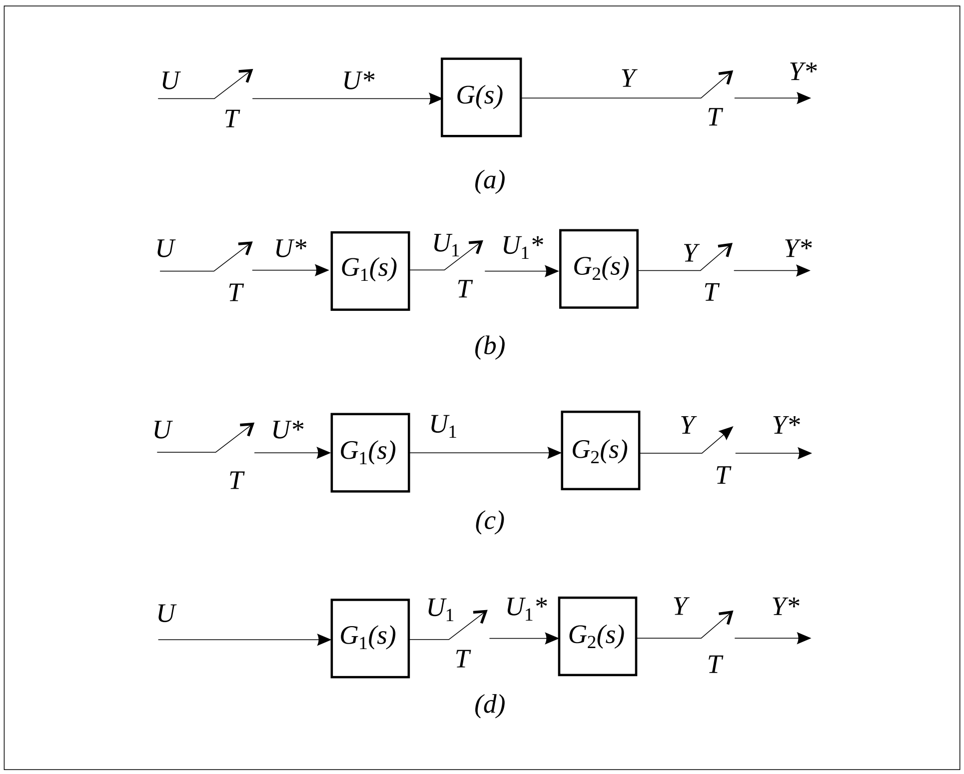

Figure 2.23: Possible cascade connections in a sampled data system

From Figure 2.23a, it follows that

$$
\begin{aligned}
Y(s) &= G(s)U^*(s) \\
Y^*(s) &= [G(s)U^*(s)]^* = G^*(s)U^*(s)
\end{aligned}
\tag{2.40}
$$

Formula (2.40) indicates one of the main properties of the starred Laplace transform defined in Appendix B in (b.11), see also (b.16). Since the starred

Laplace transform of a signal is equal to its $\mathcal{Z}$-transform (see (b.12) in Appendix B), it follows from the above equation that

$$
\begin{aligned}
Y(z) &= G(z)U(z) \\
\frac{Y(z)}{U(z)} &= G(z) = G^*(s)_{|s=\frac{1}{T}\ln z}
\end{aligned}
\tag{2.41}
$$

where T is the sampling period, and $G(z)$ the required transfer function.

Serial connection of two cascaded blocks with samplers, presented in Figure 2.23b, can be described by the following set of equations

$$
\begin{aligned}
U_1(s) &= G_1(s)U^*(s) \\
Y(s) &= G_2(s)U_1^*(s)
\end{aligned}
\tag{2.42}
$$

Using the same procedure as in the previous example, we have

$$
\begin{aligned}
U_1^*(s) &= G_1^*(s)U^*(s) \\
Y^*(s) &= G_2^*(s)U_1^*(s)
\end{aligned}
\tag{2.43}
$$

and

$$
\begin{aligned}
U_1(z) &= G_1(z)U(z) \\
Y(z) &= G_2(z)U_1(z)
\end{aligned}
\tag{2.44}
$$

which yields

$$
Y(z) = G_2(z)G_1(z)U(z) \tag{2.45}
$$

so that the transfer function for this system is

$$
\frac{Y(z)}{U(z)} = G_2(z)G_1(z) \tag{2.46}
$$

Note that in this case the transfer function is equal to the product of the transfer functions of each block, as in the case of two cascaded blocks representing continuous-time systems.

A sampled data system with cascaded elements and no sampler in between is given in Figure 2.23c. Here, we have

$$
Y(s) = G_1(s)G_2(s)U^*(s) \tag{2.47}
$$

The starred Laplace transform of (2.47) gives

$$Y^*(s) = G_1G_2^*(s)U^*(s) \tag{2.48}$$

where $G_1G_2^*(s)$ stands for $[G_1(s)G_2(s)]^*$. Then, it follows

$$\begin{aligned} Y(z) &= G_1G_2(z)U(z) \\ \frac{Y(z)}{U(z)} &= G_1G_2(z) \end{aligned} \tag{2.49}$$

It is important to note that the transfer functions in (2.46) and (2.49) are not the same, i.e. in general

$$G_1(z)G_2(z) \neq G_1G_2(z) \tag{2.50}$$

The last case, given in Figure 2.23d, is a serial connection of two elements with a sampler in between. For such a structure, we have

$$\begin{gathered} U_1(s) = G_1(s)U(s) \ \Rightarrow \ U_1^*(s) = G_1U^*(s) \\ Y(s) = G_2(s)U_1^*(s) \end{gathered} \tag{2.51}$$

Equation (2.51) gives, after $Y(s)$ is starred

$$Y^*(s) = G_2^*(s)G_1U^*(s) \tag{2.52}$$

so that

$$Y(z) = G_2(z)G_1U_1(z) \tag{2.53}$$

It can be seen from (2.53) that, in this case, we are not able to identify the quantity that relates the system input and output in the frequency domain, in other words, *for this particular open-loop sampled data structure the transfer function does not exist.*

The above discussion suggests that similar rules for continuous- and discrete-time block diagrams of the same structure are valid if there exists a sampler in front of each block of a sampled-data system (see Figure 2.23b).

2.5.2 Closed-Loop Transfer Functions

Typical structures for closed-loop sampled data control systems are given in Figure 2.24. We will find that in two cases the closed-loop transfer function

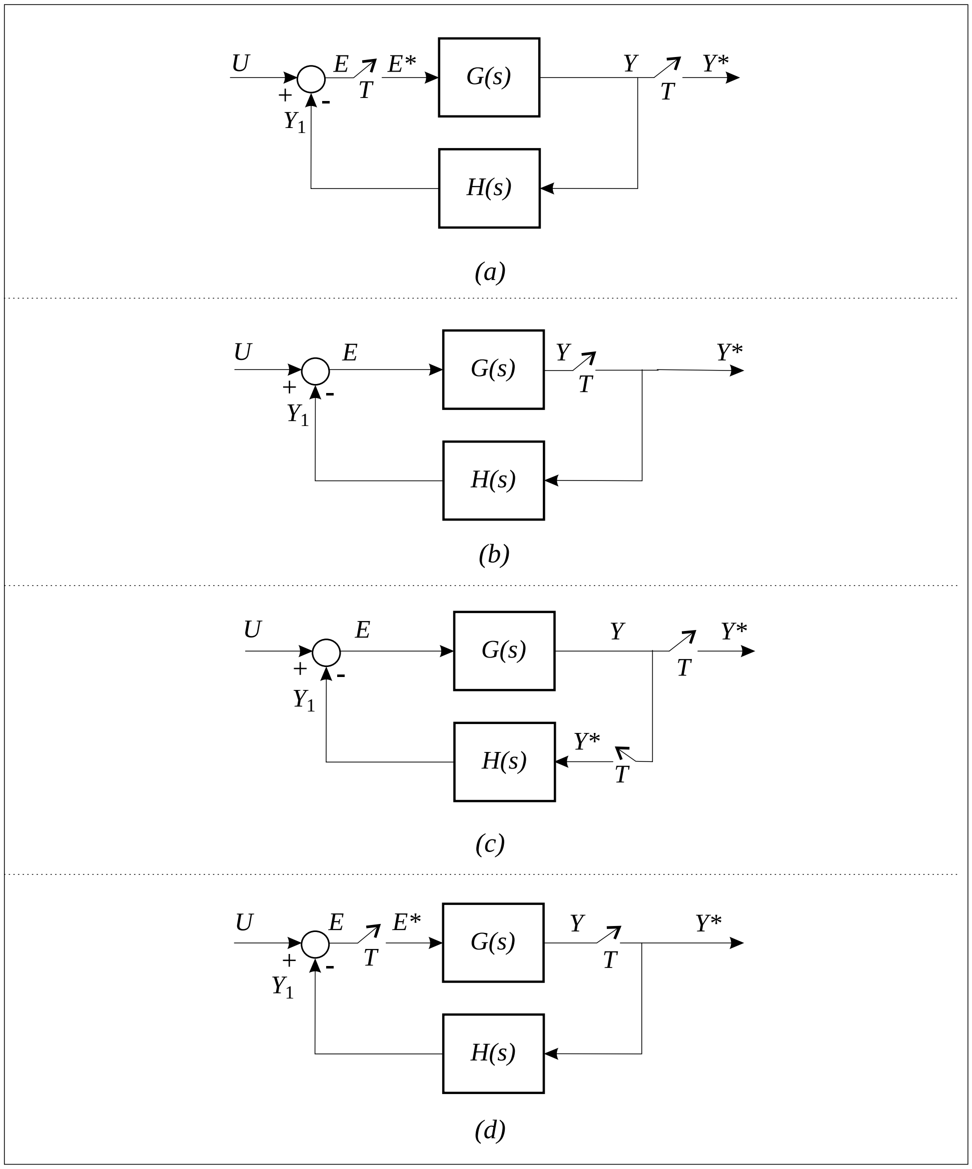

Figure 2.24: Four possible positions of a sampler in a closed-loop system

does not exist, and that in one case it has the same form as for the closed-loop continuous-time system given by (2.15). However, in this case, the transfer function depends on the complex frequency z.

According to Figure 2.24a, we have

$$\begin{gathered} Y(s) = G(s)E^*(s) \\ E(s) = U(s) - Y_1(s) = U(s) - G(s)H(s)E^*(s) \end{gathered} \tag{2.54}$$

By taking the starred Laplace transform, we obtain

$$E^*(s) = U^*(s) - GH^*(s)E^*(s) \tag{2.55}$$

or

$$E^*(s) = \frac{U^*(s)}{1 + GH^*(s)} \tag{2.56}$$

Since

$$Y^*(s) = G^*(s)E^*(s) \tag{2.57}$$

it follows

$$Y^*(s) = \frac{G^*(s)}{1 + GH^*(s)}U^*(s) \tag{2.58}$$

In terms of the $\mathcal{Z}$-transform notation, the output is given by

$$Y(z) = \frac{G(z)}{1 + GH(z)}U(z) \tag{2.59}$$

and the transfer function for the closed-loop system in Figure 2.24a is

$$\frac{Y(z)}{U(z)} = \frac{G(z)}{1 + GH(z)} \tag{2.60}$$

In a similar way, we obtain for the system in Figure 2.24b

$$Y(z) = \frac{GU(z)}{1 + GH(z)} \tag{2.61}$$

which means that *it is not possible to determine the transfer function for this sampled data feedback system configuration.* The same expression for the output as the one in (2.61) is obtained for the system given in Figure 2.24c.

Using the above procedure it is easy to find that the closed-loop system in Figure 2.24d has the transfer function given by

$$\frac{Y(z)}{U(z)} = \frac{G(z)}{1 + G(z)H(z)} \tag{2.62}$$

which corresponds to (2.15), obtained for continuous-time closed-loop systems.

The study of this section shows that *the position of the sampler has a very important role*, because it determines whether or not the input signal $U(s)$ can be separated from the system dynamics. Therefore, we have seen from these examples that the transfer function exists if the sampler is just behind the comparator (subtracter), and conversely, does not exist if the sampler is in any other place in the system.

It is important to note that in the case of sampled data systems, it is forbidden for a sampler and a continuous-time system element in cascade to mutually interchange their positions, because in that case the performance of the control system is changed. In other words, the *commutative law* is not, in general, applicable for such systems.

2.5.3 Transfer Functions of Digital Control Systems

A typical block diagram of a digital control system (digital computer controlled) is shown in Figure 2.25. One part of this configuration is a plant representing a continuous-time process that has to be controlled. Another part is a digital

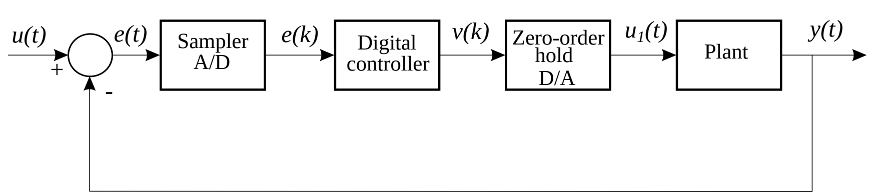

Figure 2.25: Block diagram of a digital control system

controller, usually a digital computer. The interfacing of these two parts is realized by a sampler (A/D converter), which converts analog signals into digital ones, and a zero-order hold (D/A converter), which converts digital signals into continuous-time signals. The continuous-time signals obtained are fed to the plant as control inputs. The block diagram showing transfer functions of all blocks in the above digital control system is given in Figure 2.26.

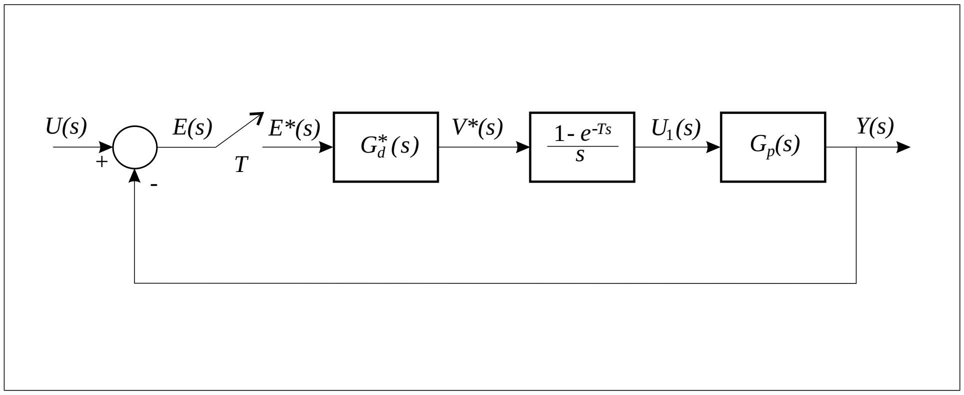

Figure 2.26: Transfer functions in a digital control system

In this system the error signal $e(t) = u(t) - y(t)$ is sampled so that the analog signal $e(t)$ is converted into a digital signal by an A/D device. The digital form of the error signal $e(k)$ is fed to the digital controller, whose transfer function is $G_d(z)$. After the controller has solved the difference equation described by this transfer function, the control signal $v(k)$ is fed to a zero-order hold. The zero-order hold in time domain is represented by a unit pulse of duration T so that its the transfer function is

$$G_h(s) = \mathcal{L}\{h(t) - h(t-T)\} = \frac{1}{s} - \frac{1}{s}e^{-Ts} = \frac{1}{s}\big(1 - e^{-sT}\big) \tag{2.63}$$

The plant transfer function is $G_p(s)$. From Figure 2.26 it follows

$$Y(s) = G_h(s)G_p(s)G_d^*(s)E^*(s) = G(s)G_d^*(s)E^*(s), \quad G(s) = G_h(s)G_p(s) \tag{2.64}$$

By the property of the starred Laplace transform given in (2.40), we have

$$Y^*(s) = G^*(s)G_d^*(s)E^*(s) \tag{2.65}$$

which in the z-domain produces

$$Y(z) = G(z)G_d(z)E(z) \tag{2.66}$$

Since

$$E(z) = U(z) - Y(z) \tag{2.67}$$

the last two equations give the required closed-loop transfer function of the system in Figure 2.26 as

$$\frac{Y(z)}{U(z)} = \frac{G_d(z)G(z)}{1 + G_d(z)G(z)} \tag{2.68}$$

Note that in this introductory control theory course we will not pay attention to sampled data control systems. Material presented in this section is used to demonstrate the procedures for finding transfer functions of a class of linear discrete-time control systems known as sampled-data control systems and to indicate that special care has to be taken while finding the corresponding transfer functions. However, in this book we will study fundamental concepts and methods for discrete-time linear control systems mostly by using dualities with continuous-time linear control systems.

2.6 Transfer Function MATLAB Laboratory Experiment

Part 1. Consider the continuous-time system represented by its transfer function

$$G(s) = \frac{s+1}{s^2 + 5s + 6}$$

Using MATLAB plot:

(a) The impulse response of the system. Use `impulse(num,den,t)` with `t=0:0.1:5`.
(b) The step response of the system. Use the function `step(num,den,t)` with `t=0:0.1:5`.
(c) The system output response due to the input $\sin(2t)$. Use the function `lsim(num,den,u,t)` with `t=0:0.2:20` and `u=sin(2*t)`.
(d) The system output response due to the input e^{-t}. Use the function `lsim(num,den,u,t)` with `t=0:0.1:5` and `u=exp(-t)`.

Part 2. A discrete-time system is described by the transfer function

$$G(z) = \frac{z-2}{z^2 - 2.5z + 1}$$

Using MATLAB plot:

(a) The impulse response of the system. Use `dimpulse(num,den)` and `axis([0 10 0 1.5])`.
(b) The step response of the system. Use `dstep(num,den)` and `axis([0 10 0 3])`.
(c) The response due to the input $\sin(2k)$. Use `dlsim(num,den,u)` with `k=0:1:50` and `u=sin(2*k)`.
(d) The steady state response due to the input 2^{-k}. Use `dlsim(num,den,u)` with `u=2.^(-k)` and `axis([0 11 0 1.1])`. Note that "." after 2 in MATLAB indicates a pointwise operation.

Part 3. Consider a flexible beam system (Qiu and Davison, 1993) whose linearized model has the transfer function

$$G(s) = \frac{1.65s^4 - 0.331s^3 - 576s^2 + 90.6s + 19080}{s^6 + 0.996s^5 + 463s^4 + 97.8s^3 + 12131s^2 + 8.11s}$$

Use MATLAB in order to find:

(a) The system open-loop poles and zeros. Use the functions `roots(num)` and `roots(den)`.
(b) The system closed-loop transfer function assuming unity negative feedback. Use `[numc,denc]=cloop(num,den,-1)`.
(c) The closed-loop poles and zeros and compare them to the open-loop poles and zeros found in (a). Use `roots(numc)` and `roots(denc)`.

Part 4. The block diagram of a simple positioning control system using a field-controlled DC motor is shown in Figure 2.27.

(a) Using the rules of block diagram algebra reduce this system to the basic feedback system shown in Figure 2.2a, and find the system transfer function $Y(s)/U(s)$.
(b) Find the transfer function of this system using Mason's gain formula.
(c) If $K = 2$, $k_1 = -0.05$, $k_2 = 0.16$, and $k_3 = 0.24$, write the MATLAB script to find the transfer function of this system (see Example 2.5).

(d) Using MATLAB functions, find the poles of the closed-loop system.

(e) For the given system poles find the partial fraction expansion using the MATLAB function `residue`. From the partial fraction expansion find analytically the system response to a unit step input. Check the obtained results using the MATLAB function `step`.

(f) Plot the unit step response of the system for the time interval of 10 seconds.

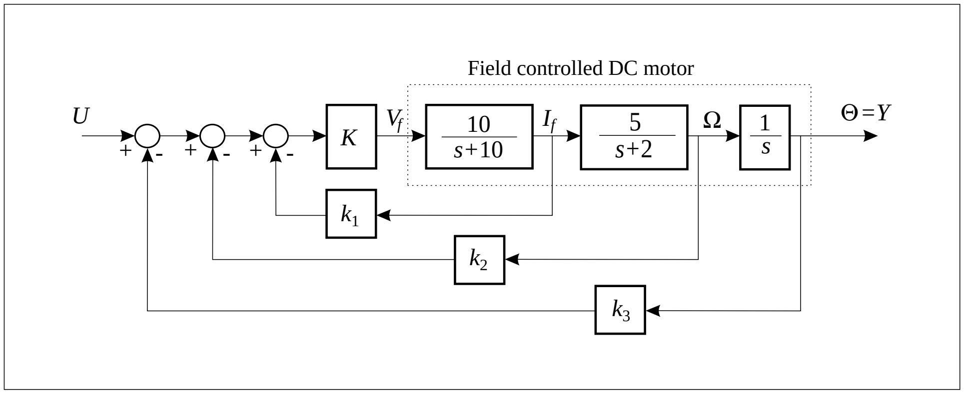

Figure 2.27: Position control system

2.7 References

Astrom, K. and B. Wittenmark, *Computer Controlled Systems: Theory and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Franklin, G., J. Powel, and M. Workman, *Digital Control of Dynamic Systems*, Addison-Wesley, Reading, Massachusetts, 1990.

Kamen, E., *Introduction to Signals and Systems*, Macmillan, New York, 1990.

Kuo, B., *Digital Control Systems*, Saunders College Publishing, New York, 1992.

Kuo, B., *Automatic Control Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Mason, S., "Feedback theory—some properties of signal flow graphs," *Proc. IRE*, vol. 41, 1144–1156, 1953.

Mason, S., "Feedback theory—further properties of signal flow graphs," *Proc. IRE*, vol. 44, 920–926, 1956.

Ogata, K., *Discrete-Time Control Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1987.

Phillips, C. and H. Nagle, *Digital Control System Analysis and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Qiu, L. and E. Davison, "Performance limitations of non-minimum phase systems in the servomechanism problem," *Automatica*, vol. 29, 337–349, 1993.

Rao, M. and G. Koshy, "Mason's formula: A new and simplified approach," *Control—Theory and Advanced Technology*, vol. 7, 355–363, 1991.

Robichaud, L., M. Boisvert, and J. Robert, *Signal Flow Graphs and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1962.

Younger, D., "A simple derivation of Mason's gain formula," *Proc. IEEE*, vol. 51, 1043–1044, 1963.

2.8 Problems

2.1 A control system is described by the following set of equations

$$\frac{d^2y(t)}{dt^2} + 2\frac{dy(t)}{dt} = e(t), \qquad e(t) = u(t) - y(t)$$

(a) Find the transfer function of this system and its impulse response.
(b) Using the inverse Laplace transform, find the response of the system to a unit step input and zero initial conditions.
(c) Using the final value theorem, calculate the steady-state error due to a unit step input, i.e. find $e(t)$ for $t \to \infty$.
(d) Plot the unit step response of this system by using the MATLAB function `step(num,den,t)` for the time interval of 10 seconds. Take `t=0:0.1:10`.

2.2 Consider the electrical circuit given in Figure 2.28.

(a) Find the voltage transfer function $V_0(s)/V_i(s)$.
(b) Suppose that an inductor L_1 is connected in parallel to resistor R_2. Find the voltage transfer function for the modified circuit.

(c) If a constant input voltage $v_i(t) = 5\,\mathrm{V}$ is applied to the circuit, find the output voltages $v_o(t)$ for the circuits given in (a) and (b) by using the inverse Laplace transform.

(d) Using the final value theorem of the Laplace transform, find the steady state values of the output voltages in cases (a) and (b).

(e) If $R_1 = 1, C_1 = 1, R_2 = 2, L_1 = 1$, plot the output $v_o(t)$ for the circuits given in (a) and (b) by using MATLAB.

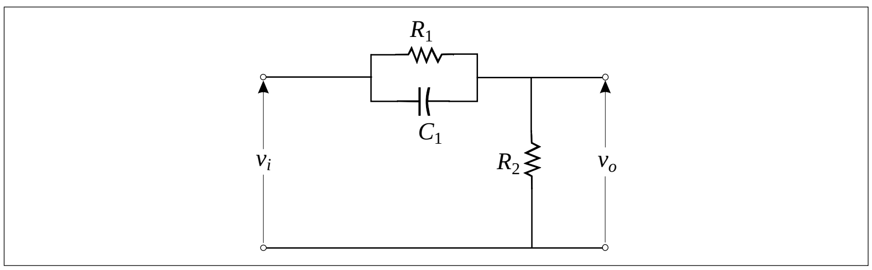

Figure 2.28: An RC network

2.3 Find the transfer function of the following continuous-time system

$$y^{(5)} + 3y^{(4)} + 2y^{(3)} + y^{(2)} + 5y^{(1)} + y = 3u^{(2)} + 2u^{(1)} + u$$

2.4 Consider the continuous-time linear system represented by the transfer function

$$G(s) = \frac{2s^5 + s^3 - 3s^2 + s + 4}{5s^8 + 2s^7 - s^6 - 3s^5 + 5s^4 + 2s^3 - 4s^2 + 2s - 1}$$

Use MATLAB to find:

(a) The zeros and poles of the system.

(b) The inverse Laplace transform of the transfer function. Use the `residue` function.

(c) The system closed-loop transfer function assuming a negative unit feedback and find the corresponding closed-loop poles.

2.5 Consider the control system given in Figure 2.29.

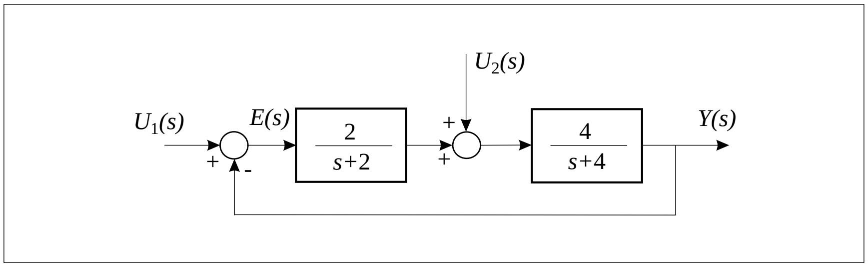

Figure 2.29: A control system with two inputs

(a) Assuming that the initial conditions are zero, find the Laplace transform of the system output due to inputs: $u_1(t) = 2e^{-t}$, $u_2(t) = 4h(t)$, where $h(t)$ is a unit step function.

(b) Find $y(t)$ by using the partial fraction expansion and the inverse Laplace transform.

(c) Using MATLAB, find $y_1(t) = y(t)$ for $u_2(t) = 0$, and $y_2(t) = y(t)$ for $u_1(t) = 0$. Then, using superposition, find $y(t)$ as a response to both $u_1(t)$ and $u_2(t)$. Plot the outputs, $y_1(t), y_2(t)$, and $y(t)$.

2.6 Using the Laplace transform, find the transfer function of the electric network given in Figure 1.7, i.e. find $V_{c_2}(s)/E_i(s)$.

2.7 Find the transfer functions $\Omega(s)/V_a(s)$ and $\Omega(s)/T_d(s)$ of the armature-controlled DC motor given in Figure 2.6.

2.8 Use formula (2.21) to find the closed-loop transfer function of a multivariable system represented by

$$\mathbf{G}(s) = \begin{bmatrix} \frac{s+1}{s^2+6s+8} & \frac{s}{s^2+6s+8} \\ \frac{-s}{s^2+6s+8} & \frac{s+3}{s^2+6s+8} \end{bmatrix}, \quad \mathbf{H}(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{1}{s+2} \end{bmatrix}$$

2.9 The block diagram of a control system is shown in Figure 2.30.

(a) Reduce this system using block diagram algebra rules and find its transfer function $Y(s)/U(s)$.

(b) Draw the signal flow diagram of the system and find the transfer function by Mason's rule.

(c) Find the transfer function of the system using the MATLAB functions `series` and `feedback` repeatedly.

(d) Find the unit step response of this system using MATLAB.

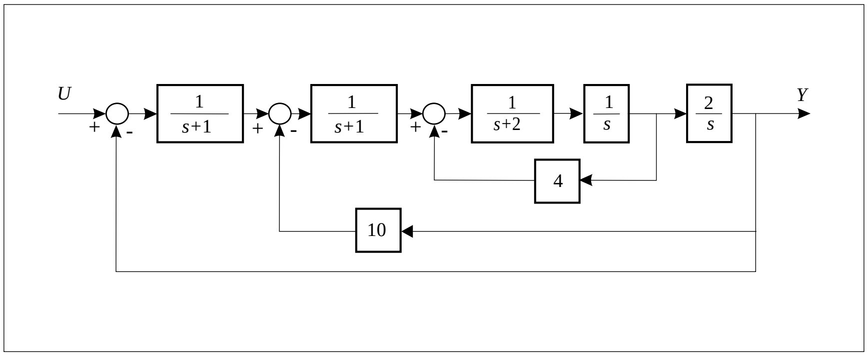

Figure 2.30: Block diagram of a control system

2.10 Using block diagram algebra rules, simplify the block diagrams shown in Figure 2.31 and find their transfer functions (the matrix transfer function for multi-input multi-output case).

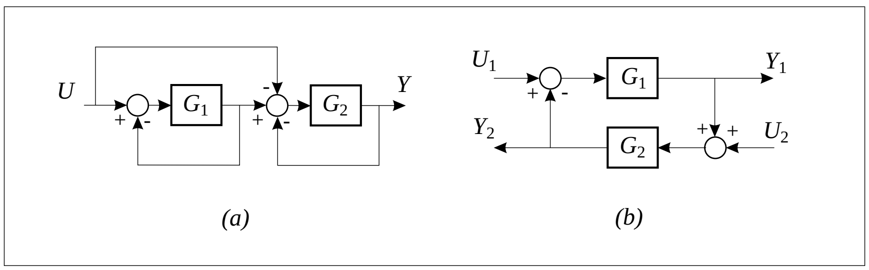

Figure 2.31: Block diagrams

(a) Repeat the procedure using Mason's gain formula.
(b) Find the transfer functions using MATLAB with

$$G_1(s) = \frac{3}{s(s+1)}, \quad G_2(s) = \frac{5}{s^2+2}$$

2.11 Verify that the block diagrams shown in Figure 2.32 are equivalent, i.e. show that they have identical transfer functions $Y(s)/U(s)$ and $Y(s)/D(s)$.

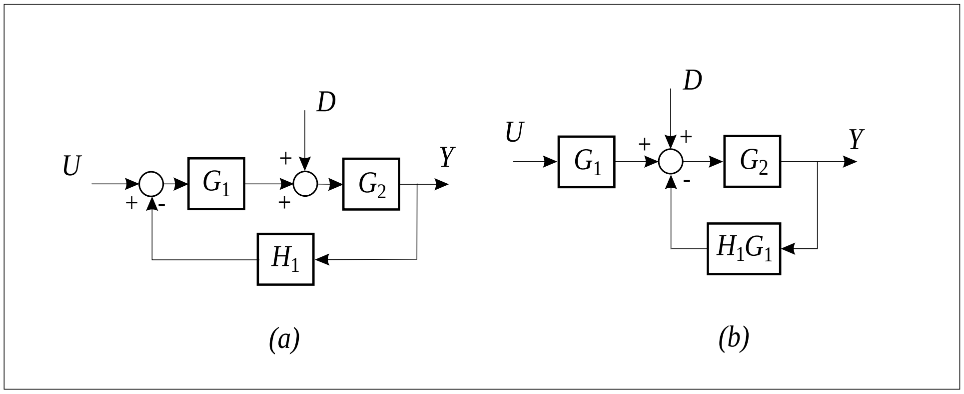

Figure 2.32: Equivalent block diagrams

2.12 Show that the transfer functions of the block diagrams given in Figures 2.33a and 2.33b are identical, i.e. conclude that the block diagrams are equivalent. Note that this equivalence has been used to move from Figure 2.14b to Figure 2.14c in Example 2.8.

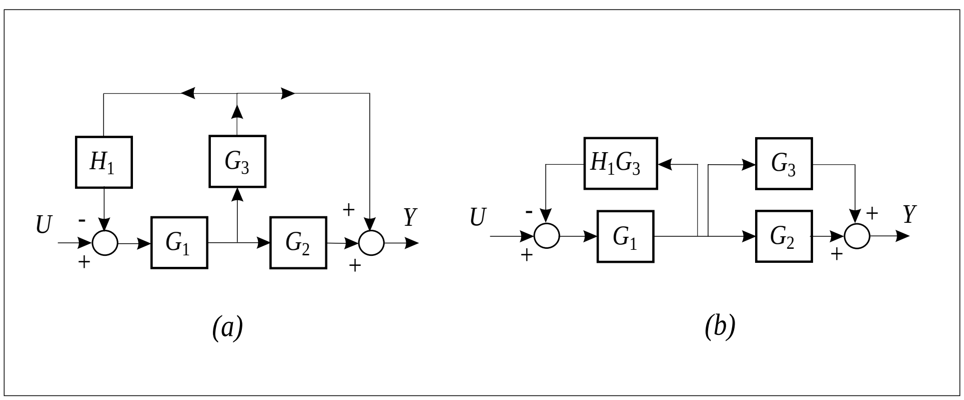

Figure 2.33: Equivalent block diagrams

2.13 Show that both transfer functions $Y(s)/U_1(s)$ and $Y(s)/U_2(s)$ in the block diagrams given in Figures 2.34a and 2.34b are identical, i.e. conclude that the block diagrams are equivalent. Note that this equivalence has been used to move from Figure 2.14c to Figure 2.14d in Example 2.8.

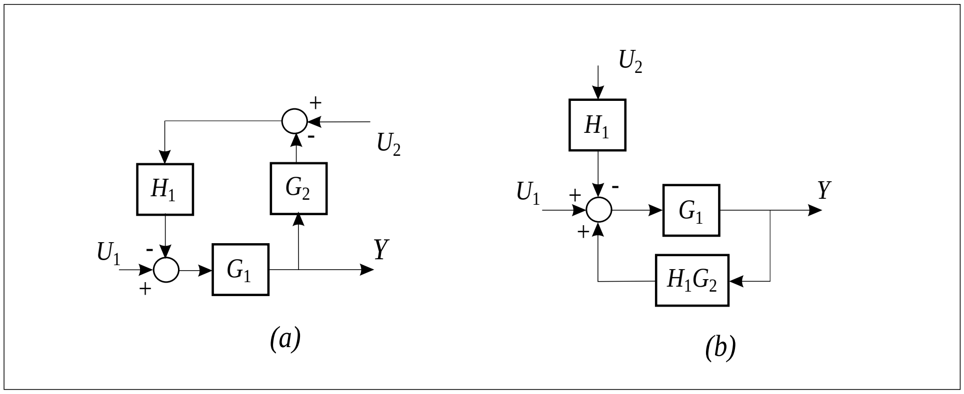

Figure 2.34: Equivalent block diagrams

2.14 Find the transfer function between the nodes E_2 and E_5, i.e. find the ratio $E_5(s)/E_2(s)$ for the signal flow graph given in Figure 2.17.

2.15 SIMULINK can be used for drawing block diagrams. Once a block diagram has been drawn and a name to it given, we can obtain the state space model by invoking the SIMULINK function `linmod('block diagram name')`. From the state space representation, the MATLAB function `ss2tf` produces the corresponding transfer function. Find the transfer functions for the block diagrams considered in Problems 2.9–2.13 by using the SIMULINK package.

2.16 Omit the branch containing block G_5 in the block diagram presented in Figure 2.14a and find the system transfer function by using:

(a) Mason's rule.
(b) Block diagram algebra.

2.17 Using Mason's gain formula, find the transfer functions of the systems whose signal flow graphs are shown in Figure 2.35.

2.18 Consider the discrete-time linear system represented by the transfer function

$$G(z) = \frac{z^4 - 3z^3 + 5z^2 + 2z}{2z^7 + 5z^5 - 3z^4 + z^3 - 2z^2 + 3z - 1}$$

Use MATLAB in order to find:

(a) The zeros and poles of the system.

(b) The inverse $\mathcal{Z}$-transform of the transfer function. Hint: Use the `residue` function.

(c) The closed-loop zeros and poles.

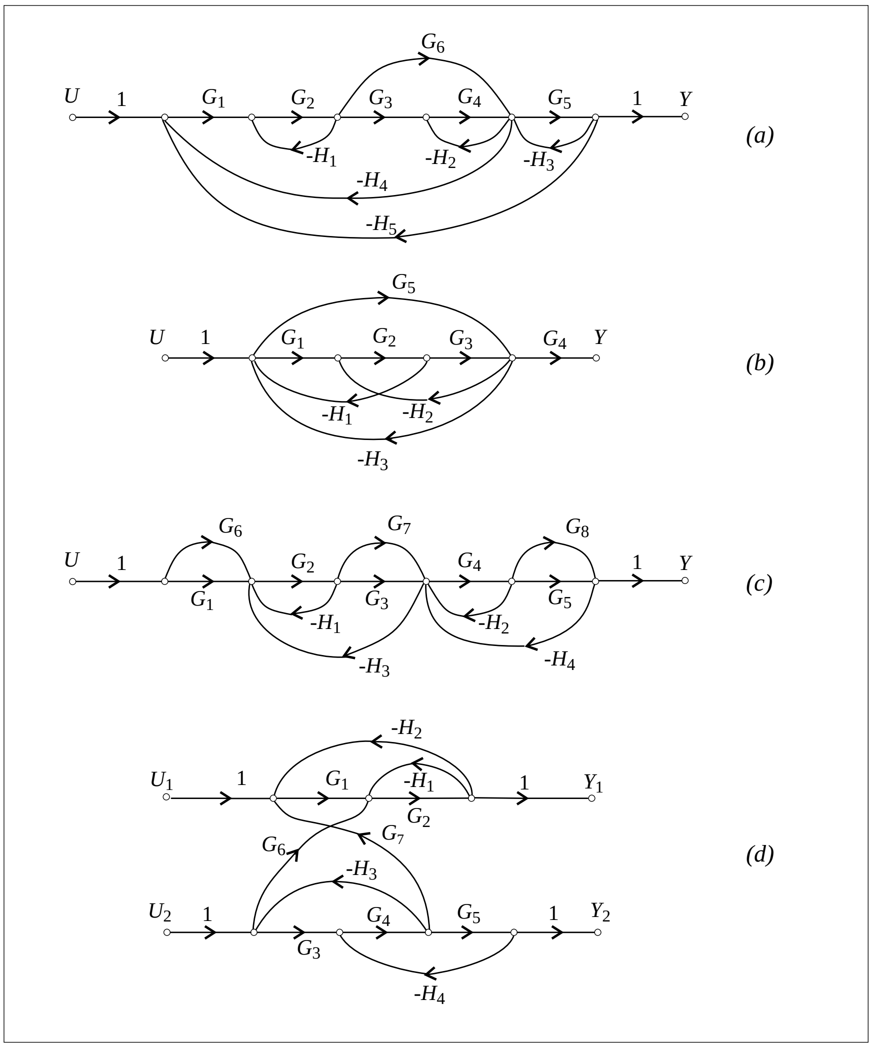

Figure 2.35: Signal flow graphs of control systems

2.19 Find the inverse $\mathcal{Z}$-transform for the following discrete transfer functions

$$H(z) = \frac{3z^2 - z}{z^2 + 2z + 5}, \quad H(z) = \frac{2z^2 + z}{(z + 0.4)(z - 1)(z - 0.2)}$$

Using MATLAB, find the first 10 samples of system outputs $y(k)$ if the system input is $u(k) = 1, k = 0, 1, 2, ...,$ and $u(k) = 0$ for $k < 0$.

2.20 For the digital control system shown in Figure 2.26 the plant transfer function is

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

(a) Find the discrete transfer function of the closed-loop system if the sampling interval is $T = 0.5\,\mathrm{s}$, and the digital controller is

$$G_d(z) = K_p + \frac{K_i}{1 - z^{-1}}$$

(b) Using MATLAB, find the poles and zeros of the closed-loop system if $K_p = 1, K_i = 1$. Plot the output response to a unit step input.