8.3: Realization of Transfer Function Models

8.3.1 Simulation Diagrams

We consider the problem of realizing a given transfer function model as a state variable model. As the choice of state variables is not unique, this can be done in multiple ways. The process is facilitated by first drawing a simulation diagram for the system.

A simulation diagram realizes an ODE model into a block diagram representation using static gains, integrators, summing nodes, and feedback loops. Historically, such diagrams were used to simulate dynamic system models on analog computers.

Given a transfer function, its two common realizations are described below.

**Serial Realization**

A serial chain of integrators can be used to realize the output of an ODE. To illustrate this idea, we consider a general second-order system model described as:

\[
\frac{y(s)}{u(s)} = \frac{K}{s^2 + 2\zeta \omega_n s + \omega_n^2}
\]

The transfer function is converted into an ODE representation by cross multiplying followed by inverse Laplace transform to obtain:

\[
\ddot{y}(t) + 2\zeta \omega_n \dot{y}(t) + \omega_n^2 y(t) = Ku(t)
\]

The above equation is rearranged to form the highest derivative as:
The simulation diagram realizes the highest derivative using a summing node and uses a series of integrators to obtain the lower order derivatives up to the output (Figure 8.3).

**Parallel Realization**

Consider a second-order transfer function with real roots:

\[
\frac{y(s)}{u(s)} = \frac{K}{s^2 + (\sigma_1 + \sigma_2)s + \sigma_1\sigma_2}
\]

The transfer function is expanded into partial fractions expansion (PFE) to obtain:

\[
y(s) = \frac{K_1}{s + \sigma_1}u(s) + \frac{K_2}{s + \sigma_2}u(s)
\]

Thus, in parallel realization, the output is expressed as:

\[
y(t) = y_1(t) + y_2(t) = K_iu(t), \quad i = 1, 2
\]

These ODEs are realized into a parallel simulation diagram (Figure 8.3).

Figure 8.3: Simulation diagrams to realize a second-order transfer function: serial realization (top); parallel realization (bottom).

Having drawn a simulation diagram, we designate the outputs of the integrators as state variables and express integrator inputs as first-order differential equations, referred as the state equations.

A serial realization acquires a controller form structure; a parallel realization acquires a modal structure. These are discussed below.

### 8.3.2 Controller Form

A controller form state variable structure results from a serial realization of the transfer function model. This is illustrated using a third-order transfer function model:

\[
G(s) = \frac{b_{1} s^{2} + b_{2} s + b_{3}}{s^{3} + a_{1} s^{2} + a_{2} s + a_{3}}
\]

The above transfer function corresponds to the following ODE:

\[
\dddot{y}(t) + a_{1} \ddot{y}(t) + a_{2} \dot{y}(t) + a_{3} y(t) = b_{1} \ddot{u}(t) + b_{2} \dot{u}(t) + b_{3} u(t)
\]

To remove the derivatives of the input variable from the simulation diagram, an auxiliary variable, \(v(t)\), is used to split the ODE as:

\[
\dddot{v}(t) = -a_{1} \ddot{v}(t) - a_{2} \dot{v}(t) - a_{3} v(t) + u(t)
\]

The highest derivative term in the first ODE is realized using a summing node:

\[
\dddot{v}(t) = -a_{1} \ddot{v}(t) - a_{2} \dot{v}(t) - a_{3} v(t) + u(t)
\]
The lower order derivative terms, $\ddot{v}(t)$ and $\dot{v}(t)$, and the output, $v(t)$, are realized using integrators.

The second ODE can be realized by summing the outputs of the integrators using coefficients as weights (Figure 8.4).

Next, the state variables are designated as: $x_{1}(t) = v(t), x_{2}(t) = \dot{v}(t), x_{3}(t) = \ddot{v}(t)$.

The resulting state equations are: 
\[
\begin{align*}
\dot{x}_{1} &= x_{2}, \\
\dot{x}_{2} &= x_{3}, \\
\dot{x}_{3} &= -a_{3}x_{1} - a_{2}x_{2} - a_{1}x_{3} + u
\end{align*}
\]

The output equation is given as: 
\[
y = b_{1}x_{1} + b_{2}x_{2} + b_{3}x_{3}
\]

In vector–matrix form, the state variable model is given as:
\[
\frac{d}{dt} \begin{bmatrix} x_{1} \\ x_{2} \\ x_{3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_{3} & -a_{2} & -a_{1} \end{bmatrix} \begin{bmatrix} x_{1} \\ x_{2} \\ x_{3} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} b_{1} \\ b_{2} \\ b_{3} \end{bmatrix} \begin{bmatrix} x_{1} \\ x_{2} \\ x_{3} \end{bmatrix}
\]

Figure 8.4: Simulation diagram for controller form realization of the transfer function model.

**Generalization to \(n\) Variables**

The controller form realization is generalized to \(n\) variables as follows: let \(G(s) = \frac{n(s)}{d(s)}\), where \(d(s) = s^{n} + a_{1}s^{n-1} + \ldots + a_{n-1}s + a_{n}\) represents the denominator polynomial.

By selecting the output variable and its derivatives: \(y(t), \dot{y}(t), \ldots, y^{(n-1)}(t)\), as state variables, we obtain the following controller form structure:

\[
A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ -a_{n} & -a_{n-1} & \ddots & \cdots & 1 \\ \vdots & \vdots & \ddots & \cdots & \ddots \\ b & b_{1} & \cdots & b_{n-1} & b_{n} \end{bmatrix}, \quad b = \begin{bmatrix} b_{0} \\ b_{1} \\ \vdots \\ b_{n-1} \\ b_{n} \end{bmatrix}
\]

Further, let \(n(s) = b_{0}s^{m} + b_{1}s^{m-1} + \ldots + b_{m}\) represent the numerator polynomial; then, the output matrix is formed as: 
\[
\begin{bmatrix} c_{1} & c_{2} & \cdots & c_{m} \end{bmatrix}
\]

In the above controller form structure, the coefficients of the characteristic polynomial appear in reverse order in the last row of the system matrix, whereas the output matrix contains the coefficients of the numerator polynomial. Hence, given a proper transfer function, \(G(s)\), the controller form can be written by inspection.

**Example**

A mass–spring–damper system is described by the following transfer function: 
\(G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^{2} + 2s + 2}\). The controller form state variable model is given as:

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x_{1} \\ x_{2} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} x_{1} \\ x_{2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\
y &= \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \begin{bmatrix} x_{1} \\ x_{2} \end{bmatrix}
\end{align*}
\]
Alternate Controller Form Structure

The integrator outputs in the simulation diagram can be alternatively numbered left to right; this reorders the state variables whereby the coefficients of the characteristic polynomial appear in the first row of the matrix. For a third-order model, the resulting equations are:

\[
\begin{align*}
\dot{x}_3 &= -a_1 x_1 - a_2 x_2 - a_3 x_3 + u, \\
\dot{x}_2 &= x_1 \\
\dot{x}_3 &= x_2
\end{align*}
\]

The alternate controller form is obtained by using 'ss' command with a given transfer function in the MATLAB Control Systems Toolbox.

Example \[\PageIndex{2}\]

A mass–spring–damper system is described by the following transfer function: \(G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2 + 2s + 2}\). The controller form state variable model is given as:

\[
\frac{\text{d}}{\text{d}t} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \\
y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

The state variables are numbered in reverse order. Hence, \(x_1\) represents the velocity and \(x_2\) the position of the inertial mass.

8.3.3 Dual (Observer Form) Realization

Assume that a state variable realization of the transfer function model is given as:

\[
\frac{\text{d}}{\text{d}t} x(t) = Ax(t) + bu(t), \quad y(t) = c^T x(t)
\]

Its dual realization is defined by the following state variable model:

\[
\frac{\text{d}}{\text{d}t} z(t) = A^T z(t) + cu(t), \quad y(t) = b^T z(t)
\]

Both the original and the dual models share the same scalar transfer function, i.e.,

\[
G(s) = c^T (sI - A)^{-1} b = b^T (sI - A^T)^{-1} c = G^T(s)
\]

When the original state variable model is in controller form, its dual appears in the ‘observer form,’ named so because of its use in the design of state observers.

In the MATLAB Control Systems Toolbox, the 'canon' command with 'companion' option is used to obtain the observer form realization.
G=tf(num, den); % define plant transfer function

canon(G,'companion') % transform into companion form

Example \((\text{PageIndex}(3))\)

A mass–spring–damper system is described by the following transfer function: \(G(s)=\frac{y(s)}{u(s)}=\frac{1}{s^2+2s+2}\). The observer form state variable model is given as:

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 & -2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \\
y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\end{align*}
\]

From the output equation, the state variable \(x_2\) represent the position of the inertial mass. From the first state equation, \(\dot{x}_1=-2x_2+u\), the variable \(x_1\) includes an integral of position.

### 8.3.4 Modal Realization

A modal realization has a block diagonal structure consisting of \(1\times 1\) and \(2\times 2\) blocks that contain real and complex eigenvalues. A PFE is used to isolate first and second-order factors in the transfer function model.

In the PFE, the denominator of a second-order factor, expressed as \(\left(s+\sigma\right)^2 + \omega^2\), can be realized as a \(2\times 2\) block containing the real and imaginary parts of the eigenvalue as: \(A_i = \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix}\). Alternately, a second-order factor can be realized in the serial form, as illustrated in the following example.

In the MATLAB Control Systems Toolbox, a modal realization is obtained by using the ‘canon’ command with ‘modal’ option.

G=tf(num, den); % define plant transfer function

canon(G,‘modal’) % obtain modal form realization

Example \((\text{PageIndex}(4))\)

Consider the following third-order model with one real and two complex poles. Use the PFE to obtain:

\[
G(s)=\frac{2(s^2+s+1)}{(s+2)(s^2+2s+2)} = \frac{3}{s+2} - \frac{s+2}{(s^2+2s+2)}
\]

A serial realization of the second-order term results in the following modal realization:

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} &= \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u, \\
y &= \begin{bmatrix} 3 & -2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
\end{align*}
\]
Alternately, the modal realization includes the real and imaginary parts of the complex eigenvalue. The resulting state variable model is given as:

\[
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 3 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
\]

We can verify that the two modal realizations share the same transfer function.

**Diagonalization and Decoupling**

When the denominator polynomial in the transfer function model has real and distinct roots, its modal matrix is a diagonal matrix with eigenvalues on the main diagonal. The resulting state equations are described by a set of decoupled first-order ODEs that can be easily integrated.

Example \(\PageIndex{5}\)

The transfer function model of a small DC motor, given as: \(G(s) = \frac{500}{s^2+10s+1025}\).

A PFE of the transfer function gives: \(G(s) = 5.59 \left( \frac{1}{s+10.28} - \frac{1}{s+99.72} \right)\).

A modal realization is obtained as:

\[
\begin{bmatrix} \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -99.72 & 0 \\ 0 & -10.28 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} V_a, \quad \omega = \begin{bmatrix} -5.59 & 5.59 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

The decoupled state and output equations representing the DC motor model are given as:

\[\dot{x}_1(t) = -99.72 x_1(t) + V_a\]
\[\dot{x}_2(t) = -10.28 x_2(t) + V_a\]
\[\omega(t) = 5.59 (x_2(t) - x_1(t))\]

The state variables in the above diagonal realization represent linear combinations of the original variables.