with *ss* results in the controller companion form, which we know to be ill-conditioned. Hence, we should avoid converting a transfer matrix to state-space representation using the command *ss*, unless we are dealing with a low order system.

3.5 Block Building in Linear, Time-Invariant State-Space

Control systems are generally interconnections of various sub-systems. If we have a state-space representation for each sub-system, we should know how to obtain the state-space representation of the entire system. Figure 2.55 shows three of the most common types of interconnections, namely the *series*, *parallel*, and *feedback* arrangement. Rather than using the transfer matrix description of Figure 2.55, we would like to depict the three common arrangements in state-space, as shown in Figure 3.7.

The series arrangement in Figure 3.7(a) is described by the following matrix equations:

$$\mathbf{x}_1^{(1)}(t) = \mathbf{A}_1\mathbf{x}_1(t) + \mathbf{B}_1\mathbf{u}(t) \tag{3.131}$$

$$\mathbf{y}_1(t) = \mathbf{C}_1\mathbf{x}_1(t) + \mathbf{D}_1\mathbf{u}(t) \tag{3.132}$$

$$\mathbf{x}_2^{(1)}(t) = \mathbf{A}_2\mathbf{x}_2(t) + \mathbf{B}_2\mathbf{y}_1(t) \tag{3.133}$$

$$\mathbf{y}(t) = \mathbf{C}_2\mathbf{x}_2(t) + \mathbf{D}_2\mathbf{y}_1(t) \tag{3.134}$$

where the state-space representation of the first sub-system is $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$, while that of the second subsystem is $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$. The input to the system, $\mathbf{u}(t)$, is also

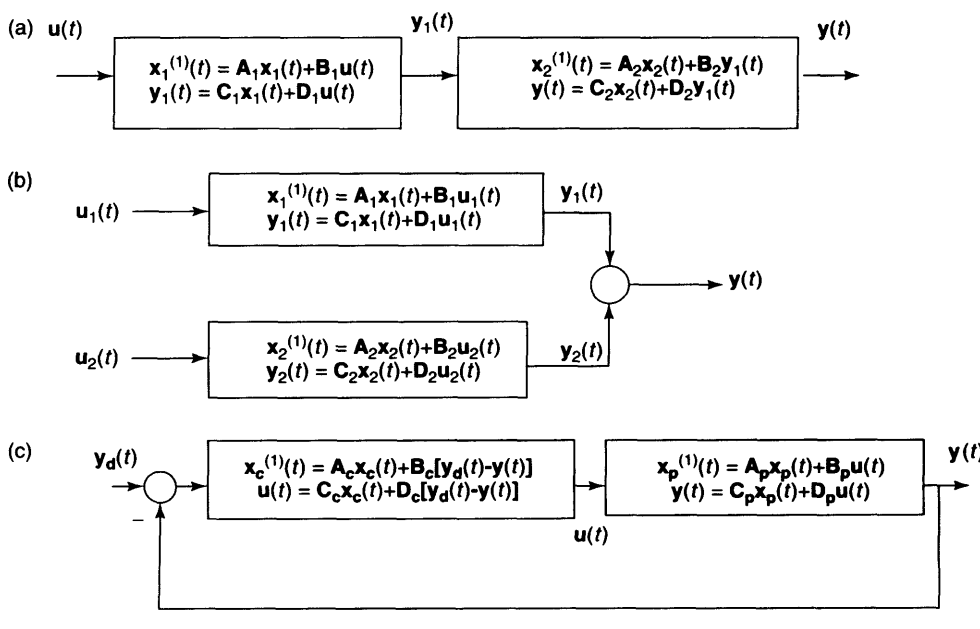

Figure 3.7 Three common arrangements of sub-systems models in state-space

the input to the first sub-system, while the system's output, $\mathbf{y}(t)$, is the output of the second sub-system. The output of the first sub-system, $\mathbf{y}_1(t)$, is the input to the second sub-system. Substitution of Eq. (3.132) int Eq. (3.133) yields:

$$\mathbf{x}_2^{(1)}(t) = \mathbf{A}_2\mathbf{x}_2(t) + \mathbf{B}_2\mathbf{C}_1\mathbf{x}_1(t) + \mathbf{B}_2\mathbf{D}_1\mathbf{u}(t) \tag{3.135}$$

and substituting Eq. (3.132) into Eq. (3.134), we get

$$\mathbf{y}(t) = \mathbf{C}_2\mathbf{x}_2(t) + \mathbf{D}_2\mathbf{C}_1\mathbf{x}_1(t) + \mathbf{D}_2\mathbf{D}_1\mathbf{u}(t) \tag{3.136}$$

If we define the state-vector of the system as $\mathbf{x}(t) = [\mathbf{x}_1^T(t); \mathbf{x}_2^T(t)]^T$, Eqs. (3.131) and (3.135) can be expressed as the following state-equation of the system:

$$\mathbf{x}^{(1)}(t) = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2\mathbf{D}_1 \end{bmatrix} \mathbf{u}(t) \tag{3.137}$$

and the output equation is Eq. (3.136), re-written as follows:

$$\mathbf{y}(t) = [\mathbf{D}_2\mathbf{C}_1 \quad \mathbf{C}_2]\mathbf{x}(t) + \mathbf{D}_2\mathbf{D}_1\mathbf{u}(t) \tag{3.138}$$

The MATLAB (CST) command *series* allows you to connect two sub-systems in series using Eqs. (3.137) and (3.138) as follows:

```
>>sys = series(sys1,sys2) <enter>
```

The command *series* allows connecting the sub-systems when only some of the outputs of the first sub-system are going as inputs into the second sub-system (type *help series* ⟨*enter*⟩ for details; also see Example 2.28). Note that the sequence of the sub-systems is crucial. We will get an *entirely different* system by switching the sequence of the sub-systems in Figure 3.7(a), unless the two sub-systems are identical.

Deriving the state and output equations for the parallel connection of sub-systems in Figure 3.7(b) is left to you as an exercise. For connecting two parallel sub-systems, MATLAB (CST) has the command *parallel*, which is used in a manner similar to the command *series*.

The feedback control system arrangement of Figure 3.7(c) is more complicated than the series or parallel arrangements. Here, a controller with state-space representation $(\mathbf{A_c}, \mathbf{B_c}, \mathbf{C_c}, \mathbf{D_c})$ is connected in series with the plant $(\mathbf{A_p}, \mathbf{B_p}, \mathbf{C_p}, \mathbf{D_p})$ and the feedback loop from the plant output, $\mathbf{y}(t)$, to the summing junction is closed. The input to the closed-loop system is the desired output, $\mathbf{y_d}(t)$. The input to the controller is the error $[\mathbf{y_d}(t) - \mathbf{y}(t)]$, while its output is the input to the plant, $\mathbf{u}(t)$. The state and output equations of the plant and the controller are, thus, given by

$$\mathbf{x_p}^{(1)}(t) = \mathbf{A_p}\mathbf{x_p}(t) + \mathbf{B_p}\mathbf{u}(t) \tag{3.139}$$

$$\mathbf{y}(t) = \mathbf{C_p}\mathbf{x_p}(t) + \mathbf{D_p}\mathbf{u}(t) \tag{3.140}$$

$$\mathbf{x_c}^{(1)}(t) = \mathbf{A_c}\mathbf{x_c}(t) + \mathbf{B_c}[\mathbf{y_d}(t) - \mathbf{y}(t)] \tag{3.141}$$

$$\mathbf{u}(t) = \mathbf{C_c}\mathbf{x_c}(t) + \mathbf{D_c}[\mathbf{y_d}(t) - \mathbf{y}(t)] \tag{3.142}$$

Substituting Eq. (3.142) into Eqs. (3.139) and (3.140) yields the following:

$$\mathbf{x_p}^{(1)}(t) = \mathbf{A_p x_p}(t) + \mathbf{B_p C_c x_c}(t) + \mathbf{B_p D_c}[\mathbf{y_d}(t) - \mathbf{y}(t)] \tag{3.143}$$

$$\mathbf{y}(t) = \mathbf{C_p x_p}(t) + \mathbf{D_p C_c x_c}(t) + \mathbf{D_p D_c}[\mathbf{y_d}(t) - \mathbf{y}(t)] \tag{3.144}$$

Equation (3.144) can be expressed as

$$\mathbf{y}(t) = (\mathbf{I} + \mathbf{D_p D_c})^{-1}[\mathbf{C_p x_p}(t) + \mathbf{D_p C_c x_c}(t)] + (\mathbf{I} + \mathbf{D_p D_c})^{-1}\mathbf{D_p D_c y_d}(t) \tag{3.145}$$

provided the square matrix $(\mathbf{I} + \mathbf{D_p D_c})$ is non-singular. Substituting Eq. (3.145) into Eq. (3.143) yields the following state-equation of the closed-loop system:

$$\mathbf{x}^{(1)}(t) = \mathbf{Ax}(t) + \mathbf{By_d}(t) \tag{3.146}$$

and the output equation of the closed-loop system is Eq. (3.145) re-written as:

$$\mathbf{y}(t) = \mathbf{Cx}(t) + \mathbf{Dy_d}(t) \tag{3.147}$$

where

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x_p}(t) \\ \mathbf{x_c}(t) \end{bmatrix};$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_p} - \mathbf{B_p D_c}(\mathbf{I} + \mathbf{D_p D_c})^{-1}\mathbf{C_p} & \mathbf{B_p C_c} - \mathbf{B_p D_c}(\mathbf{I} + \mathbf{D_p D_c})^{-1}\mathbf{D_p C_c} \\ -\mathbf{B_c}(\mathbf{I} + \mathbf{D_p D_c})^{-1}\mathbf{C_p} & \mathbf{A_c} - \mathbf{B_c}(\mathbf{I} + \mathbf{D_p D_c})^{-1}\mathbf{D_p C_c} \end{bmatrix}$$

$$\mathbf{C} = (\mathbf{I} + \mathbf{D_p D_c})^{-1}\begin{bmatrix} \mathbf{C_p} & \mathbf{D_p C_c} \end{bmatrix};$$

$$\mathbf{D} = (\mathbf{I} + \mathbf{D_p D_c})^{-1}\mathbf{D_p D_c}; \quad \mathbf{B} = \begin{bmatrix} \mathbf{B_p D_c}(\mathbf{I} - \mathbf{D}) \\ \mathbf{B_c}(\mathbf{I} - \mathbf{D}) \end{bmatrix} \tag{3.148}$$

Using MATLAB (CST), the closed-loop system given by Eqs. (3.146)–(3.148) can be derived as follows:

```
>>sys0 = series(sysc,sysp) % series connection of LTI blocks sysc
  and sysp <enter>

>>sys1=ss(eye(size(sys0))) % state-space model (A=B=C=0, D=I) of
  the feedback block, sys1 <enter>

>>sysCL= feedback(sys0, sys1) % negative feedback from output to
  input of sys0 <enter>
```

where *sys0* is the state-space representation of the controller, *sysc*, in series with the plant, *sysp*, *sys1* is the state-space representation ($\mathbf{A} = \mathbf{B} = \mathbf{C} = \mathbf{0}$, $\mathbf{D} = \mathbf{I}$) of the feedback block in Figure 3.7(c), and *sysCL* is the state-space representation of the closed-loop system. Note that *sys0* is the *open-loop system* of Figure 3.7(c), i.e. the system when the feedback loop is absent.

Example 3.18

Let us derive the state-space representation of an interesting system, whose block-diagram is shown in Figure 3.8. The system represents a missile tracking

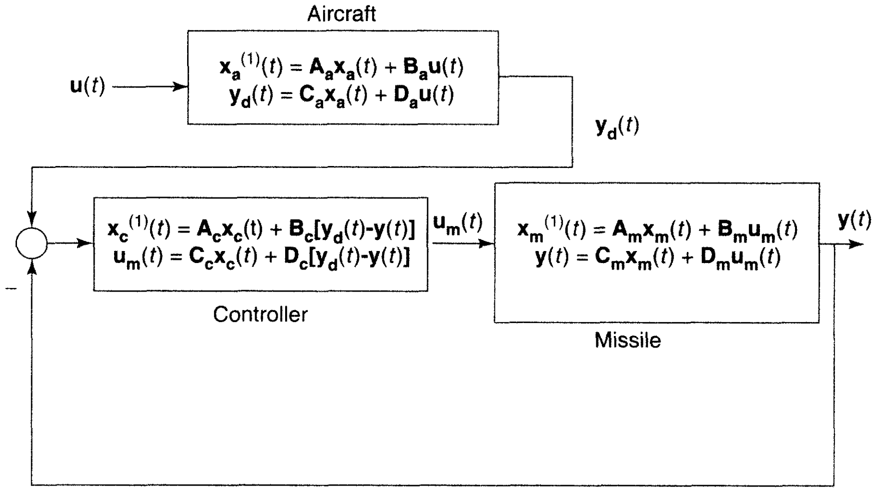

Figure 3.8 Block diagram for the aircraft-missile control system of Example 3.18

a maneuvering aircraft. The pilot of the aircraft provides an input vector, $\mathbf{u}(t)$, to the aircraft represented as $(\mathbf{A_a}, \mathbf{B_a}, \mathbf{C_a}, \mathbf{D_a})$. The input vector, $\mathbf{u}(t)$, consists of the aircraft pilot's deflection of the *rudder* and the *aileron*. The motion of the aircraft is described by the vector, $\mathbf{y_d}(t)$, which is the desired output of the missile, i.e. the missile's motion – described by the output vector, $\mathbf{y}(t)$ – should closely follow that of the aircraft. The output vector, $\mathbf{y}(t)$, consists of the missile's *linear* and *angular velocities* with respect to *three mutually perpendicular axes* attached to the missile's *center of gravity* – a total of six output variables. The state-space representation for the missile is $(\mathbf{A_m}, \mathbf{B_m}, \mathbf{C_m}, \mathbf{D_m})$. The missile is controlled by a feedback controller with the state-space representation $(\mathbf{A_c}, \mathbf{B_c}, \mathbf{C_c}, \mathbf{D_c})$ whose task is to ultimately make $\mathbf{y}(t) = \mathbf{y_d}(t)$, i.e. cause the missile to hit the maneuvering aircraft.

The matrices representing the aircraft, missile, and controller are as follows:

$$\mathbf{A_a} = \begin{bmatrix} -0.0100 & -0.1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.4158 & 1.0250 & 0 & 0 & 0 & 0 \\ 0 & 0.0500 & -0.8302 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5600 & -1.0000 & 0.0800 & 0.0400 \\ 0 & 0 & 0 & 0.6000 & -0.1200 & -0.3000 & 0 \\ 0 & 0 & 0 & -3.0000 & 0.4000 & -0.4700 & 0 \\ 0 & 0 & 0 & 0 & 0.0800 & 1.0000 & 0 \end{bmatrix}$$

$$\mathbf{B_a} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.0730 & 0.0001 \\ -4.8000 & 1.2000 \\ 1.5000 & 10.0000 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C_a} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 250 & 0 & 0 & 0 \\ 0 & 250 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}; \quad \mathbf{D_a} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{A_m} = \begin{bmatrix} 0.4743 & 0 & 0.0073 & 0 & 0 & 0 \\ 0 & -0.4960 & 0 & 0 & 0 & 0 \\ -0.0368 & 0 & -0.4960 & 0 & 0 & 0 \\ 0 & -0.0015 & 0 & -0.0008 & 0 & 0.0002 \\ 0 & 0 & -0.2094 & 0 & -0.0005 & 0 \\ 0 & -0.2094 & 0 & 0 & 0 & -0.0005 \end{bmatrix}$$

$$\mathbf{B_m} = \begin{bmatrix} 0 & 0 & 0 \\ 191.1918 & 0 & 0 \\ 0 & 191.1918 & 0 \\ 0 & 0 & 1.0000 \\ 0 & 232.5772 & 0 \\ 232.5772 & 0 & 0 \end{bmatrix}; \quad \mathbf{C_m} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D_m} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A_c} = \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ -1.0 & 0 & 0 & -0.3 & 0 & 0 \\ 0 & -1.0 & 0 & 0 & -0.3 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & -0.3 \end{bmatrix}$$

$$\mathbf{B_c} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0001 \\ 0.6774 & 0.0000 & 0.0052 & 0.0000 & -0.0001 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

$$\mathbf{C_c} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{D_c} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that the aircraft is a *seventh* order sub-system, while the missile and the controller are *sixth* order sub-systems. The state-space representation of the entire system is obtained as follows:

```
>>sysc=ss(Ac,Bc,Cc,Dc); sysm=ss(Am,Bm,Cm,Dm); sys0 = series(sysc,sysm);
  <enter>

>>sys1=ss(eye(size(sys0))); <enter>

>>sysCL=feedback(sys0,sys1); <enter>

>>sysa=ss(Aa,Ba,Ca,Da); syst=series(sysa, sysCL) <enter>

a =
        x1          x2           x3          x4             x5
   x1   0.4743      0            0.0073258   0      0
   x2   0           -0.49601     0       0   0
   x3   -0.036786   0            -0.49601    0      0
   x4   0           -0.0015497   0           -0.00082279    0
   x5   0       0          -0.20939      0         -0.00048754
   x6   0           -0.20939     0           -8.2279e-006   0
   x7   0       0   0        0   0
   x8   0       0   0        0   0
   x9   0       0   0        0   0
   x10  0       0   0        0   0
   x11  -0.6774     0            -0.0052     0              0.0001
   x12  0       0   0        0   0
   x13  0       0   0        0   0
   x14  0       0   0        0   0
   x15  0       0   0        0   0
   x16  0       0   0        0   0
   x17  0       0   0        0   0
   x18  0       0   0        0   0
   x19  0       0   0        0   0

        x6            x7           x8           x9     x10
   x1   0             0        0   0        0
   x2   0             191.19       0        0   0
   x3   0             0            191.19       0      0
   x4   0.00017749    0        0   1        0
   x5   0             0            232.58       0      0
   x6   -0.00048754   232.58       0        0   0
   x7   0             0        0   0        1
   x8   0             0        0   0        0
   x9   0             0        0   0        0
   x10  -0.0001       -1           0        0   -0.3
   x11  0             0            -1           0      0
   x12  0             0        0            -1  0
   x13  0             0        0   0        0
   x14  0             0        0   0        0
   x15  0             0        0   0        0
   x16  0             0        0   0        0
   x17  0             0        0   0        0
   x18  0             0        0   0        0
   x19  0             0        0   0        0
```

	x11	x12	x13	x14	x15
x1	0	0	0	0	0
x2	0	0	0	0	0
x3	0	0	0	0	0
x4	0	0	0	0	0
x5	0	0	0	0	0
x6	0	0	0	0	0
x7	0	0	0	0	0
x8	1	0	0	0	0
x9	0	1	0	0	0
x10	0	0	0	0	0
x11	-0.3	0	0.6774	1.3	-0.0001
x12	0	-0.3	0	0	0
x13	0	0	-0.01	-0.1	0
x14	0	0	0	-0.4158	1.025
x15	0	0	0	0.05	-0.8302
x16	0	0	0	0	0
x17	0	0	0	0	0
x18	0	0	0	0	0
x19	0	0	0	0	0

	x16	x17	x18	x19
x1	0	0	0	0
x2	0	0	0	0
x3	0	0	0	0
x4	0	0	0	0
x5	0	0	0	0
x6	0	0	0	0
x7	0	0	0	0
x8	0	0	0	0
x9	0	0	0	0
x10	0	0.0001	0	0
x11	0	0	0	0
x12	0	0	0	0
x13	0	0	0	0
x14	0	0	0	0
x15	0	0	0	0
x16	-0.56	-1	0.08	0.04
x17	0.6	-0.12	-0.3	0
x18	-3	0.4	-0.47	0
x19	0	0.08	1	0

b =

	u1	u2
x1	0	0
x2	0	0
x3	0	0
x4	0	0
x5	0	0
x6	0	0
x7	0	0
x8	0	0
x9	0	0
x10	0	0
x11	0	0

```
  x12    0       0
  x13    0       0
  x14    0       0
  x15    0       0
  x16  0.073   0.0001
  x17   -4.8     1.2
  x18   1.5      10
  x19    0       0

c =
          x1      x2      x3     x4     x5
   y1     1       0       0      0      0
   y2     0       1       0      0      0
   y3     0       0       1      0      0
   y4     0       0       0      1      0
   y5     0       0       0      0      1
   y6     0       0       0      0      0

          x6      x7      x8     x9    x10
   y1     0       0       0      0      0
   y2     0       0       0      0      0
   y3     0       0       0      0      0
   y4     0       0       0      0      0
   y5     0       0       0      0      0
   y6     1       0       0      0      0

         x11     x12     x13    x14    x15
   y1     0       0       0      0      0
   y2     0       0       0      0      0
   y3     0       0       0      0      0
   y4     0       0       0      0      0
   y5     0       0       0      0      0
   y6     0       0       0      0      0

         x16     x17     x18    x19
   y1     0       0       0      0
   y2     0       0       0      0
   y3     0       0       0      0
   y4     0       0       0      0
   y5     0       0       0      0
   y6     0       0       0      0

d =
          u1      u2
   y1     0       0
   y2     0       0
   y3     0       0
   y4     0       0
   y5     0       0
   y6     0       0

Continuous-time model.
```

The total system, *syst*, is of order 19, which is the sum of the individual orders of the sub-systems. If the entire system, *syst*, is asymptotically stable, the missile