

```
#1:-----
      s^6+28.24s^5+258.3s^4+1468s^3+8214s^2+8801s+1631
      -0.6716s^5+804.5s^4+2.781e004s^3+8.085e004s^2+1.214e006s
      +1.082e006
#2:-----
      s^6+28.24s^5+258.3s^4+1468s^3+8214s^2+8801s+1631
```

Therefore,

$$\mathbf{G}_1(s) = \begin{bmatrix} (0.04829s^5 + 1.876s^4 + 1.949s^3 - 1228s^2 - 5762s - 2385)/d(s) \\ (-0.4919s^5 + 9.483s^4 - 49.05s^3 + 551.6s^2 - 939.6s + 907.6)/d(s) \end{bmatrix}$$

$$\mathbf{G}_2(s) = \begin{bmatrix} (0.02915s^5 + 1.289s^4 - 0.3041s^3 - 2.388 \times 10^4 s^2 - 8.29 \times 10^5 s - 9.544 \times 10^5)/d(s) \\ (-0.6716s^5 + 804.5s^4 + 27810s^3 + 8085s^2 + 1.214 \times 10^6 s + 1.082 \times 10^6)/d(s) \end{bmatrix} \quad (3.119)$$

where

$$d(s) = (s^6 + 28.24s^5 + 258.3s^4 + 1468s^3 + 8214s^2 + 8801s + 1631) \quad (3.120)$$

For brevity, the coefficients have been rounded off to four significant digits. If the MATLAB's *long* format is used to report the results, a greater accuracy is possible. Note that all the elements of the transfer matrix $\mathbf{G}(s)$ have a common denominator polynomial, whose roots are the poles (or eigenvalues) of the system. This is confirmed by using the coefficients of $d(s)$ rounded to five significant digits with the intrinsic MATLAB function *roots* as follows:

```
>>roots([1 28.238 258.31 1467.9 8214.5 8801.2 1630.6]) <enter>

ans =
-15.8746
-10.3862
-0.3491+6.3444i
-0.3491-6.3444i
-1.0444
-0.2346
```

3.4 Special State-Space Representations: The Canonical Forms

In Section 3.1 we saw how some special state-space representations can be obtained, such as the controller and observer companion forms. The companion forms are members of a special set of state-space representations, called *canonical forms*. In addition to the companion forms, another canonical form is the *Jordan canonical form*, which is derived from the partial fraction expansion of the system's transfer matrix as described below. In Section 3.3, we saw how the transfer matrix can be obtained from a state-space

representation. Now we will address the inverse problem, namely deriving special state-space representations from the transfer matrix. For simplicity, consider a single-input, single-output system with the transfer function given by the following partial fraction expansion:

$$Y(s)/U(s) = k_0 + k_1/(s - p_1) + k_2/(s - p_2) + \cdots + k_i/(s - p_i) + k_{i+1}/(s - p_i)^2 + \cdots + k_{i+m}/(s - p_i)^{m+1} + \cdots + k_n/(s - p_n) \quad (3.121)$$

where n is the order of the system, and all poles, except $s = p_i$, are simple poles. The pole, $s = p_i$, is of multiplicity $(m + 1)$, i.e. $(s - p_i)$ occurs as a power $(m + 1)$ in the transfer function. Let us select the state-variables of the system as follows:

$$\begin{aligned} X_1(s)/U(s) &= 1/(s - p_1); X_2(s)/U(s) = 1/(s - p_2); \dots; \\ X_i(s)/U(s) &= 1/(s - p_i); X_{i+1}(s)/U(s) = 1/(s - p_i)^2; \dots; \\ X_{i+m}(s)/U(s) &= 1/(s - p_i)^{m+1}; \dots; X_n(s)/U(s) = 1/(s - p_n) \end{aligned} \quad (3.122)$$

Taking the inverse Laplace transform of Eq. (3.122), we get the following state-equations:

$$\begin{aligned} x_1^{(1)}(t) &= p_1 x_1(t) + u(t) \\ x_2^{(1)}(t) &= p_2 x_2(t) + u(t) \\ &\vdots \\ x_i^{(1)}(t) &= p_i x_i(t) + u(t) \\ x_{i+1}^{(1)}(t) &= x_i(t) + p_i x_{i+1}(t) \\ &\vdots \\ x_{i+m}^{(1)}(t) &= x_{i+m-1}(t) + p_i x_{i+m}(t) \\ &\vdots \\ x_n^{(1)}(t) &= p_n x_n(t) + u(t) \end{aligned} \quad (3.123)$$

and the output equation is given by

$$\begin{aligned} y(t) &= k_1 x_1(t) + k_2 x_2(t) + \cdots + k_i x_i(t) + k_{i+1} x_{i+1}(t) \\ &+ \cdots + k_{i+m} x_{i+m}(t) + \cdots + k_n x_n(t) + k_0 u(t) \end{aligned} \quad (3.124)$$

(Note that in deriving the state-equations corresponding to the repeated pole, $s = p_i$, we have used the relationship $X_{i+1}(s) = X_i/(s - p_i)$, $X_{i+2}(s) = X_{i+1}/(s - p_i)$, and so on.) The state-space representation given by Eqs. (3.123) and (3.124) is called the *Jordan canonical form*. The state coefficient matrices of the Jordan canonical form are, thus, the

following:

$$\mathbf{A} = \begin{bmatrix} p_1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & p_2 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & p_i & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 & p_i & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & p_i & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & p_i & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & p_n \end{bmatrix}$$

$$\mathbf{B} = [1 \quad 1 \quad \dots \quad 1 \quad 0 \quad \dots \quad 0 \quad 1 \quad \dots \quad 1]^T$$

$$\mathbf{C} = [k_1 \quad k_2 \quad \dots \quad k_i \quad k_{i+1} \quad \dots \quad k_{i+m} \quad \dots \quad k_n]$$

$$\mathbf{D} = k_0 \quad (3.125)$$

Note the particular structure of the $\mathbf{A}$ matrix in Eq. (3.125). The system's poles (i.e. the eigenvalues of $\mathbf{A}$) occur on the main diagonal of $\mathbf{A}$, with the repeated pole, p_i , occurring as many times as the multiplicity of the pole. A square block associated with the repeated pole, p_i , is marked by a dashed border, and is known as the *Jordan block* of pole, p_i . The diagonal *below* the main diagonal – called the *subdiagonal* – of this block has all elements equal to 1. All other elements of $\mathbf{A}$ are zeros. The matrix $\mathbf{B}$ also has a particular structure: all elements associated with the simple poles are *ones*, the first element of the Jordan block of repeated pole, p_i – shown in dashed border – is one, while the remaining elements of the Jordan block of p_i are zeros. The elements of matrix $\mathbf{C}$ are simply the residues corresponding to the poles in the partial fraction expansion (Eq. (1.121)), while the matrix $\mathbf{D}$ is equal to the direct term, k_0 , in the partial fraction expansion. If a system has more than one repeated poles, then there is a Jordan block associated with each repeated pole of the same structure as in Eq. (3.125). If none of the poles are repeated, then $\mathbf{A}$ is a diagonal matrix. The Jordan canonical form can be also obtained similarly for multi-input, multi-output systems.

Example 3.15

Let us find the Jordan canonical form of the following system:

$$Y(s)/U(s) = (s + 1)/[(s - 1)^2(s - 3)] \quad (3.126)$$

The partial fraction expansion of Eq. (3.126) is the following:

$$Y(s)/U(s) = 1/(s-3) - 1/(s-1) - 1/(s-1)^2 \quad (3.127)$$

Comparing Eq. (3.127) with Eq. (3.121) and using Eq. (3.125), we get the following Jordan canonical form:

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & \boxed{1} & 0 \\ 0 & \boxed{1} & \boxed{1} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & \boxed{-1} & \boxed{-1} \end{bmatrix}; \quad \mathbf{D} = 0 \quad (3.128)$$

The Jordan block associated with the repeated pole, $s = 1$, is shown in dashed borders.

While Jordan canonical form is an easy way of obtaining the state-space representation of a system, it has a major drawback: for the Jordan canonical form to be a *practical* representation, all the poles of the system (i.e. eigenvalues of $\mathbf{A}$) must be *real*. After all, the purpose of having state-space representations is to practically implement a control system, using electrical circuits or mechanical devices. A state-space representation with complex coefficient matrices cannot be implemented in a hardware (have you ever heard of a *complex electrical resistance*, or a spring with *complex stiffness*?). To make some sense out of Jordan canonical form for a system with complex poles, we can combine the partial fractions corresponding to each pair of complex conjugate poles, $p_{1,2} = \sigma \pm i\omega$, into a second order *real* sub-system as follows:

$$k_1/(s-p_1) + k_2/(s-p_2) = 2[\alpha s - (\alpha\sigma + \beta\omega)]/(s^2 - 2\sigma s + \sigma^2 + \omega^2) \quad (3.129)$$

where $k_{1,2} = \alpha \pm i\beta$ are the residues corresponding to the poles, $p_{1,2}$. Remember that the residues corresponding to complex conjugate poles are also complex conjugates. Since the complex poles *always* occur as complex conjugates, their combination into real second order sub-systems using Eq. (3.129) will lead to a real state-space representation. From Eq. (3.129), it can be shown that the real Jordan block in $\mathbf{A}$ corresponding to a pair of complex conjugate poles, $p_{1,2} = \sigma \pm i\omega$, is a 2×2 block with real parts on the diagonal, and the imaginary parts off the diagonal as follows:

$$\text{Jordan block of } p_{1,2} = \sigma \pm i\omega \text{ in } \mathbf{A} = \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix} \quad (3.130)$$

The MATLAB Control System Toolbox function *canon* provides an easy derivation of the canonical forms, using the methodology presented above. The function is used as follows:

```
>>[csys,T] = canon(sys,'type') <enter>
```

where *sys* is an LTI object of the system (either transfer matrix, or state-space), '*type*' is either '*modal*' for Jordan canonical form, or '*companion*' for the observer companion

form, *csys* is the returned canonical form, and **T** is the returned state-transformation matrix which transforms the state-space representation from *sys* to *csys* (**T** is meaningful only if *sys* is a state-space representation, and not the transfer matrix). The matrix **A** of the Jordan canonical form obtained using *canon* has each pair of complex eigenvalues in a real Jordan block given by Eq. (3.130).

Example 3.16

For the system of Example 3.14, let us obtain the canonical forms. For the Jordan canonical form, the MATLAB (CST) command *canon* is used as follows:

```
>> syst=ss(A, B, C, D); [jsyst,T] = canon(syst, 'modal') <enter>
a =
```

	x1	x2	x3	x4	x5
x1	-15.873	0	0	0	0
x2	0	-10.387	0	0	0
x3	0	0	-0.34925	6.3444	0
x4	0	0	-6.3444	-0.34925	0
x5	0	0	0	0	-1.0444
x6	0	0	0	0	0

	x6
x1	0
x2	0
x3	0
x4	0
x5	0
x6	-0.23455

b =

	u1	u2
x1	0.50702	-20.055
x2	-0.36131	30.035
x3	0.092163	-5.577
x4	0.13959	13.23
x5	-0.17417	8.7113
x6	0.021513	14.876

c =

	x1	x2	x3	x4	x5
y1	0.86988	2.3105	2.7643	6.459	2.8803
y2	-7.9857	-11.128	-0.19075	-0.78991	3.2141

	x6
y1	-9.885
y2	10.406

d =

	u1	u2
y1	0	0
y2	0	0

Continuous-time model.

T =

Columns 1 through 2

-1.67367664062877	0.82071823417532
0.93271689676861	-1.37011949477365
0.00703777816771	0.01235017857500
-0.01678546516033	0.05675504485937
0.01434336645020	0.32743224909474
0.00550594309107	-0.03421867428744

Columns 3 through 4

-0.64877657888967	0.02185375479665
1.30954638127904	0.00863637157274
0.06144994008648	1.42410332092227
-0.12409538851455	0.03844790932243
1.10820137234454	0.07909786976484
0.01382978986462	-0.04137902672467

Columns 5 through 6

0.15933965118530	-0.03863614709032
-0.32626410503493	0.08854860917358
-0.00622411675595	0.07244625298095
-1.41411419523105	0.02977656775124
-0.08538437917204	0.21852854369639
-0.01967206771160	1.01709518519537

Note the 2×2 real Jordan block in **A** corresponding to the complex eigenvalues $p_{3,4} = -0.34925 \pm 6.3444i$. Also, note that the corresponding terms in the matrices **B** and **C** are also real. The transformation matrix, **T**, has been reported in long format for greater accuracy in calculations. Next, we calculate the observer canonical form of the system as follows:

```
>>[csyst, T] = canon(syst, 'companion') <enter>
```

a =

	x1	x2	x3	x4	x5
x1	9.6034e-015	-1.3017e-013	1.6822e-012	-4.8772e-011	1.5862e-009
x2	1	5.6899e-014	1.6964e-012	-3.4476e-011	7.7762e-010
x3	7.7716e-016	1	2.6068e-013	-4.6541e-012	1.0425e-010
x4	4.5103e-017	2.4997e-015	1	-8.3311e-013	1.9014e-011
x5	1.5179e-018	2.7864e-016	5.5303e-015	1	2.1529e-012
x6	5.421e-020	1.0103e-017	1.9602e-016	-3.2543e-015	1

	x6
x1	-1630.6
x2	-8801.2
x3	-8214.5
x4	-1467.9
x5	-258.31
x6	-28.238

b =

	u1	u2
x1	1	945.61
x2	0	1128.8
x3	0	201.9

```

x4  0  36.481
x5  0  4.0669
x6  0  0.1451

```

c =

```

      x1      x2      x3      x4      x5
y1  0.04829  0.51209 -24.985 -725.97 20044
y2 -0.49194  23.374  -581.99 11670  -2.1041e+005

      x6
y1  -3.488e+005
y2  3.5944e+006

```

d =

```

      u1      u2
y1  0      0
y2  0      0

```

Continuous-time model.

T =

Columns 1 through 2

```

0.36705413565042  -1.48647885543276
0.48941400292950  0.46746500786628
0.06992034388180  0.10572273608335
0.01406952155565  0.02165321186650
0.00129963404010  0.00361303652580
0.00003944900770  0.00015766993850

```

Columns 3 through 4

```

3.02790751537344  -2.47039343978461
10.58180159205530 -2.47081762640110
1.83987823178029  -0.61525593744681
0.34386037047772  -0.10310276871196
0.04003837042044  -0.00922166413447
0.00145527317768  -0.00028916486064

```

Columns 5 through 6

```

-1.39030096046515  63.91103372168656
-2.11561239460537  73.88483216381728
-0.39195667361013  13.19012284392485
-0.09863522809937  2.36076797272696
-0.01208541334668  0.26314010184181
-0.00043627032173  0.00940569645931

```

In the **A** matrix of the computed observer companion form, all the elements except those in the last column and the subdiagonal are negligible, and can be assumed to be zeros. We can also derive the controller companion form merely by taking the transposes of **A**, **B**, and **C** of the observer form computed above. The controller companion form is thus denoted by the coefficient set (**A^T**, **C^T**, **B^T**, **D**).

The Jordan canonical form is useful for *decoupling* the state-equations of systems with distinct eigenvalues; such systems have a diagonal **A** matrix of the Jordan canonical form.

The companion forms are useful in designing control systems. However, a great disadvantage of the companion forms (both controller and observer) is that they are *ill-conditioned*, which means that the eigenvalues and eigenvectors of the matrix $\mathbf{A}$ are very sensitive to *perturbations* in the elements of $\mathbf{A}$. This results in large inaccuracies in the computed eigenvalues (and eigenvectors), even if there is a small error in calculating $\mathbf{A}$. Since a system's characteristics are governed by the eigenvalues and eigenvectors of $\mathbf{A}$, an ill-conditioned $\mathbf{A}$ matrix is undesirable. The ill-conditioning of companion forms generally gets worse as the order of the system increases. Hence, we should normally avoid using the companion forms as state-space representations, especially for large order systems. MATLAB assigns a *condition number* to each square matrix. The condition number indicates how close a matrix is to being *singular* (i.e. determinant of the matrix being zero). A larger condition number means that the matrix is closer to being singular. With MATLAB we can assess the condition number of a square matrix, $\mathbf{A}$, using the command `cond(A)`. If `cond(A)` is small, it indicates that $\mathbf{A}$ is *well-conditioned*. If `cond(A)` is very large, it implies an *ill-conditioned* $\mathbf{A}$. Whenever we try to invert an ill-conditioned matrix, MATLAB issues a warning that the matrix is ill-conditioned and the results may be inaccurate.

Example 3.17

Let us compare the condition numbers for the Jordan canonical form and the observer companion form derived in Example 3.16. The condition number for the Jordan canonical form is calculated by first retrieving the state coefficient matrices from the LTI object `jsyst` using the Control System Toolbox (CST) function `ssdata`, and then applying `cond` to matrix $\mathbf{A}$ as follows:

```
>> [Aj,Bj,Cj,Dj]=ssdata(jsyst); cond(Aj) <enter>
ans =
    67.6741
```

while the condition number for the companion form is the following:

```
>> [Ac,Bc,Cc,Dc]=ssdata(csyst); cond(Ac) <enter>
ans =
    9.1881e+004
```

The condition number for the companion form is, thus, very large in comparison to that of the Jordan canonical form, confirming that the former is ill-conditioned. Why is a companion form ill-conditioned while the Jordan canonical form is not? The answer lies in all the diagonal elements, except the last, being zeros in the matrix $\mathbf{A}$ of the companion forms. In contrast, the Jordan form's matrix $\mathbf{A}$ has a *populated* diagonal, i.e. none of the diagonal elements are zeros.

For conversion of a transfer matrix into a state-space representation, you can use the MATLAB (CST) LTI object `ss`. However, the state-space conversion of a transfer matrix