

在讨论了基于 MATLAB 的现代控制系统状态空间模型后,就要根据对象的状态空间模型对系统进行分析,其目的就是要揭示系统的运动规律和基本特性。系统分析一般有定性分析和定量分析两种。定性分析主要分析现代控制系统的能控性、能观性和稳定性,而定量分析则是对现代控制系统的运动规律进行精确的研究,定量地确定系统由初始状态和外部激励所引起的响应,即在知道了系统的初始状态和外部输入信号后,如何根据状态空间模型确定系统未来的状态或输出,以了解系统的运动状态。

给定现代控制系统的连续时间状态空间模型

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (3.0.1a)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (3.0.1b)$$

或离散时间状态空间模型

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (3.0.2a)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (3.0.2b)$$

其中, \mathbf{x} 是系统的 n 维状态向量, \mathbf{u} 是 m 维控制输入, \mathbf{y} 是 r 维测量输出, \mathbf{A} 、 \mathbf{B} 、 \mathbf{C} 和 \mathbf{D} 是适当维数的实常数矩阵。对于连续时间状态空间模型(见式(3.0.1)), 变量 $t \geq 0$ 是时间变量; 对于离散时间状态空间模型(见式(3.0.2)), 变量 $k = 0, 1, 2, \dots$ 是离散采样时刻变量。为了方便起见, 对于时不变系统, 设系统的初始时刻 $t_0 = 0$ ($k_0 = 0$); 若 $t_0 \neq 0$ ($k_0 \neq 0$), 则只需在相应结果中以 $t - t_0$ ($k - k_0$) 代替 t (k), t_0 (k_0) 代替 0, 初始状态 $\mathbf{x}(0) = \mathbf{x}_0$ 。系统的运动分析就是在给定的输入信号 \mathbf{u} , 了解系统状态和输出随时间变化的情况, 即系统状态和输出的时间响应, 从而评判系统的性能。这样一个问题在数学上归结为对给定的初始条件 $\mathbf{x}(0) = \mathbf{x}_0$ 和函数 \mathbf{u} , 求解式(3.0.1)或式(3.0.2)。

本章将介绍 MATLAB 环境下基于状态空间模型的线性时不变系统的定性和定量分析, 特别是借助于 MATLAB 软件, 可以很方便地绘制出系统的状态和输出对初始状态和一些特殊输入信号的时间响应, 从而可

以有效地反映出系统变量的动态和稳态变化情况。进一步,分析系统的状态能控性、输出能控性、状态能观性和稳定性等性质。

3.1 状态空间模型的运动响应分析

考虑连续时间状态空间模型(见式(3.0.1)),其状态方程和输出方程的解分别为

$$\mathbf{x}(t) = e^{At} \mathbf{x}(0) + e^{At} \int_0^t e^{-A\tau} \mathbf{B} \mathbf{u}(\tau) d\tau = e^{At} \mathbf{x}(0) + \int_0^t e^{A(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (3.1.1)$$

$$\mathbf{y}(t) = \mathbf{C} e^{At} \mathbf{x}(0) + \mathbf{C} \int_0^t e^{A(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau + \mathbf{D} \mathbf{u}(t) \quad (3.1.2)$$

相应地,离散时间状态空间模型(见式(3.0.2))的状态方程和输出方程的解分别为

$$\mathbf{x}(k) = \mathbf{G}^k \mathbf{x}(0) + \sum_{i=0}^{k-1} \mathbf{G}^{k-i-1} \mathbf{H} \mathbf{u}(i), \quad k = 1, 2, 3, \dots \quad (3.1.3)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{G}^k \mathbf{x}(0) + \mathbf{C} \sum_{i=0}^{k-1} \mathbf{G}^{k-i-1} \mathbf{H} \mathbf{u}(i) + \mathbf{D} \mathbf{u}(k) \quad (3.1.4)$$

状态方程的解 \mathbf{x} 包括两部分:第一部分是由系统自由运动引起的,是初始状态对系统运动的影响;第二部分是由控制输入引起的,反映了输入对系统状态的影响。两部分的叠加构成了系统的状态响应。输出方程的解 \mathbf{y} 由三部分组成:第一部分是当外部输入等于零时,由初始状态引起的,故为系统的零输入响应;第二部分是当初始状态为零时,由外部输入引起的,故为系统的外部输入响应;第三部分是系统输入的直接传输部分。因此,根据系统的解,只要知道系统的初始状态和初始时刻之后的输入信号,就可以求出系统在初始时刻之后任意时刻处的状态解和输出解,可以定量分析系统输出的性能。由于输入信号是由设计者确定的,因此,可通过适当选取控制输入,使得系统响应满足所期望的要求。

3.1.1 单位阶跃响应分析

在 MATLAB 软件中,函数 `step` 和 `dstep` 分别给出了由式(3.0.1)描述的连续时间系统和由式(3.0.2)描述的离散时间系统的单位阶跃响应曲线,其格式和功能如下。

1) 连续时间系统单位阶跃响应函数 `step`

<code>step(A, B, C, D)</code>	% A、B、C、D 为系统状态空间模型的相应矩阵
<code>[y, x, t] = step(A, B, C, D)</code>	% 返回系统输出 y 、状态 x 以及相应的时间 t
<code>[y, x, t] = step(A, B, C, D, iu)</code>	% iu 表示输入变量的序号
<code>[y, x, t] = step(A, B, C, D, iu, t)</code>	% t 表示自定义时间

2) 离散时间系统单位阶跃响应函数 dstep

```

dstep(A,B,C,D)           % A、B、C、D为系统状态空间模型的相应矩阵
[y,x,t] = dstep(A,B,C,D) % 返回系统输出 y、状态 x 以及相应的时间 t
[y,x,t] = dstep(A,B,C,D,iu) % iu 表示输入变量的序号
[y,x,t] = dstep(A,B,C,D,iu,t) % t 表示自定义时间

```

【例 3.1.1】 考虑以下系统：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 6.5 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

试给出该系统的单位阶跃响应曲线。

解：这是一个具有 2 个输入 2 个输出的系统，系统的传递函数矩阵为

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s+1 & 1 \\ -6.5 & s \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \\ &= \frac{1}{s^2 + s + 6.5} \begin{bmatrix} s & -1 \\ 6.5 & s+1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \\ &= \frac{1}{s^2 + s + 6.5} \begin{bmatrix} s-1 & s \\ s+7.5 & 6.5 \end{bmatrix} \end{aligned}$$

因此，由不同输入对不同输出的 4 个传递函数分别为

$$\frac{Y_1(s)}{U_1(s)} = \frac{s-1}{s^2 + s + 6.5}, \quad \frac{Y_1(s)}{U_2(s)} = \frac{s}{s^2 + s + 6.5}$$

$$\frac{Y_2(s)}{U_1(s)} = \frac{s+7.5}{s^2 + s + 6.5}, \quad \frac{Y_2(s)}{U_2(s)} = \frac{6.5}{s^2 + s + 6.5}$$

在考虑输入信号 u_1 时，假设 u_2 取零；反之，考虑输入信号 u_2 时，假设 u_1 取零。在 MATLAB 软件中，编写以下 M 文件(Example311.m)：

```

A = [-1 -1; 6.5 0];           % 系统矩阵 A
B = [1 1; 1 0];             % 系统矩阵 B
C = [1 0; 0 1];             % 系统矩阵 C
D = [0 0; 0 0];             % 系统矩阵 D
step(A,B,C,D)                % 输出阶跃响应

```

运行 Example311.m 文件，运行结果如图 3.1.1 所示。

也可以将同一个输入的两条响应曲线绘在同一张图上，此时，可以采用以下的函数：

```
[y,x,t] = step(A,B,C,D,iu)
```

或

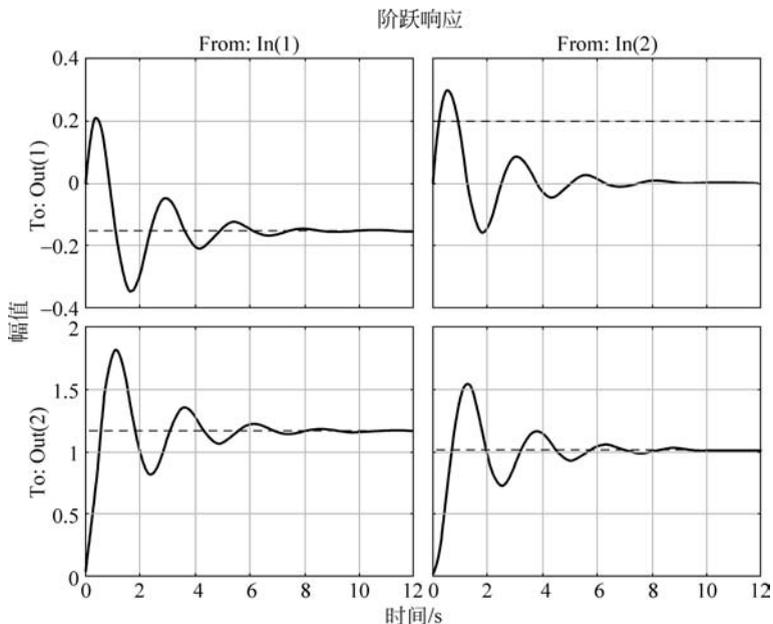


图 3.1.1 单位阶跃响应曲线

$$[y, x, t] = \text{step}(A, B, C, D, iu, t)$$

其中, i_u 表示第 i 个输入, t 是用户确定的时间, 矩阵 y 和 x 分别包含系统在各个时刻 t 处计算出的输出量和状态值(y 和 x 的列数分别与输出变量和状态变量的个数相同, y 和 x 的每一行是对应时刻 t 的计算值), 进而应用绘图命令 `plot` 绘出相应的响应曲线。

【例 3.1.2】 考虑例 3.1.1 所示系统, 试给出控制输入 u_1 对应的单位阶跃响应曲线。

解: 在 MATLAB 软件中, 编写以下的 M 文件(Example312.m):

```

% 输入状态空间表达式中的矩阵参数
A = [-1 -1; 6.5 0];
B = [1 1; 1 0];
C = [1 0; 0 1];
D = [0 0; 0 0];
[y, x, t] = step(A, B, C, D, 1); % 生成第 1 个控制输入对应的输出和状态轨迹
plot(t, y(:, 1), t, y(:, 2)) % 绘图第 1 个控制输入对应的输出轨迹
grid % 绘制网格
title('阶跃响应曲线: 输入 = u1 (u2 = 0)') % 标题
xlabel('时间/s') % x 轴标签
ylabel('输出') % y 轴标签
text(3.4, -0.06, 'y1') % 在指定坐标处加标注
text(3.4, 1.4, 'y2') % 在指定坐标处加标注
    
```

运行 Example312.m 文件,得到对应控制输入 u_1 的两条输出变量阶跃响应曲线如图 3.1.2 所示,虚线为输出变量 y_2 阶跃响应曲线,实线为输出变量 y_1 阶跃响应曲线。两条曲线的确定可以通过求解平衡态确定。

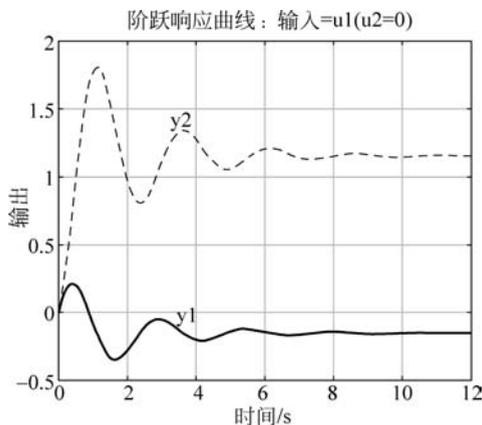


图 3.1.2 单位阶跃响应曲线(u_1 为单位阶跃输入, $u_2=0$)

【例 3.1.3】 考虑例 3.1.1 所示系统,试给出控制输入 u_1 在时间段 $0 \leq t \leq 10$ 上的对应单位阶跃响应曲线。

解: 在 MATLAB 软件中,编写以下 M 文件(Example313.m):

```
% 在 MATLAB 工作空间生成以 0.01 为间隔的时间段  $0 \leq t \leq 10$ 
t = 0:0.01:10;
% 输入状态空间表达式中的矩阵
A = [-1 -1; 6.5 0];
B = [1 1; 1 0];
C = [1 0; 0 1];
D = [0 0; 0 0];
% 生成控制输入  $u_1$  在时间段  $0 \leq t \leq 10$  上的对应单位阶跃响应曲线
[y, x, t] = step(A, B, C, D, 1, t); % 阶跃响应返回 y, x
plot(t, y(:,1), t, y(:,2)) % 同时绘制 y1, y2
grid % 网格
title('阶跃响应曲线: 输入 = u1 (u2 = 0)') % 标题
xlabel('时间/s') % x 轴标签
ylabel('输出') % y 轴标签
text(3.4, -0.06, 'y1') % 在指定坐标位置加标注
text(3.4, 1.4, 'y2') % 在指定坐标位置加标注
```

运行 Example313.m 文件,得到对应于控制输入 u_1 在时间段 $0 \leq t \leq 10$ 上的两条输出变量单位阶跃响应曲线如图 3.1.3 所示,虚线为输出变量 y_2 阶跃响应曲线,实线为输出变量 y_1 阶跃响应曲线。两条曲线的确定可以通过求解平衡态确定。

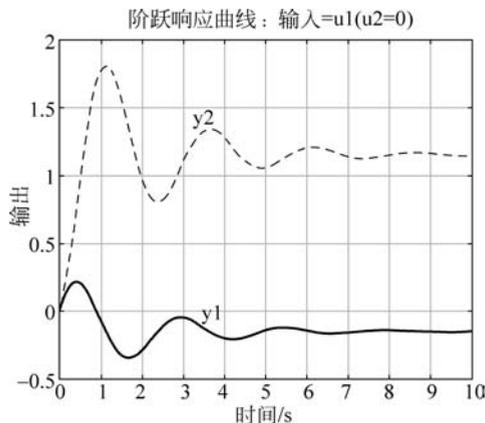


图 3.1.3 具有给定时间段的单位阶跃响应曲线(u_1 为单位阶跃输入, $u_2=0$)

3.1.2 单位脉冲响应分析

在 MATLAB 软件中, 函数 `impulse` 和 `dimpulse` 分别给出了由式(3.0.1)描述的连续时间系统和由式(3.0.2)描述的离散时间系统的单位脉冲响应曲线, 其格式和功能如下。

1) 连续时间系统脉冲响应函数 `impulse`

```
impulse(A, B, C, D)           % A、B、C、D 为系统状态空间模型的相应矩阵
[y, x, t] = impulse(A, B, C, D) % 返回系统输出 y、状态 x 以及相应的时间 t
[y, x, t] = impulse(A, B, C, D, iu) % iu 表示输入变量的序号
[y, x, t] = impulse(A, B, C, D, iu, t) % t 表示自定义时间
```

2) 离散时间系统脉冲响应函数 `dimpulse`

```
impulse(A, B, C, D)           % A、B、C、D 为系统状态空间模型的相应矩阵
[y, x, t] = impulse(A, B, C, D) % 返回系统输出 y、状态 x 以及相应的时间 t
[y, x, t] = impulse(A, B, C, D, iu) % iu 表示输入变量的序号
[y, x, t] = impulse(A, B, C, D, iu, t) % t 表示自定义时间
```

【例 3.1.4】 试求系统

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

的单位脉冲响应。

解：在 MATLAB 软件中, 编写以下的 M 文件(Example314.m)：

```
% 输入状态空间表达式中的矩阵
```

```

A = [0 1; -1 -1];
B = [0;1];
C = [1 0];
D = [0];
impulse(A,B,C,D)           % 绘制单位脉冲响应
grid                       % 网格
title('单位脉冲响应')     % 标题

```

运行 Example314.m 文件,运行结果如图 3.1.4 所示。

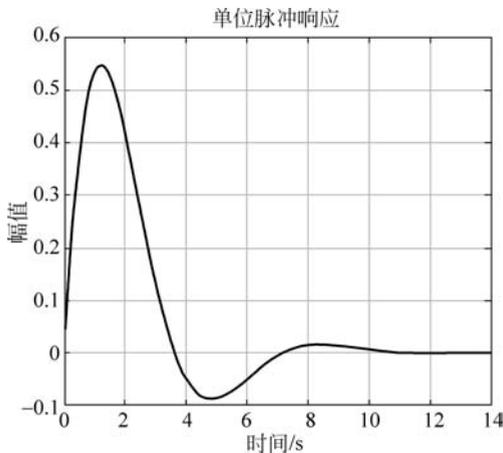


图 3.1.4 单位脉冲响应曲线

3.1.3 初始状态响应分析

在诸如系统的稳定性分析和检验等问题中,需要了解系统在没有外部输入的情况下,系统的状态或输出对初始状态的时间响应。在 MATLAB 软件中,函数 `initial` 和 `dinitial` 分别给出了由式(3.0.1)描述的连续时间系统和由式(3.0.2)描述的离散时间系统输出对初始状态 x_0 的时间响应曲线,其格式和功能如下。

1) 连续时间系统初始状态响应函数 `initial`

```
initial(A,B,C,D,x0,t)
```

2) 离散时间系统初始状态响应函数 `dinitial`

```
dinitial(A,B,C,D,x0,t)
```

其中,A、B、C 和 D 是描述系统状态空间模型的系数矩阵,t 是可由用户确定的时间区间, x_0 是系统的初始状态。类似于前面的讨论,也可以将响应曲线画在同一张图中,以下用一个例子来说明这一函数的应用。

【例 3.1.5】 考虑由以下状态方程描述的系统：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

求该系统对初始状态的时间响应。

解：在 MATLAB 软件中,编写以下 M 文件(Example315.m)：

```
A = [0 1; -10 -5]; % 系统矩阵
B = [0;0];
C = [1 0;0 1];
D = [0;0];
x0 = [2;1]; % 初始状态
t = 0:0.05:3; % 指定时间范围,采样间隔 0.05s
[y,x,t] = initial(A,B,C,D,x0,t); % 初始状态响应
plot(t,x(:,1),t,x(:,2)) % 同时绘制状态 x1,x2
grid % 网格
title('初始条件响应') % 标题
xlabel('时间/s') % x 轴标签
ylabel('x1, x2') % y 轴标签
text(0.55,1.15,'x1') % 在指定坐标位置加标注
text(0.4,-2.9,'x2') % 在指定坐标位置加标注
```

运行 Example315.m 文件,运行结果如图 3.1.5 所示。

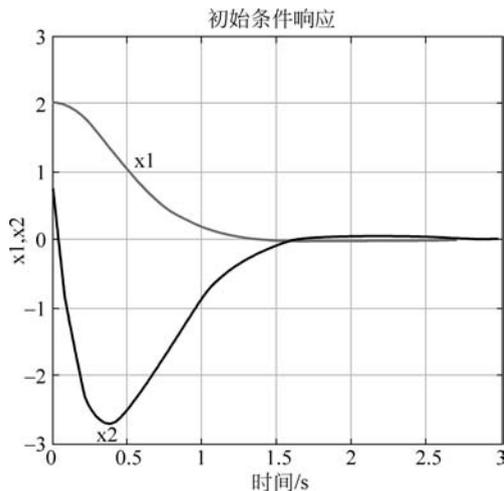


图 3.1.5 系统初始状态响应曲线

3.1.4 任意输入信号响应分析

在 MATLAB 软件中,函数 `lsim` 和 `dlsim` 分别给出了由式(3.0.1)描述的连续时间系统和由式(3.0.2)描述的离散时间系统对任意输入信号的时间响应曲线,其格式和功能如下。

1) 连续时间系统任意输入信号响应函数 `lsim`

```
lsim(sys,u,t,x0)           % 系统对初始状态 x0 和输入 u 的响应
lsim(A,B,C,D,u,t,x0)      % 系统矩阵为(A,B,C,D)系统
[y,t] = lsim(sys,u,t,x0)   % 返回响应输出
```

2) 离散时间系统任意输入信号响应函数 `dlsim`

```
dlsim(sys,u,t,x0)         % 系统对初始状态 x0 和输入 u 的响应
dlsim(A,B,C,D,u,t,x0)    % 系统矩阵为(A,B,C,D)系统
[y,t] = dlsim(sys,u,t,x0) % 返回响应输出
```

其中, `sys` 表示储存在计算机内的状态空间模型,它可以由函数 `sys = ss(A,B,C,D)` 来得到, `x0` 是初始状态,时间区间由用户给定。若初始状态是零,则可以省略 `x0`。下面用一个例子来说明这一函数的应用。

【例 3.1.6】 试求如下系统:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -1 & 0.5 \\ -1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0] \mathbf{x}(t)$$

在输入 $u = e^{-t}$ 下的输出响应,假定系统的初始状态 $\mathbf{x}(0) = 0$ 。

解: 在 MATLAB 软件中,编写以下 M 文件(Example316.m):

```
t = 0:0.1:12;           % 指定时间范围,采样间隔 0.1s
A = [-1 0.5; -1 0];    % 系统矩阵
B = [0;1];
C = [1 0];
D = [0];
sys = ss(A,B,C,D);      % 状态空间模型
u = exp(-t);            % 指定输入
[y,t] = lsim(sys,exp(-t),t);
plot(t,y)               % 绘制系统响应曲线
grid                    % 网格
title('指数输入信号响应(u = exp^-t)') % 标题
xlabel('时间/s')        % x 轴标签
ylabel('输出')          % y 轴标签
```

运行 Example316.m 文件,运行结果如图 3.1.6 所示。

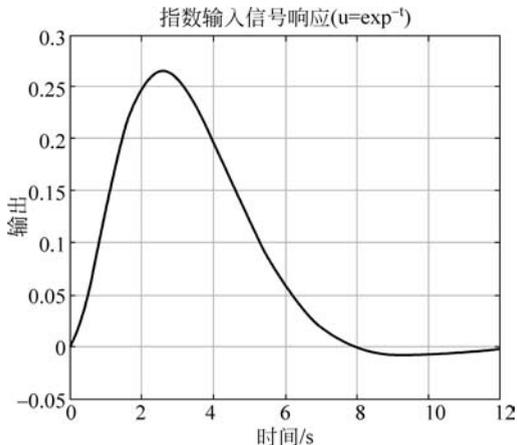


图 3.1.6 给定输入信号下的状态响应曲线

3.2 状态空间模型的能控性和能观性分析

系统的状态变量反映了系统内部的全部动态特征,系统的运动分析揭示了系统状态变量的运动行为。然而,当系统的运动状况不满意时,能否通过系统的控制输入来改变系统的动态变化行为呢?这就需要检验输入对系统状态/输出的影响或控制能力,这种对状态/输出的控制能力就是系统的状态/输出能控性。另一方面,要实现所设计的反馈控制,需要系统的信息,可利用的系统信息越多,所能达到的系统性能往往就越好。系统能直接测量得到的信息是系统的输出,而系统内部的全部动态信息由状态反映。那么,系统的输出能否反映系统状态的信息呢?这就是系统的状态能观测性问题。状态/输出能控性反映了输入对系统状态/输出的影响和控制能力,能观性反映了输出对系统状态的识别能力,它们反映了系统本身的内在特性。这两个概念是卡尔曼在 20 世纪 60 年代提出的,是现代控制理论中的两个基本概念。本节将给出基于 MATLAB 软件的系统状态/输出能控性和能观性的分析方法。

3.2.1 状态能控性分析

由式(3.0.1)描述的连续时间系统状态能控的充分必要条件是

$$\text{rank}(\Gamma_c[\mathbf{A}, \mathbf{B}]) = \text{rank}([\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}]) = n$$

其中,整数 n 是系统状态向量的维数。为了判别系统的状态能控性,只需检验由状态空间模型中的状态矩阵 \mathbf{A} 和输入矩阵 \mathbf{B} 构成的矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 是否满秩。由此也可以看出,矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 在系统状态能控性检验中起着重要作用,故将矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 称为该系统的状态能控性判别矩阵,简称状态能控性矩阵。

状态能控性矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 只依赖系统状态方程中的状态矩阵和输入矩阵,与状态能控性定义中的终端时间 T 无关。这表明一个系统若是状态能控的,则对任意给定的时间间隔 $[0, T]$,都存在使得在该时间段内将初始状态转移到零状态的控制律。

如何来有效判断状态能控性矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 是否满秩呢?对于单输入系统, $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 是一个 $n \times n$ 维的矩阵,可以通过判断 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 的行列式是否为零来确定它是否满秩。而对一个具有 m 个输入的系统, $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 是一个 $n \times nm$ 维的矩阵,而 $(\Gamma_c[\mathbf{A}, \mathbf{B}])(\Gamma_c[\mathbf{A}, \mathbf{B}])^T$ 是一个 $n \times n$ 维的矩阵。由线性代数的知识可知: $\text{rank}(\Gamma_c[\mathbf{A}, \mathbf{B}]) = \text{rank}((\Gamma_c[\mathbf{A}, \mathbf{B}])(\Gamma_c[\mathbf{A}, \mathbf{B}])^T)$ 。故可以通过检验 $n \times n$ 维矩阵 $(\Gamma_c[\mathbf{A}, \mathbf{B}])(\Gamma_c[\mathbf{A}, \mathbf{B}])^T$ 的行列式是否为零来判断矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 是否满秩。

对给定的连续时间状态空间模型(见式(3.0.1)),MATLAB给出了求系统状态能控性矩阵的函数 $\text{ctrb}(\mathbf{A}, \mathbf{B})$ 。因此,对于单输入的系统,可以根据 $\det(\text{ctrb}(\mathbf{A}, \mathbf{B}))$ 是否等于零来判别系统的能控性;而对多输入系统,可以用 $\det(\text{ctrb}(\mathbf{A}, \mathbf{B}) * \text{ctrb}(\mathbf{A}, \mathbf{B})')$ 是否等于零来判别系统的能控性,当然这一方法也适用于单输入系统。此外,可以用秩函数 rank 直接给出能控性矩阵的秩 $\text{rank}(\text{ctrb}(\mathbf{A}, \mathbf{B}))$ 。下面用一个例子来说明这一方法的应用。

【例 3.2.1】 判断线性定常系统

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & 3 & 2 \\ 0 & 2 & 0 \\ 0 & 1 & 3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} \mathbf{u}$$

的能控性,其中, \mathbf{x} 是三维的状态向量, \mathbf{u} 是二维的控制向量。

解: 在 MATLAB 软件命令行窗口输入以下指令:

```
>> A = [1 3 2; 0 2 0; 0 1 3];
>> B = [2 1; 1 1; -1 -1];
>> rank(ctrb(A, B))
```

运行结果如下:

```
ans =
     2
```

即能控性矩阵的秩等于 2,小于系统的阶数 3,故系统是状态不能控的。

尽管状态能控性矩阵是对连续时间状态空间模型导出的,但这个结论对离散时间系统也是成立的。

3.2.2 输出能控性分析

由式(3.0.1)描述的连续时间系统输出完全能控的充分必要条件是 $p \times (nm + m)$ 维输出能控性矩阵

$$[\mathbf{CB} \quad \mathbf{CAB} \quad \mathbf{CA}^2\mathbf{B} \quad \cdots \quad \mathbf{CA}^{n-1}\mathbf{B} \quad \mathbf{D}]$$

是行满秩的,即秩等于 p 。注意 p 是输出变量的个数。

【例 3.2.2】 试判断以下系统的状态和输出能控性:

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \\ y &= [1 \quad 0] \mathbf{x}\end{aligned}$$

解:系统的状态能控性矩阵为

$$\Gamma_c[\mathbf{A}, \mathbf{B}] = [\mathbf{B} \quad \mathbf{AB}] = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

由于 $\det(\Gamma_c[\mathbf{A}, \mathbf{B}]) = 0$,故系统是状态不能控的。进一步,系统的输出能控性矩阵为

$$\mathbf{S} = [\mathbf{CB} \quad \mathbf{CAB} \quad \mathbf{D}] = [1 \quad -1 \quad 0]$$

显然,输出能控性矩阵 \mathbf{S} 是行满秩的,故系统是输出能控的。

这个例子说明了系统的状态能控性和输出能控性没有必然的因果关系,即对任意一个系统,其输出能控不一定状态能控。另一方面,状态能控也不一定输出能控。

3.2.3 状态能观性分析

由式(3.0.1)描述的连续时间系统状态能观的充分必要条件是

$$\text{rank}(\Gamma_o[\mathbf{C}, \mathbf{A}]) = \text{rank} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} = n \quad (3.2.1)$$

其中,整数 n 是系统状态向量的维数。为了判别系统的状态能观性,只需检验由状态空间模型中的状态矩阵 \mathbf{A} 和输出矩阵 \mathbf{C} 构成的矩阵 $\Gamma_o[\mathbf{C}, \mathbf{A}]$ 是否满秩。由此也可以看出,矩阵 $\Gamma_o[\mathbf{C}, \mathbf{A}]$ 在系统状态能观性检验中起着重要作用,故将矩阵 $\Gamma_o[\mathbf{C}, \mathbf{A}]$ 称为该系统的状态能观性判别矩阵,简称状态能观性矩阵。

对给定的连续时间状态空间模型(见式(3.0.1)),MATLAB 软件提供了生成矩阵 $\Gamma_o[\mathbf{C}, \mathbf{A}]$ 的函数 `obsv`,它的一般形式为 `obsv(A, C)`。因此,对于单输入系统,可以根据 $\det(\text{obsv}(A, C))$ 是否等于零来判别系统的能观性;而对于多输入系统,可以用 $\det(\text{obsv}(A, C)' * \text{obsv}(A, C))$ 是否等于零来判别系统的能观性。此外,可以用秩函数 `rank` 直接给出能观性矩阵的秩 `rank(obsv(A, C))`。

3.3 现代控制系统的稳定性分析

在控制工程中,设计者往往希望所设计的系统在受到扰动后,尽管系统会偏离处于平衡状态的工作点,但在扰动消失后,它有能力自动回到并保持在工作点附近,如倒立摆装置中,当摆杆受扰动而偏离垂直位置后,系统仍能使摆杆回到垂直位置,并能始终保持在

垂直位置附近。这就是系统稳定的基本含义。稳定性是一个控制系统能正常工作的基本要求,系统只有在稳定的前提下才能进一步探讨其他特性。因此,稳定性问题一直是自动控制理论中的一个最基本和最重要的问题,控制系统的稳定性分析是系统分析的首要任务。

李雅普诺夫(Lyapunov)第二方法是一种定性方法,它无须求解复杂的系统微分方程,而是通过构造一个类似于能量函数的标量李雅普诺夫函数,然后再根据李雅普诺夫函数的性质来直接判定系统的稳定性。因此,它特别适合于那些难以求解的非线性系统和时变系统。由于这一方法无须求解系统微分方程的解就可直接判定系统稳定性,故称其为李雅普诺夫直接法。李雅普诺夫第二方法不仅可用于系统稳定性分析,而且还可用于系统过渡过程特性的评价以及参数最优化问题的求解,该方法的最大优点是它可用于控制系统的设计,从而使得该方法在自动控制的各个分支中都有广泛的应用,是控制理论中最重要的理论和方法之一。

考虑系统 $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$ 的平衡状态 $\boldsymbol{x}_e = \mathbf{0}$, 如果对任意给定的 $\epsilon > 0$, 存在一个 $\delta > 0$ (与 ϵ 和初始时刻 t_0 有关), 使得从球域 $S(\delta)$ 内任一初始状态出发的状态轨线始终都保持在球域 $S(\epsilon)$ 内, 则称平衡状态 $\boldsymbol{x}_e = \mathbf{0}$ 是李雅普诺夫意义下稳定的。

考虑系统 $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$ 的平衡状态 $\boldsymbol{x}_e = \mathbf{0}$, 如果平衡状态 $\boldsymbol{x}_e = \mathbf{0}$ 是李雅普诺夫意义下稳定的, 并且当 $t \rightarrow \infty$ 时, 始于原点邻域中的轨线 $\boldsymbol{x}(t) \rightarrow \mathbf{0}$, 则平衡状态 $\boldsymbol{x}_e = \mathbf{0}$ 称为在李雅普诺夫意义下是渐近稳定的。

若由式(3.0.1)描述的连续时间系统是稳定的, 且对系统的任意状态, 以该状态为初始状态的状态轨线随着时间的推移都收敛到平衡状态 $\boldsymbol{x}_e = \mathbf{0}$, 则系统称为是大范围渐近稳定的。

由于从状态空间中任意点出发的状态轨线都要收敛于原点, 因此, 大范围渐近稳定的系统在整个状态空间中只能有一个平衡状态, 这也是系统大范围渐近稳定的必要条件。

如果存在某个实数 $\epsilon > 0$, 对不管多么小的 $\delta > 0$, 在球域 $S(\delta)$ 内总存在一个状态 \boldsymbol{x}_0 , 使得始于这一状态的状态轨线最终会离开球域 $S(\epsilon)$, 则平衡状态 $\boldsymbol{x}_e = \mathbf{0}$ 称为不稳定的。

3.3.1 连续时间系统的稳定性分析

考虑线性时不变自治系统

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \quad (3.3.1)$$

其中, \boldsymbol{x} 是系统的 n 维状态向量, \boldsymbol{A} 是 $n \times n$ 维状态矩阵。显然, $\boldsymbol{x}_e = \mathbf{0}$ 是系统的平衡状态。

考虑线性时不变自治系统(见式(3.3.1)), 其在平衡点 $\boldsymbol{x}_e = \mathbf{0}$ 处渐近稳定的充分必要条件是

$$\boldsymbol{A}^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} < \mathbf{0} \quad (3.3.2)$$

存在一个对称正定矩阵 \boldsymbol{P} , 使得式(3.3.2)成立。选取一个对称正定矩阵 \boldsymbol{Q} , 若矩阵方程

$$\boldsymbol{A}^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} = -\boldsymbol{Q} \quad (3.3.3)$$

有一个对称正定矩阵解 \boldsymbol{P} , 则该对称正定矩阵 \boldsymbol{P} 满足式(3.3.2)。而对于给定的对称正定

矩阵 Q , 式(3.3.3)是关于矩阵 P 的元素的一个线性方程组, 从而可以应用求解线性方程组的方法从式(3.3.3)中求取解矩阵 P 。

在求解式(3.3.3)时, 需要首先给定一个对称正定矩阵 Q 。那么是否会出现对某个给定的矩阵 Q , 式(3.3.3)无解, 而对另一个给定的矩阵 Q , 式(3.3.3)又有解呢? 理论上可以证明, 式(3.3.3)的可解性不依赖矩阵 Q 的选取, 即若对某一个矩阵 Q , 式(3.3.3)是可解的, 则对所有的对称正定矩阵 Q , 式(3.3.3)都是可解的, 尽管在计算上会有一些的差异。基于这一事实, 为了方便起见, 在具体系统的稳定性分析中常将矩阵 Q 选为单位矩阵, 即 $Q=I$ 。

以上分析表明了可以通过求解一个线性方程组(见式(3.3.3)), 并检验由此得到的矩阵 P 的正定性来判别线性时不变自治系统(见式(3.3.1))是否是渐近稳定的, 即线性时不变自治系统(见式(3.3.1))在平衡点 $x_e=0$ 处渐近稳定的充分必要条件是对任意给定的对称正定矩阵 Q , 存在一个对称正定矩阵 P , 使得矩阵方程式(3.3.3)成立。

矩阵方程式(3.3.3)在检验线性时不变系统稳定性中起着重要的作用, 因此给它一个特殊的名字——李雅普诺夫矩阵方程, 简称李雅普诺夫方程, 而不等式(3.3.2)则称为是线性时不变自治系统的李雅普诺夫矩阵不等式, 相应的矩阵 P 称为是线性时不变自治系统的一个李雅普诺夫矩阵, 由矩阵 P 可以确定系统的一个李雅普诺夫函数 $V(x)=x^T P x$, 同时也可以得到 $dV(x)/dt=-x^T Q x$ 。通过求解李雅普诺夫方程(见式(3.3.3))来判别系统稳定性的方法称为是稳定性分析的李雅普诺夫方程处理方法。

MATLAB 软件给出了求解李雅普诺夫方程的函数, 它的一般形式是

```
P = lyap(A', Q) % 求解矩阵方程式(3.3.3), 返回 P 矩阵
```

注意, 为了求解形如式(3.3.3)的李雅普诺夫方程, 在函数 lyap 的输入量中用的是 A' 。特别地, 给出矩阵方程

$$AP + PB = -Q \quad (3.3.4)$$

可用以下函数形式进行求解:

```
P = lyap(A, B, Q) % 求解矩阵方程式(3.3.4), 返回 P 矩阵
```

求解李雅普诺夫方程的函数还有 lyap2, 它主要用于连续系统李雅普诺夫方程的符号解法。

【例 3.3.1】 设二阶线性时不变系统的状态方程为

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

试分析系统平衡状态的稳定性。

解: 在 MATLAB 命令行窗口中输入如下指令:

```
>> A = [0 1; -1 -1];
>> P = lyap(A', eye(2))
```

运行结果如下:

```
P =
    1.5000    0.5000
    0.5000    1.0000
```

进一步,在命令行窗口中输入

```
>> eig(P)
```

可得矩阵 P 的特征值,运行结果如下:

```
ans =
    1.8090
    0.6910
```

由于矩阵 P 的所有特征值都是正的,故矩阵 P 是正定的,从而可以得到该系统平衡状态是渐近稳定的结论。

为了分析例 3.3.1 中系统平衡状态的稳定性,根据李雅普诺夫稳定性理论,就是要检验是否存在一个 2×2 维的对称矩阵 P ,使得线性矩阵不等式(Linear Matrix Inequality, LMI)

$$\begin{cases} PA + A^T P < 0 \\ P > 0 \end{cases} \quad (3.3.5)$$

是可行的,即存在一个正定对称矩阵解 P ,满足线性矩阵不等式(3.3.5)。如果线性矩阵不等式(3.3.5)是可行的,则该系统平衡状态是渐近稳定的。为此,应用 LMI 工具箱提供的相关命令和函数来检验线性矩阵不等式(3.3.5)的可行性。

在 MATLAB 软件中,编写以下的 M 文件(Example332.m):

```
% 输入系统状态矩阵
A = [0 1; -1 -1];
% 以命令 setlmis 开始描述一个线性矩阵不等式
setlmis([])
% 定义线性矩阵不等式中的决策变量 P
P = lmivar(1,[2 1]);
% 依次描述所涉及的线性矩阵不等式
% 1st LMI
lmiterm([1 1 1 P],A',1,'s');
% 2nd LMI
lmiterm([2 1 1 P],-1,1);
% 以命令 getlmis 结束线性矩阵不等式系统的描述,并命名为 lmis
lmis = getlmis;
% 调用线性矩阵不等式系统可行性问题的求解器 feasp
[tmin,xfeas] = feasp(lmis);
% 将得到的决策变量值转化为矩阵形式
PP = dec2mat(lmis,xfeas,P)
```

运行 Example332.m 文件,可得相应的线性矩阵不等式(3.3.5)是可行的,且该不等式的一个可行解为

```

PP =
    93.7314    27.4336
    27.4336    70.8701

```

进一步,命令行窗口中输入

```
>> eig(P)
```

可得矩阵 P 的特征值,运行结果如下:

```

ans =
    112.0205
     52.5810

```

因此,得到的解矩阵 P 是正定的。根据 Lyapunov 稳定性理论,该系统平衡状态是渐近稳定的。

3.3.2 离散时间系统的稳定性分析

考虑离散时间线性时不变系统 $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$, 假设原点是该系统的一个平衡状态。则该系统在原点处渐近稳定的充分必要条件是对任意给定的对称正定矩阵 Q , 矩阵方程

$$\mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{P} = -\mathbf{Q} \quad (3.3.6)$$

存在对称正定解矩阵 P 。

式(3.3.6)称为离散李雅普诺夫矩阵方程,其解矩阵 P 称为系统(3.3.1)的李雅普诺夫矩阵。对于给定的矩阵 Q , 式(3.3.6)是关于矩阵 P 中元素的一个线性方程组,因此,可以通过求解线性方程组的方法来求解离散李雅普诺夫方程。

离散李雅普诺夫矩阵方程式(3.3.6)的可解性并不依赖于矩阵 Q 的选取,因此,在具体应用时,可选取 $Q = I$, 然后求解方程

$$\mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{P} = -\mathbf{I}$$

进而检验所得到的解矩阵 P 是否正定,从而确定系统的稳定性。

MATLAB 软件也提供了求解离散李雅普诺夫矩阵方程式(3.3.6)的函数,即

```
P = dlyap(A', Q) % 求解矩阵方程式(3.3.6), 返回 P 矩阵
```

其用法与 lyap 相似,此处不再赘述。