

第5章

实例1：打印机驱动控制系统分析与设计

打印机(Printer)是计算机的重要输出设备之一,是由约翰·沃特和戴夫·唐纳德合作发明的,用于将计算机的运算结果或中间结果以人所能识别的数字、字母、符号和图像等,依照规定的格式打印在相关介质上的设备。打印分辨率、打印速度和噪声是衡量打印机质量的重要指标。打印机的分类很多,如按打印元件对介质是否有击打动作,分为击打式打印机与非击打式打印机;按打印字符结构,分为全形字打印机和点阵字符打印机;按一行字在纸上形成的方式,分为串式打印机与行式打印机;按采用的实现技术,分为喷墨式、激光式、热敏式、静电式、磁式、发光二极管式等打印机类型。目前,打印机正向轻、薄、短、小、低功耗、高速度和智能化方向发展。

打印机驱动系统是负责打印速度的重要单元,本章以打印机皮带驱动系统为例,采用 MATLAB 软件实现打印机驱动系统现代控制方法设计。首先,介绍打印机皮带驱动系统的状态空间模型,再对建立的状态空间模型分析打印机皮带驱动系统的稳定性、能控性、能观性等性能。进一步,运用极点配置状态反馈控制方法,设计打印机皮带驱动系统的状态反馈控制器和跟踪控制器,提高打印机皮带驱动系统的动态性能和鲁棒性能。最后,考虑打印机皮带驱动系统难以直接测量状态变量,设计系统的状态观测器,估计该系统中难以测量的状态变量,减少实际应用中传感器的使用数量,节约打印机生产成本和维护成本。

5.1 打印机皮带驱动系统模型

5.1.1 驱动系统微分方程

计算机常用的低价位打印机都有皮带驱动器,用来驱动打印设备沿打印页面横向移动。图 5.1.1 给出了一个装有直流电机的打印机皮带驱动系统示意图,其中,光传感器用来测量打印设备的位置,皮带张力变化用来调整滑轮的转动。

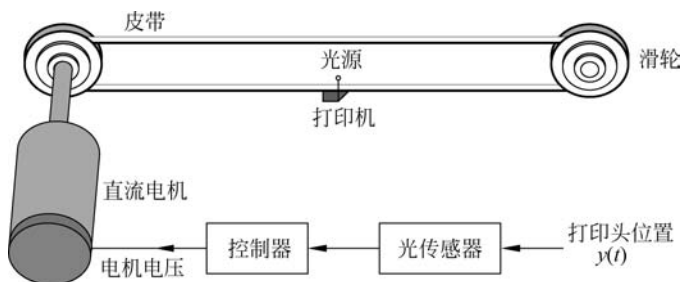


图 5.1.1 打印机皮带驱动系统示意图

为建立如图 5.1.1 所示的打印机皮带驱动系统的微分方程模型,设定打印机皮带驱动系统的皮带弹性系数为 k ,滑轮半径为 r ,电机轴转角为 θ ,右滑轮转角为 θ_p ,打印设备质量为 m 。打印设备在时间 t 的位置值为 $y(t)$,可以通过光传感器测量位置变量 y 。光传感器的输出电压 $v_1 = k_1 y$,控制器的输出电压为 v_2 , v_2 与电机激励磁场相关。此外,控制器输出电压 v_2 是传感器输出电压 v_1 的函数,通常满足如下线性关系:

$$v_2 = - \left[k_2 \frac{dv_1}{dt} + k_3 v_1 \right] \quad (5.1.1)$$

其中, k_2 为微分系数, k_3 为比例系数。具体参数和受力分析如图 5.1.2 所示,其中, T_1 和 T_2 分别为皮带上下侧的张力, θ 和 θ_p 分别为电机和滑轮的转角。令电机和滑轮的总转动惯量为 $J = J_{\text{电机}} + J_{\text{滑轮}}$ 。通常采用中等功率的直流电机,如选定电机功率为典型的 $\frac{1}{8}$ 马力 (1 马力 = 735.499W) 情况下,有 $J = 0.01 \text{ kg} \cdot \text{m}^2$ 。进一步,假设磁场电感忽略不计,磁场电阻 $R = 2 \Omega$,电机常数 $K_m = 2 \text{ N} \cdot \text{m/A}$,而电机和滑轮的摩擦系数 $b = 0.25 \text{ N} \cdot \text{ms/rad}$,滑轮半径 $r = 0.15 \text{ m}$ 。

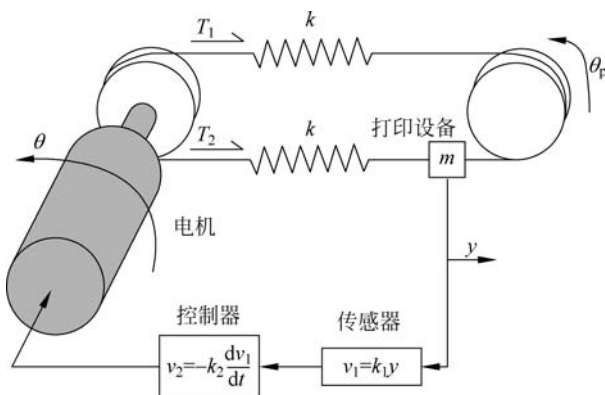


图 5.1.2 打印机皮带驱动系统模型简图

现在列写打印机皮带驱动系统的运动方程。注意到 $y = r\theta_p$, 于是皮带张力 T_1 可表示为

$$T_1 = k(r\theta - r\theta_p) = k(r\theta - y) \quad (5.1.2)$$

张力 T_2 为 $T_2 = k(y - r\theta)$, 作用在质量 m 上的净张力为

$$T_1 - T_2 = m \frac{d^2 y}{dt^2} \quad (5.1.3)$$

且 $T_1 - T_2 = k(r\theta - y) - k(y - r\theta) = 2k(r\theta - y) = 2kx_1$, 其中第一个状态变量为 $x_1 = (r\theta - y)$ 。令第二个状态变量为 $x_2 = dy/dt$, 并利用式(5.1.2)和式(5.1.3)可得

$$\frac{dx_2}{dt} = \frac{2k}{m}x_1 \quad (5.1.4)$$

当选择第三个状态变量为 $x_3 = d\theta/dt$ 时, x_1 的一阶导数为

$$\frac{dx_1}{dt} = r \frac{d\theta}{dt} - \frac{dy}{dt} = rx_3 - x_2 \quad (5.1.5)$$

接下来, 寻求描述电机旋转运动的微分方程。当 $L=0$ 时, 有电机磁场电流 $i = v_2/R$ 和电机转矩 $T_m = K_m i$ 。因此

$$T_m = \frac{K_m}{R}v_2 \quad (5.1.6)$$

而电机输出转矩提供了驱动皮带的转矩与扰动或者不希望的负载转矩之和, 因此 $T_m = T + T_d$ 。转矩 T 驱动电机轴带动滑轮运动, 满足

$$T = J \frac{d^2 \theta}{dt^2} + b \frac{d\theta}{dt} + r(T_1 - T_2) \quad (5.1.7)$$

因此

$$\frac{dx_3}{dt} = \frac{d^2 \theta}{dt^2} \quad (5.1.8)$$

所以有

$$\frac{dx_3}{dt} = \frac{-K_m k_1 k_2}{JR}x_2 - \frac{b}{J}x_3 - \frac{2kr}{J}x_1 - \frac{T_d}{J} \quad (5.1.9)$$

5.1.2 驱动系统状态空间模型

式(5.1.4)、式(5.1.5)、式(5.1.9)是描述系统的3个一阶微分方程, 其矩阵形式为

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & -1 & r \\ \frac{2k}{m} & 0 & 0 \\ -\frac{2kr}{J} & -\frac{K_m k_1 k_2}{JR} & -\frac{b}{J} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J} \end{bmatrix} T_d \quad (5.1.10)$$

假设某个打印机皮带驱动系统的结构参数取值如下: 质量 $m = 0.2\text{kg}$, 光传感器 $k_1 =$

1V/m, 半径 $r=0.15\text{m}$, 电感 $L=0$, 摩擦系数 $b=0.25\text{N}\cdot\text{ms/rad}$, 电阻 $R=2\Omega$, 常数 $K_m=2\text{N}\cdot\text{m/A}$, 惯量 $J=0.01\text{kg}\cdot\text{m}^2$, 弹性系数 $k=20$ 和 $k_2=0.1$, 且选择使用 $k_2=0.1$ 和 $k_3=0$ (速度反馈)。把以上参数代入打印机皮带驱动系统的状态方程, 可得

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & -1 & 0.15 \\ 200 & 0 & 0 \\ -600 & -10 & -25 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ -100 \end{bmatrix} u \quad (5.1.11)$$

定义系统输出变量为

$$y = -x_3$$

则连续时间驱动系统的状态空间模型为

$$\begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} 0 & -1 & 0.15 \\ 200 & 0 & 0 \\ -600 & -10 & -25 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ -100 \end{bmatrix} u \\ y = [0 \ 0 \ -1] \mathbf{x} \end{cases} \quad (5.1.12)$$

5.1.3 驱动系统状态空间模型的离散化

以 $T=0.05\text{s}$ 为采样周期, 采用 MATLAB 软件中的 c2d 函数, 将打印机皮带驱动系统状态空间模型离散化。在 MATLAB 命令窗口中输入如下指令:

```
>> A = [0 -1 0.15; 200 0 0; -600 -10 -25]; % 系统的状态矩阵
>> B = [0; 0; -100]; % 系统的输入矩阵
>> [G, H] = c2d(A, B, 0.05) % 调用 c2d 函数
```

运行结果如下:

```
G =
    0.6871    -0.0458    0.0037
    8.9114     0.7594    0.0241
   -16.2917     0.2372    0.2359
H =
   -0.0120
   -0.0452
   -2.7492
```

因此, 所求的离散化状态空间模型为

$$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} 0.6871 & -0.0458 & 0.0037 \\ 8.9114 & 0.7594 & 0.0241 \\ -16.2917 & 0.2372 & 0.2359 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} -0.0120 \\ -0.0452 \\ -2.7492 \end{bmatrix} u(k) \\ y(k) = [0 \ 0 \ -1] \mathbf{x}(k) \end{cases} \quad (5.1.13)$$

若取采样周期为 $T=0.01\text{s}$, 则类似可得相应的离散状态空间模型为

$$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} 0.9858 & -0.01 & 0.0013 \\ 1.9905 & 0.99 & 0.0014 \\ -5.3739 & -0.0605 & 0.775 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} -0.0007 \\ -0.0005 \\ -0.8835 \end{bmatrix} u(k) \\ y(k) = [0 \quad 0 \quad -1] \mathbf{x}(k) \end{cases} \quad (5.1.14)$$

从以上两个离散状态空间模型可以看出,不同的采样周期所得到的离散模型是不同的。这也验证了离散状态空间模型依赖于所选取的采样周期的结论。

5.1.4 驱动系统状态空间模型转换为传递函数模型

采用 MATLAB 软件中的 ss2tf 函数,将打印机皮带驱动系统状态空间模型转换为传递函数模型。在 MATLAB 命令窗口中输入如下指令:

```
>> A = [0 -1 0.15; 200 0 0; -600 -10 -25]; % 系统的状态矩阵
>> B = [0; 0; -100]; % 系统的输入矩阵
>> C = [0 0 -1]; % 系统的输出矩阵
>> D = 0; % 系统的直接转移矩阵
>> [num, den] = ss2tf(A, B, C, D) % 调用 ss2tf 函数
```

运行结果如下:

```
num =
    1.0e+04 *
         0         0.0100         0         2.0000
den =
    1.0e+03 *
    0.0010    0.0250    0.2900    5.3000
```

因此,所求系统的传递函数为

$$G(s) = \frac{100s^2 + 20000}{s^3 + 25s^2 + 290s + 5300} \quad (5.1.15)$$

5.2 打印机皮带驱动系统性能分析

5.2.1 驱动系统运动响应分析

1. 驱动系统单位脉冲响应分析

为了获得打印机皮带驱动系统的单位脉冲响应,在 MATLAB 软件中调用 impulse 函数,编写如下 M 文件(Example521.m):

```
% 系统的状态矩阵
A = [0 -1 0.15; 200 0 0; -600 -10 -25];
```

```

% 系统的输入矩阵
B = [0;0;-100];
% 系统的输出矩阵
C = [0 0 -1];
% 系统的直接转移矩阵
D = [0];
% 调用 impulse 函数
impulse(A,B,C,D)
ylabel('y')
xlabel('时间/s')
grid on

```

运行 Example521.m 文件,结果如图 5.2.1 所示。

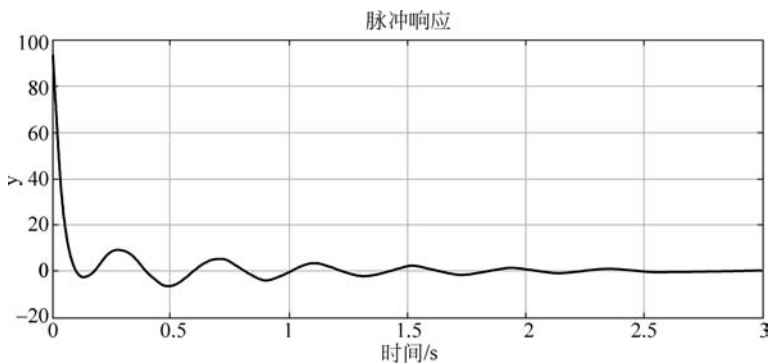


图 5.2.1 系统的单位脉冲响应曲线

2. 驱动系统初始状态响应分析

为了获得打印机皮带驱动系统的初始状态响应,在 MATLAB 软件中调用 initial 函数,编写如下 M 文件(Example522.m):

```

% 系统的状态矩阵
A = [0 -1 0.15;200 0 0;-600 -10 -25];
% 系统的输入矩阵
B = [0;0;-100];
% 系统的输出矩阵
C = [0 0 -1];
% 系统的直接转移矩阵
D = [0];
% 系统的初始状态
x0 = [2,1,1];
% 调用 initial 函数
[y,x] = initial(A,B,C,D,x0);
t = 0:0.05:55.20;
% 创建子图
subplot(3,1,1)

```

```

% 调用 plot 画图函数
plot(t,x(:,1))
% 给 x 轴加标注
xlabel('时间/s')
% 给 y 轴加标注
ylabel('x_1')
subplot(3,1,2)
plot(t,x(:,2))
xlabel('时间/s')
ylabel('x_2')
subplot(3,1,3)
plot(t,x(:,3))
xlabel('time/s')
ylabel('x_3')

```

运行 Example522.m 文件,结果如图 5.2.2 所示。

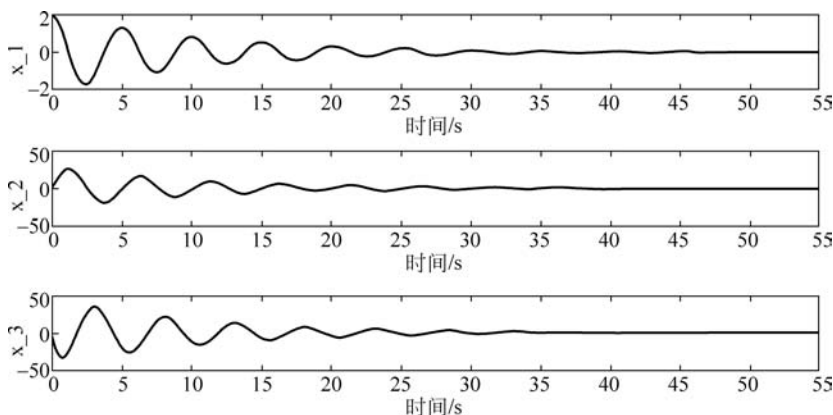


图 5.2.2 系统的初始状态响应曲线

5.2.2 驱动系统能控性和能观性分析

(1) 系统状态能控的充分必要条件为

$$\text{rank}(\Gamma_c[\mathbf{A}, \mathbf{B}]) = \text{rank}([\mathbf{B} \quad \mathbf{AB} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}]) = n \quad (5.2.1)$$

即状态能控性判别矩阵 $\Gamma_c[\mathbf{A}, \mathbf{B}]$ 满秩意味着系统是能控的,系统能控意味着系统出现偏差时,总可以通过控制使得偏差为零。

在 MATLAB 软件命令行窗口输入如下指令:

```

% 系统的状态矩阵
>> A = [0 -1 0.15; 200 0 0; -600 -10 -25];
% 系统的输入矩阵
>> B = [0; 0; -100];
% 系统的输出矩阵

```

```
>> C = [0 0 -1];
% 系统的直接转移矩阵
>> D = 0;
% 获取能控性判别矩阵的秩
>> n = rank(ctrb(A,B))
```

运行结果如下：

```
n =
    3
```

因此可得系统的状态能控性判别矩阵为

$$\text{rank}(\Gamma_c [A, B]) = \text{rank} \left(\begin{bmatrix} 0 & -15 & 375 \\ 0 & 0 & -3000 \\ -100 & 2500 & -53500 \end{bmatrix} \right) = 3 \quad (5.2.2)$$

可以看出,该状态能控性判别矩阵是满秩的,所以系统是能控的。

(2) 系统状态能观的充分必要条件为

$$\text{rank}(\Gamma_o [A, C]) = \text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n \quad (5.2.3)$$

在 MATLAB 软件命令行窗口输入如下指令：

```
% 系统的状态矩阵
>> A = [0 -1 0.15; 200 0 0; -600 -10 -25];
% 系统的输入矩阵
>> B = [0; 0; -100];
% 系统的输出矩阵
>> C = [0 0 -1];
% 系统的直接转移矩阵
>> D = 0;
% 获取能观性判别矩阵的秩
>> n = rank(observ(A,C))
```

运行结果如下：

```
n =
    3
```

因此可得系统的状态能观性判别矩阵为

$$\text{rank}(\Gamma_o [A, C]) = \text{rank} \left(\begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & -1.0000 & 0.1500 \\ -100 & 2500 & -53500 \end{bmatrix} \right) = 3 \quad (5.2.4)$$

可以看出,该状态能观性判别矩阵是列满秩的(即矩阵的秩等于列数),所以系统是状态能观

的,即可以通过系统的输出来观测系统中未知的状态,为观测器的设计提供了理论基础。

5.2.3 驱动系统稳定性分析

对于该系统,采用 Lyapunov 方程处理方法,即线性时不变系统在平衡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 ,存在一个对称正定矩阵 P ,使得方程

$$A^T P + PA = -Q \quad (5.2.5)$$

成立,那么该系统是渐近稳定的。

取 Q 为三阶单位矩阵,在 MATLAB 命令窗口中输入如下指令:

```
% 系统的状态矩阵
>> A = [0 -1 0.15; 200 0 0; -600 -10 -25];
% 对称正定的矩阵,这里取为三阶单位矩阵
>> Q = eye(3);
% 调用 lyap 函数
>> P = lyap(A', Q)
% 调用 eig 函数求特征值
>> val = eig(P)
```

运行结果如下:

```
P =
    137.8737    0.6317    0.2114
     0.6317    0.6604   -0.0132
     0.2114   -0.0132    0.0213
val =
     0.0206
     0.6578
    137.8769
```

由运算结果可得,矩阵 P 的特征值都是正的,所以矩阵 P 正定,所以系统在原点处的平衡状态是渐近稳定的。

5.3 打印机皮带驱动系统控制器设计

5.3.1 驱动系统极点配置状态反馈控制器设计

运用现代控制理论中的状态反馈和极点配置方法,对打印机皮带驱动系统设计状态反馈控制器。因为驱动系统是状态能控的,因此该系统可以通过状态反馈任意配置闭环极点。所设定的闭环极点可根据不同的性能指标进行修改。设计由式(5.1.10)描述的驱动系统的状态反馈控制器为 $u = -Kx$,其中 K 的值由所配置的极点决定。

设计者希望通过引入状态反馈控制器使输出超调量 $\sigma \leq 5\%$,峰值时间 $t_p \leq 0.5s$ 。对

此,设系统有一对主导极点,并且有

$$\lambda_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} \quad (5.3.1)$$

其中, ξ 和 ω_n 是二阶系统的阻尼比和无阻尼自振频率。利用二阶系统的阻尼比 ξ 、无阻尼自振频率 ω_n 等参数和超调量 σ 、峰值时间 t_p 的关系,由

$$\sigma = \exp\left(-\frac{\xi\pi}{\sqrt{1-\xi^2}}\right) \leq 5\%, \quad t_p = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} \leq 0.5 \quad (5.3.2)$$

可得

$$\xi \geq 0.707, \quad \omega_n \geq 9$$

通过式(5.3.2)求得闭环系统的主导极点,而第3个极点选取在远离主导极点的左半开复平面中。因此,选取第3个极点为

$$\text{pole} = [-7.07 + 7.07j \quad -7.07 - 7.07j \quad -100] \quad (5.3.3)$$

采用爱克曼公式法可以得到要设计的状态反馈控制器为

$$u = [-81.5980 \quad 4.3770 \quad -0.8914] \mathbf{x} \quad (5.3.4)$$

从而得到校正后的闭环系统为

$$\begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} 0 & -1 & 0.15 \\ 200 & 0 & 0 \\ -8759.8 & 427.7 & -114.1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ -100 \end{bmatrix} u \\ \mathbf{y} = [0 \quad 0 \quad -1] \mathbf{x} \end{cases} \quad (5.3.5)$$

在 MATLAB 软件中,编写如下 M 文件(Example531.m):

```
% 系统状态矩阵
A = [0 -1 0.15; 200 0 0; -600 -10 -25];
% 系统输入矩阵
B = [0; 0; -100];
% 系统输出矩阵
C = [0 0 -1];
% 系统直接转移矩阵
D = [0];
% 系统的极点配置
J = [-100 -7.07 + 7.07 * j -7.07 - 7.07 * j];
% 调用 acker 函数求取状态反馈增益矩阵
K = acker(A, B, J);
t = 0:0.01:4;
% 调用 step 函数,求配置极点后的系统阶跃响应
[y, x1, t] = step(A - B * K, B, C, D, 1, t);
% 调用 step 函数,求原系统阶跃响应
[yy, x2, t] = step(A, B, C, D, 1, t);
subplot(2, 1, 1)
plot(t, yy)
xlabel('时间/s')
ylabel('yy')
```

```

grid on
subplot(2,1,2);
plot(t,y)
grid on
xlabel('时间/s')
ylabel('y')

```

运行 Example531.m 文件,结果如图 5.3.1 所示。

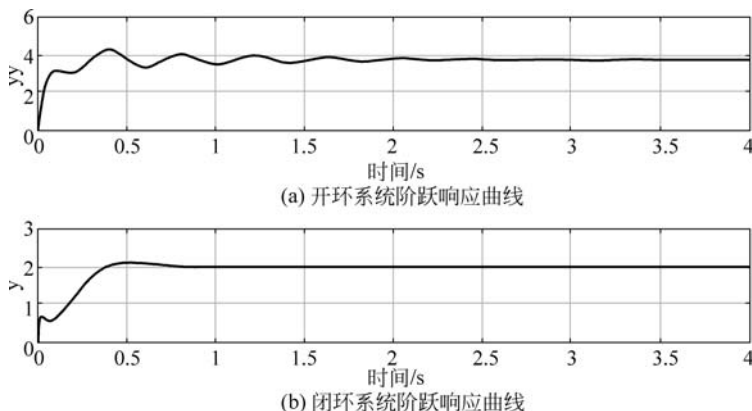


图 5.3.1 开环系统与闭环系统的阶跃响应曲线

从图 5.3.1 可以看出,原系统阶跃响应有较为明显的振荡情况,从图 5.3.1(a)可以大致判断其调节时间约为 2.3s ($\Delta=5$),超调量为 15% 。从控制对象来考虑,打印机打印头的控制容错率非常低,因此无论从调节时间还是超调量来看,原系统动态性能都非常差。因此需要配置极点使闭环系统具有更好的过渡过程性能,在根据需求配置极点后,得到图 5.3.1(b)的响应曲线。可以明显看到,配置极点后系统超调量非常小,调节时间不到 0.5s ,系统动态性能显著提升,但此时的稳态值不再是原系统阶跃响应稳态值 3.774 ,出现了稳态误差。

5.3.2 驱动系统跟踪控制器设计

由原打印机驱动系统的阶跃响应和极点配置后闭环系统的阶跃响应曲线图 5.3.1 可知,校正后的系统存在稳态误差,即极点配置方法可能会使一个原来没有稳态误差的系统产生稳态误差。另一方面,实际驱动系统不可避免地存在外部扰动,比如外力碰撞打印机等。对于其中的确定性扰动,扰动的存在使得打印机驱动系统在稳态时不能很好地跟踪参考输入,从而产生输出稳态误差。因此,需要对打印机驱动系统设计能够实现无静差跟踪阶跃参考输入信号的渐近跟踪调节器,即跟踪控制器。

首先,建立原打印机驱动系统的增广系统

$$\begin{cases} \begin{bmatrix} \dot{x} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ q \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \\ y = [C \quad 0] \begin{bmatrix} x \\ q \end{bmatrix} \end{cases} \quad (5.3.6)$$

为了减小增广系统所增加的动态环节对原打印机驱动闭环系统性能的影响,要将增加的期望极点配置在打印机驱动闭环极点的左边,则再选择一个极点为-9。

在 MATLAB 软件中,编写如下 M 文件(Example532.m):

```
% 增广系统极点配置
A = [0 -1 0.15; 200 0 0; -600 -10 -25]; B = [0; 0; -100]; C = [0 0 -1]; D = [0];
% 增广系统的状态矩阵
AA = [A zeros(3,1); C 0];
% 增广系统的输入矩阵
BB = [B; 0];
% 增广系统极点配置
JJ = [-100 -7.07 + 7.07 * 1i -7.07 -7.07 * 1i -9];
% 调用 acker 函数求取状态反馈增益矩阵
KK = acker(AA, BB, JJ);
% 闭环系统参数矩阵
K1 = [KK(1) KK(2) KK(3)]; K2 = KK(4);
% 增广闭环系统状态矩阵
Ac = [A - B * K1 - B * K2; C 0];
% 增广闭环系统输入矩阵
Bc = [0; 0; 0; -3.774];
% 增广闭环系统输出矩阵
Cc = [C 0];
% 增广闭环系统直接转移矩阵
Dc = 0;
% 绘制闭环输出响应曲线
t = 0:0.01:4;
[yyy, x3, t] = step(Ac, Bc, Cc, Dc, 1, t);
plot(t, yyy)
grid
xlabel('时间/s')
ylabel('y')
```

运行 Example532.m 文件,结果如图 5.3.2 所示。

在程序代码中,将跟踪的参考输出量设定为原系统阶跃响应稳态值 3.774,这样可以方便与原系统阶跃响应曲线进行比较。事实上,在实际系统中可以任意调整参考输出量来控制最后输出稳态值。从图 5.3.2 所示曲线可以看出,改进后的系统完美继承了原来由配置极点方法所设计闭环系统的动态性能,而且消除了稳态误差,从而进一步改善了系统性能。值得一提的是,通过设计跟踪控制器,原先由极点配置控制器得到的阶跃响应曲线

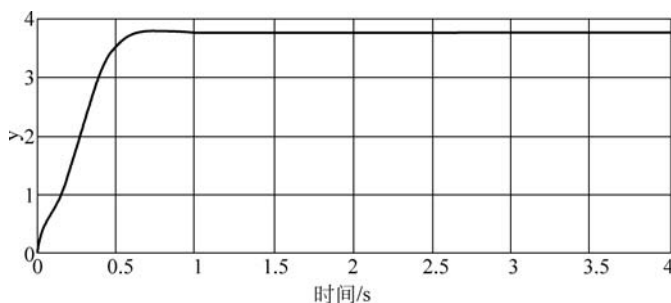


图 5.3.2 加入跟踪控制器后闭环系统的阶跃响应曲线

中无法消除的小尖峰也消除了,由此可得跟踪控制器增加的新极点对系统稳定性也有一定帮助。

5.3.3 驱动系统状态观测器设计

对打印机皮带驱动系统模型进行分析,可以发现系统内部的状态信号不完全是可以从外部直接测量得到的,或者某些信号非常不容易测量,如角加速度难以直接通过传感器测量,而且引入过多的传感器也会在一定程度上降低控制系统的可靠性。所以直接进行状态反馈的控制是不合适的。

从系统的性能分析中可知该系统是状态能观的,那么它所有的状态信息都必定在输出中得到反映,因此可以设计一个状态观测器,用系统的外部输入输出信息来确定其状态量,避免直接测量打印机皮带驱动系统的状态变量。

考虑由式(5.1.10)描述的打印机皮带驱动系统,设计龙伯格观测器为

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) = (A - LC)\hat{x} + Bu + Ly \quad (5.3.7)$$

其中,误差 $e = x - \hat{x}$ 的动态变化为

$$\dot{e} = (A - LC)e \quad (5.3.8)$$

在建立观测器后,可以进一步通过仿真得出误差曲线来检验观测器的效果。

在 MATLAB 命令窗口中输入如下指令:

```
% 系统的状态矩阵
>> A = [0 -1 0.15; 200 0 0; -600 -10 -25];
% 系统的输出矩阵
>> C = [0 0 -1];
% V 为观测器极点
>> V = [-200 -14.14 + j*14.14 -14.14 - j*14.14];
% L 为观测器增益矩阵
>> L = (acker(A',C',V))';
```

得到观测器的增益矩阵为

$$L = \begin{bmatrix} 9.9993 \\ -23.3688 \\ -203.2800 \end{bmatrix}$$

且 $A-LC$ 得

$$AA = \begin{bmatrix} 0 & -1.0000 & 10.1493 \\ 200.0000 & 0 & -23.3688 \\ -600.0000 & -10.0000 & -228.2800 \end{bmatrix}$$

相应的观测器为

$$\dot{\bar{x}} = \begin{bmatrix} 0 & -1 & 10.1493 \\ 200 & 0 & -23.3688 \\ -600 & -10 & -228.2800 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ 0 \\ -100 \end{bmatrix} u + \begin{bmatrix} 9.9993 \\ -23.3688 \\ -203.2800 \end{bmatrix} y \quad (5.3.9)$$

状态估计的误差动态方程为

$$\dot{e} = (A-LC)e = \begin{bmatrix} 0 & -1 & 10.1493 \\ 200 & 0 & -23.3688 \\ -600 & -10 & -228.2800 \end{bmatrix} e \quad (5.3.10)$$

下面进一步通过仿真来检验观测器的效果。取初始误差为

$$e(0) = [1 \quad 2 \quad 1.5]^T$$

在 MATLAB 软件中,编写如下 M 文件(Example534.m):

```
% AA 为误差系统状态矩阵
AA = [0 -1 10.1493; 200 0 -23.3688; -600 -10 -228.28];
% BB 为误差系统输入矩阵
BB = [0; 0; 0];
C = [0 0 -1];
D = 0;
e0 = [1; 2; 1.5]; % 初始误差向量
t = 0:0.01:4;
sys = ss(AA, BB, C, D); % 误差的状态空间模型
% 获得误差系统的初始状态响应
[y, t, e] = initial(sys, e0, t);
% 画出状态变量 1 的误差曲线
subplot(3, 1, 1), plot(t, e(:, 1))
grid
ylabel('e1')
xlabel('时间/s')
subplot(3, 1, 2), plot(t, e(:, 2)) % 画出状态变量 2 的误差曲线
grid
ylabel('e2')
xlabel('时间/s')
subplot(3, 1, 3)
```

```

plot(t,e(:,3))           % 画出状态变量 3 的误差曲线
grid
ylabel('e3')
xlabel('时间/s')

```

运行 Example534.m 文件,可得到状态估计的误差曲线如图 5.3.3 所示。

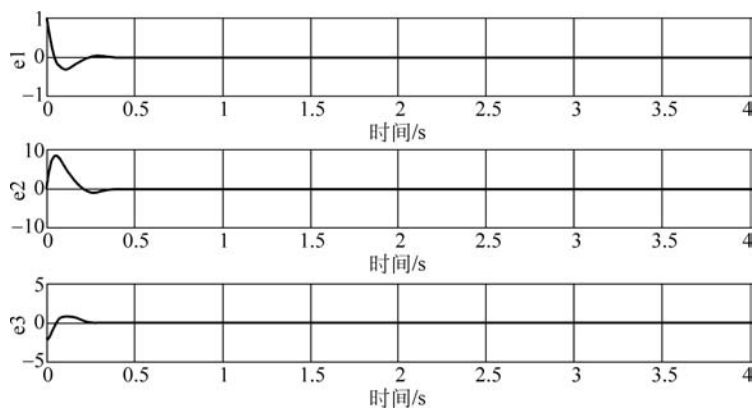


图 5.3.3 状态估计的误差曲线

从图 5.3.3 中可以看出,尽管系统的真实状态和观测器状态的初值有误差,但是随着时间的推移,它们之间的误差将衰减到零。由此可以得出,该观测器符合期望要求。采用这个观测器时,只需要测量输出 x_3 即角速度就可以估计系统的所有状态信息。