

3.1 控制输入受限下的 LMI 控制算法设计

在实际控制系统的设计中,通常面临控制输入受限的问题。采用 LMI 方法,可实现控制输入受限下的控制算法设计^[1~4]。

3.1.1 系统描述

考虑状态方程

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.1)$$

其中 $\mathbf{x} = [x_1 \ x_2]^T$, u 为控制输入。

控制器设计为

$$u = \mathbf{K}\mathbf{x} \quad (3.2)$$

其中, $\mathbf{K} = [k_1 \ k_2]$ 。

控制目标为通过设计 LMI 求解 $\mathbf{K}$, 实现 $\mathbf{x} \rightarrow 0$, $|u| \leq u_{\max}$ 。

3.1.2 控制器的设计与分析

设计 Lyapunov 函数如下:

$$V = \mathbf{x}^T \mathbf{P} \mathbf{x}$$

其中, $\mathbf{P} > 0$, $\mathbf{P} = \mathbf{P}^T$ 。通过 $\mathbf{P}$ 的设计可有效地调节 $\mathbf{x}$ 的收敛效果,并有利于 LMI 的求解。

则

$$\begin{aligned} \dot{V} &= \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} \\ &= (\mathbf{A}\mathbf{x} + \mathbf{B}u)^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} (\mathbf{A}\mathbf{x} + \mathbf{B}u) \\ &= (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}\mathbf{x})^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}\mathbf{x}) \\ &= \mathbf{x}^T (\mathbf{A} + \mathbf{B}\mathbf{K})^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} (\mathbf{A} + \mathbf{B}\mathbf{K}) \mathbf{x} \\ &= \mathbf{x}^T \mathbf{Q}_1^T \mathbf{x} + \mathbf{x}^T \mathbf{Q}_1 \mathbf{x} = \mathbf{x}^T \mathbf{Q} \mathbf{x} \end{aligned}$$

其中, $\mathbf{Q}_1 = \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{K})$, $\mathbf{Q} = \mathbf{Q}_1^T + \mathbf{Q}_1$ 。

为了实现指数收敛,即 $\dot{V} \leq -\alpha V$, 取

$$\alpha V + \dot{V} = \alpha x^T P x + x^T Q x = x^T (\alpha P + Q) x$$

取 $\alpha P + Q < 0, \alpha > 0$, 则 $\alpha V + \dot{V} \leq 0$, 即 $\dot{V} \leq -\alpha V$, 由 $\dot{V} \leq -\alpha V$, 可得解为

$$V(t) \leq V(0) \exp(-\alpha t) \leq V(0)$$

如果 $t \rightarrow \infty$, 则 $V(t) \rightarrow 0$, 从而 $x \rightarrow 0$ 且指数收敛。

3.1.3 LMI 的设计与求解

由于 $V(0) = x_0^T P x_0$, 如果存在正定对称阵 P 和 $\bar{\omega} > 0$, 使得 $x_0^T P x_0 \leq \bar{\omega}$ 成立, 则可保证 $V(0) \leq \bar{\omega}$, 从而保证 $V(t) \leq \bar{\omega}$ 。

取 $K^T K \leq \bar{\omega}^{-1} u_{\max}^2 P$, 由 $u = Kx$ 可得

$$u^2 = (Kx)^T Kx = x^T K^T Kx \leq x^T \bar{\omega}^{-1} u_{\max}^2 P x = \bar{\omega}^{-1} u_{\max}^2 V \leq u_{\max}^2$$

则

$$|u| \leq u_{\max}$$

通过上述分析, 构造两个 LMI 为

$$\alpha P + Q < 0 \quad (3.3)$$

$$K^T K - \bar{\omega}^{-1} u_{\max}^2 P \leq 0 \quad (3.4)$$

不等式(3.3)中, Q 中含有 P 和 K , 式(3.4)中也含有 P 和 K , 故不能独立 Q 存在, 将式(3.3)中的 Q 展开如下:

$$\alpha P + PA + PBK + A^T P + K^T B^T P < 0$$

左右同乘以 P^{-1} , 可得

$$\alpha P^{-1} + AP^{-1} + BKP^{-1} + P^{-1} A^T + P^{-1} K^T B^T < 0 \quad (3.5)$$

不等式(3.4)中含有非线性项, 必须转化为线性矩阵不等式才能求解。取 $k_0 = \bar{\omega}^{-1} u_{\max}^2$, 则不等式(3.4)变为 $K^T K \leq k_0 P$ 。根据 Schur 补定理, 式(3.4)变换为

$$\begin{bmatrix} k_0 P & K^T \\ K & 1 \end{bmatrix} \geq 0$$

左右同乘以 $\begin{bmatrix} P^{-1} & 0 \\ 0 & 1 \end{bmatrix}$, 可得

$$\begin{bmatrix} k_0 P^{-1} & P^{-1} K^T \\ KP^{-1} & 1 \end{bmatrix} \geq 0 \quad (3.6)$$

考虑式(3.5)和式(3.6), 令 $F = KP^{-1}$ 和 $N = P^{-1}$, 则 $P^{-1} K^T = F^T$, 由式(3.5)和式(3.6)可得第1个和第2个 LMI 为

$$\alpha N + AN + BF + NA^T + F^T B^T < 0 \quad (3.7)$$

$$\begin{bmatrix} k_0 N & F^T \\ F & 1 \end{bmatrix} \geq 0 \quad (3.8)$$

根据 P 的定义, 可设计第3个 LMI 为

$$P > 0, \quad P = P^T \quad (3.9)$$

要满足 $\mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \leq \bar{\omega}$, 根据 Schur 补定理, 可设计第 4 个 LMI 为

$$\begin{bmatrix} \bar{\omega} & \mathbf{x}_0^T \\ \mathbf{x}_0 & \mathbf{N} \end{bmatrix} \geq 0 \quad (3.10)$$

通过上面 4 个 LMI, 即式(3.7)至式(3.10), 通过设计合适的 $u_{\max}$ 和 α 值, 可求得有效的 $\mathbf{K}$ 。

3.1.4 仿真实例

实际模型为

$$\ddot{\theta} = -25\dot{\theta} + 133u(t)$$

考虑模型式(3.1), 取 $x_1 = \theta, x_2 = \dot{\theta}$, 则对应于式 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, 有 $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{J} \end{bmatrix}$,

$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}$ 。取 $J = \frac{1}{133}, b = \frac{25}{133}$, 初始状态值为 $\mathbf{x}(0) = [1 \ 0]$ 。

取 $\bar{\omega} = 0.10, \alpha = 10, u_{\max} = 1.0$, 采用 LMI 程序 chap3_1LMI.m, 求解 LMI 式(3.7)至式(3.10), MATLAB 运行后显示有可行解, 解为 $\mathbf{K} = [-0.9528 \ -0.0381]$ 。控制律采用式(3.2), 将求得的 $\mathbf{K}$ 代入控制器程序 chap3_1ctrl.m, 仿真结果如图 3.1 和图 3.2 所示。为了保证有可行解, 可取较大的 $u_{\max}$ 值, 并取 α 为较小的值。

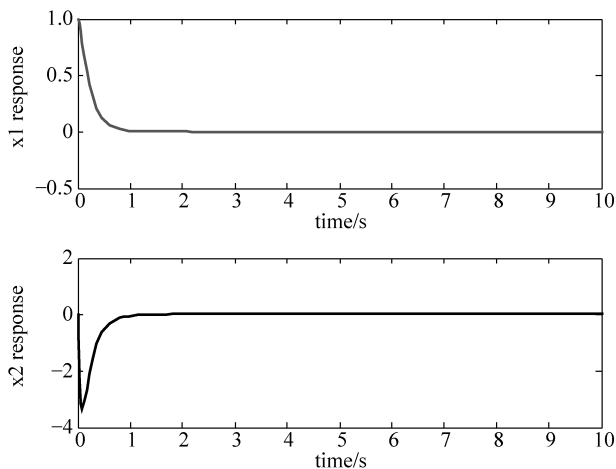


图 3.1 状态响应

仿真程序:

(1) LMI 不等式求 $\mathbf{K}$ 程序: chap3_1LMI.m

```
clear all;
close all;
```

```
J = 1/133; b = 25/133;
```

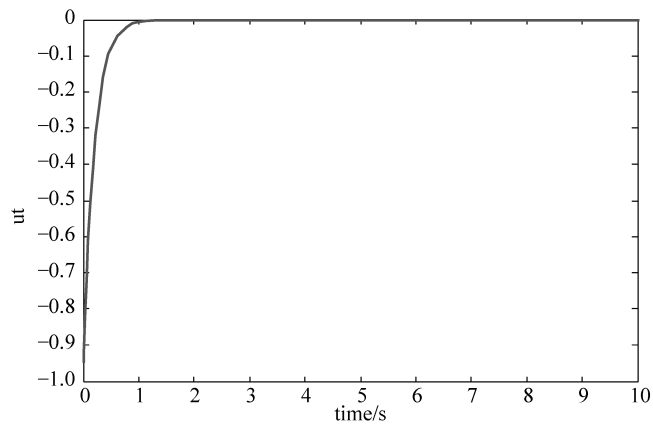


图 3.2 控制输入信号

```

A = [0 1; 0 -b/J];
B = [0 1/J]';

F = sdpvar(1,2);
P = sdpvar(2,2,'symmetric');
N = sdpvar(2,2,'symmetric');

umax = 1.0;
alfa = 10; w_bar = 0.10;
x0 = [1 0]';

%First LMI
L1 = set((alfa * N + A * N + B * F + N * A' + F' * B') < 0);

%Second LMI
k0 = umax^2/w_bar;
M1 = [k0 * N F'; F 1];
L2 = set(M1 > 0);

%Third LMI
L3 = set(N > 0);

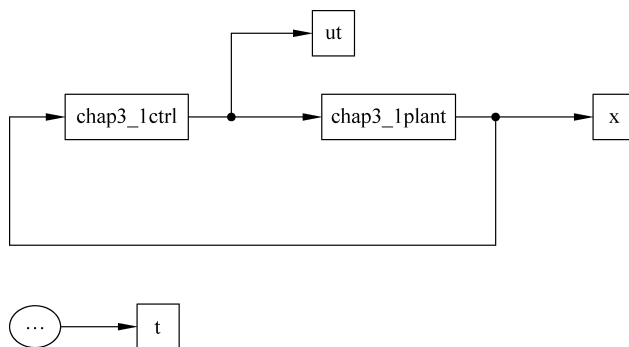
%Fourth LMI
M2 = [w_bar x0'; x0 N];
L4 = set(M2 > 0);

L = L1 + L2 + L3 + L4;
solvesdp(L);

F = double(F);
N = double(N);

P = inv(N)
K = F * P
    
```

(2) Simulink 主程序: chap3_1sim.mdl



(3) 被控对象 S 函数: chap3_1plant.m

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
    sizes = simsizes;
    sizes.NumContStates = 2;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 2;
    sizes.NumInputs = 1;
    sizes.DirFeedthrough = 0;
    sizes.NumSampleTimes = 0;
    sys = simsizes(sizes);
    x0 = [1 0];
    str = [];
    ts = [];
    function sys = mdlDerivatives(t,x,u)
        A = [0 1;0 -25];
        B = [0 133]';

        ut = u(1);

        dx = A * x + B * ut;

        sys(1) = dx(1);
        sys(2) = dx(2);
    function sys = mdlOutputs(t,x,u)
```

```
sys(1) = x(1);
sys(2) = x(2);
```

(4) 控制器 S 函数: chap3_lctrl.m

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0 0];
function sys = mdlOutputs(t,x,u)
x1 = u(1);
x2 = u(2);
X = [x1 x2]';

K = [ -0.9528 -0.0381];

ut = K * X;
sys(1) = ut;
```

(5) 作图程序: chap3_1plot.m

```
close all;

figure(1);
subplot(211);
plot(t,x(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('x1 response');
subplot(212);
plot(t,x(:,2),'b','linewidth',2);
xlabel('time(s)');ylabel('x2 response');

figure(2);
plot(t,ut(:,1),'r','linewidth',2);
```

```
xlabel('time(s)');ylabel('ut');
```

3.2 控制输入受限下位置跟踪 LMI 控制算法

3.2.1 系统描述

考虑电机-负载模型如下:

$$J\ddot{\theta} = -b\dot{\theta} + u(t) \quad (3.11)$$

其中, θ 为角度, J 为转动惯量, b 为黏性系数, u 为控制输入, $|u(t)| \leq u_{\max}$ 。

式(3.11)可写为

$$\ddot{\theta} = -\frac{b}{J}\dot{\theta} + \frac{1}{J}u(t)$$

取角度指令为 θ_d , 则角度跟踪误差为 $x_1 = \theta - \theta_d$, 角速度跟踪误差为 $x_2 = \dot{\theta} - \dot{\theta}_d$, 则控制目标在控制输入受限条件下, 实现角度和角速度跟踪, 即 $t \rightarrow \infty$ 时, $x_1 \rightarrow 0, x_2 \rightarrow 0, |u(t)| \leq u_{\max}$ 。

由于

$$\begin{aligned} \dot{x}_2 &= -\frac{b}{J}\dot{\theta} + \frac{1}{J}u - \ddot{\theta}_d = -\frac{b}{J}(x_2 + \dot{\theta}_d) + \frac{1}{J}u - \ddot{\theta}_d \\ &= -\frac{b}{J}x_2 + \frac{1}{J}u - \ddot{\theta}_d - \frac{b}{J}\dot{\theta}_d \\ &= -\frac{b}{J}x_2 + \frac{1}{J}(u - J\ddot{\theta}_d - b\dot{\theta}_d) \end{aligned}$$

取 $\tau = u - J\ddot{\theta}_d - b\dot{\theta}_d$, 即

$$u = \tau + J\ddot{\theta}_d + b\dot{\theta}_d \quad (3.12)$$

则 $|u(t)| \leq u_{\max}$ 转化为 $|\tau + J\ddot{\theta}_d + b\dot{\theta}_d| \leq u_{\max}$, 即 $-u_{\max} \leq \tau + J\ddot{\theta}_d + b\dot{\theta}_d \leq u_{\max}$, 从而

$$\tau \leq u_{\max} - J\ddot{\theta}_d - b\dot{\theta}_d \leq u_{\max} + J|\ddot{\theta}_d|_{\max} + b|\dot{\theta}_d|_{\max}$$

即

$$\tau_{\max} = u_{\max} + J|\ddot{\theta}_d|_{\max} + b|\dot{\theta}_d|_{\max}$$

从而由 $u_{\max}$ 可得到 $\tau_{\max}$ 。

由 $\dot{x}_2 = -\frac{b}{J}x_2 + \frac{1}{J}\tau$, 可得

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{b}{J}x_2 + \frac{1}{J}\tau \end{aligned}$$

则误差状态方程为

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\tau \quad (3.13)$$

其中, $\mathbf{x} = [x_1 \quad x_2]^T$, $\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{b}{J} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}$ 。

控制器设计为

$$\tau = \mathbf{K}\mathbf{x} \quad (3.14)$$

其中, $\mathbf{K} = [k_1 \quad k_2]$ 。

控制目标转化为通过设计 LMI, 实现 $\mathbf{x} \rightarrow 0$, $|\tau| \leq \tau_{\max}$ 。

针对模型式(3.13)进行控制器的设计、收敛性分析及 LMI 的设计, 与 3.1 节相同。

3.2.2 仿真实例

实际模型为

$$\ddot{\theta} = -25\dot{\theta} + 133u(t)$$

被控对象角度和角速度的初始值为 $[1.0 \quad 0]^T$, 取角度指令为 $\theta_d = \sin t$, 则 $\dot{\theta}_d = \cos t$, 角度跟踪误差为 $x_1(0) = \theta(0) - \theta_d(0) = 1.0$, 角速度跟踪误差为 $x_2(0) = \dot{\theta}(0) - \dot{\theta}_d(0) = -1.0$, $\mathbf{x}(0) = [1 \quad -1]^T$ 。

取 $\bar{\omega} = 0.10$, $\alpha = 10$, $u_{\max} = 1.0$, 采用 LMI 程序 chap3_2LMI.m, 求解 LMI 式(3.7)至式(3.10), MATLAB 运行后显示有可行解, 解为 $\mathbf{K} = [-0.9870 \quad -0.0293]$ 。控制律采用式(3.12), 将求得的 $\mathbf{K}$ 代入控制器程序 chap3_2ctrl.m, 仿真结果如图 3.3 和图 3.4 所示。为了保证有可行解, 可取 $u_{\max}$ 为较大的值, 并取 α 为较小的值。

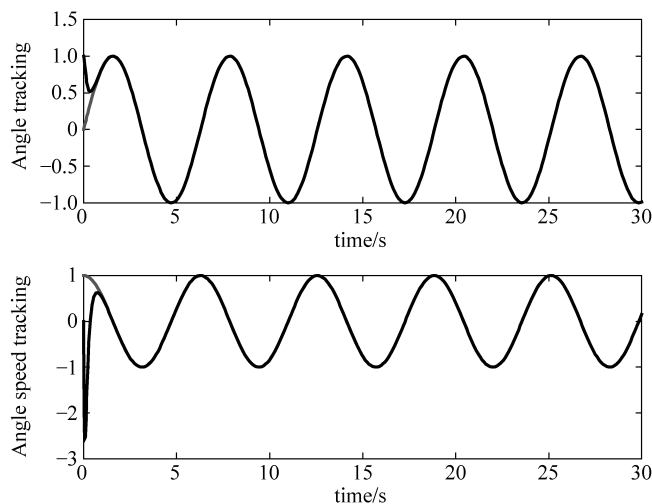


图 3.3 角度和角速度跟踪

仿真程序:

(1) LMI 不等式求 $\mathbf{K}$ 程序: chap3_2LMI.m

```
clear all;
```

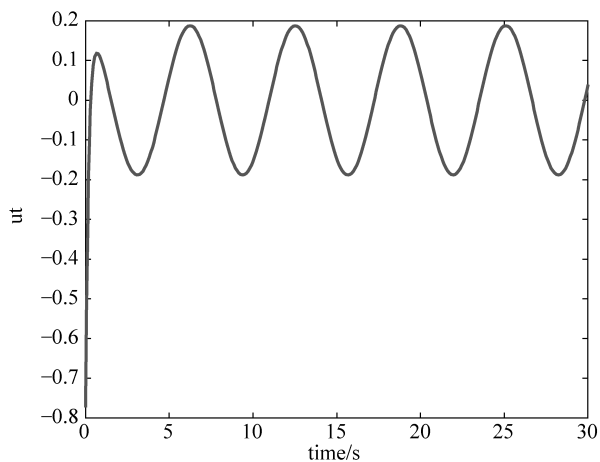


图 3.4 控制输入信号

```

close all;

J = 1/133;
b = 25/133;

A = [0 0; 0 -b/J];
B = [0 1/J]';

K = sdpvar(1,2);

M = sdpvar(3,3);
F = sdpvar(1,2);

P = sdpvar(2,2,'symmetric');
N = sdpvar(2,2,'symmetric');

umax = 1.0;
tol_max = umax + J + b;
alfa = 10; w_bar = 1.0;

% First LMI
x0 = [1 -1]';
L1 = set((alfa * N + A * N + B * F + N * A' + F' * B') < 0);

% Second LMI
k0 = tol_max^2 / w_bar;
M = [k0 * N F'; F 1];
L2 = set(M > 0);

% Third LMI
L3 = set(N > 0);

```

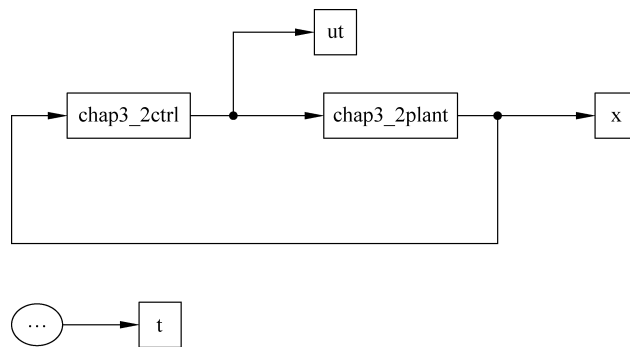
```
% Fourth LMI
M1 = [w_bar x0'; x0 N];
L4 = set(M1 > 0);
```

```
L = L1 + L2 + L3 + L4;
solvesdp(L);
```

```
F = double(F);
N = double(N);
```

```
P = inv(N)
K = F * P
```

(2) Simulink 主程序: chap3_2sim.mdl



(3) 被控对象 S 函数: chap3_2plant.m

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 2;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 2;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
```

```

sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0 = [1 0];
str = [];
ts = [];
function sys = mdlDerivatives(t,x,u)
A = [0 1;0 -25];
B = [0 133]';

```

```

ut = u(1);

```

```

dx = A * x + B * ut;

```

```

sys(1) = dx(1);
sys(2) = dx(2);
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);
sys(2) = x(2);

```

(4) 控制器 S 函数: chap3_2ctrl.m

```

function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0 0];
function sys = mdlOutputs(t,x,u)
x1 = u(1);
x2 = u(2);

xd = sin(t);
dxd = cos(t);

```

```

ddxd = - sin(t);
e = x1 - xd;
de = x2 - dxd;

K = [ - 0.9870  - 0.0293];

tol = K * [e de]';

ut = tol + 1/133 * dxd + 25/133 * dxd;

sys(1) = ut;

(5) 作图程序: chap3_2plot.m

close all;

figure(1);
subplot(211);
plot(t, sin(t), 'r', t, x(:,1), 'b', 'linewidth', 2);
xlabel('time(s)'); ylabel('Angle tracking');
subplot(212);
plot(t, cos(t), 'r', t, x(:,2), 'b', 'linewidth', 2);
xlabel('time(s)'); ylabel('Angle speed tracking');

figure(2);
plot(t, ut(:,1), 'r', 'linewidth', 2);
xlabel('time(s)'); ylabel('ut');

```

3.3 控制输入受限下带扰动控制系统 LMI 控制算法设计

3.3.1 系统描述

考虑状态方程

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(u + d) \quad (3.15)$$

其中, $\mathbf{x} = [x_1 \ x_2]^T$, u 为控制输入, $|u| \leq u_{\max}$, $d(t)$ 为加在控制输入上的积分有界扰动。

控制器设计为

$$u = \mathbf{K}\mathbf{x} \quad (3.16)$$

其中, $\mathbf{K} = [k_1 \ k_2]$ 。

控制目标为在满足 $|u| \leq u_{\max}$ 条件下, 通过设计 LMI 求解 $\mathbf{K}$, 实现 $\mathbf{x} \rightarrow 0$ 。

3.3.2 基于 H_∞ 指标控制器的设计与分析

设计 Lyapunov 函数如下:

$$V = \mathbf{x}^T \mathbf{P} \mathbf{x}$$

其中, $P > 0, P = P^T$ 。

通过 P 的设计, 可有效地调节 x 的收敛效果, 并有利于 LMI 的求解。则

$$\begin{aligned}\dot{V} &= \dot{x}^T P x + x^T P \dot{x} \\ &= (Ax + Bu + Bd)^T P x + x^T P (Ax + Bu + Bd) \\ &= (Ax + BKx + Bd)^T P x + x^T P (Ax + BKx + Bd) \\ &= x^T (A + BK)^T P x + x^T P (A + BK) x + (Bd)^T P x + x^T P (Bd) \\ &= x^T Q_1^T x + x^T Q_1 x + (B^T P x + x^T P B) d\end{aligned}$$

其中, $Q_1 = P(A + BK), Q = Q_1^T + Q_1$ 。

令 $\eta = [x^T \quad d]^T$, 则

$$\eta = \begin{bmatrix} x_1 \\ x_2 \\ d \end{bmatrix} = \begin{bmatrix} x \\ d \end{bmatrix}, \quad \eta^T = [x^T \quad d] = [x_1 \quad x_2 \quad d]$$

从而

$$\dot{V} = x^T Q x + \eta^T \begin{bmatrix} 0 & PB \\ B^T P & 0 \end{bmatrix} \eta = \eta^T \begin{bmatrix} Q & PB \\ B^T P & 0 \end{bmatrix} \eta$$

其中,

$$\begin{aligned}x^T Q x &= [x^T \quad d] \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} = \eta^T \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \eta \\ (B^T P x + x^T P B) d &= [dB^T P \quad x^T P B] \begin{bmatrix} x \\ d \end{bmatrix} \\ &= [x^T \quad d] \begin{bmatrix} 0 & PB \\ B^T P & 0 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} = \eta^T \begin{bmatrix} 0 & PB \\ B^T P & 0 \end{bmatrix} \eta\end{aligned}$$

输出为 $Z = Cx$, H_∞ 指标取

$$\int_0^t Z^T Z dt < \int_0^t \gamma^2 d^2(t) dt + V(0) \quad (3.17)$$

其中, $\gamma > 0, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 。

由于

$$\begin{aligned}Z^T Z - \gamma^2 d^2 &= x^T C^T C x - \gamma^2 d^2 \\ \eta^T \begin{bmatrix} C^T C & 0 \\ 0 & -\gamma^2 \end{bmatrix} \eta &= [x^T \quad d] \begin{bmatrix} C^T C & 0 \\ 0 & -\gamma^2 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} \\ &= [x^T C^T C \quad -\gamma^2 d] \begin{bmatrix} x \\ d \end{bmatrix} = x^T C^T C x - \gamma^2 d^2\end{aligned}$$

则

$$Z^T Z - \gamma^2 d^2 = \eta^T \begin{bmatrix} C^T C & 0 \\ 0 & -\gamma^2 \end{bmatrix} \eta$$

从而

$$\dot{V} + \mathbf{Z}^T \mathbf{Z} - \gamma^2 d^2 = \boldsymbol{\eta}^T \begin{bmatrix} \mathbf{Q} + \mathbf{C}^T \mathbf{C} & \mathbf{P}\mathbf{B} \\ (\mathbf{P}\mathbf{B})^T & -\gamma^2 \end{bmatrix} \boldsymbol{\eta}$$

取

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{Q} + \mathbf{C}^T \mathbf{C} & \mathbf{P}\mathbf{B} \\ (\mathbf{P}\mathbf{B})^T & -\gamma^2 \end{bmatrix} < 0 \quad (3.18)$$

则

$$\dot{V} + \mathbf{Z}^T \mathbf{Z} - \gamma^2 d^2 \leq 0$$

对上式积分,可得

$$V + \int_0^t \mathbf{Z}^T \mathbf{Z} dt \leq \int_0^t \gamma^2 d^2 dt + V(0)$$

假设 d 为递减的扰动信号,即积分有界扰动^[4],取

$$\int_0^\infty d^2 dt \leq \gamma^{-2} v_{\max}$$

由于 $\int_0^t \mathbf{Z}^T \mathbf{Z} dt \geq 0$, 则

$$V(t) \leq \bar{\omega}$$

其中, $v_{\max} + V(0) \leq \bar{\omega}$ 。

由 $V(t) \leq \bar{\omega}$ 可得

$$\mathbf{P}_{\min} \|\mathbf{x}\|^2 \leq \mathbf{x}^T \mathbf{P} \mathbf{x} \leq \bar{\omega}$$

则收敛结果为

$$\|\mathbf{x}\|^2 \leq \frac{v_{\max} + V(0)}{\mathbf{P}_{\min}}$$

3.3.3 闭环系统 LMI 的设计

由式(3.18)展开,可得

$$\begin{bmatrix} \mathbf{P}\mathbf{A} + \mathbf{P}\mathbf{B}\mathbf{K} + \mathbf{A}^T \mathbf{P} + \mathbf{K}^T \mathbf{B}^T \mathbf{P} & \mathbf{P}\mathbf{B} \\ (\mathbf{P}\mathbf{B})^T & -\gamma^2 \end{bmatrix} < 0$$

左右同乘以 $\begin{bmatrix} \mathbf{P}^{-1} & 0 \\ 0 & 1 \end{bmatrix}$, 可得

$$\begin{bmatrix} \mathbf{A}\mathbf{P}^{-1} + \mathbf{B}\mathbf{K}\mathbf{P}^{-1} + \mathbf{P}^{-1} \mathbf{A}^T + \mathbf{P}^{-1} \mathbf{K}^T \mathbf{B}^T & \mathbf{B} \\ \mathbf{B}^T & -\gamma^2 \end{bmatrix} < 0 \quad (3.19)$$

令 $\mathbf{F} = \mathbf{K}\mathbf{P}^{-1}$, $\mathbf{N} = \mathbf{P}^{-1}$, 则 $\mathbf{P}^{-1} \mathbf{K}^T = \mathbf{F}^T$, 由式(3.19), 可得第 1 个 LMI 为

$$\begin{bmatrix} \mathbf{A}\mathbf{N} + \mathbf{B}\mathbf{F} + \mathbf{N}\mathbf{A}^T + \mathbf{F}^T \mathbf{B}^T & \mathbf{B} \\ \mathbf{B}^T & -\gamma^2 \end{bmatrix} < 0 \quad (3.20)$$

根据 $\mathbf{P}$ 的定义, 可设计第 2 个 LMI 为

$$\mathbf{N} > 0, \quad \mathbf{N} = \mathbf{N}^T \quad (3.21)$$

3.3.4 控制输入受限下 LMI 的设计

由于 $V(0) = \mathbf{x}_0^T \mathbf{P} \mathbf{x}_0$, 如果存在正定对称阵 $\mathbf{P}$ 和 $\bar{\omega} > 0$, 使得 $\mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \leq \bar{\omega}$ 成立, 则可保证 $V(0) \leq \bar{\omega}$, 从而保证 $V(t) \leq \bar{\omega}$ 。

取 $\mathbf{K}^T \mathbf{K} \leq \bar{\omega}^{-1} u_{\max}^2 \mathbf{P}$, 由 $u = \mathbf{K} \mathbf{x}$ 可得

$$u^2 = (\mathbf{K} \mathbf{x})^T \mathbf{K} \mathbf{x} = \mathbf{x}^T \mathbf{K}^T \mathbf{K} \mathbf{x} \leq \mathbf{x}^T \bar{\omega}^{-1} u_{\max}^2 \mathbf{P} \mathbf{x} = \bar{\omega}^{-1} u_{\max}^2 V \leq u_{\max}^2$$

则

$$|u| \leq u_{\max}$$

通过上述分析, 构造 LMI 如下:

$$\mathbf{K}^T \mathbf{K} - \bar{\omega}^{-1} u_{\max}^2 \mathbf{P} \leq 0 \quad (3.22)$$

不等式(3.22)中含有非线性项, 必须转化为线性矩阵不等式才能求解。取 $k_0 = \bar{\omega}^{-1} u_{\max}^2$, 则不等式(3.22)变为 $\mathbf{K}^T \mathbf{K} \leq k_0 \mathbf{P}$ 。根据 Schur 补定理, 式(3.22)变换为

$$\begin{bmatrix} k_0 \mathbf{P} & \mathbf{K}^T \\ \mathbf{K} & 1 \end{bmatrix} \geq 0$$

左右同乘以 $\begin{bmatrix} \mathbf{P}^{-1} & 0 \\ 0 & 1 \end{bmatrix}$, 可得

$$\begin{bmatrix} k_0 \mathbf{P}^{-1} & \mathbf{P}^{-1} \mathbf{K}^T \\ \mathbf{K} \mathbf{P}^{-1} & 1 \end{bmatrix} \geq 0 \quad (3.23)$$

令 $\mathbf{F} = \mathbf{K} \mathbf{P}^{-1}$ 和 $\mathbf{N} = \mathbf{P}^{-1}$, 则 $\mathbf{P}^{-1} \mathbf{K}^T = \mathbf{F}^T$, 由式(3.23), 可得第3个 LMI 为

$$\begin{bmatrix} k_0 \mathbf{N} & \mathbf{F}^T \\ \mathbf{F} & 1 \end{bmatrix} \geq 0 \quad (3.24)$$

要满足 $\mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \leq \bar{\omega}$, 根据 Schur 补定理, 可将其设计为第4个 LMI 为

$$\begin{bmatrix} \bar{\omega} & \mathbf{x}_0^T \\ \mathbf{x}_0 & \mathbf{N} \end{bmatrix} \geq 0 \quad (3.25)$$

通过上面4个 LMI, 设计合适的 $u_{\max}$, 可求得有效的 $\mathbf{K}$ 。

3.3.5 仿真实例

实际模型为

$$\ddot{\theta} = -25\dot{\theta} + 133[u(t) + d(t)]$$

考虑模型式(3.1), 取 $x_1 = \theta, x_2 = \dot{\theta}$, 则对应于式 $\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B}(u + d)$, 有 $\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{b}{J} \end{bmatrix}$,

$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}$ 。取 $J = \frac{1}{133}, b = \frac{25}{133}, d = 0.1e^{-5t}$, 初始状态值为 $\mathbf{x}(0) = [0.1 \ 0]$ 。

取 $\gamma=3.0, \bar{\omega}=0.10, u_{\max}=0.50$, 采用 LMI 程序 chap3_3LMI.m, 求解 LMI 式(3.20)、式(3.21)、式(3.24)和式(3.25), MATLAB 运行后显示有可行解, 解为 $K = [-0.0099 \quad 0.1809]$ 。控制律采用式(3.16), 将求得的 K 代入控制器程序 chap3_3ctrl.m, 仿真结果如图 3.5 和图 3.6 所示。

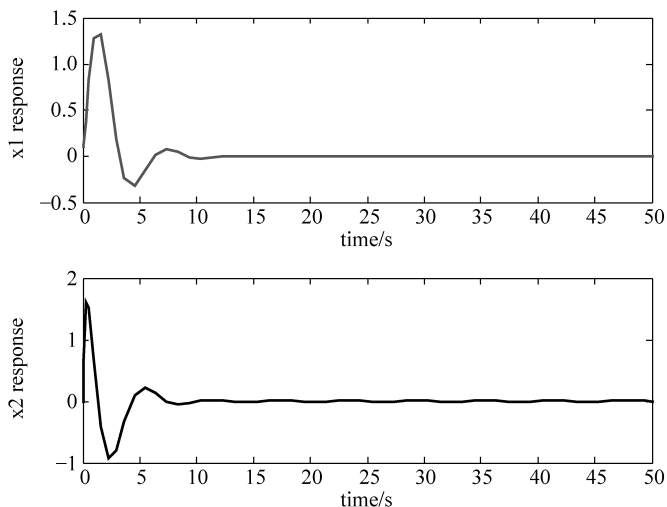


图 3.5 状态响应

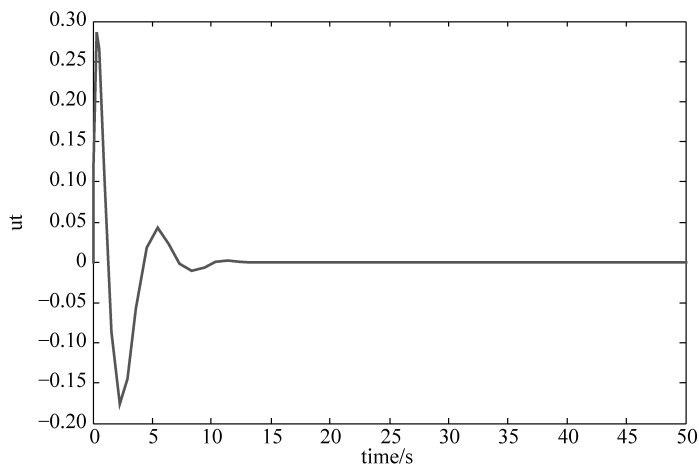


图 3.6 控制输入信号

仿真程序:

(1) LMI 不等式求 K 程序: chap3_3LMI.m

```
clear all;
close all;
```

```
J = 1/133; b = 25/133;
A = [0 1; 0 -b/J];
B = [0 1/J]';
```

```

F = sdpvar(1,2);
P = sdpvar(2,2,'symmetric');
N = sdpvar(2,2,'symmetric');

% First LMI
gama = 3.0;
M = [A * N + B * F + N * A' + F' * B' B; B' - gama^2];
L1 = set(M < 0);

% Second LMI
L2 = set(N > 0);

umax = 0.50;
w_bar = 0.10;
x0 = [1 0]';
% Third LMI
k0 = umax^2/w_bar;
M1 = [k0 * N F'; F 1];
L3 = set(M1 > 0);
% Fourth LMI
M2 = [w_bar x0'; x0 N];
L4 = set(M2 > 0);

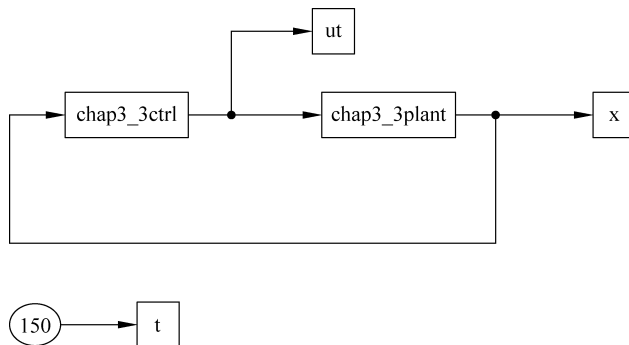
L = L1 + L2 + L3 + L4;
solvesdp(L);

F = double(F);
N = double(N);

P = inv(N);
K = F * P

```

(2) Simulink 主程序: chap3_3sim.mdl



(3) 被控对象 S 函数: chap3_3plant.m

```

function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);

```

```

case 3,
sys = mdlOutputs(t,x,u);
case {2,4,9}
sys = [];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 2;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 2;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0 = [0.1 0];
str = [];
ts = [];
function sys = mdlDerivatives(t,x,u)
A = [0 1;0 -25];
B = [0 133]';

ut = u(1);
dt = 0.10 * exp(-5 * t);

dx = A * x + B * (ut + dt);

sys(1) = dx(1);
sys(2) = dx(2);
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);
sys(2) = x(2);

```

(4) 控制器 S 函数: chap3_3ctrl.m

```

function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
[sys,x0,str,ts] = mdlInitializeSizes;
case 3,
sys = mdlOutputs(t,x,u);
case {2,4,9}
sys = [];
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;

```

```

sizes.NumOutputs    = 1;
sizes.NumInputs     = 2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0 0];
function sys = mdlOutputs(t,x,u)
x1 = u(1);
x2 = u(2);
X = [x1 x2]';

K = [-0.0099 0.1809];
ut = K * X;
sys(1) = ut;

```

(5) 作图程序: chap3_3plot.m

```

close all;

figure(1);
subplot(211);
plot(t,x(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('x1 response');
subplot(212);
plot(t,x(:,2),'b','linewidth',2);
xlabel('time(s)');ylabel('x2 response');

figure(2);
plot(t,ut(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('ut');

```

3.4 控制输入受限下带扰动控制系统 LMI 跟踪控制算法设计

3.4.1 系统描述

考虑电机-负载模型

$$J\ddot{\theta} = -b\dot{\theta} + u(t) + d(t) \quad (3.26)$$

其中, θ 为角度, J 为转动惯量, b 为黏性系数, u 为控制输入, $d(t)$ 为加在控制输入上的积分有界扰动。

式(3.26)可写为

$$\ddot{\theta} = -\frac{b}{J}\dot{\theta} + \frac{1}{J}[u(t) + d(t)]$$

取角度指令为 θ_d , 则角度跟踪误差为 $x_1 = \theta - \theta_d$, 角速度跟踪误差为 $x_2 = \dot{\theta} - \dot{\theta}_d$, 则控制

目标为角度和角速度的跟踪,即 $t \rightarrow \infty$ 时, $x_1 \rightarrow 0, x_2 \rightarrow 0$ 。

由于

$$\begin{aligned}\dot{x}_2 &= -\frac{b}{J}\dot{\theta} + \frac{1}{J}(u+d) - \ddot{\theta}_d = -\frac{b}{J}(x_2 + \dot{\theta}_d) + \frac{1}{J}(u+d) - \ddot{\theta}_d \\ &= -\frac{b}{J}x_2 + \frac{1}{J}(u+d) - \ddot{\theta}_d - \frac{b}{J}\dot{\theta}_d \\ &= -\frac{b}{J}x_2 + \frac{1}{J}(u+d - J\ddot{\theta}_d - b\dot{\theta}_d)\end{aligned}$$

取 $\tau = u - J\ddot{\theta}_d - b\dot{\theta}_d$, 即 $u = \tau + J\ddot{\theta}_d + b\dot{\theta}_d$ 。由 $\dot{x}_2 = -\frac{b}{J}x_2 + \frac{1}{J}(\tau + d)$, 可得

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{b}{J}x_2 + \frac{1}{J}(\tau + d)\end{aligned}$$

则误差状态方程为

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(\tau + d) \quad (3.27)$$

$$\text{其中, } \mathbf{x} = [x_1 \quad x_2]^T, \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{J} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}。$$

控制器设计为

$$\tau = \mathbf{K}\mathbf{x} \quad (3.28)$$

其中, $\mathbf{K} = [k_1 \quad k_2]$ 。

控制目标转化为通过设计 LMI 求 $\mathbf{K}$, 实现 $t \rightarrow \infty$ 时, $x \rightarrow 0$ 。

针对模型式(3.27)进行控制器的设计、收敛性分析及 LMI 的设计,与 3.3 节“带扰动的控制系统 LMI 控制算法设计”相同。

3.4.2 仿真实例

实际模型为

$$\ddot{\theta} = -25\dot{\theta} + 133[u(t) + d(t)]$$

取 $x_1 = \theta, x_2 = \dot{\theta}$, 则可得式 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(\tau + d)$, 取 $d = e^{-5t}$, 初始状态值为 $[1.0 \quad 0]^T$ 。

取角度指令为 $\theta_d = \sin t$, 则 $\dot{\theta}_d = \cos t$, 角度跟踪误差为 $x_1(0) = \theta(0) - \theta_d(0) = 1.0$, 角速度跟踪误差为 $x_2(0) = \dot{\theta}(0) - \dot{\theta}_d(0) = -1.0, \mathbf{x}(0) = [1 \quad -1]^T$ 。

取 $\gamma = 3.0$, 采用 LMI 程序 chap3_4LMI.m, 求解 LMI 式, 求解 LMI 式(3.20)、式(3.21)、式(3.24)和式(3.25), MATLAB 运行后显示有可行解, 解为 $\mathbf{K} = [-0.0099 \quad 0.1809]$ 。控制律采用式(3.28)和 $u = \tau + J\ddot{\theta}_d + b\dot{\theta}_d$, 将求得的 $\mathbf{K}$ 代入控制器程序 chap3_4ctrl.m 中, 仿真结果如图 3.7 和图 3.8 所示。

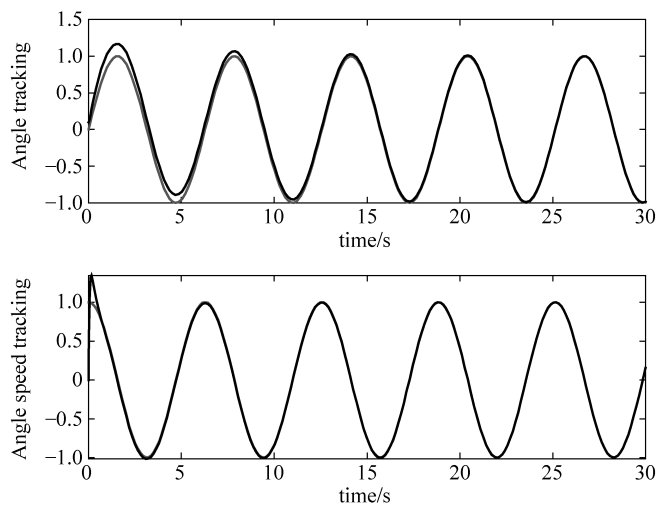


图 3.7 角度及角速度跟踪

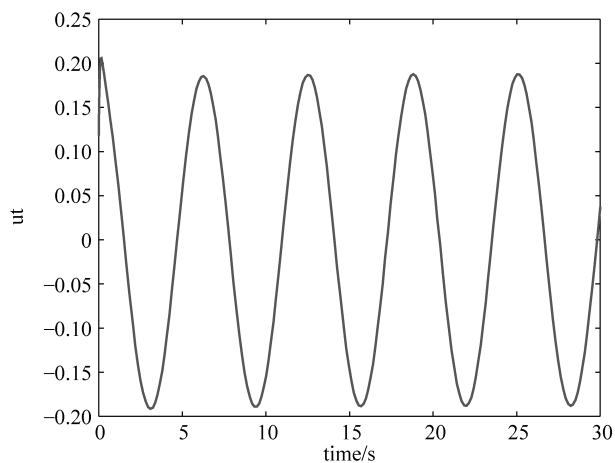


图 3.8 控制输入信号

仿真程序：

(1) LMI 不等式求 K 程序：chap3_4LMI.m

```
clear all;
close all;

J = 1/133; b = 25/133;
A = [0 1; 0 -b/J];
B = [0 1/J]';

F = sdpvar(1,2);
P = sdpvar(2,2,'symmetric');
N = sdpvar(2,2,'symmetric');

% First LMI
gamma = 3.0;
```

```

M = [A * N + B * F + N * A' + F' * B' B; B' - gama^2];
L1 = set(M < 0);

% Second LMI
L2 = set(N > 0);

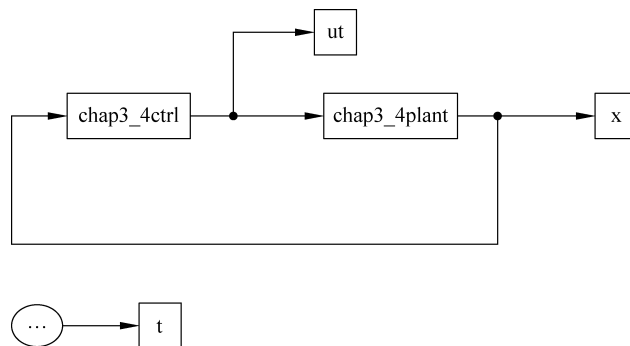
umax = 0.50;
w_bar = 0.10;
x0 = [1 0]';
% Third LMI
k0 = umax^2/w_bar;
M1 = [k0 * N F'; F 1];
L3 = set(M1 > 0);
% Fourth LMI
M2 = [w_bar x0'; x0 N];
L4 = set(M2 > 0);

L = L1 + L2 + L3 + L4;
solvesdp(L);

F = double(F);
N = double(N);

P = inv(N);
K = F * P
    
```

(2) Simulink 主程序: chap3_4sim.mdl



(3) 被控对象 S 函数: chap3_4plant.m

```

function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error('Invalid flag in spacemodel');
    end
    
```

```

        error(['Unhandled flag = ',num2str(flag)]);
    end
    function [sys,x0,str,ts] = mdlInitializeSizes
    sizes = simsizes;
    sizes.NumContStates = 2;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 2;
    sizes.NumInputs = 1;
    sizes.DirFeedthrough = 0;
    sizes.NumSampleTimes = 0;
    sys = simsizes(sizes);
    x0 = [0.1 0];
    str = [];
    ts = [];
    function sys = mdlDerivatives(t,x,u)
    A = [0 1;0 -25];
    B = [0 133]';

    ut = u(1);
    dt = 0.10 * exp(-5 * t);

    dx = A * x + B * (ut + dt);

    sys(1) = dx(1);
    sys(2) = dx(2);
    function sys = mdlOutputs(t,x,u)
    sys(1) = x(1);
    sys(2) = x(2);

```

(4) 控制器 S 函数: chap3_4ctrl.m

```

function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];

```

```

str = [];
ts = [0 0];
function sys = mdlOutputs(t,x,u)
x1 = u(1);
x2 = u(2);
X = [x1 x2]';

xd = sin(t);dxd = cos(t);
ddxd = -sin(t);

e = x1 - xd;de = x2 - dxd;
X = [e de]';

% K = [ -0.0099 0.1809];
% K = [ -0.0305 0.1277];
K = [ -0.0168 0.0667];

tol = K * X;
ut = tol + 1/133 * ddxd + 25/133 * dxd;

sys(1) = ut;

```

(5) 作图程序: chap3_4plot.m

```

close all;

figure(1);
subplot(211);
plot(t,sin(t),'r',t,x(:,1),'b','linewidth',2);
xlabel('time(s)');ylabel('Angle tracking');
subplot(212);
plot(t,cos(t),'r',t,x(:,2),'b','linewidth',2);
xlabel('time(s)');ylabel('Angle speed tracking');

figure(2);
plot(t,ut(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('ut');

```

参考文献

- [1] Grimm G, Hatfield J, Postlethwaite I, et al. Antiwindup for stable linear systems with input saturation: An LMI-based synthesis[J]. IEEE Transactions on Automatic Control, 2003, 48(9): 1509-1525.
- [2] Hu T S, Teel A, Zaccarian L. Stability and performance for saturated systems via quadratic and non-quadratic lyapunov functions[J]. IEEE Transactions on Automatic Control, 2006, 51(3): 1770-1786.
- [3] Wu H N, Zhu H Y, Wang J W. H_{∞} Fuzzy control for a class of nonlinear coupled ODE-PDE systems with input constraint[J]. IEEE Transactions on Fuzzy Systems, 2015, 23(3): 393-604.
- [4] Fang H, Lin Z, Hu T. Analysis of linear systems in the presence of actuator saturation and L2-disturbances[J]. Automatica, 2004, 40(7): 1229-1238.