第5章 基于干扰及输出测量延迟观测器的滑模控制

5.1 基于慢时变干扰观测器的连续滑模控制

5.1.1 系统描述

考虑带有慢干扰的二阶系统：

$$\ddot{\theta} = -b\dot{\theta} + au - d \tag{5.1}$$

其中，$a>0$，$b>0$，a 和 b 为已知值；d 为慢干扰时变信号。

5.1.2 观测器设计

针对二阶系统式(5.1)，设计观测器为[1]

$$\dot{\hat{d}} = k_1(\hat{\omega} - \dot{\theta}) \tag{5.2}$$

$$\dot{\hat{\omega}} = -\hat{d} + au - k_2(\hat{\omega} - \dot{\theta}) - b\dot{\theta} \tag{5.3}$$

其中，$\hat{d}$ 为对 d 项的估计，$\hat{\omega}$ 为对 $\dot{\theta}$ 的估计，$k_1>0$，$k_2>0$。

由于 $d=-\ddot{\theta}-b\dot{\theta}+au$，$\hat{d}=-\dot{\hat{\omega}}-b\dot{\theta}+au-k_2(\hat{\omega}-\dot{\theta})$，则 $\tilde{d}=(\dot{\hat{\omega}}+\ddot{\theta})+k_2(\hat{\omega}-\dot{\theta})=-\dot{\tilde{\omega}}-k_2\tilde{\omega}$。

稳定性分析如下：

定义 Lyapunov 函数为

$$V_1 = \frac{1}{2k_1}\tilde{d}^2 + \frac{1}{2}\tilde{\omega}^2 \tag{5.4}$$

其中，$\tilde{d}=d-\hat{d}$，$\tilde{\omega}=\dot{\theta}-\hat{\omega}$。

于是

$$\dot{V}_1 = \frac{1}{k_1}\tilde{d}\dot{\tilde{d}} + \tilde{\omega}\dot{\tilde{\omega}} = \frac{1}{k_1}\tilde{d}(\dot{d} - \dot{\hat{d}}) + \tilde{\omega}(\ddot{\theta} - \dot{\hat{\omega}}) \tag{5.5}$$

假设干扰 d 为慢时变信号，$\dot{d}$ 很小，当取 k_1 较大值时，可认为

$$\frac{1}{k_1}\dot{d} = 0 \tag{5.6}$$

将式(5.2)、式(5.3)和式(5.6)代入式(5.5)，得

$$
\begin{aligned}
\dot V_1 &= \frac{1}{k_1}\tilde d\dot d - \frac{1}{k_1}\tilde d\dot{\hat d} + \tilde\omega(\ddot\theta - (-\hat d + au - k_2(\hat\omega - \dot\theta) - b\dot\theta)) \\
&= \frac{1}{k_1}\tilde d\dot d - \frac{1}{k_1}\tilde d k_1(\hat\omega - \dot\theta) + \tilde\omega(-b\dot\theta + au - d - \\
&\quad (-\hat d + au - k_2(\hat\omega - \dot\theta) - b\dot\theta)) \\
&= \frac{1}{k_1}\tilde d\dot d - \tilde d(\hat\omega - \dot\theta) + \tilde\omega(-d + \hat d + k_2(\hat\omega - \dot\theta)) \\
&= \frac{1}{k_1}\tilde d\dot d + \tilde d\tilde\omega + \tilde\omega(-\tilde d - k_2\tilde\omega) = \frac{1}{k_1}\tilde d\dot d - k_2\tilde\omega^2 = -k_2\tilde\omega^2 \leqslant 0
\end{aligned} \tag{5.7}
$$

当 $\dot V_1 \equiv 0$ 时，$\tilde\omega \equiv 0$，$\dot{\tilde\omega} \equiv 0$，$\tilde d \equiv 0$。根据 LaSalle 不变性定理，$t \to \infty$时，$\tilde d \to 0$。通过采用本观测器，对 d 项进行有效的观测，从而实现补偿。

5.1.3 仿真实例

考虑带有慢干扰的二阶系统：

$$\ddot\theta = -b\dot\theta + au - d$$

其中，$a=5$，$b=0.15$，$d=150\sin(0.1t)$。

采用观测器式(5.2)和式(5.3)，取 $k_1=500$，$k_2=200$。仿真结果如图 5.1 所示。

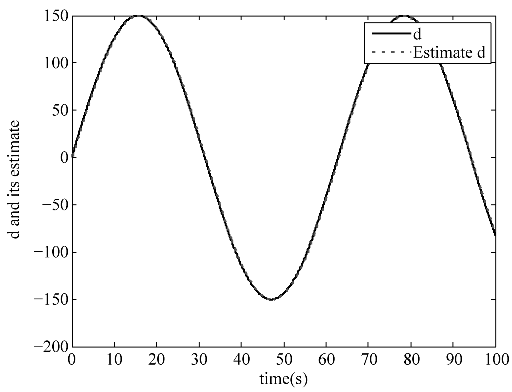
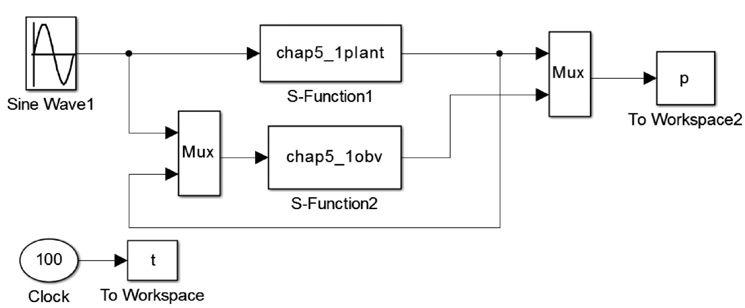


图 5.1 干扰及观测结果

仿真程序：

(1) Simulink 主程序：chap5_1sim.mdl



(2) 观测器 S 函数：chap5_1obv.m

```
function [sys,x0,str,ts] = s_function(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      = 4;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0  = [0;0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
ut = u(1);
dth = u(3);

k1 = 1000;
k2 = 200;

a = 5;b = 0.15;
sys(1) = k1 * (x(2) - dth);
sys(2) = - x(1) + a * ut - k2 * (x(2) - dth) - b * dth;
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);          % 估算
sys(2) = x(2);          % 速度估算
```

(3) 被控对象 S 函数：chap5_1plant.m

```
function [sys,x0,str,ts] = s_function(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       = 3;
sizes.NumInputs        = 1;
sizes.DirFeedthrough   = 0;
sizes.NumSampleTimes   = 0;
sys = simsizes(sizes);
x0  = [0;0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
ut = u(1);
b = 0.15;
a = 5;

d = 150 * sin(0.1 * t);
ddth = - b * x(2) + a * ut - d;

sys(1) = x(2);
sys(2) = ddth;
function sys = mdlOutputs(t,x,u)
d = 150 * sin(0.1 * t);
sys(1) = x(1);
sys(2) = x(2);
sys(3) = d;
```

(4) 作图程序：chap5_1plot.m

```
close all;

figure(1);
plot(t,p(:,3),'k',t,p(:,4),'r:','linewidth',2);
xlabel('time(s)');ylabel('d and its estimate');
legend('d','Estimate d');
```

5.1.4 基于慢时变干扰观测器的连续滑模控制

针对式(5.1)，取位置指令为 θ_d，误差为 $e=\theta_d-\theta$。滑模函数为

$$s=\dot{e}+ce$$

其中，$c>0$，则

$$\dot{s}=\ddot{e}+c\dot{e}=\ddot{\theta}_d-\ddot{\theta}+c\dot{e}=\ddot{\theta}_d+b\dot{\theta}-au+d+c\dot{e}$$

基于干扰补偿的滑模控制器设计为

$$u=\frac{1}{a}[\ddot{\theta}_{\rm d}+b\dot{\theta}+c\dot{e}+\hat{d}+\eta\,\mathrm{sgn}(s)] \tag{5.8}$$

定义 $\tilde{d}=d-\hat{d}$，$|\tilde{d}|\leqslant\eta$。

取 Lyapunov 函数为

$$V_2=\frac{1}{2}s^2$$

则

$$\begin{aligned}\dot{V}_2&=s\dot{s}=s(\ddot{\theta}_{\rm d}+b\dot{\theta}-au+d+c\dot{e})\\&=s(\ddot{\theta}_{\rm d}+b\dot{\theta}-(\ddot{\theta}_{\rm d}+b\dot{\theta}+c\dot{e}+\hat{d}+\eta\,\mathrm{sgn}(s))+d+c\dot{e})\\&=s(d-\hat{d}-\eta\,\mathrm{sgn}(s))\\&=\tilde{d}s-\eta\mid s\mid\leqslant 0\end{aligned} \tag{5.9}$$

取闭环系统的 Lyapunov 函数为

$$V=V_1+V_2=\frac{1}{2k_1}\tilde{d}^2+\frac{1}{2}\tilde{\omega}^2+\frac{1}{2}s^2$$

由式(5.7)和式(5.9)，可得 $\dot{V}=-k_2\tilde{\omega}^2+\tilde{d}s-\eta|s|$。

由于$|\tilde{d}|\leqslant\eta$，则存在 $\eta_1\geqslant 0$，使 $\tilde{d}s-\eta|s|=-\eta_1|s|$成立，则 $\dot{V}=-k_2\tilde{\omega}^2-\eta_1|s|$。

当 $\dot{V}\equiv 0$ 时，$s\equiv 0$。根据 LaSalle 不变性定理，$t\to\infty$时，$s\to 0$，$e\to 0$，$\dot{e}\to 0$。

为了降低抖振，采用饱和函数 sat(s)代替符号函数 sgn(s)。

5.1.5　仿真实例

考虑带有慢干扰的二阶系统：

$$\ddot{\theta}=-b\dot{\theta}+au-d$$

其中，$a=5$，$b=0.15$，$d=150\sin(0.1t)$。

位置指令为 $\theta_{\rm d}=\sin t$，控制器采用式(5.8)，观测器采用式(5.2)和式(5.3)。取 $k_1=500$，$k_2=200$，取 $\eta=5.0$，$c=15$，$\Delta=0.10$。仿真结果如图 5.2～图 5.4 所示。

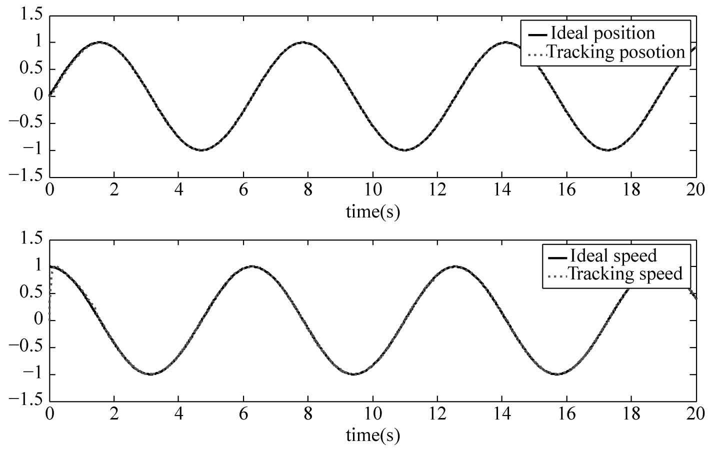


图 5.2　位置和速度跟踪

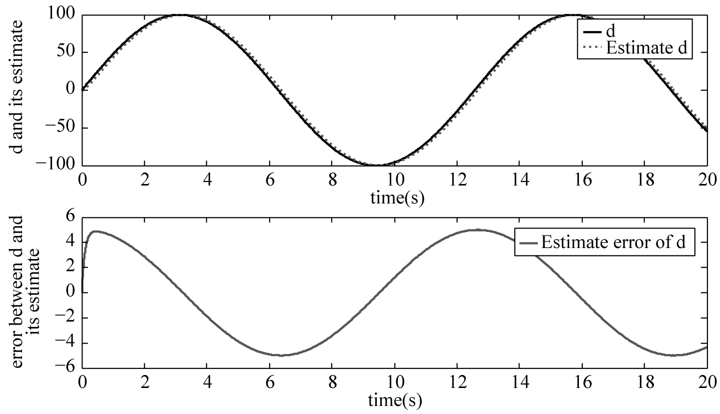


图 5.3　干扰及观测结果

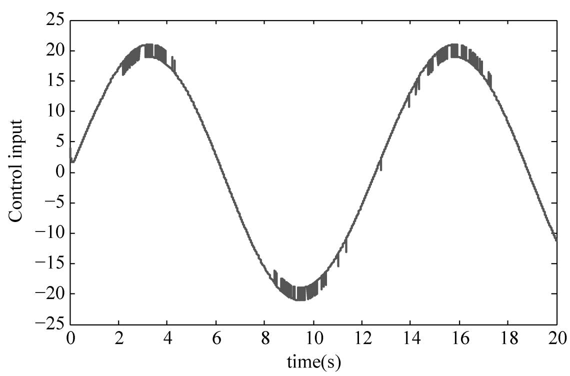
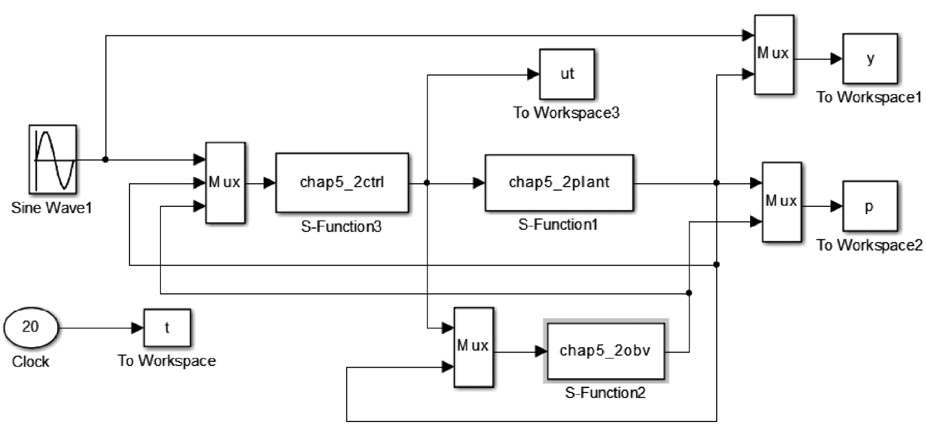


图 5.4　控制输入

仿真程序：

(1) Simulink 主程序：chap5_2sim.mdl



(2) 控制器 S 函数：chap5_2ctrl.m

```
function [sys,x0,str,ts] = s_function(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 {1,2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumDiscStates   = 0;
sizes.NumOutputs      = 1;
sizes.NumInputs       = 5;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 0;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [];
function sys = mdlOutputs(t,x,u)
thd = u(1);dthd = cos(t);ddthd = - sin(t);
th = u(2);
dth = u(3);
dp = u(5);
b = 0.15;a = 5;

e = thd - th;
de = dthd - dth;
c = 15;
s = c * e + de;

xite = 5.0;
M = 2;
if M == 1
    ut = 1/a * (ddthd + b * dth + c * de + dp + xite * sign(s));
elseif M == 2          % Saturated function
      delta = 0.10;
      kk = 1/delta;
    if abs(s)> delta
      sats = sign(s);
    else
      sats = kk * s;
    end
    ut = 1/a * (ddthd + b * dth + c * de + dp + xite * sign(s));
```

```
end
sys(1) = ut;
```

（3）观测器 S 函数：chap5_2obv.m

```
function [sys,x0,str,ts] = s_function(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      = 1;
sizes.NumInputs       = 4;
sizes.DirFeedthrough  = 0;
sizes.NumSampleTimes  = 0;
sys = simsizes(sizes);
x0  = [0;0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
ut = u(1);
dth = u(3);

k1 = 5000;
k2 = 500;

a = 5;b = 0.15;
sys(1) = k1 * (x(2) - dth);
sys(2) = - x(1) + a * ut - k2 * (x(2) - dth) - b * dth;
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);          % d 估算
```

（4）被控对象 S 函数：chap5_2plant.m

```
function [sys,x0,str,ts] = s_function(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       = 3;
sizes.NumInputs        = 1;
sizes.DirFeedthrough   = 0;
sizes.NumSampleTimes   = 0;
sys = simsizes(sizes);
x0  = [0;0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
ut = u(1);
b = 0.15;
a = 5;

d = 100 * sin(0.5 * t);
ddth = - b * x(2) + a * ut - d;

sys(1) = x(2);
sys(2) = ddth;
function sys = mdlOutputs(t,x,u)
d = 100 * sin(0.5 * t);
sys(1) = x(1);
sys(2) = x(2);
sys(3) = d;
```

（5）作图程序：chap5_2plot.m

```
close all;

figure(1);
subplot(211);
plot(t,y(:,1),'k',t,y(:,2),'r:','linewidth',2);
legend('ideal position','tracking posotion');
subplot(212);
plot(t,cos(t),'k',t,y(:,3),'r:','linewidth',2);
legend('ideal speed','tracking speed');

figure(2);
subplot(211);
plot(t,p(:,3),'k',t,p(:,4),'r:','linewidth',2);
xlabel('time(s)');ylabel('d and its estimate');
legend('d','Estimate d');
```

```
subplot(212);
plot(t,p(:,3) - p(:,4),'r','linewidth',2);
xlabel('time(s)');ylabel('error between d and its estimate');
legend('Estimate error of d');

figure(3);
plot(t,ut(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('Control input');
```

5.2 基于指数收敛干扰观测器的滑模控制

5.2.1 系统描述

考虑 SISO 系统动态方程：

$$J\ddot{\theta}+b\dot{\theta}=u+d \tag{5.10}$$

其中，J 为转动惯量；b 为阻尼系数；u 为控制输入；θ、$\dot{\theta}$ 分别代表角度、角速度；d 为外界干扰，且 $J>0$，$b>0$。

5.2.2 指数收敛干扰观测器的问题提出

由式(5.10)得

$$d=J\ddot{\theta}+b\dot{\theta}-u \tag{5.11}$$

设计观测器或估计器的基本思想就是用估计输出与实际输出的差值对估计值进行修正。因此，将干扰观测器设计为

$$\dot{\hat{d}}=K(d-\hat{d})=-K\hat{d}+Kd=-K\hat{d}+K(J\ddot{\theta}+b\dot{\theta}-u) \tag{5.12}$$

其中，$K>0$。

一般没有干扰 d 的微分的先验知识，相对于观测器的动态特性，干扰 d 的变化是缓慢的[2]，即

$$\dot{d}=0$$

令观测误差为

$$\tilde{d}=d-\hat{d}$$

则

$$\dot{\tilde{d}}=-\dot{\hat{d}}=-K(d-\hat{d})=-K\tilde{d}$$

即观测误差满足如下约束：

$$\dot{\tilde{d}}+K\tilde{d}=0 \tag{5.13}$$

则观测器是指数收敛的，且收敛速率可通过选择 K 值来确定。

在实际工程中，由于观测噪声，很难通过速度信号求微分来得到加速度信号。因此，观测器式(5.12)在实际工程中不能实现，但它为设计非线性观测器提供了基础。

5.2.3 指数收敛干扰观测器的设计

取 $\dot{\hat{d}}=K(d-\hat{d})$，定义辅助参数向量[2]为

$$z=\hat{d}-KJ\dot{\theta} \tag{5.14}$$

则

$$\dot{z}=\dot{\hat{d}}-KJ\ddot{\theta}$$

由于 $\dot{\hat{d}}=K(d-\hat{d})=K(J\ddot{\theta}+b\dot{\theta}-u)-K\hat{d}$，则

$$\dot{z}=K(J\ddot{\theta}+b\dot{\theta}-u)-K\hat{d}-KJ\ddot{\theta}=K(b\dot{\theta}-u)-K\hat{d}$$

干扰观测器设计为

$$\begin{cases}\dot{z}=K(b\dot{\theta}-u)-K\hat{d}\\ \hat{d}=z+KJ\dot{\theta}\end{cases} \tag{5.15}$$

则

$$\dot{z}=K(b\dot{\theta}-T)-K(z+KJ\dot{\theta})=K(b\dot{\theta}-T-KJv)-Kz$$

针对常值干扰或慢干扰，可假设 $\dot{d}=0$[2]，则

$$\dot{\tilde{d}}=\dot{d}-\dot{\hat{d}}=-\dot{\hat{d}}=-\dot{z}-KJ\ddot{\theta}$$

将 $\dot{z}$ 代入上式，得

$$\begin{aligned}\dot{\tilde{d}}&=-(K(b\dot{\theta}-T-KJ\dot{\theta})-Kz)-KJ\ddot{\theta}=-K(b\dot{\theta}-T-KJ\dot{\theta})+Kz-KJ\ddot{\theta}\\ &=K(z+KJ\dot{\theta})-K(J\ddot{\theta}+b\dot{\theta}-T)\\ &=K\hat{d}-K(J\ddot{\theta}+b\dot{\theta}-T)=K(\hat{d}-d)=-K\tilde{d}\end{aligned}$$

因而得到观测误差方程为

$$\dot{\tilde{d}}+K\tilde{d}=0$$

解为

$$\tilde{d}(t)=\tilde{d}(t_0)\mathrm{e}^{-Kt}$$

由于 $\tilde{d}(t_0)$ 的值是确定的，可见，观测器的收敛精度取决于参数 K 值。通过设计参数 K，使估计值 $\hat{d}$ 按指数逼近干扰 d。由观测器式(5.15)可知，该观测器不需要 $\ddot{\boldsymbol{\theta}}$ 信息。

5.2.4 滑模控制器的设计与分析

采用观测器式(5.15)观测干扰 d，在滑模控制中对干扰进行补偿，可有效地降低切换增益，从而有效地降低抖振。

取控制目标为 $\theta\rightarrow\theta_{\mathrm{d}}$，$\dot{\theta}\rightarrow\dot{\theta}_{\mathrm{d}}$。针对模型式(5.10)，设计滑模函数

$$s=ce+\dot{e} \tag{5.16}$$

其中，$c>0$，$e=\theta_{\mathrm{d}}-\theta$。

由于 $\ddot{\theta}=\frac{1}{J}(-b\dot{\theta}+u+d)$，则

$$\ddot{e}=\ddot{\theta}_{\mathrm{d}}-\ddot{\theta}=\ddot{\theta}_{\mathrm{d}}-\frac{1}{J}(-b\dot{\theta}+u+d)$$

$$\dot{s}=c\dot{e}+\ddot{e}=c\dot{e}+\ddot{\theta}_{\mathrm{d}}-\frac{1}{J}(-b\dot{\theta}+u+d)$$

$$=c\dot{e}+\ddot{\theta}_{\mathrm{d}}+\frac{b}{J}\dot{\theta}-\frac{1}{J}u-\frac{1}{J}d$$

取控制律为

$$u(t)=J\left(c\dot{e}+\ddot{\theta}_{\mathrm{d}}+\frac{b}{J}\dot{\theta}-\frac{1}{J}\hat{d}+k_0 s+\eta\mathrm{sgn}s\right) \tag{5.17}$$

其中，$k_0>0$。

则

$$\dot{s}=c\dot{e}+\ddot{\theta}_{\mathrm{d}}+\frac{b}{J}\dot{\theta}-\left(c\dot{e}+\ddot{\theta}_{\mathrm{d}}+\frac{b}{J}\dot{\theta}-\frac{1}{J}\hat{d}+k_0 s+\eta\mathrm{sgn}s\right)-\frac{1}{J}d$$

$$=\frac{1}{J}\hat{d}-k_0 s-\eta\mathrm{sgn}s-\frac{1}{J}d=-k_0 s-\eta\mathrm{sgn}s-\frac{1}{J}\tilde{d}$$

取闭环系统的 Lyapunov 函数为

$$V=\frac{1}{2}s^2+\frac{1}{2}\tilde{d}^2$$

则

$$\dot{V}=s\dot{s}+\tilde{d}\dot{\tilde{d}}=s\left(-k_0 s-\eta\mathrm{sgn}s-\frac{1}{J}\tilde{d}\right)-K\tilde{d}^2$$

$$=-k_0 s^2-\eta\mid s\mid-\frac{1}{J}\tilde{d}s-K\tilde{d}^2\leqslant 0$$

其中，$\eta\geqslant\frac{1}{J}|\tilde{d}|_{\max}$。

由于观测器初始观测误差为 $\tilde{d}(0)$，则可取 $|\tilde{d}(0)|=|\tilde{d}|_{\max}$，从而可取 $\eta\geqslant\frac{1}{J}|\tilde{d}(0)|$。

控制系统收敛性分析如下：

由于

$$\dot{V}=-k_0 s^2-\eta\mid s\mid-\frac{1}{J}\tilde{d}s-K\tilde{d}^2\leqslant -k_0 s^2-K\tilde{d}^2\leqslant -k_1\left(\frac{1}{2}s^2+\frac{1}{2}\tilde{d}^2\right)=-k_1 V$$

其中，$k_1=2\min\{k_0,K\}$。

采用引理 1.1，不等式 $\dot{V}\leqslant -k_1 V$ 的解为

$$V(t)\leqslant \mathrm{e}^{-k_1(t-t_0)}V(t_0)$$

可见，控制系统指数收敛，收敛精度取决于参数 k_1 值，$t\to\infty$ 时，$s\to 0$，$\tilde{d}\to 0$，从而 $e\to 0$，$\dot{e}\to 0$。

5.2.5　仿真实例

模型为 $\ddot{\theta}=-25\dot{\theta}+133(u+d)$，对比 $J\ddot{\theta}+b\dot{\theta}=u+d$，可知 $J=\frac{1}{133}$，$b=\frac{25}{133}$。

1. 仿真实例(1)：干扰观测器的测试

分别取 $d(t)=-5$ 和 $d(t)=0.05\sin t$。取参数 $K=50$，采用观测器式(5.15)，仿真结果如图 5.5 和图 5.6 所示。为了提高收敛精度，可在 Simulink 环境中将 ODE45 迭代法的 Relative tolerance 精度取 1e−6 或更小。

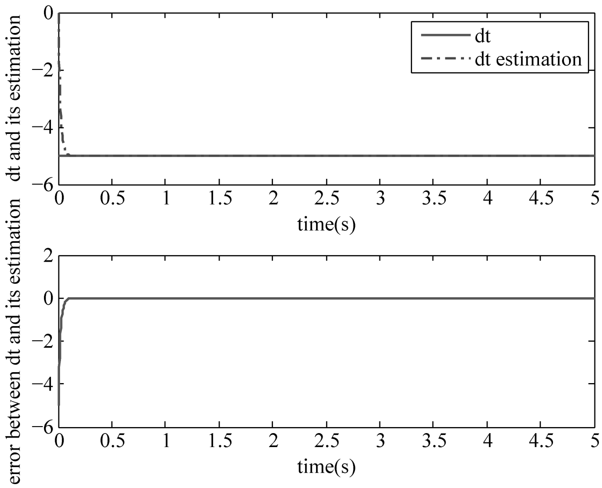


图 5.5　$d(t)=-5$ 的干扰观测结果

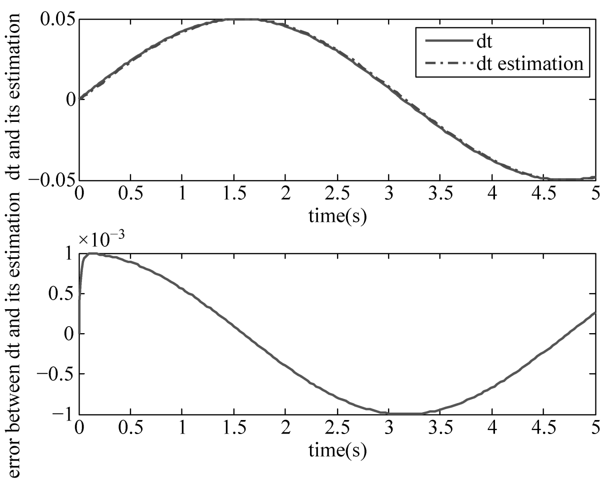


图 5.6　$d(t)=0.05\sin t$ 的干扰观测结果

2. 仿真实例(2)：闭环控制仿真实例

针对模型式(5.10)，取 $d(t)=-5$。位置指令为 $\theta_{\mathrm{d}}=\sin t$，观测器采用式(5.15)，控制器采用式(5.17)，取 $c=10$，$K=50$，根据干扰值及观测器初始值，取 $\eta=5.0$。采用饱和函数代替连续函数，取边界层厚度为 $\Delta=0.05$，仿真结果如图 5.7～图 5.9 所示。

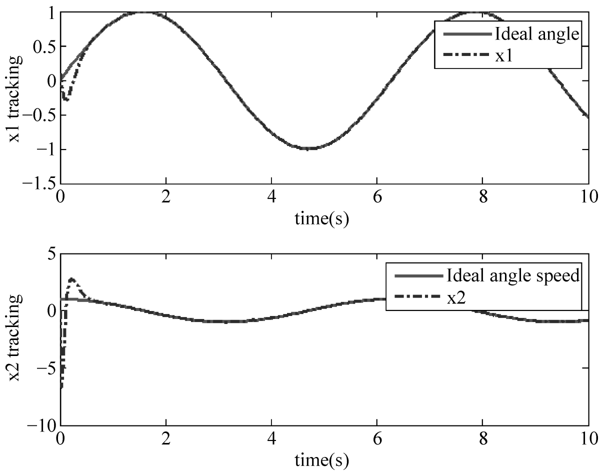


图 5.7　角度和角速度跟踪

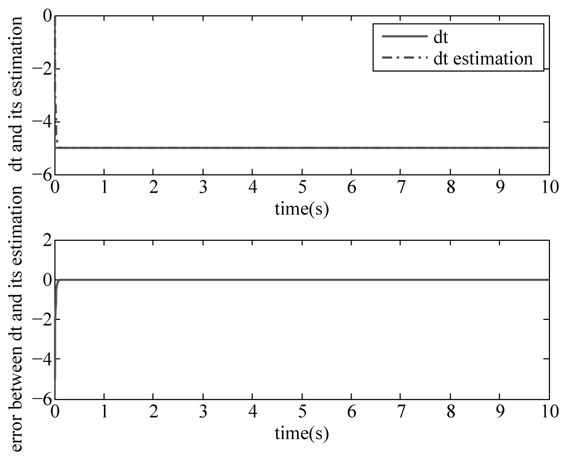


图 5.8　干扰及观测结果

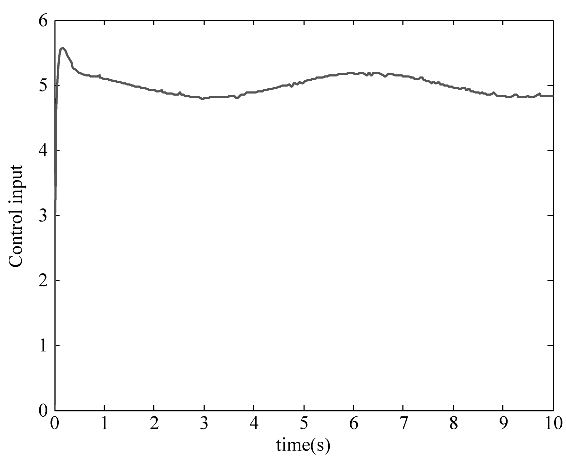
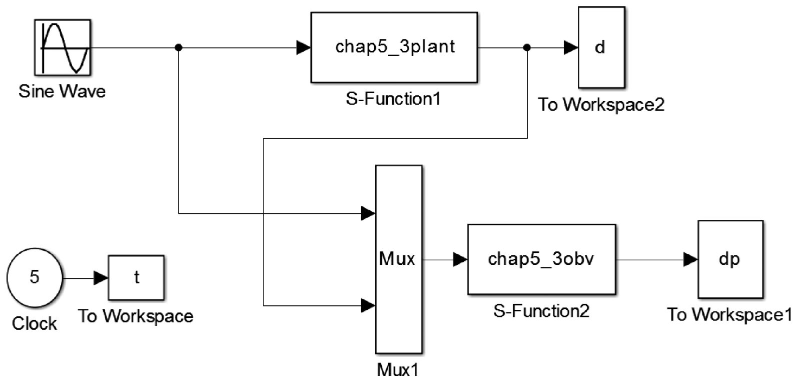


图 5.9　控制输入

仿真程序：

仿真实例(1)：观测器仿真程序。

(1) Simulink 主程序：chap5_3sim. mdl



(2) 被控对象程序：chap5_3plant. m

```
function [sys,x0,str,ts] = NDO_plant(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      = 3;
sizes.NumInputs       = 1;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 0;
sys = simsizes(sizes);
x0  = [0.1,0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
ut = u(1);
dt = -5;
%dt = 0.05*sin(t);
sys(1) = x(2);
sys(2) = -25*x(2) + 133*(ut + dt);
function sys = mdlOutputs(t,x,u)
```

```
dt = -5;
%dt = 0.05*sin(t);
sys(1) = x(1);
sys(2) = x(2);
sys(3) = dt;
```

(3) 干扰观测器程序：chap5_3obv.m

```
function [sys,x0,str,ts] = NDO(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   = 1;
sizes.NumDiscStates   = 0;
sizes.NumOutputs      = 1;
sizes.NumInputs       = 4;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 0;
sys = simsizes(sizes);
x0  = [0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
K = 50;
J = 1/133;
b = 25/133;

ut = u(1);

dth = u(3);
z = x(1);
dp = z + K*J*dth;

dz = K*(b*dth-ut) - K*dp;
sys(1) = dz;
function sys = mdlOutputs(t,x,u)
K = 50;
J = 1/133;
dth = u(3);
z = x(1);
```

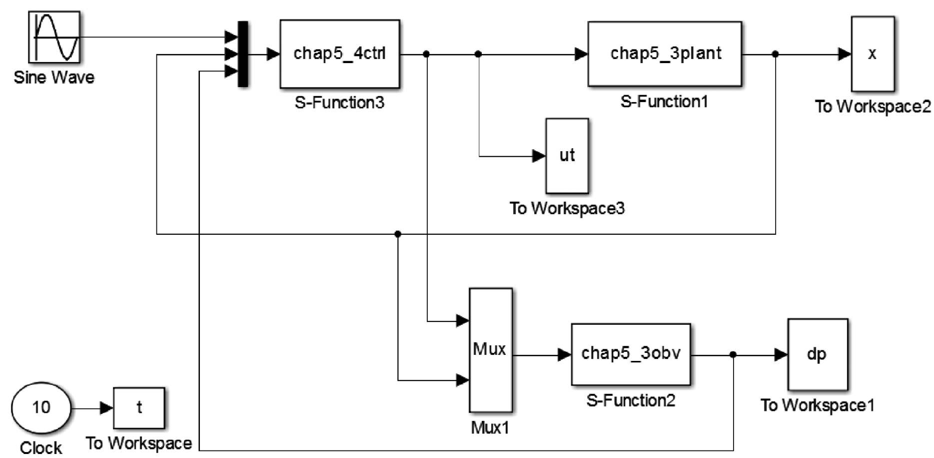
```
dp = z + K * J * dth;

sys(1) = dp;
```

(4) 作图程序：chap5_3plot.m

```
close all;

figure(1);
subplot(211);
plot(t,d(:,3),'r',t,dp(:,1),'-.b','linewidth',2);
xlabel('time(s)');ylabel('dt and its estimation');
legend('dt','dt estimation');
subplot(212);
plot(t,d(:,3)-dp(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('error between dt and its estimation');
```

仿真实例(2)：控制系统仿真程序。

(1) Simulink 主程序：chap5_4sim.mdl



(2) 被控对象程序：chap5_3plant.m(见仿真实例(1))

(3) 控制器程序：chap5_4ctrl.m

```
function [sys,x0,str,ts] = s_function(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {1,2,4,9}
    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       = 5;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [-1 0];
function sys = mdlOutputs(t,x,u)
J = 1/133;
b = 25/133;

thd = u(1);
dthd = cos(t);
ddthd = -sin(t);

th = u(2);
dth = u(3);
dp = u(5);

e = thd - th;
de = dthd - dth;

c = 10;
xite = 5.0;
s = c*e + de;

%Saturated function
delta = 0.05;
kk = 1/delta;
if abs(s)>delta
    sats = sign(s);
else
    sats = kk*s;
end

k0 = 10;
%ut = J*(c*de + ddthd + b/J*dth - 1/J*dp + k0*s + xite*sign(s));
ut = J*(c*de + ddthd + b/J*dth - 1/J*dp + k0*s + xite*sats);

sys(1) = ut;
```

(4) 干扰观测器程序：chap5_3obv.m(见仿真实例(1))

(5) 作图程序：chap5_4plot.m

```
close all;
figure(1);
subplot(211);
plot(t,sin(t),'r',t,x(:,1),'-.b','linewidth',2);
xlabel('time(s)');ylabel('x1 tracking');
```

```
legend('ideal angle','x1');
subplot(212);
plot(t,cos(t),'r',t,x(:,2),'-.b','linewidth',2);
xlabel('time(s)');ylabel('x2 tracking');
legend('ideal angle speed','x2');

figure(2);
subplot(211);
plot(t,x(:,3),'r',t,dp(:,1),'-.b','linewidth',2);
xlabel('time(s)');ylabel('dt and its estimation');
legend('dt','dt estimation');
subplot(212);
plot(t,x(:,3)-dp(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('error between dt and its estimation');

figure(3);
plot(t,ut(:,1),'r','linewidth',2);
xlabel('time(s)');ylabel('Control input');
```

5.3 基于输出延时观测器的滑模控制

在运动控制系统中，由于测量传感器的因素，会造成位置和速度信号的测量延迟，通过设计输出延时观测器，可很好地对测量信号进行校正。国内外学者在输出延时观测器方面取得了很大的进展。针对线性系统，文献[3]基于时滞微分方程设计了线性系统的有输出延时的观测器，文献[4]针对线性系统中输出延时做了进一步研究，在时变延时的情况下设计了延时观测器。

文献[5]针对非线性系统输出延时的情况，设计了一种链式观测器，这类观测器由很多个观测器串联组成，每个子观测器负责观测出指定的一小段延时信号，由最后一个子观测器观测出正确的无延时信号，其缺点是每个子观测器的参数都是不同的，给工程实践造成了不便。针对这一情况，文献[6]提出了另一种结构的链式延时观测器，其中每个子观测器都具有相同的结构和参数，便于工程实现，并且具有指数收敛的良好稳定性能。本节针对简单的线性系统，设计一种具有固定测量延迟的观测器，在此基础上设计了一种滑模控制方法。

5.3.1 系统描述

考虑对象：

$$G(s)=\frac{k}{s^2+as+b} \tag{5.18}$$

上式可表示为

$$\ddot{\theta}(t)=-a\dot{\theta}(t)-b\theta(t)+ku(t) \tag{5.19}$$

其中，$\theta(t)$为角度信号，$u(t)$为控制输入。

取 $\boldsymbol{z}=[\theta \quad \dot{\theta}]^{\mathrm{T}}$，式(5.19)可表示为

$$\dot{\boldsymbol{z}}(t)=\boldsymbol{A}\boldsymbol{z}(t)+\boldsymbol{H}u(t) \tag{5.20}$$

其中，$\boldsymbol{A}=\begin{bmatrix}0 & 1\\ -b & -a\end{bmatrix}$，$\boldsymbol{H}=[0 \quad k]^{\mathrm{T}}$。

假设输出信号有延迟，Δ 为输出的测量时间延迟，则实际输出可表示为

$$\bar{y}(t)=\theta(t-\Delta)=\boldsymbol{C}z(t-\Delta) \tag{5.21}$$

其中，$\boldsymbol{C}=[1 \quad 0]$。

观测的目标为：当 $t\to\infty$ 时，$\hat{\theta}(t)\to\theta(t)$，$\dot{\hat{\theta}}\to\dot{\theta}(t)$。

5.3.2 输出延迟观测器的设计

引理 5.1[7,8]：对于线性延迟系统

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{x}(t-\Delta) \tag{5.22}$$

其稳定性条件为

$$\sigma\boldsymbol{I}-\boldsymbol{A}-\boldsymbol{B}\mathrm{e}^{-\Delta\sigma}=0 \tag{5.23}$$

特征根的实部为负，则延迟系统式(5.22)为指数稳定。

针对本延迟系统式(5.20)，取$\hat{\boldsymbol{z}}=[\hat{\theta} \quad \hat{\dot{\theta}}]^{\mathrm{T}}$，设计如下延迟观测器：

$$\dot{\hat{\boldsymbol{z}}}(t)=\boldsymbol{A}\hat{\boldsymbol{z}}(t)+\boldsymbol{H}u(t)+\boldsymbol{K}[\bar{y}(t)-\boldsymbol{C}\hat{\boldsymbol{z}}(t-\Delta)] \tag{5.24}$$

其中，$\hat{\boldsymbol{z}}(t-\Delta)$是$\hat{\boldsymbol{z}}(t)$的延迟信号。

由式(5.20)～式(5.24)可得

$$\dot{\boldsymbol{\delta}}(t)=\boldsymbol{A}\boldsymbol{\delta}(t)-\boldsymbol{K}\boldsymbol{C}\boldsymbol{\delta}(t-\Delta) \tag{5.25}$$

其中，$\boldsymbol{\delta}(t)=\boldsymbol{z}(t)-\hat{\boldsymbol{z}}(t)$。

则根据引理 5.1，延迟观测器的稳定性条件为：选择合适的 $\boldsymbol{K}$，使式(5.25)的特征根的实部为负，则延迟系统式(5.25)为指数稳定，即 $t\to\infty$时，$\delta(t)$指数收敛于零。

根据引理 5.1，针对线性延迟系统式(5.25)，其稳定性条件方程为

$$\sigma\boldsymbol{I}-\boldsymbol{A}+\boldsymbol{K}\boldsymbol{C}\mathrm{e}^{-\Delta s}=0 \tag{5.26}$$

的特征根 σ 在负半面。

仿真中首先根据经验给出 $\boldsymbol{K}$ 值，然后采用 MATLAB 函数“fsolve”来解方程式(5.26)中的根 σ，使其在负半面，从而验证 $\boldsymbol{K}$。

为了简便，以下推导中，省略变量中的(t)。

5.3.3 滑模控制器的设计与分析

控制的目标为 $\theta\to\theta_{\mathrm{d}}$。针对模型式(5.18)，设计滑模函数

$$s=ce+\dot{e} \tag{5.27}$$

其中，$c>0$，$e=\theta_{\mathrm{d}}-\theta$。

采用观测器式(5.24)求 $\hat{\theta}$ 和 $\hat{\dot{\theta}}$，设计控制律为

$$u(t)=\frac{1}{k}(\ddot{\theta}_{\mathrm{d}}+a\hat{\dot{\theta}}+b\hat{\theta}+\eta\hat{s}+c\hat{\dot{e}}) \tag{5.28}$$

其中，$\eta>0$，令 $\hat{e}=\theta_{\mathrm{d}}-\hat{\theta}$，$\hat{s}=c\hat{e}+\dot{\hat{e}}$。

取滑模控制的 Lyapunov 函数为

$$V=\frac{1}{2}s^2$$

由于

$$\ddot{e}=\ddot{\theta}_{\mathrm{d}}-\ddot{\theta}=\ddot{\theta}_{\mathrm{d}}+a\dot{\theta}+b\theta-ku$$

$$\dot{s}=c\dot{e}+\ddot{e}=c\dot{e}+\ddot{\theta}_{\mathrm{d}}+a\dot{\theta}+b\theta-ku$$

则

$$\begin{aligned}\dot{s}&=c\dot{e}+\ddot{\theta}_{\mathrm{d}}+a\dot{\theta}+b\theta-(\ddot{\theta}_{\mathrm{d}}+a\dot{\hat{\theta}}+b\hat{\theta}+\eta\hat{s}+c\dot{\hat{e}})\\&=c\dot{\tilde{e}}+a\dot{\tilde{\theta}}+b\tilde{\theta}-\eta\hat{s}\\&=-\eta s+\eta\tilde{s}+c\dot{\tilde{e}}+a\dot{\tilde{\theta}}+b\tilde{\theta}\\&=-\eta s+\eta(-c\tilde{\theta}-\dot{\tilde{\theta}})+c(-\dot{\tilde{\theta}})+a\dot{\tilde{\theta}}+b\tilde{\theta}\\&=-\eta s+\eta(-c\tilde{\theta}-\dot{\tilde{\theta}})+c(-\dot{\tilde{\theta}})+a\dot{\tilde{\theta}}+b\tilde{\theta}\\&=-\eta s+(b-\eta c)\tilde{\theta}+(a-\eta-c)\dot{\tilde{\theta}}\end{aligned}$$

其中，$\tilde{\theta}=\theta-\hat{\theta}$，$\dot{\tilde{\theta}}=\dot{\theta}-\dot{\hat{\theta}}$，$\tilde{e}=e-\hat{e}=-\theta+\hat{\theta}=-\tilde{\theta}$，$\dot{\tilde{e}}=-\dot{\tilde{\theta}}$，$\tilde{s}=s-\hat{s}=c\tilde{e}+\dot{\tilde{e}}=-c\tilde{\theta}-\dot{\tilde{\theta}}$。

于是

$$\dot{V}=-\eta s^2+s((b-\eta c)\tilde{\theta}+(a-\eta-c)\dot{\tilde{\theta}})=-\eta s^2+k_1 s\tilde{\theta}+k_2 s\dot{\tilde{\theta}}$$

其中，$k_1=b-\eta c$，$k_2=a-\eta-c$。

由于 $k_1 s\tilde{\theta}\leqslant\frac{1}{2}s^2+\frac{1}{2}k_1^2\tilde{\theta}^2$，$k_2 s\dot{\tilde{\theta}}\leqslant\frac{1}{2}s^2+\frac{1}{2}k_2^2\dot{\tilde{\theta}}^2$，则

$$\dot{V}\leqslant-\eta s^2+\frac{1}{2}s^2+\frac{1}{2}k_1^2\tilde{\theta}^2+\frac{1}{2}s^2+\frac{1}{2}k_2^2\dot{\tilde{\theta}}^2=-(\eta-1)s^2+\frac{1}{2}k_1^2\tilde{\theta}^2+\frac{1}{2}k_2^2\dot{\tilde{\theta}}^2$$

其中，$\eta>1$。

由于观测器指数收敛，所以

$$\dot{V}\leqslant-\eta_1 V+\chi(\cdot)\mathrm{e}^{-\sigma_0(t-t_0)}\leqslant-\eta_1 V+\chi(\cdot)$$

其中，$\eta_1=\eta-1>0$，$\chi(\cdot)$是 $\|\tilde{\boldsymbol{z}}(t_0)\|$ 的 K 类函数，取 $\boldsymbol{z}=[\theta\quad\dot{\theta}]^{\mathrm{T}}$。

采用引理 1.1，不等式方程 $\dot{V}\leqslant-\eta_1 V+\chi(\cdot)$的解为

$$\begin{aligned}V(t)&\leqslant\mathrm{e}^{-\eta_1(t-t_0)}V(t_0)+\chi(\cdot)\mathrm{e}^{-\eta_1 t}\int_{t_0}^{t}\mathrm{e}^{\eta_1\tau}\mathrm{d}\tau\\&=\mathrm{e}^{-\eta_1(t-t_0)}V(t_0)+\frac{\chi(\cdot)\mathrm{e}^{-\eta_1 t}}{\eta_1}(\mathrm{e}^{\eta_1 t}-\mathrm{e}^{\eta_1 t_0})\\&=\mathrm{e}^{-\eta_1(t-t_0)}V(t_0)+\frac{\chi(\cdot)}{\eta_1}(1-\mathrm{e}^{-\eta_1(t-t_0)})\end{aligned}$$

即

$$\lim_{t\to\infty}V(t)\leqslant\frac{1}{\eta_1}\chi(\cdot)$$

且 $V(t)$渐进收敛，收敛精度取决于 η_1。

5.3.4 仿真实例

考虑对象：

$$G(s)=\frac{1}{s^2+10s+1}$$

该对象可表示为

$$\ddot{\theta}=-10\dot{\theta}-\theta+u(t)$$

取角度延迟时间为 $\Delta=3.0$。延迟观测器中，取 $\boldsymbol{K}=[0.1 \quad 0.1]$，采用 MATLAB 函数“fsolve”求方程式(5.26)的根为 $\sigma=-0.3661$，根据引理 5.1，满足稳定性要求。

取 $u(t)=\sin t$，系统初始状态为 $[0.2 \quad 0]$，延迟观测器式(5.24)的初始值 $\hat{z}(t-\Delta)=[0 \quad 0]^{\mathrm{T}}$。延迟观测器的观测结果如图 5.10 和图 5.11 所示。可见，采用延迟观测器，可实现角度和角速度的理想观测。

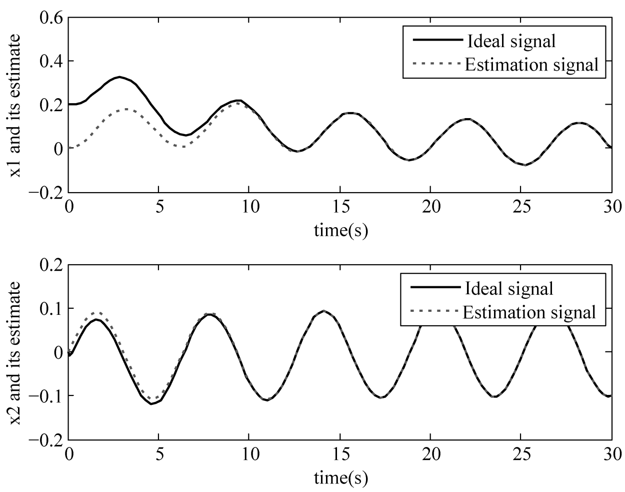


图 5.10 角度和角速度的观测

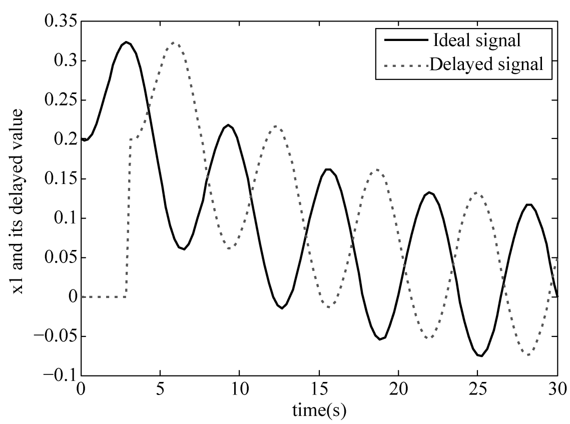


图 5.11 理想角度信号及其实测延迟信号

采用基于延迟观测器的滑模控制，控制器取式(5.28)，取 $c=10$，$k=1$，$\eta=15$，控制效果如图 5.12 和图 5.13 所示。可见，通过采用输出延迟观测器，可获得良好的跟踪性能。

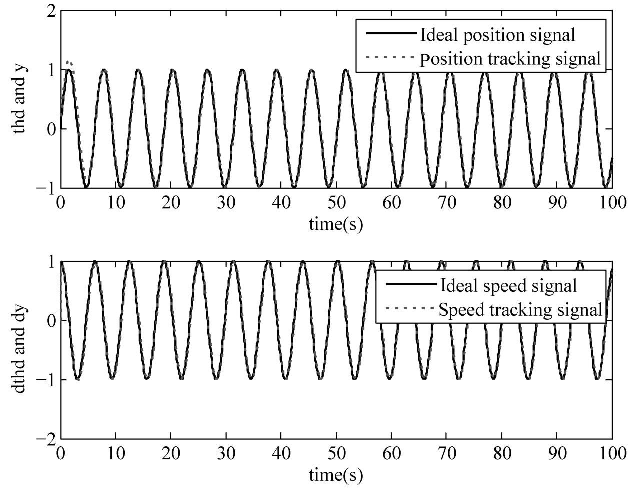


图 5.12　基于延迟观测器的角度、角速度跟踪

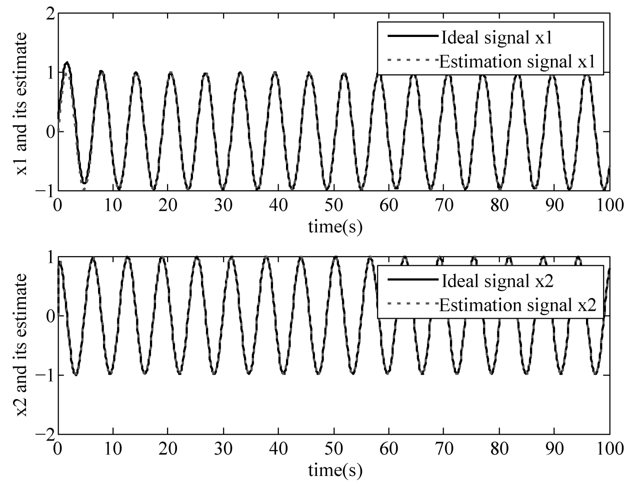


图 5.13　延迟观测器的观测结果

仿真程序：

1. 延迟观测器的验证

(1) K 的验证主程序：design_K. m

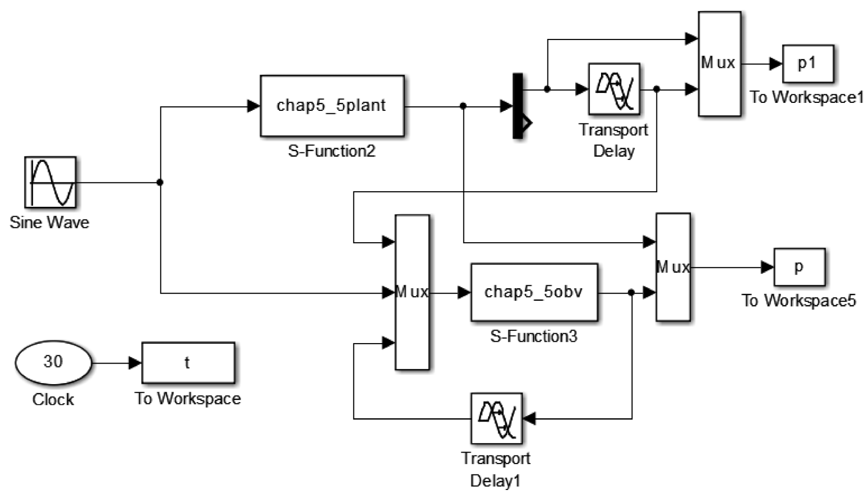
```
close all;

x0 = 0;
options = foptions;
options(1) = 1;
x = fsolve('fun_x',x0,options)
```

(2) K 的验证子程序：fun_x.m

```
function F = fun(x)
tol = 3;
k1 = 0.1;k2 = 0.1;

K = [k1,k2]';
C = [1,0];
A = [0 1; - 1  - 10];

F = det(x * eye(2) - A + K * C * exp( - tol * x));
```

2. 延迟观测器

(1) 主程序：chap5_5sim.mdl



(2) 对象 S 函数：chap5_5plant.m

```
function [sys,x0,str,ts] = s_function(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  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [0.2 0];
str = [];
ts  = [-1 0];
function sys = mdlDerivatives(t,x,u)
sys(1) = x(2);
sys(2) = -10 * x(2) - x(1) + u(1);
function sys = mdlOutputs(t,x,u)
th = x(1);w = x(2);

sys(1) = th;
sys(2) = w;
```

(3) 观测器 S 函数：chap5_5obv.m

```
function [sys,x0,str,ts] = s_function(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       = 4;
sizes.DirFeedthrough  = 0;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [0 0];
str = [];
ts  = [-1 0];
function sys = mdlDerivatives(t,x,u)
tol = 3;
th_tol = u(1);
yp = th_tol;

ut = u(2);

z_tol = [u(3);u(4)];
```

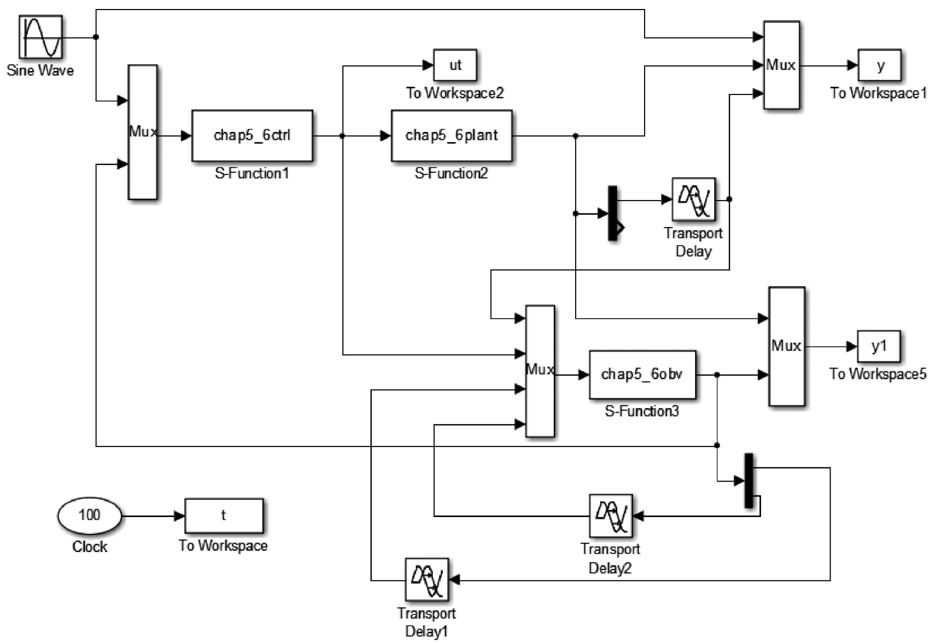
```
thp = x(1);wp = x(2);
%%%%%%%%%%%%%%%%%%%
A = [0 1; -1 -10];
C = [1 0];

H = [0;1];

k1 = 0.1;k2 = 0.1; % Verify by design_K.m
z = [thp wp]';
%%%%%%%%%%%%%%%%%%
K = [k1 k2]';

dz = A * z + H * ut + K * (yp - C * z_tol);

for i = 1:2
    sys(i) = dz(i);

end
function sys = mdlOutputs(t,x,u)
thp = x(1);wp = x(2);

sys(1) = thp;
sys(2) = wp;
```

(4) 作图程序：chap7_11plot.m

```
close all;

figure(1);
subplot(211);
plot(t,p(:,1),'k',t,p(:,3),'r:','linewidth',2);
xlabel('time(s)');ylabel('x1 and its estimate');
legend('ideal signal','estimation signal');
subplot(212);
plot(t,p(:,2),'k',t,p(:,4),'r:','linewidth',2);
xlabel('time(s)');ylabel('x2 and its estimate');
legend('ideal signal','estimation signal');

figure(2);
subplot(211);
plot(t,p(:,1) - p(:,3),'r','linewidth',2);
xlabel('time(s)');ylabel('error of x1 and its estimate');
subplot(212);
plot(t,p(:,2) - p(:,4),'r','linewidth',2);
xlabel('time(s)');ylabel('error of x2 and its estimate');

figure(3);
plot(t,p1(:,1),'k',t,p1(:,2),'r:','linewidth',2);
xlabel('time(s)');ylabel('x1 and its delayed value');
legend('ideal signal','delayed signal');
```

3. 滑模控制系统仿真程序

(1) 主程序：chap5_6sim.mdl



(2) 控制器 S 函数：chap5_6ctrl.m

```
function [sys,x0,str,ts] = s_function(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {1,2,4,9}
    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       = 3;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [-1 0];
```

```
function sys = mdlOutputs(t,x,u)
tol = 3;
thd = sin(t);
wd = cos(t);
ddthd = - sin(t);

thp = u(2);
wp = u(3);

e1p = thd - thp;
e2p = wd - wp;

k = 1;a = 10;b = 1;
c = 10;
xite = 15;
sp = c * e1p + e2p;
ut = 1/k * (ddthd + a * wp + b * thp + xite * sp + c * e2p);

sys(1) = ut;
```

(3）对象 S 函数：chap5_6plant.m

```
function [sys,x0,str,ts] = s_function(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  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [0.2 0];
str = [];
ts  = [-1 0];
function sys = mdlDerivatives(t,x,u)
sys(1) = x(2);
```

```
sys(2) = -10 * x(2) - x(1) + u(1);
function sys = mdlOutputs(t,x,u)
th = x(1);w = x(2);

sys(1) = th;
sys(2) = w;
```

(4) 观测器 S 函数：chap5_6obv.m

```
function [sys,x0,str,ts] = s_function(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        = 4;
sizes.DirFeedthrough   = 0;
sizes.NumSampleTimes   = 1;
sys = simsizes(sizes);
x0  = [0 0];
str = [];
ts  = [-1 0];
function sys = mdlDerivatives(t,x,u)
tol = 3;
th_tol = u(1);
yp = th_tol;

ut = u(2);

z_tol = [u(3);u(4)];

thp = x(1);wp = x(2);
%%%%%%%%%%%%%%%%%%
A = [0 1;-1 -10];
C = [1 0];

H = [0;1];

k1 = 0.1;k2 = 0.1; % Verify by design_K.m
```

```
z = [thp wp]';
%%%%%%%%%%%%%%%%%%
K = [k1 k2]';

dz = A * z + H * ut + K * (yp - C * z_tol);

for i = 1:2
    sys(i) = dz(i);

end
function sys = mdlOutputs(t,x,u)
thp = x(1);wp = x(2);

sys(1) = thp;
sys(2) = wp;
```

(5) 作图程序：chap5_6plot. m

```
close all;
figure(1);
plot(t,y(:,1),'k',t,y(:,3),'r:','linewidth',2);
xlabel('time(s)');ylabel('thd and y');
legend('ideal position signal','delayed position signal');

figure(2);
subplot(211);
plot(t,y1(:,1),'k',t,y1(:,3),'r:','linewidth',2);
xlabel('time(s)');ylabel('x1 and its estimate');
legend('ideal signal x1','estimation signal x1');
subplot(212);
plot(t,y1(:,2),'k',t,y1(:,4),'r:','linewidth',2);
xlabel('time(s)');ylabel('x2 and its estimate');
legend('ideal signal x2','estimation signal x2');

figure(3);
subplot(211);
plot(t,y(:,1),'k',t,y(:,2),'r:','linewidth',2);
xlabel('time(s)');ylabel('thd and y');
legend('ideal position signal','position tracking signal');
subplot(212);
plot(t,cos(t),'k',t,y(:,3),'r:','linewidth',2);
xlabel('time(s)');ylabel('dthd and dy');
legend('ideal speed signal','speed tracking signal');

figure(4);
subplot(211);
plot(t,y(:,1) - y(:,2),'r','linewidth',2);
xlabel('time(s)');ylabel('error between thd and y');
legend('position tracking error');
subplot(212);
plot(t,cos(t) - y(:,3),'r','linewidth',2);
```

```
xlabel('time(s)');ylabel('error between dthd and dy');
legend('speed tracking error');

figure(5);
plot(t,ut(:,1),'k','linewidth',2);
xlabel('time(s)');ylabel('Control input');
```

5.4 一种时变测量延迟观测器及滑模控制

在实际过程中，测量延迟信号往往是时变的，此时需要设计时变测量延迟观测器。文献[9,10]给出了满足 Lipschitz 条件的两类测量延迟观测器的设计方法。本节针对满足 Lipschitz 条件的非线性系统，设计一种具有时变测量延迟的观测器[9]。

5.4.1 系统描述

假设二阶非线性系统为

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{M}(x,t,u) \tag{5.29}$$

其中，$\boldsymbol{x}=[x_1 \quad x_2]^{\mathrm{T}}$，$x_1$ 为位置信号，u 为控制输入，$\boldsymbol{A}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$，$\boldsymbol{M}(x,t,u)=\boldsymbol{B}\boldsymbol{\chi}(x,t,u)$，$\boldsymbol{B}=[0 \quad 1]^{\mathrm{T}}$。

假设 1：$\boldsymbol{M}(x,t,u)$为 Lipschitz 条件；

假设 2：$\delta(t)$为时变的，$\delta(t)\in[0,\Delta]$。

假设输出信号有延迟，$\delta(t)$为输出位置的时间延迟，则实际输出可表示为

$$\bar{y}(t)=x_1(t-\delta)=\boldsymbol{C}\boldsymbol{x}(t-\delta(t)) \tag{5.30}$$

其中，$\boldsymbol{C}=[1 \quad 0]$。

观测的目标为：当 $t\rightarrow\infty$时，$\hat{x}_1(t)\rightarrow x_1(t)$，$\hat{x}_2(t)\rightarrow x_2(t)$。

5.4.2 输出延迟观测器的设计

定理 5.1[9]：针对非线性系统式(5.29)，假设 $x_1(t-\delta)$为实测的延迟信号，则延迟观测器设计为

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}}(t)=\boldsymbol{A}\hat{\boldsymbol{x}}(t)+\boldsymbol{M}(\hat{\boldsymbol{x}},t,u)+\boldsymbol{K}(\bar{y}(t)\\ -\boldsymbol{C}\hat{\boldsymbol{x}}(t-\delta))\end{aligned} \tag{5.31}$$

针对二阶系统，观测器具体表示为

$$\dot{\hat{x}}_1=\hat{x}_2+k_1(x_1(t-\delta)-\hat{x}_1(t-\delta))$$

$$\dot{\hat{x}}_2=\boldsymbol{\chi}(\hat{x},t,u)+k_2(x_1(t-\delta)-\hat{x}_1(t-\delta))$$

其中，$\hat{\boldsymbol{x}}(t-\delta)$是$\hat{\boldsymbol{x}}(t)$的延迟信号，$\boldsymbol{K}=[k_1 \quad k_2]^{\mathrm{T}}$，通过 $\boldsymbol{K}$ 的设计，使 $\boldsymbol{A}-\boldsymbol{KC}$ 满足 Hurwitz 条件。

根据定理 5.1，延迟观测器式(5.31)为渐进收敛，即当 $t\rightarrow\infty$时，$\hat{x}_1(t)\rightarrow x_1(t)$，

$\hat{x}_2(t) \to x_2(t)$。该定理的分析和证明见文献[9]。

5.4.3　按 $A-KC$ 为 Hurwitz 进行 K 的设计

观测器稳定条件为($\boldsymbol{A}-\boldsymbol{KC}$)为 Hurwitz。由于

$$\bar{\boldsymbol{A}}=\boldsymbol{A}-\boldsymbol{KC}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}-\begin{bmatrix}k_1\\ k_2\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}$$

$$=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}-\begin{bmatrix}k_1 & 0\\ k_2 & 0\end{bmatrix}=\begin{bmatrix}-k_1 & 1\\ -k_2 & 0\end{bmatrix}$$

其特征方程为

$$|\lambda\boldsymbol{I}-\bar{\boldsymbol{A}}|=\begin{vmatrix}\lambda+k_1 & 1\\ -k_2 & \lambda\end{vmatrix}=\lambda^2+k_1\lambda+k_2=0$$

由 $(\lambda+k)^2=0$ 得 $\lambda^2+2k\lambda+k^2=0, k>0$,从而

$$k_1=2k,\quad k_2=k^2 \tag{5.32}$$

通过 $\boldsymbol{K}$ 的设计,使 $\boldsymbol{A}-\boldsymbol{KC}$ 满足 Hurwitz 条件。

5.4.4　观测器仿真实例

单力臂机械手动力学方程为

$$\ddot{x}=-\frac{1}{I}(d\dot{x}+mgl\cos x)+\frac{1}{I}u$$

该动态方程可写为

$$\dot{x}_1=x_2$$

$$\dot{x}_2=-\frac{d}{I}x_2-\frac{mgl}{I}\cos x_1+\frac{1}{I}u$$

其中,x_1 和 x_2 分别为角度和角速度;输出为 $\tilde{y}(t)=x_1(t-\Delta)$;$u$ 为控制输入。模型物理参数取 $g=9.8, m=1, l=0.25, d=2.0$。

上式可整理为

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{M}(x,t,u)$$

其中,$\boldsymbol{x}=[x_1\quad x_2]^{\mathrm{T}}$,$x_1$ 为角度信号,$u(t)$为控制输入,$\boldsymbol{A}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$,$\boldsymbol{M}(x,t,u)=\boldsymbol{B}\boldsymbol{\chi}(x,t,u)$,$\boldsymbol{B}=[0\quad 1]^{\mathrm{T}}$,$\boldsymbol{\chi}(x,t,u)=\begin{bmatrix}0 & -\frac{d}{I}x_2-\frac{mgl}{I}\cos x_1+\frac{1}{I}u\end{bmatrix}^{\mathrm{T}}$。

可见,$\boldsymbol{M}(x,t,u)$为 Lipschitz 的。仿真时,取 $\Delta=1.0$,则延迟时间取值范围为 $\delta(t)\in[0,1.0]$,$\delta(t)$取随机值,仿真中由程序 delta.m 实现。延迟观测器中,按式(5.32)设计 $\boldsymbol{K}$,取 $k=1$,则 $\boldsymbol{K}=[2\quad 1]^{\mathrm{T}}$。取输入 $u(t)=\sin t$,模型式(5.29)初始状态为$[0.50\quad 0]^{\mathrm{T}}$,延迟观测器式(5.31)的初始值$\hat{\boldsymbol{x}}(t-\delta)=[0\quad 0]^{\mathrm{T}}$。延迟观测器的观测结果如图 5.14～图 5.16 所示。可见,采用延迟观测器,可实现带角度测量延迟的角度和角速度理想观测。

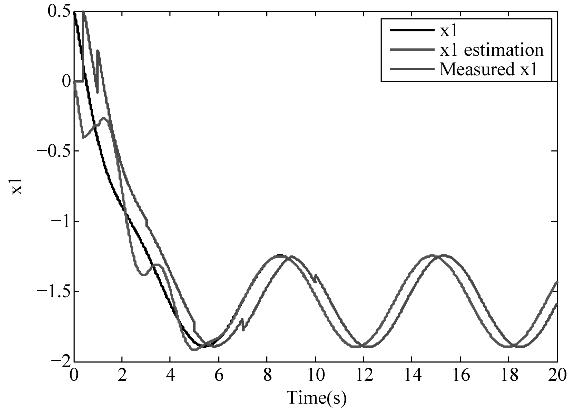


图 5.14　理想位置信号、观测值及实测延迟信号

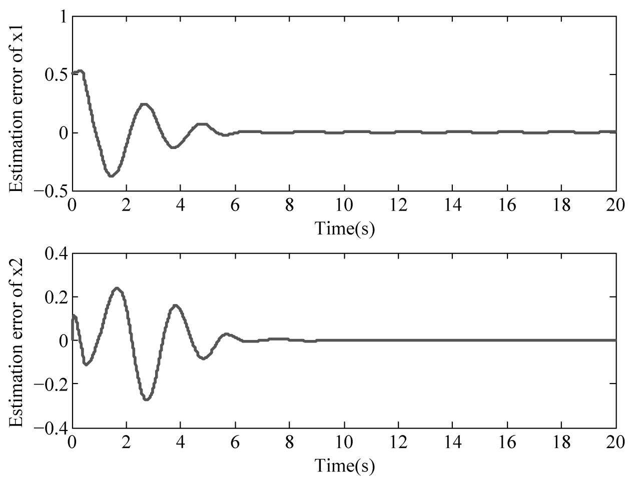


图 5.15　位置和速度的观测误差

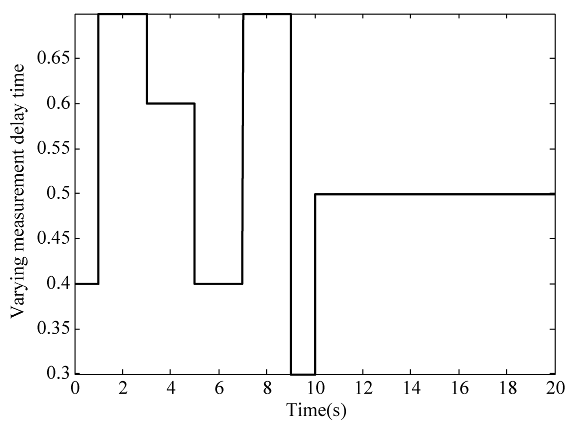


图 5.16　时变的测量延迟

仿真程序：

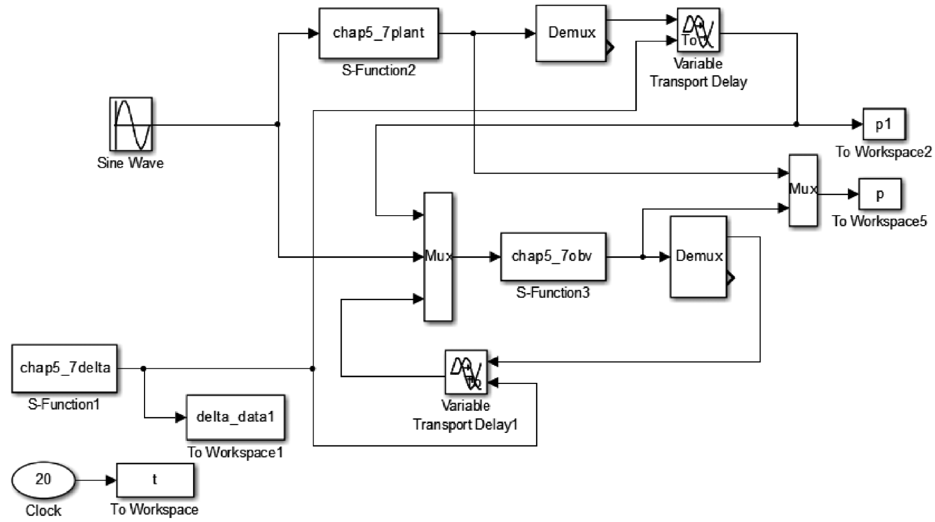
（1）测量延迟信号程序：chap5_7delta. m

```
function [sys,x0,str,ts] = s_function(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  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 0;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [-1 0];
function sys = mdlOutputs(t,x,u)
sys(1) = delta(t);
```

（2）测量延迟信号产生函数：delta. m

```
function [tout] = tolt(tin)
if tin <= 1.0
    tout = 0.40;
elseif tin <= 3.0
    tout = 0.70;
elseif tin <= 5.0
    tout = 0.60;
elseif tin <= 7.0
    tout = 0.40;
elseif tin <= 9.0
    tout = 0.70;
elseif tin <= 10.0
    tout = 0.30;
else
    tout = 0.50;
end
tout = tout;
```

(3) 主程序：chap5_7sim. mdl



(4) 对象 S 函数：chap5_7plant. m

```
function [sys,x0,str,ts] = s_function(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  = [0.5 0];
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
g = 9.8;m = 1;l = 0.25;d = 2.0;
I = 4/3 * m * l^2;

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

```
sys(2) = 1/I * ( - d * x(2) - m * g * l * cos(x(1))) + 1/I * u;
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);
sys(2) = x(2);
```

(5) 观测器 S 函数：chap5_7obv. m

```
function [sys,x0,str,ts] = s_function(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       = 3;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [0 0];
str = [];
ts  = [ - 1 0];
function sys = mdlDerivatives(t,x,u)
x1p = x(1);
x2p = x(2);

y = u(1);
ut = u(2);
yp = u(3);

k1 = 2;k2 = 1;
K = [k1 k2]';

g = 9.8;m = 1;l = 0.25;d = 2.0;
I = 4/3 * m * l^2;

X = 1/I * ( - d * x2p - m * g * l * cos(x1p)) + 1/I * ut;

sys(1) = x2p + k1 * (y - yp);
sys(2) = X + k2 * (y - yp);
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);
sys(2) = x(2);
```

（6）作图程序：chap5_7plot.m

```
close all;
figure(1);
plot(t,p(:,1),'k',t,p(:,3),'r',t,p1(:,1),'b','linewidth',2);
xlabel('Time/s');ylabel('x1');
legend('x1','x1 estimation','measured x1');

figure(2);
subplot(211);
plot(t,p(:,1)-p(:,3),'r','linewidth',2);
xlabel('Time/s');ylabel('Estimation error of x1');
subplot(212);
plot(t,p(:,2)-p(:,4),'r','linewidth',2);
xlabel('Time/s');ylabel('Estimation error of x2');

figure(3);
plot(t,delta_data1,'k','linewidth',2);
xlabel('Time/s');ylabel('Varying measurement delay time');
```

5.4.5 基于时变测量输出延迟观测器的滑模控制

考虑5.4.4节的单力臂机械手动力学方程：

$$\begin{cases}\dot{x}_1=x_2\\ \dot{x}_2=-\dfrac{d}{I}x_2-\dfrac{mgl}{I}\cos x_1+\dfrac{1}{I}u\end{cases}\tag{5.33}$$

输出为 $\tilde{y}(t)=x_1(t-\delta)$，$\delta$ 为时变，采用观测器(5.31)求 $\hat{x}_1$ 和 $\hat{x}_2$。控制目标为 $x_1\to x_{1\mathrm{d}}$，$x_2\to\dot{x}_{1\mathrm{d}}$。设计滑模函数为

$$s=ce+\dot{e}$$

其中，$c>0$，$e=x_1-x_{1\mathrm{d}}$。

取Lyapunov函数为

$$V=\frac{1}{2}s^2$$

由于

$$\ddot{e}=\ddot{x}_1-\ddot{x}_{1\mathrm{d}}=-\frac{d}{I}x_2-\frac{mgl}{I}\cos x_1+\frac{1}{I}u-\ddot{x}_{1\mathrm{d}}$$

则

$$\dot{s}=c\dot{e}+\ddot{e}=c\dot{e}-\frac{d}{I}x_2-\frac{mgl}{I}\cos x_1+\frac{1}{I}u-\ddot{x}_{1\mathrm{d}}$$

采用观测器式(5.31)求 $\hat{x}_1$ 和 $\hat{x}_2$，设计控制律为

$$u(t)=I\left(\frac{d}{I}\hat{x}_2+\frac{mgl}{I}\cos\hat{x}_1+\ddot{x}_{1\mathrm{d}}-c\hat{\dot{e}}-\eta\hat{s}\right)\tag{5.34}$$

其中，$\eta>0$，$\hat{e}=\hat{x}_1-x_{1\mathrm{d}}$，$\hat{s}=c\hat{e}+\hat{\dot{e}}$。

取 $\tilde{s}=\hat{s}-s$，则

$$\begin{aligned}\dot{s}&=c\dot{e}+\ddot{e}=c\dot{e}-\frac{d}{I}x_2-\frac{mgl}{I}\cos x_1+\frac{1}{I}u-\ddot{x}_{1d}\\&=c\dot{e}-\frac{d}{I}x_2-\frac{mgl}{I}\cos x_1+\frac{d}{I}\hat{x}_2+\frac{mgl}{I}\cos\hat{x}_1-c\dot{\hat{e}}-\eta\hat{s}\\&=-\eta s-\eta\tilde{s}-c\dot{\tilde{e}}+\frac{d}{I}\tilde{x}_2+\frac{mgl}{I}(\cos\hat{x}_1-\cos x_1)\\&=-\eta s-\eta(c\tilde{x}_1+\tilde{x}_2)-c\tilde{x}_2+\frac{d}{I}\tilde{x}_2+\frac{mgl}{I}(\cos\hat{x}_1-\cos x_1)\end{aligned}$$

其中，令 $\tilde{e}=e-\hat{e}=x_1-\hat{x}_1=\tilde{x}_1$，$\dot{\tilde{e}}=\dot{e}-\dot{\hat{e}}=x_2-\hat{x}_2=\tilde{x}_2$，$\tilde{s}=\hat{s}-s=c\tilde{e}+\dot{\tilde{e}}=c\tilde{x}_1+\tilde{x}_2$。

于是

$$\begin{aligned}\dot{V}&=-\eta s^2+s\left(-\eta(c\tilde{x}_1+\tilde{x}_2)-c\tilde{x}_2+\frac{d}{I}\tilde{x}_2+\frac{mgl}{I}(\cos\hat{x}_1-\cos x_1)\right)\\&\leqslant-\eta s^2+s\left(-\eta(c\tilde{x}_1+\tilde{x}_2)-c\tilde{x}_2+\frac{d}{I}\tilde{x}_2+\frac{2mgl}{I}\right)\\&=-\eta s^2+s\left(O(\tilde{x}_1,\tilde{x}_2)+\frac{2mgl}{I}\right)\end{aligned}$$

其中，$O(\tilde{x}_1,\tilde{x}_2)=-\eta(c\tilde{x}_1+\tilde{x}_2)-c\tilde{x}_2+\frac{d}{I}\tilde{x}_2$。

由于观测器渐进收敛，则 $O(\tilde{x}_1,\tilde{x}_2)+\frac{2mgl}{I}\leqslant O_{\max}$，于是

$$\begin{aligned}\dot{V}&\leqslant-\eta s^2+0.5(s^2+O_{\max}^2)=-(\eta-0.5)s^2+0.5O_{\max}^2\\&=-(2\eta-1)V+0.5O_{\max}^2\end{aligned}$$

取 $\eta_1=2\eta-1>0$，采用引理1.1，不等式方程 $\dot{V}\leqslant-\eta_1V+0.5O_{\max}^2$ 的解为

$$V(t)\leqslant e^{-\eta_1 t}V(t_0)+0.5O_{\max}^2\int_0^t e^{-\eta_1(t-\tau)}d\tau$$

由于 $\int_0^t e^{-\eta_1(t-\tau)}d\tau=\frac{1}{\eta_1}e^{-\eta_1 t}\int_0^t e^{\eta_1\tau}d\eta_1\tau=\frac{1}{\eta_1}e^{-\eta_1 t}e^{\eta_1 t}=\frac{1}{\eta_1}$，则

$$V(t)\leqslant e^{-\eta_1 t}V(t_0)+\frac{1}{2\eta_1}O_{\max}^2$$

当 $t\to\infty$ 时，$V(t)\to\frac{1}{2\eta_1}O_{\max}^2$。可见，跟踪误差收敛结果取决于 η 和 $O_{\max}$，当 η_1 足够大，$O_{\max}\to0$ 时，$s\to0$，$e\to0$，$\dot{e}\to0$。

5.4.6 闭环控制仿真实例

采用与观测器仿真实例相同的单力臂机械手动力学方程为被控对象模型。仿真时，取 $\Delta=1.0$，则延迟时间取值范围为 $\delta(t)\in[0,1.0]$。延迟观测器中，取 $k=1$，则 $\boldsymbol{K}=[2\quad1]^{\mathrm{T}}$。取输入 $u(t)=\sin t$，模型式(5.29)初始状态为 $[0.5\quad0]^{\mathrm{T}}$，延迟观测器式(5.31)的初始值 $\hat{\boldsymbol{x}}(t-\delta)=[0\quad0]^{\mathrm{T}}$。

采用控制器式(5.34)，取 $c=50$，$\eta=30$，仿真结果如图5.17～图5.19所示。可见，采用基于延迟观测器的滑模控制方法，可实现角度和角速度的高精度控制。

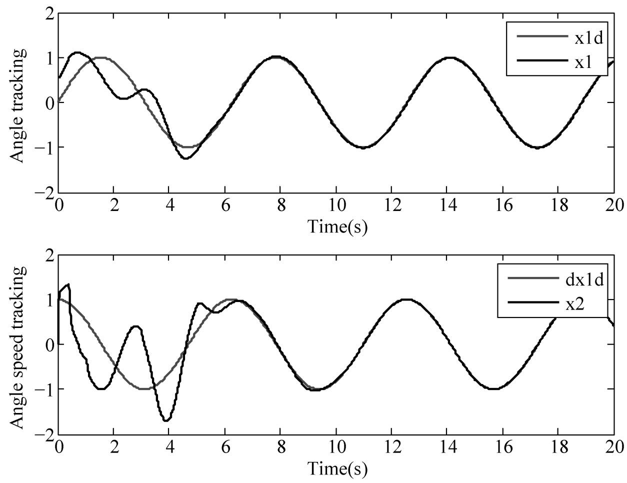


图 5.17　角度和角速度跟踪

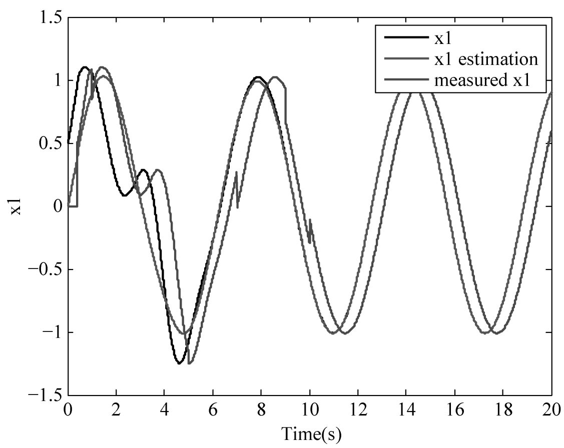


图 5.18　理想角度信号、观测值及实测延迟信号

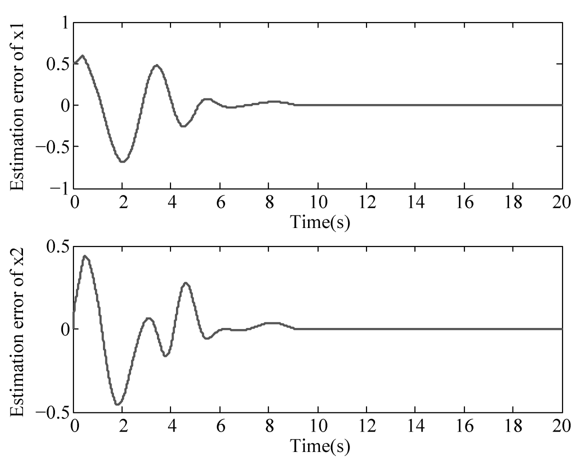


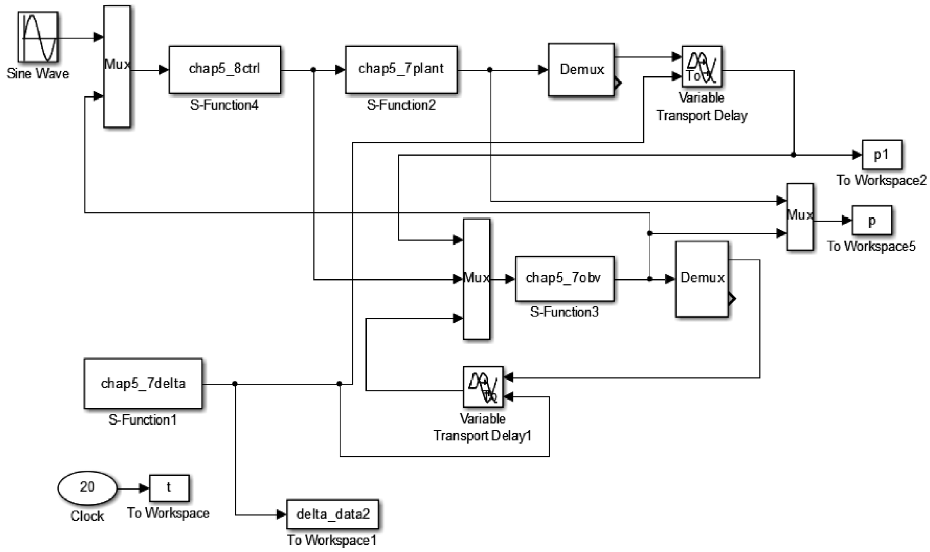
图 5.19　角度和角速度的观测误差

仿真程序：

(1) 测量延迟信号程序[①]：chap5_7delta. m

(2) 测量延迟信号产生函数：delta. m

(3) 主程序：chap5_8sim. mdl



(4) 控制器 S 函数：chap5_8ctrl. m

```
function [sys,x0,str,ts] = s_function(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {1,2,4,9}
    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      = 3;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [-1 0];
```

① 为节约篇幅，这类略去的程序代码可到清华大学出版社网站本书所在页面下载。

```
function sys = mdlOutputs(t,x,u)
x1d = sin(t);
dx1d = cos(t);
ddx1d = - sin(t);

x1p = u(2);
x2p = u(3);

e1p = x1p - x1d;
e2p = x2p - dx1d;

c = 50;
xite = 30;
sp = c * e1p + e2p;
g = 9.8;m = 1;l = 0.25;d = 2.0;I = 4/3 * m * l^2;

ut = I * (d/I * x2p + m * g * l * cos(x1p) + ddx1d - c * e2p - xite * sp);

sys(1) = ut;
```

（5）被控对象 S 函数：chap5_7plant. m

（6）观测器 S 函数：chap5_7obv. m

（7）作图程序：chap5_8plot. m

```
close all;
figure(1);
subplot(211);
plot(t,sin(t),t,p(:,1),'k','linewidth',2);
xlabel('Time/s');ylabel('angle tracking');
legend('x1d','x1');
subplot(212);
plot(t,cos(t),t,p(:,2),'k','linewidth',2);
xlabel('Time/s');ylabel('angle speed tracking');
legend('dx1d','x2');

figure(2);
plot(t,p(:,1),'k',t,p(:,3),'r',t,p1(:,1),'b','linewidth',2);
xlabel('Time/s');ylabel('x1');
legend('x1','x1 estimation','measured x1');

figure(3);
subplot(211);
plot(t,p(:,1) - p(:,3),'r','linewidth',2);
xlabel('Time/s');ylabel('Estimation error of x1');
subplot(212);
plot(t,p(:,2) - p(:,4),'r','linewidth',2);
xlabel('Time/s');ylabel('Estimation error of x2');

figure(4);
plot(t,delta_data2,'k','linewidth',2);
xlabel('Time/s');ylabel('Varying measurement delay time');
```

参考文献

[1] Atsuo K, Hiroshi I, Kiyoshi S. Chattering Reduction of Disturbance Observer Based Sliding Mode Control[J]. IEEE Transactions on Industry Applications, 1994, 30(2): 456-461.

[2] Chen W H, Balance D J, Gawthrop P J, et al. A nonlinear disturbance observer for robotic manipulator[J]. IEEE Transactions on Industrial Electronics, 2000, 47(4): 932-938.

[3] Michiels W, Roose D. Time-delay Compensation in Unstable Plants Using Delayed[C]. Proceedings of the 40th IEEE Conference on Decision and Control, 2001, 2: 1433-1437.

[4] Subbarao K, Muralidhar P C. State Observer for Linear Systems with Piece-Wise Constant Output Delays[J]. IET Control Theory and Applications, 2001, 3: 1017-1022.

[5] Germani A, Manes C, Pepe P. A New Approach to State Observation of Nonlinear Systems with Delayed Output[J]. Automatic Control, 2002, 47(1): 96-101.

[6] Kazantzis N, Wright R A. Nonlinear Observer Design in the Presence of Delayed Output Measurements[J]. Systems & Control Letters, 2005, 54(9): 877-886.

[7] Sun Leping. Stability Criteria for Delay Differential Equations[J]. Journal of Shanghai Teachers University, 1998, 27(3): 1-6.

[8] Desoer C A, Vidyasagar M. Feedback System: Input-output Properties[M]. New York: Academic Press, 1977.

[9] He Qing, Liu Jinkun. An observer for a velocity-sensorless VTOL aircraft with time-varying measurement delay[J/OL]. International Journal of Systems Science, 2014. http://dx.doi.org/10.1080/00207721.2014.900135.

[10] He Qing, Liu Jinkun. Sliding mode observer for a class of globally Lipschitz nonlinear systems with time-varying delay and noise in its output[J/OL]. IET Control Theory & Applications, 2014. http://doi: 10.1049/iet-cta.2013.1004.