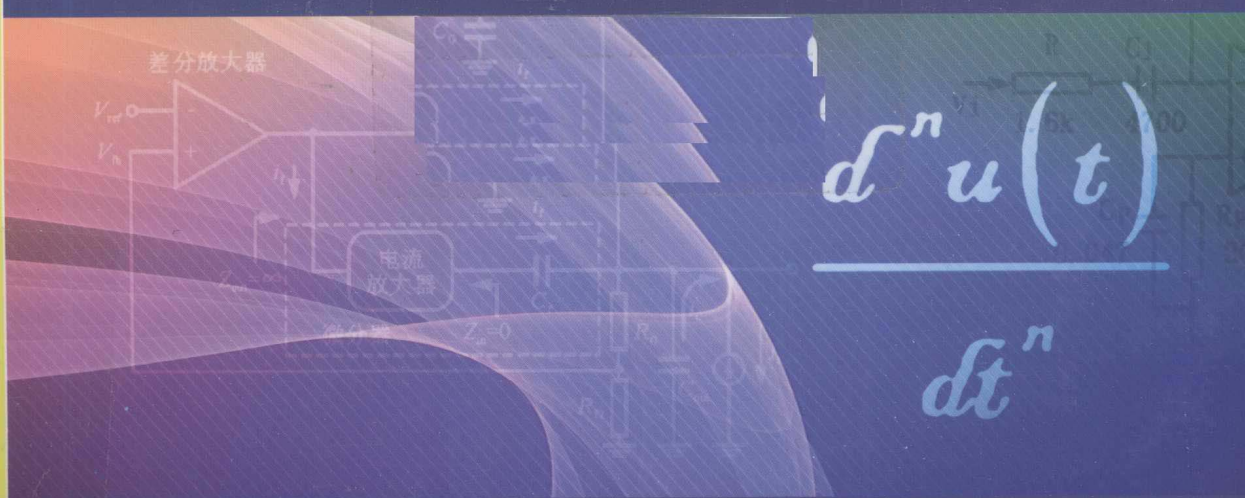


电子工程技术丛书

# 微分器设计与应用 —— 信号滤波与求导

Differentiator Design and Application  
—— Signal Filtering and Differentiation

● 王新华 刘金琨 著



The background of the lower half of the cover features several technical diagrams. On the left, there is a circuit diagram of a differential amplifier with inputs labeled  $V_{id}$  and  $V_{in}$ . In the center, there is a diagram of a current amplifier with a feedback loop and a label '电流放大器'. On the right, there is a large mathematical formula for the  $n$ -th derivative of a function  $u(t)$  with respect to time  $t$ .

$$\frac{d^n u(t)}{dt^n}$$



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

电子工程技术丛书

# 微分器设计与应用

——信号滤波与求导

*Differentiator Design and Application*

——*Signal Filtering and Differentiation*

王新华 刘金琨 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书系统地论述了微分器的理论、滤波与求导、设计方法和工程应用，是作者多年来从事微分器理论研究与应用工程实际成果的总结。书中有大量仿真程序及相关 MATLAB 仿真程序。

全书以微分器的理论为基础，结合实际工程问题设计各种形式的微分器，共分 8 章，包括有限时间收敛微分器的原理及其设计方法，混合微分器的概念、原理及其设计方法，基于描述函数法的微分器频域分析，微分器输出中存在的峰值现象分析及其抑制方法，高阶积分链式微分器，微分器对系统不确定项和未知状态的充分逼近及其在反馈控制中的应用等。

本书适用于从事航空航天自动化、计算机应用、通信工程和电气自动化等领域工作的工程技术人员阅读，也可作为高等院校工业自动化、自动控制、自动化仪表、计算机应用等专业的教学参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目(CIP)数据

微分器设计与应用：信号滤波与求导 / 王新华，刘金琨著. — 北京：电子工业出版社，2010.4  
ISBN 978-7-121-10596-8

I. ①微… II. ①王… ②刘… III. ①微分器—设计 IV. ①TP342

中国版本图书馆 CIP 数据核字(2010)第 050451 号

策划编辑：赵丽松

责任编辑：李蕊

印刷：北京东光印刷厂

装订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本：787×1 092 1/16 印张：14.75 字数：378 千字

印次：2010 年 4 月第 1 次印刷

印数：4000 册 定价：38.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

# 前 言

迅速精确地获取被跟踪目标的速度和加速度对于一些系统至关重要，如防御系统中的导弹拦截系统。在大多数情况下，采用差分方法来近似估计信号的导数。通常情况下，由于几乎噪声存在于所有的信号中，所以通过这种方法不能正确地估计出信号的导数。虽然卡尔曼滤波器可以被用来抑制扰动，同时求取信号的导数，但是需要有对象的模型，这限制了信号的通用性，需要对被跟踪目标详细了解。本书采用不基于对象模型的信号导数求取方法，设计了各种适合于实际工程的微分器。

本书的研究目的是继续发展微分器理论、设计及应用。本书系统地论述了微分器的理论、设计方法和工程应用，是作者多年来从事微分器理论研究与工程实际应用成果的总结。本书的特点是：

(1)从基本概念、基本理论和基本方法入手，论述微分器的理论和设计方法，浅显易懂，而且给出了完整的理论证明。

(2)针对每种算法给出了完整的 MATLAB 仿真程序，并给出了程序的说明和仿真结果，具有很强的可读性(基于 MATLAB 6.5 环境下开发)。

(3)着重从应用领域角度出发，突出了理论联系实际，面向广大工程技术人员，具有很强的工程性和实用性。书中有大量应用实例及其结果分析，为读者提供了有益的借鉴。

(4)所给出的各种微分器算法完整，程序设计、结构设计力求简单明了，便于自学和进一步开发。

(5)所介绍的方法并不局限于书中实例的控制，同时也适用于解决运动控制领域其他背景的信号估计和控制问题。

全书以微分器的理论为基础，结合实际工程问题设计各种形式的微分器，共分 8 章。第 1 章为绪论，介绍微分器的由来、基本结构，以及目前国际上经典的微分器；第 2 章为基础知识，介绍本书中用到的数学基础；第 3 章论述有限时间收敛微分器的原理及其设计方法；第 4 章给出了混合微分器的概念、原理及其设计方法，并分析了其消除噪声的本质；第 5 章采用描述函数法对微分器进行分析，从频域角度分析微分器参数对系统动态性能的影响；第 6 章分析了微分器输出中存在的峰值现象，并设计了几种变摄动参数的方法来抑制峰值现象的发生；第 7 章设计了高阶积分链式微分器，用于求取加速度等信息，具有更强的快速收敛性和消除扰动的能力；第 8 章实现了微分器的充分逼近、扩张观测器的充分逼近及其在反馈控制中的应用，并把该逼近方法与经典的模糊逼近、RBF(Radial-Based-Function)神经网络逼近的方法进行了比较。

本书既对理论进行了详细的分析，又紧密结合了工程实际，适用于从事航空航天自动化、计算机应用、通信工程和电气自动化等领域工作的工程技术人员阅读，也可作为大专院校工业自动化、自动控制、自动化仪表、计算机应用等专业的教学参考书。

书中所有算法的仿真程序都可在 <http://yydz.phei.com.cn> 网站的资源下载栏目下载，也可与本书策划编辑赵丽松(zls@phei.com.cn)联系索取，以方便读者的阅读。



本书的研究工作得到了国家自然科学基金(编号: 60774008)资助。

由于作者水平有限, 书中难免存在一些不足和错误之处, 真诚欢迎广大读者批评指正。若读者有问题指正或需与作者商讨, 或对算法及仿真程序有疑问, 请通过电子邮件 wangxin-hua04@gmail.com 或者 ljk@buaa.edu.cn 与作者联系。作者相信, 通过与广大同行的交流, 作者会得到许多新的有益的建议, 从而将本书写得更好。

# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 研究目的 .....	1
1.2 状态观测器 .....	1
1.2.1 全状态观测器 .....	1
1.2.2 扩张观测器 .....	5
1.3 问题提出 .....	11
1.3.1 一阶惯性环节滤波 .....	11
1.3.2 二阶振荡环节滤波分析 .....	12
1.3.3 二阶振荡环节求导分析 .....	14
1.4 微分器的由来：不基于对象模型 .....	17
1.5 微分器介绍 .....	18
1.5.1 线性微分器 .....	18
1.5.2 非线性微分器 .....	25
1.6 微分器的离散化 .....	36
1.7 微分器的主要用途 .....	41
1.8 本书结构安排 .....	42
参考文献 .....	42
<b>第 2 章 基础知识</b> .....	46
2.1 有限时间稳定基础 .....	46
2.1.1 三种稳定形式 .....	46
2.1.2 有限时间稳定概念 .....	49
2.1.3 Lyapunov 原理 .....	50
2.1.4 系统齐次性引理 .....	52
2.2 奇异摄动理论 .....	52
2.3 传递函数的频域特性扫频测试 .....	53
2.4 系统描述函数法分析 .....	56
参考文献 .....	59
<b>第 3 章 有限时间收敛微分器</b> .....	60
3.1 引言 .....	60
3.2 有限时间收敛微分器原理 .....	60
3.2.1 有限时间收敛微分器设计 .....	62
3.2.2 有限时间收敛微分器抗扰动性分析 .....	64
3.2.3 有限时间收敛二阶微分器 .....	66
3.2.4 有限时间收敛高阶微分器 .....	67



3.3 结论 .....	74
参考文献 .....	74
<b>第4章 混合微分器</b> .....	<b>76</b>
4.1 引言 .....	76
4.2 混合微分器的主要结果 .....	76
4.2.1 线性微分器设计 .....	76
4.2.2 非线性微分器设计 .....	78
4.2.3 混合微分器设计 .....	82
4.3 三种微分器收敛性能比较 .....	86
4.4 基于扫频方法的微分器频域分析 .....	96
4.4.1 不同微分器的频域测试比较 .....	96
4.4.2 不同 $\epsilon$ 对混合微分器频域特性的影响 .....	102
4.5 滑模微分器和混合微分器抑制噪声比较 .....	107
4.6 微分器对系统不确定项的估计 .....	112
4.6.1 在无噪声情况下的不确定项的估计 .....	112
4.6.2 带有噪声的不确定系统不确定项的估计 .....	115
4.7 结论 .....	118
参考文献 .....	118
<b>第5章 基于描述函数法的微分器频域分析</b> .....	<b>120</b>
5.1 引言 .....	120
5.2 线性微分器频域分析 .....	121
5.3 基于描述函数法的非线性微分器频域分析 .....	129
5.4 基于描述函数法的混合微分器频域分析 .....	135
5.5 结论 .....	143
参考文献 .....	143
<b>第6章 微分器峰值现象分析与抑制</b> .....	<b>144</b>
6.1 引言 .....	144
6.2 峰值问题 .....	144
6.3 峰值现象分析 .....	146
6.4 峰值现象抑制 .....	152
6.5 结论 .....	155
参考文献 .....	155
<b>第7章 高阶积分链式微分器及其在加速度反馈中的应用</b> .....	<b>157</b>
7.1 引言 .....	157
7.2 积分链式微分器与高增益微分器的等价性 .....	157
7.3 非线性积分链式微分器 .....	160
7.4 基于三阶微分器的带噪声的不确定系统反馈控制 .....	162
7.5 结论 .....	178
参考文献 .....	178



第 8 章 微分器的充分逼近及其在反馈控制中的应用 .....	180
8.1 引言 .....	180
8.2 问题分析 .....	181
8.3 基于模糊系统的控制器设计 .....	182
8.3.1 基于模糊系统的不确定项逼近 .....	182
8.3.2 自适应模糊控制器设计 .....	182
8.3.3 自适应模糊控制的稳定性分析 .....	183
8.4 基于 RBF 神经网络的自适应控制 .....	191
8.4.1 基于 RBF 神经网络的不确定项逼近 .....	191
8.4.2 自适应神经网络控制的设计与分析 .....	191
8.5 基于扩张观测器逼近的输出反馈控制 .....	197
8.5.1 基于扩张观测器的逼近 .....	197
8.5.2 基于扩张观测器逼近的输出反馈控制器设计 .....	199
8.6 基于积分链式微分器逼近的输出反馈控制 .....	207
8.6.1 基于积分链式微分器的逼近 .....	207
8.6.2 基于积分链式微分器逼近的控制器设计 .....	208
8.7 结论 .....	224
参考文献 .....	224



# 第1章 绪论



## 1.1 研究目的

本书的研究目的是继续发展微分器理论、设计及其应用。信号导数的求取是一个传统而众所周知的问题，并且近年来吸引了更多学者来研究这一领域。迅速、精确地获取被跟踪目标的速度和加速度对于一些系统至关重要，如防御系统中的导弹拦截系统，需要得到对方目标的速度和加速度，才能进行更好的拦截。所以在雷达系统等目标跟踪应用中，由传感器测到的目标位置信息来估计其速度、加速度仍然是一个困难的任务和具有挑战性的问题<sup>[1]</sup>。信号对于时间的导数在其他领域都是要求取的量<sup>[2]~[6]</sup>。在对人体运动分析时，内力和力矩的测定要求估计身体某些部位的加速度，而这些加速度需要由位移数据的两次求导得到。在热工业应用中，需要由测量的温度值估计出温度的变化率。

大家知道，用解析数学表达式表示的信号，如基础函数等，能够用数学方法求导。在大多数情况下，信号是没有数学表达式的，所以不能用数学方法直接求导，通常采用差分方法来近似地估计信号的导数。通常情况下，由于几乎所有的信号中都存在扰动，所以通过差分方法不能正确地估计出信号的速度。虽然卡尔曼滤波器可以被用来抑制扰动，同时求取信号的导数，但是需要有对象的模型，这限制了信号的类型。所以，构造含有滤波功能又不基于对象模型的微分器势在必行。



## 1.2 状态观测器

### 1.2.1 全状态观测器

在对一个系统进行控制时，除了利用输出反馈以外，主要利用状态反馈来配置极点。为了利用状态进行反馈，必须用传感器来测量状态变量，但不是所有状态变量在物理上都可以测量，于是提出用状态观测器给出状态估计的问题<sup>[43],[44]</sup>。针对各种问题出现了不同的观测器：时滞系统观测器<sup>[40]</sup>、针对离散系统的观测器<sup>[41]</sup>、针对系统中有扰动的系统状态观测器<sup>[42]</sup>、针对非线性非最小相位系统内模动态的观测器<sup>[46]</sup>等。这些观测器都是在系统模型已知的情况下设计的。

对于线性系统

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}\mathbf{x}\end{aligned}\quad (1.1)$$

其中， $\mathbf{x}$  为系统状态向量， $\mathbf{y}$  系统测量输出向量， $\mathbf{A}$ 、 $\mathbf{B}$ 、 $\mathbf{C}$  为相应维数的常数矩阵。因此基于上述系统的观测器设计为

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{K}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \quad (1.2)$$

实际上，令  $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ ，则式(1.2)和式(1.1)的误差系统为



$$\dot{e} = (A - KC)e \quad (1.3)$$

选择矩阵  $K$  使得  $A - BK$  是 Hurwitz (或者说矩阵  $A - BK$  的特征根的实部在  $s$  平面的左平面内), 那么可以得到:  $\hat{x}$  依照全局指数收敛方式收敛于  $x$ 。这种观测器也能基于位置估计速度信号, 但是这种估计是基于模型的。

**仿真实例:** 二阶线性全状态观测器

在系统式(1.1)中, 二阶系统为

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (1.4)$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [1 \quad 0]$$

那么, 状态观测器为

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \quad (1.5)$$

$$K = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

可得

$$A - KC = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} [1 \quad 0] = \begin{bmatrix} -k_1 & 1 \\ -k_2 & 0 \end{bmatrix} \quad (1.6)$$

所以, 特征方程为

$$|sI - (A - KC)| = \begin{vmatrix} s+k_1 & -1 \\ k_2 & s \end{vmatrix} = 0 \quad (1.7)$$

即

$$s^2 + k_1s + k_2 = 0 \quad (1.8)$$

选择

$$(s+3)(s+4) = 0 \quad (1.9)$$

即希望系统的特征根为  $\lambda_1 = -3$  和  $\lambda_2 = -4$ , 对照式(1.8)和式(1.9), 可设计  $k_1 = 7$ ,  $k_2 = 12$ 。

系统观测器仿真曲线如图 1.1 和图 1.2 所示。

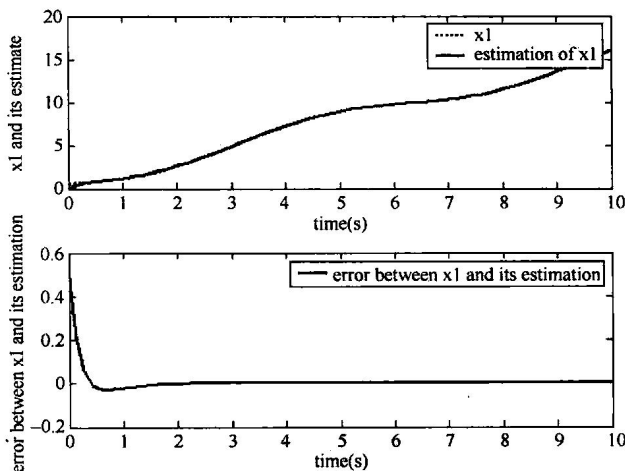
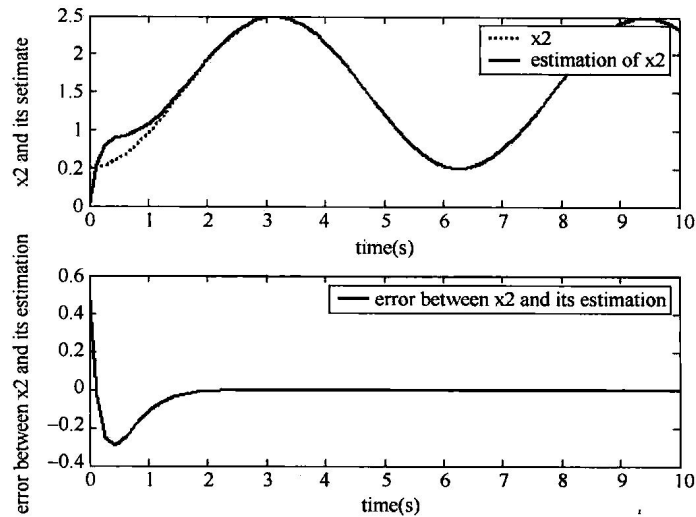


图 1.1 系统状态  $x_1$  和观测器变量  $\hat{x}_1$  及二者误差曲线

图 1.2 系统状态  $x_2$  和观测器变量  $\hat{x}_2$  及二者误差曲线

仿真程序: chap1\_1

观测器 Simulink 仿真主程序(如图 1.3 所示): chap1\_1sim.mdl

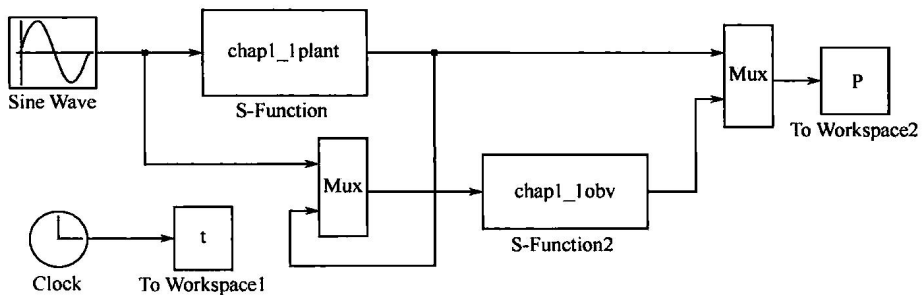


图 1.3 观测器 Simulink 仿真主程序

系统对象 S 函数: chap1\_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=2;
sizes.NumInputs=1;
sizes.DirFeedthrough=0;
sizes.NumSampleTimes=1;
sys=simsizes(sizes);
x0=[0.5 0.5];
str=[];
ts=[-1 0];
function sys=mdlDerivatives(t,x,u)
```

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

观测器 S 函数: chap1\_lobv.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)  
y=u(2);  
e=y-x(1);
```

```
k1=7;  
k2=12;  
sys(1)=x(2)+ k1* e;  
sys(2)=u(1)+ k2* e;  
function sys=mdlOutputs(t,x,u)  
sys(1)=x(1);  
sys(2)=x(2);
```

作图程序: chap1\_1plot.m

```
close all;
```

```
figure(1);  
subplot(211);  
plot(t,p(:,1),'k',t,p(:,3),'k','LineWidth',2);  
xlabel('time(s)');ylabel('x1 and its estimate');  
legend('x1','estimation of x1');  
subplot(212);  
plot(t,p(:,1)-p(:,3),'k','LineWidth',2);  
xlabel('time(s)');ylabel('error between x1 and its estimation');  
legend('error between x1 and its estimation');
```

```
figure(2);  
subplot(211);  
plot(t,p(:,2),'k',t,p(:,4),'k','LineWidth',2);  
xlabel('time(s)');ylabel('x2 and its estimate');  
legend('x2','estimation of x2');  
subplot(212);  
plot(t,p(:,2)-p(:,4),'k','LineWidth',2);  
xlabel('time(s)');ylabel('error between x2 and its estimate');  
legend('error between x2 and its estimation');
```

## 1.2.2 扩张观测器

对于系统

$$\begin{cases} \dot{x}_i = x_{i+1}, & i=1, \dots, n-1 \\ \dot{x}_n = \delta(t) + bu \\ y = x_1 \end{cases} \quad (1.10)$$



其中,  $x_1, \dots, x_n$  为系统状态,  $u \in \mathbf{R}$  为控制输入,  $y \in \mathbf{R}$  为系统测量输出,  $b$  为不等于零的常数,  $\delta(t)$  为系统的有界不确定项。

选取  $k_1, \dots, k_{n+1}$ , 使  $s^{n+1} + k_1 s^n + \dots + k_n s + k_{n+1} = 0$  的特征根实部均为负, 所设计的扩张观测器<sup>[37]</sup>为

$$\begin{cases} \dot{\hat{x}}_i = \hat{x}_{i+1} - \frac{k_i}{\epsilon^i} (\hat{x}_1 - y), & i=1, \dots, n-1 \\ \dot{\hat{x}}_n = \hat{x}_{n+1} - \frac{k_n}{\epsilon^n} (\hat{x}_1 - y) + bu \\ \dot{\hat{x}}_{n+1} = -\frac{k_{n+1}}{\epsilon^{n+1}} (\hat{x}_1 - y) \end{cases} \quad (1.11)$$

那么, 可得结论

$$\lim_{\epsilon \rightarrow 0} \hat{x}_i = x_i, \quad i=1, \dots, n \quad (1.12)$$

和

$$\lim_{\epsilon \rightarrow 0} \hat{x}_{n+1} = \delta(t) \quad (1.13)$$

**仿真实例: 二阶系统扩张观测器**

选取二阶不确定系统

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u + \delta(t) \\ y = x_1 \end{cases} \quad (1.14)$$

其中,  $\delta(t)$  为不确定项。设计的扩张观测器为

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 - \frac{k_1}{\epsilon} (\hat{x}_1 - y) \\ \dot{\hat{x}}_2 = \hat{x}_3 - \frac{k_2}{\epsilon^2} (\hat{x}_1 - y) + u \\ \dot{\hat{x}}_3 = -\frac{k_3}{\epsilon^3} (\hat{x}_1 - y) \end{cases} \quad (1.15)$$

参数选择为

$$k_1 = 6, k_2 = 11, k_3 = 6, \frac{1}{\epsilon} = R = \begin{cases} 100t^3, & 0 \leq t \leq 1 \\ 100, & t > 1 \end{cases} \quad (1.16)$$

其中,  $\frac{1}{\epsilon} = R$  的选择详见第 6 章(微分器峰值现象)。选择  $u = \sin t$ ,  $\delta(t) = 0.5 \sin t$ 。利用扩张观测器估计的状态及不确定项曲线如图 1.4、图 1.5 和图 1.6 所示。

仿真程序: chap1\_2

扩张观测器估计系统不确定项主程序(如图 1.7 所示): chap1\_2sim.mdl

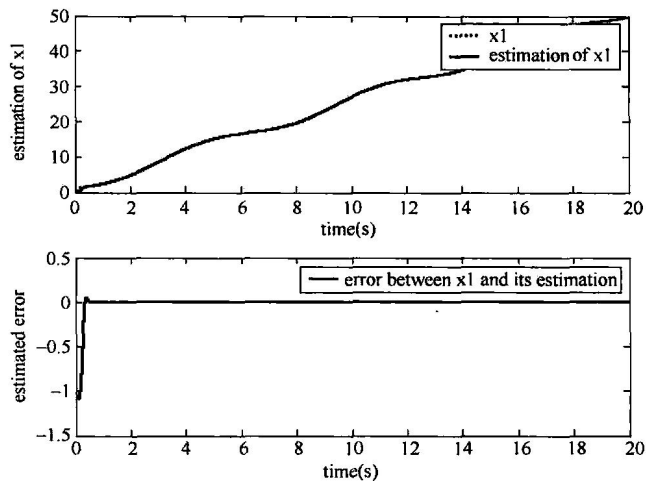


图 1.4 系统状态  $x_1$  和观测器变量  $\hat{x}_1$  及二者误差曲线

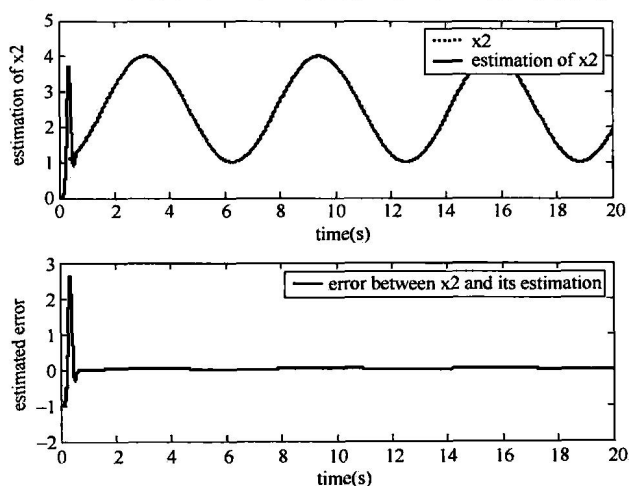


图 1.5 系统状态  $x_2$  和观测器变量  $\hat{x}_2$  及二者误差曲线

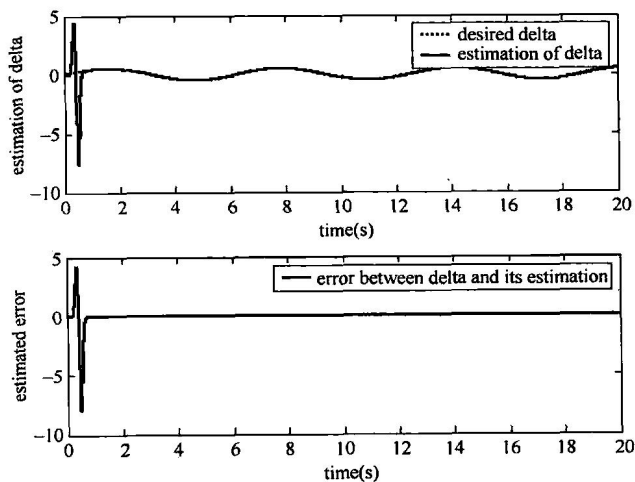


图 1.6 不确定项  $\delta$  和观测器变量  $\hat{x}_3$  及二者误差曲线

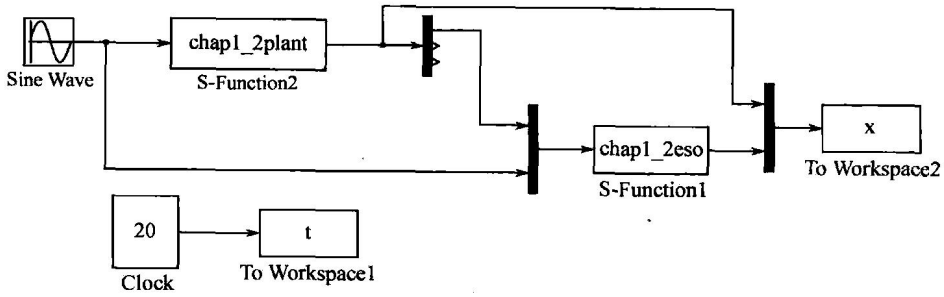


图 1.7 扩张观测器估计系统不确定项主程序

系统对象 S 函数: chapl\_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=[1 1];
str=[];
ts=[];
function sys=mdlDerivatives(t,x,u)
fx=0.5*sin(t);

sys(1)=x(2);
sys(2)=fx+u;
function sys=mdlOutputs(t,x,u)
fx=0.5*sin(t);
```





```
sys(1)=x(1);  
sys(2)=x(2);  
sys(3)=fx;
```

扩张观测器 S 函数: chap1\_2eso.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=3;  
sizes.NumDiscStates=0;  
sizes.NumOutputs=3;  
sizes.NumInputs=2;  
sizes.DirFeedthrough=0;  
sizes.NumSampleTimes=0;  
sys=simsizes(sizes);  
x0=[0 0 0];  
str=[];  
ts=[];  
function sys=mdlDerivatives(t,x,u)  
x1=u(1);  
ut=u(2);  
  
k1=6;  
k2=11;  
k3=6;  
e=x(1)-x1;  
  
tmax=1;  
if t<=tmax  
    R=100*t^3;
```