

第11章

高阶统计工具箱函数简介

高阶统计工具箱中存在大量处理非高斯信号的函数,为了掌握这些函数的基本用途,本章对这些函数进行归纳,给出每个函数的基本功能,具体的语法将在后面章节中使用时加以详细说明。

本章主要包括

- 高阶统计量非参数化估计函数
- 高阶统计量参数化估计函数
- 二次相位耦合函数
- 二阶 Volterra 系统函数
- 谐波恢复函数
- 时延估计函数
- 阵列信号处理-DCCA 函数
- 高阶时频分布函数

表 11.1 高阶统计量非参数化估计函数

函数名	功能说明
cum2x	估计互协方差
cum3x	估计三阶互累积量
cum4x	估计四阶互累积量
cumest	估计累积量
bicoher	估计双相干系数(直接方法)
bicoherx	估计互双相干系数(直接方法)
bispecd	估计双谱(直接方法)
bispecdx	估计互双谱(直接方法)
bispeci	估计双谱(间接方法)
glstat	高斯-线性检测量

表 11.2 高阶统计量参数化估计函数

函数名	功能说明
armaqs	利用 q 切片算法估计 ARMA 参数
armarts	利用残差时间序列算法估计 ARMA 参数
armasyn	产生 ARMA 合成信号
arorder	估计 AR 模型的阶次
arcest	利用相关函数或累积量估计 AR 参数
bispect	ARMA 过程的理论双谱
cumtrue	计算 ARMA 过程的理论累积量
maest	估计 MA 参数
maorder	估计 MA 的阶次
rpud	产生一独立同分布的序列
trispect	估计 ARMA 过程三谱的 2 维切片

表 11.3 二次相位耦合函数

函数名	功能说明
qpcgen	产生含噪的二次相位耦合谐波信号
qpctor	利用 TOR 方法检测二次相位耦合

表 11.4 二阶 Volterra 系统函数

函数名	功能说明
nngen	计算二阶 Volterra 系统的输出
nlpow	利用 Power 方法估计二阶 Volterra 系统的参数
nltick	利用 Tick 方法二阶 Volterra 系统的参数

表 11.5 谐波恢复函数

函数名	功能说明
harmest	估计谐波的频率
hargen	产生含高斯噪声的谐波

表 11.6 时延估计函数

函数名	功能说明
tde	利用三阶互累积量的时延估计
tdeb	利用三阶互双谱的时延估计
tdegen	产生时延估计的合成信号
tder	使用互相关的时延估计

表 11.7

阵列信号处理- DOA 函数

函数名	功能说明
doa	估计源的数目
doagen	产生用于 DOA 问题的合成信号

表 11.8

高阶时频分布函数

函数名	功能说明
wig2	Wigner-Ville 分布
wig2c	Choi-Williams 分布
wig3	Wigner 双谱的主对角切片
wig3c	加核函数的 Wigner 双谱的主对角切片
wig4	Wigner 三谱的主对角切片
wig4c	加核函数的 Wigner 三谱的主对角切片

第12章

高阶统计量的估计

高阶统计量的估计是其应用的基础,和功率谱分析方法一样,它的估计算法也分为非参数化方法和参数化方法两大类.

本章的内容主要包括:

- 高阶累积量的估计
- 双谱的估计

12.1 引言

高阶统计量估计的研究和功率谱估计的研究几乎是同时进行的,它的发展大致可分为两个阶段:第一阶段是从20世纪六七十年代开始,一些学者从数理统计的角度研究了高阶统计量的性质以及非参数化估计方法.当时由于缺乏足够的理论基础,无法完整地理解高阶累积量和高阶谱的物理意义;同时,由于在短数据下高阶统计量的估计与自相关函数的估计相比,方差很大,因此高阶统计量没有引起人们的重视.第二阶段从20世纪80年代开始,随着功率谱估计方法的不断发展,特别是参数化功率谱估计理论的日益完善以及高速计算机的迅速发展,很多学者又重新开始了高阶谱估计理论的研究,在参数化高阶谱估计等方面做了大量的工作.

从20世纪80年代以来,有关高阶谱估计方法的文献时有发表,高阶谱估计方法在非最小相位系统辨识、非线性系统辨识、谐波恢复与检测、生物医学信号处理、雷达与声纳信号处理等领域都得到了广泛的应用.在实际应用中,人们无法知道数据的真实的各阶统计量,只能得到统计量的一致估计.目前常用的估计方法包括非参数化和参数化方法两大类.但是在实际中常用的高阶统计量分析工具是高阶累积量及其对应的谱,故本章主要介绍高阶累积量和高阶谱的估计.

12.2 高阶累积量的估计

实际中,我们往往根据有限样本来估计累积量和高阶谱,因此估计的结果只能是一种统计意义上的近似,它的性能由偏差和方差来刻画.令 s 代表某个累积量,利用 N 个观测样本估计的结果记为 \hat{s}_N ,则估计值的偏差为 $E\{\hat{s}_N\}-s$,如果偏差为零,则称估计为无偏的;实际中仅仅当 $N \rightarrow \infty$ 才满足 $E\{\hat{s}_N\}=s$,称为渐进无偏的.仅利用偏差还不能完全刻画估计的性能,估计方差是另一种评价指标,为 $E\{|\hat{s}_N - s|^2\}$,如果当 $N \rightarrow \infty$ 时方差为零,则称估计是渐进一致的.

在累积量的估计过程中,为了得到好的估计结果,常常利用分段平滑以及加窗等方法.高阶统计分析工具箱中提供了估计二阶、三阶和四阶累积量的函数,但是在应用这些函数时假设分析信号的均值为零.

12.2.1 二阶累积量估计

二阶累积量的估计式为:

$$C_{xy}(k) = \frac{1}{N} \sum_{n=N_1}^{N_2} x^*(n)y(n+k), \quad (12.2.1)$$

其中 N_1, N_2 的取值与设定的最大延迟量 maxlag 有关,其计算式为

$$\begin{cases} 1 \leq n \leq N \\ 1 \leq n+k \leq N \end{cases} \Rightarrow \max\{1, 1-k\} \leq n \leq \min\{N-k, N\}. \quad (12.2.2)$$

高阶统计分析工具箱中估计二阶累积量的函数为 cum2x.m 和 cum2est.m.

函数名: cum2x

语法: y_cum=cum2x(x,y, maxlag, nsamp, overlap, flag)

说明: 估计 x,y 的互二阶累积量(当 x=y 时,估计 x 的二阶累积量)

输入参数: x,y 是分析的数据向量或矩阵,且维数相同; maxlag 是估计的最大延迟量; nsamp 是每一段的样本数; overlap 是重叠百分比; flag 为 'biased' 时表示有偏估计,为 'unbiased' 时表示无偏估计.

输出参数: y_cum 返回估计的互二阶累积量

参考函数: cum2est, cum3x, cum3est, cum4x, cum4est, cumest

函数名: cum2est

语法: y_cum=cum2est(y, maxlag, nsamp, overlap, flag)

说明: 估计 y 的二阶累积量

输入参数: y 是分析的数据列向量; maxlag 是估计的最大延迟量; nsamp 是每一段的样本数; overlap 是重叠百分比; flag 为 'biased' 时表示有偏估计,为 'unbiased' 时表示无偏估计.

输出参数: y_cum 返回估计的二阶累积量.

参考函数: cum2x, cum3x, cum3est, cum4x, cum4est, cumest

例程 12-1 估计非高斯 ARMA(1,3)过程的二阶累积量.

f12_1.m

%生成非高斯 ARMA 仿真信号

rand('seed',0);

randn('seed',0);

%%非高斯输入

u=rpiid(1024,'exp');

%%最大延迟量

maxlag=20;

%%ARMA(1,3)过程

sig=filter([1,-2],[1,-1.5,0.8,-0.2],u);

%显示 ARMA 过程信号

figure(1);

```

plot(1:length(sig),sig);
xlabel('n');
ylabel('ARMA(1,3)信号幅值')
%估计二阶累积量
%%参数设置
nsamp=128;
overlap=0;
flag='biased';
sigcum=cum2x(sig,sig,maxlag,nsamp,overlap,flag);
%%显示二阶累积量
figure(2);
subplot(2,1,1);
plot((1:length(sigcum))-maxlag-1,sigcum);
xlabel('延迟量 k');
ylabel('二阶累积量幅值');
%利用 cum2est 函数
sigcum=cum2est(sig,maxlag,nsamp,overlap,flag);
%%显示二阶累积量
subplot(2,1,2);
plot((1:length(sigcum))-maxlag-1,sigcum);
xlabel('延迟量 k');
ylabel('二阶累积量幅值');

```

首先,产生的 ARMA(1,3)过程的信号波形如图 12.2.1 所示.

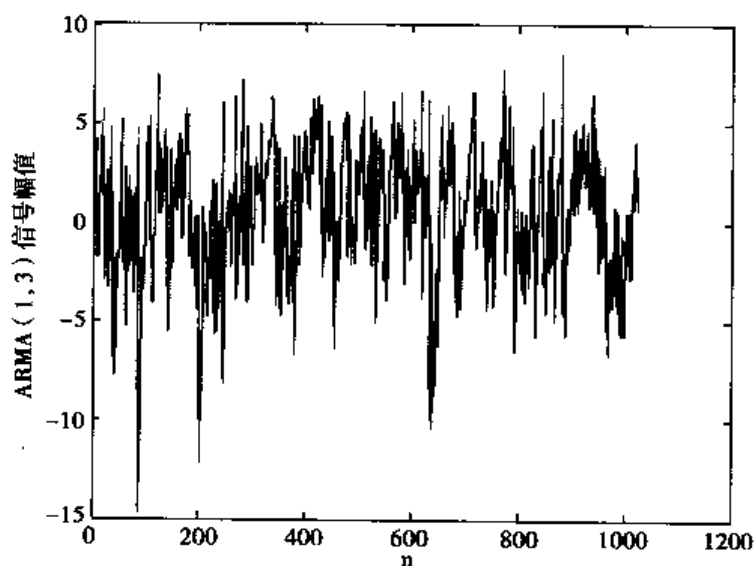


图 12.2.1 ARMA(1,3)过程的信号波形

然后,利用两种不同的函数估计出二阶累积量如图 12.2.2 所示,可见两种不同途径的计算结果是一致的.

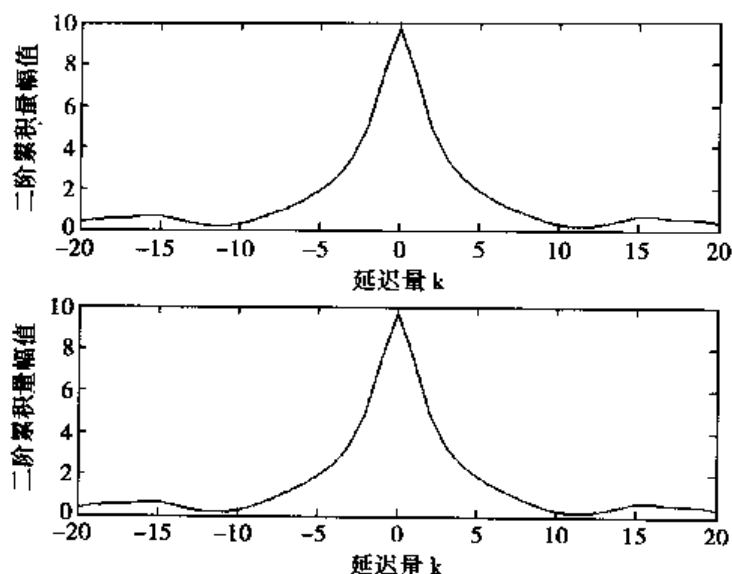


图 12.2.2 ARMA(1,3)过程的二阶累积量

12.2.2 三阶累积量估计

三阶累积量的估计式为:

$$C_{xyz}(k, l) = \frac{1}{N} \sum_{n=N_1}^{N_2} x'(n)y(n+k)z(n+l), \quad (12.2.3)$$

其中 N_1, N_2 的取值与设定的最大延迟量 maxlag 有关, 其计算式为

$$\begin{cases} 1 \leq n \leq N, \\ 1 \leq n+k \leq N \Rightarrow \max\{1, 1-k, 1-l\} \leq n \leq \min\{N-k, N-l, N\}, \\ 1 \leq n+l \leq N, \end{cases} \quad (12.2.4)$$

高阶统计分析工具箱中估计三阶累积量的函数为 cum3x.m 和 cum3est.m.

函数名: cum3x

语法: y_cum=cum3x(x,y,z, maxlag, nsamp, overlap, flag, k1)

说明: 估计 x,y,z 的互三阶累积量(当 x=y=z 时, 估计 x 的三阶累积量)

输入参数: x,y,z 是分析的数据向量或矩阵, 且维数相同; maxlag 是估计的最大延迟量; nsamp 是每一段的样本数; overlap 是重叠百分比; flag 为 'biased' 表示有偏估计, 为 'unbiased' 表示无偏估计; k1 是固定的延迟量 $c_3(m, k_1)$.

输出参数: y_cum 返回估计的互三阶累积量

参考函数: cum2x, cum2est, cum3est, cum4x, cum4est, cumest

函数名: cum3est

语法: y_cum=cum3est(y, maxlag, nsamp, overlap, flag, k1)

说明: 估计 y 的三阶累积量

输入参数: y 是分析的数据列向量; maxlag 是估计的最大延迟量; nsamp 是每一段的样本数; overlap 是重叠百分比; flag 为 'biased' 表示有偏估计, 为 'unbiased' 表示无偏估计; k1 是固定的延迟量 $c_3(m, k_1)$.

输出参数: y_cum 返回估计的三阶累积量

参考函数: cum2x, cum2est, cum3x, cum4x, cum4est, cumest

例程 12-2 估计非高斯 ARMA(1,2)过程的三阶累积量.

f12_2.m

```
%生成非高斯 ARMA 仿真信号
rand('seed',0);
randn('seed',0);
%%非高斯输入
u=rpiid(1024,'exp');
%%最大延迟量
maxlag=25;
%%ARMA(1,2)过程
sig=filter([1,-2],[1,-1.5,0.8],u);
%显示 ARMA 过程信号
figure(1);
plot(1:length(sig),sig);
xlabel('n');
ylabel('ARMA(1,2)信号幅值')
%估计三阶累积量
%%参数设置
nsamp=128;
overlap=0;
flag='biased';
for k=-maxlag:maxlag
    sig3cum(:,k+maxlag+1)=cum3x(sig,sig,sig,maxlag,nsamp,overlap,flag,k);
end
%%显示三阶累积量
figure(2);
subplot(2,2,1);
%%等高线图
contour(-maxlag:maxlag,-maxlag:maxlag,sig3cum);
xlabel('延迟量 k');
ylabel('延迟量 l');
subplot(2,2,2);
%%三维图
mesb(-maxlag:maxlag,-maxlag:maxlag,sig3cum);
xlabel('延迟量 k');
ylabel('延迟量 l');
%利用 cum3est 函数
for k=-maxlag:maxlag
    sig3cum(:,k+maxlag+1)=cum3est(sig,maxlag,nsamp,overlap,flag,k);
```



```

end
%%显示三阶累积量
subplot(2,2,3);
%%%等高线图
contour(-maxlag:maxlag,-maxlag:maxlag,sig3cum);
xlabel('延迟量 k');
ylabel('延迟量 l');
subplot(2,2,4);
%%%三维图
mesh(-maxlag:maxlag,-maxlag:maxlag,sig3cum);
xlabel('延迟量 k');
ylabel('延迟量 l');

```

首先,生成的 ARMA(1,2)过程的信号波形如图 12.2.3 所示。

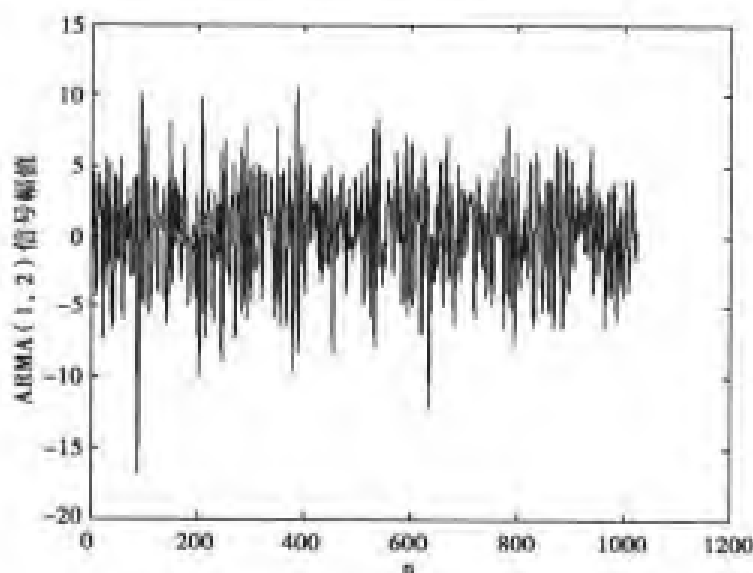


图 12.2.3 ARMA(1,2)过程的信号波形

然后,利用两种不同的函数估计出三阶累积量的等高线图和三维图如图 12.2.4 所示,可见两种不同途径的计算结果是一致的。

12.2.3 四阶累积量估计

四阶累积量的估计式为:

$$\begin{aligned} \hat{C}_{xyyz}(k, l, m) = & \frac{1}{N} \sum_{n=N_1}^{N_2} w^*(n) x(n+k) y(n+l) z^*(n+m) - \hat{C}_{xy}(k) \hat{C}_{yz}(l-m) - \\ & \hat{C}_{xy}(l) \hat{C}_{yz}(k-m) - \hat{C}_{xz}(m) \hat{C}_{xy}(l-k), \end{aligned} \quad (12.2.5)$$

其中 N_1, N_2 的取值应该保证 $x(n)$ 的求和在区间 $[1, N]$ 上。

由于四阶累积量是四维函数,因此利用常用的方法不能在三维坐标系中进行显示,为此我们使用空间的球来表示四阶累积量,其中球的中心坐标对应三个延迟量,而球的半径对应四阶累积量的大小。

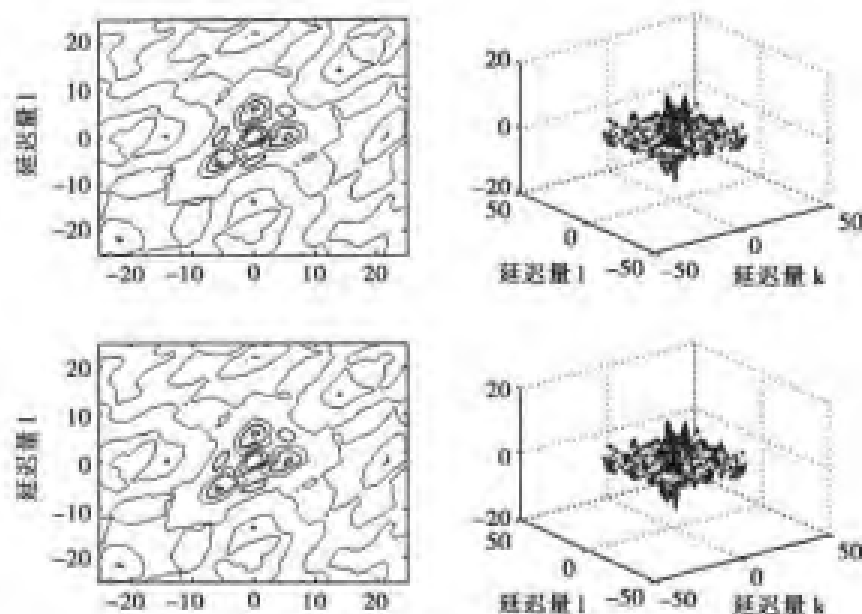


图 12.2.4 ARMA(1,2)过程的三阶累积量

高阶谱分析工具箱中估计协方差的函数为 `cum4x.m` 和 `cum4est.m`.

函数名: `cum4x`

语法: `y_cum=cum4x(w,x,y,z,maxlag,nsamp,overlap,flag,k1,k2)`

说明: 估计 w, x, y, z 的互四阶累积量(当 $w=x=y=z$ 时, 估计 x 的四阶累积量)

输入参数: w, x, y, z 是分析的数据向量或矩阵, 且维数相同; `maxlag` 是估计的最大延迟量; `nsamp` 是每一段的样本数; `overlap` 是重叠百分比; `flag` 为 'biased' 表示有偏估计, 为 'unbiased' 表示无偏估计; k_1, k_2 是固定的延迟量 $c_4(m, k_1, k_2)$.

输出参数: `y_cum` 返回估计的互四阶累积量

参考函数: `cum2x`, `cum2est`, `cum3est`, `cum3x`, `cum4est`, `cumest`

函数名: `cum4est`

语法: `y_cum=cum4est(y,maxlag,nsamp,overlap,flag,k1,k2)`

说明: 估计 y 的四阶累积量

输入参数: y 是分析的数据列向量; `maxlag` 是估计的最大延迟量; `nsamp` 是每一段的样本数; `overlap` 是重叠百分比; `flag` 为 'biased' 表示有偏估计, 为 'unbiased' 表示无偏估计; k_1, k_2 是固定的延迟量 $c_4(m, k_1, k_2)$.

输出参数: `y_cum` 返回估计的四阶累积量

参考函数: `cum2x`, `cum2est`, `cum3est`, `cum3x`, `cum4x`, `cumest`

例程 12-3 估计非高斯 ARMA(1,2) 过程的四阶累积量.

f12_3.m

%生成非高斯 ARMA 仿真信号

`rand('seed',0);`

`randn('seed',0);`

```

%%非高斯输入
u=rpiid(1024,'exp');
%%最大延迟量
maxlag=25;
%%ARMA(1,2)过程
sig=filter([1,-2],[1,-1.5,1],u);
%显示 ARMA 过程信号
figure(1);
plot(1:length(sig),sig);
xlabel('n');
ylabel('ARMA(1,2)信号幅值')
%估计四阶累积量
%%参数设置
nsamp=128;
overlap=0;
flag='biascd';
k1=10;
k2=10;
sig4cum=cum4x(sig,sig,sig,sig,maxlag,nsamp,overlap,flag,k1,k2);
r=sig4cum(5);
%举例显示一个球
figure(2);
subplot(2,2,1);
ellipsoid(5-maxlag-1,k1,k2,r,r,r);
%利用 cum4est 函数
sig4cum=cum4est(sig,maxlag,nsamp,overlap,flag,k1,k2);
r=sig4cum(5);
%举例显示一个球
subplot(2,2,2);
ellipsoid(5-maxlag-1,k1,k2,r,r,r);

```

首先,生成的 ARMA(1,2)过程的信号波形如图 12.2.5 所示.

然后,以一个点为例,分别计算了两种不同函数的四阶累积量估计结果,如图 12.2.6 所示,可见两者也是一致的.

12.3 双谱的估计

高阶谱的估计是功率谱估计的一种自然扩展,重要的区别在于对平滑的不同要求.传统的功率谱估计方法主要分为两大类:非参数化方法和参数化方法.第一类方法又分为两种:直接方法(基于观测数据的傅立叶变换)和间接方法(基于观测数据的自相关函数的傅立叶变换).第二类方法包括 MA、AR 和 ARMA 算法等.

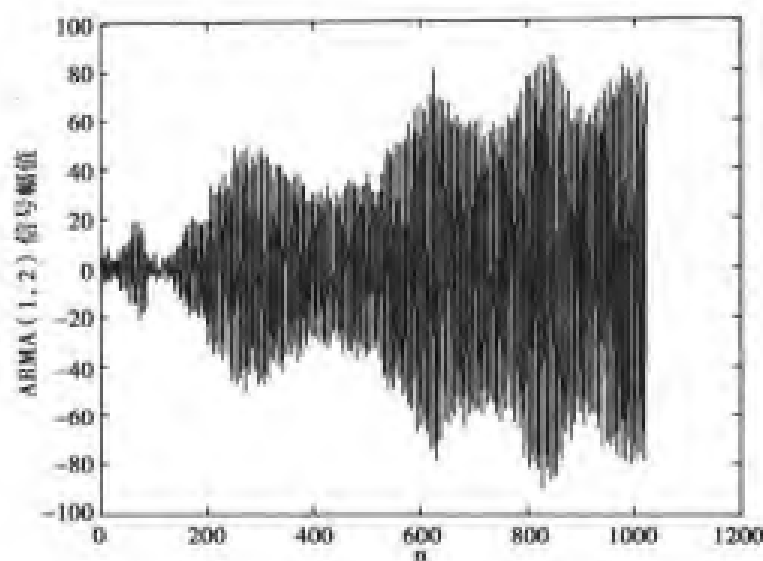


图 12.2.5 ARMA(1,2)过程的信号波形

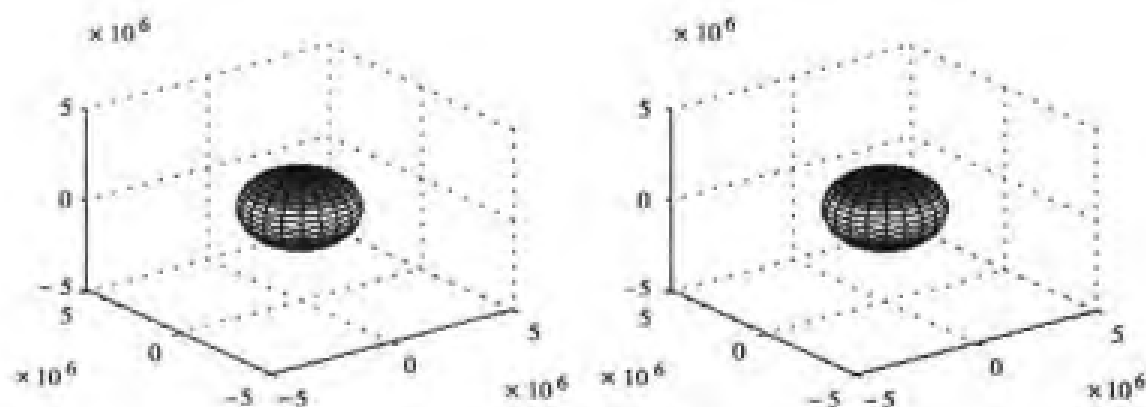


图 12.2.6 ARMA(1,2)过程的四阶累积量(一个点)

同样,高阶谱的估计也存在上述三类方法,下面介绍双谱和三谱的估计.

12.3.1 非参数化方法

12.3.1.1 直接方法

双谱的直接估计式为

$$B_{x,y,z}(f_1, f_2) = \frac{1}{N^2} X_N^*(f_1 + f_2) X_N(f_1) X_N(f_2). \quad (12.3.1)$$

高阶统计分析工具箱中双谱的直接估计函数为 `bispecd.m` 和 `bispecdx.m`.

函数名: `bispecd`

语法: `[Bspec, waxis] = bispecd(y, nfft, wind, nsamp, overlap)`

说明: 双谱的估计(直接方法)

输入参数: y 是数据向量或时间序列; $nfft$ 是傅立叶变换的长度; $wind$ 是频率平滑窗; $nsamp$ 是每一段的样本数; $overlap$ 是重叠数, 默认为 50.

输出参数: $Bspec$ 返回估计的双谱, 大小为 $nfft \times nfft$, 初始位置在矩阵中心; $waxis$ 返回频率坐标, 采样频率假设为 1.

参考函数: `bicoher`, `bicoherx`, `bispecdx`, `bispeci`

函数名: `bispecdx`

语法: `[Bspec, waxis] = bispecdx(x, y, z, nfft, wind, nsamp, overlap, plotflag)`

说明: 互双谱的估计(直接方法), 当 $x=y=z$ 时, 估计 x 的双谱

输入参数: x, y, z 是数据向量或时间序列, 且维数相同; $nfft$ 是傅立叶变换的长度; $wind$ 是频率平滑窗; $nsamp$ 是每一段的样本数; $overlap$ 是重叠数, 默认为 50; $plotflag=0$ 不显示互双谱, 默认为 1.

输出参数: $Bspec$ 返回估计的双谱, 大小为 $nfft \times nfft$, 初始位置在矩阵中心; $waxis$ 返回频率坐标, 采样频率假设为 1.

参考函数: `bicoher`, `bicoherx`, `bispecdx`, `bispeci`

例程 12-4 利用直接方法估计二次相位耦合信号的双谱.

f12_4.m

%生成二次相位耦合信号

`load qpc;`

`sig = zmat;`

`[M, N] = size(sig);`

%显示信号波形

`figure(1);`

`plot(sig(1 : M * N));`

`xlabel('n');`

`ylabel('QPC 信号');`

%估计双谱

`nfft = 128;`

`wind = 3;`

`nsamp = 64;`

`overlap = 50;`

`[b1, waxis] = bispecd(sig, nfft, wind, nsamp, overlap);`

%显示

`figure(2);`

`subplot(2, 2, 1)`

%%等高线

`contour(waxis, waxis, abs(b1));`

`xlabel('f1');`

`ylabel('f2');`

`subplot(2, 2, 2)`

%%三维图

```

mesh(waxis,waxis,abs(b1));
xlabel('f1');
ylabel('f2');
%利用互双谱
plotflag=0;
[b2,waxis]=bispecdx(sig,sig,sig,nfft,wind,nsamp,overlap,plotflag);
%显示
subplot(2,2,3)
%%等高线
contour(waxis,waxis,abs(b2));
xlabel('f1');
ylabel('f2');
subplot(2,2,4)
%%三维图
mesh(waxis,waxis,abs(b2));
xlabel('f1');
ylabel('f2');

```

生成的二次相位耦合信号波形如图 12.3.1 所示,该信号在三个频率 0.10Hz、0.15Hz 和 0.25Hz 之间存在二次相位耦合现象,另外还有一个独立频率 0.40Hz.

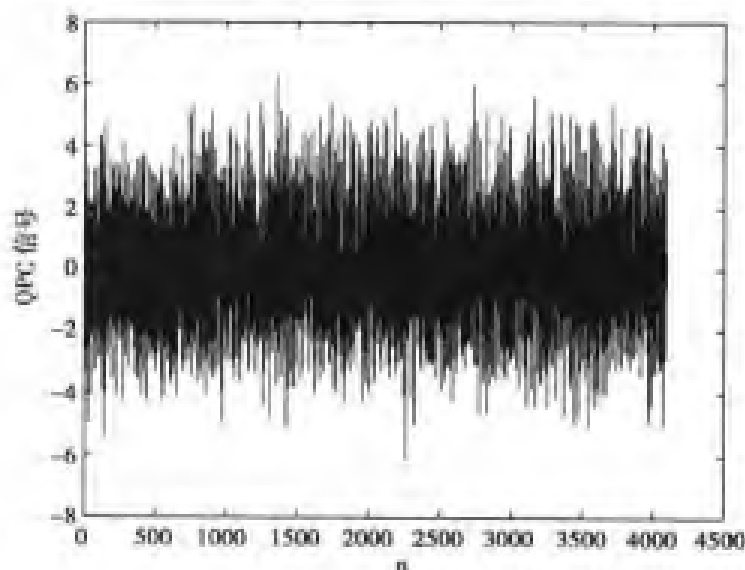


图 12.3.1 二次相位耦合信号的波形

利用直接方法估计的双谱如图 12.3.2 所示,能明显看出在频率对 (0.10, 0.15) 处存在峰值,从而揭示了二次相位耦合现象的位置,另外由对称性能得到另外 11 个峰值位置.

12.3.1.2 间接方法

双谱的间接估计式为三阶累积量的二维傅立叶变换:

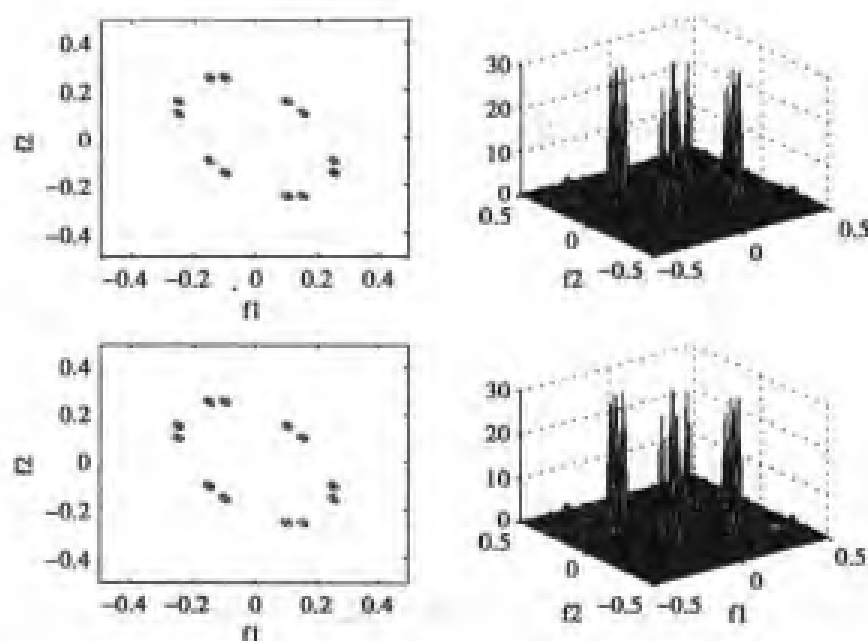


图 12.3.2 二次相位耦合信号的双谱估计(直接方法)

$$B_{x,y,z}(f_1, f_2) = \sum_{k=-N-1}^{N-1} \sum_{l=-N-1}^{N-1} \hat{C}_{xy}(k, l) e^{-j2\pi(f_1 k + f_2 l)}, \quad (12.3.2)$$

高阶统计分析工具箱中双谱的间接估计函数为 `bispecdi.m`。

函数名: `bispecdi`

语法: `[Bspec, waxis] = bispecdi(y, nlag, nsamp, overlap, flag, nfft, wind)`

说明: 双谱的估计(间接方法)

输入参数: `y` 是数据向量或时间序列; `nlag` 是延迟量的数目; `nsamp` 是每一段的样本数; `overlap` 是重叠数, 默认为 50; `flag` = 'biased' 或 'unbiased' (默认); `nfft` 是傅立叶变换的长度, 默认为 128; `wind` 是窗函数, 默认时 `wind` = 0, 窗函数为 Pazen 窗。

输出参数: `Bspec` 返回估计的双谱, 大小为 `nfft` × `nfft`, 初始位置在矩阵中心; `waxis` 返回频率坐标, 采样频率假设为 1。

参考函数: `bicoher`, `bicoherx`, `bispecdx`, `bispecd`

例程 12-5 利用间接方法估计二次相位耦合信号的双谱。

f12_5.m

%生成二次相位耦合信号

load qpc;

sig=zmat;

[M,N]=size(sig);

%估计双谱

%%设置参数

nfft=128;

wind=1;

nsamp=64;

```

overlap=50;
nlag=49;
flag='unbiased';
[b1,waxis]=bispeci(sig,nlag,nsamp,overlap,flag,nfft,wind);
%显示
subplot(2,2,1)
contour(waxis,waxis,abs(b1));
xlabel('f1');
ylabel('f2');
subplot(2,2,2)
mesh(waxis,waxis,abs(b1));
xlabel('f1');
ylabel('f2');

```

利用间接方法估计的双谱如图 12.3.3 所示,也能看出在频率对(0.10,0.15)处存在峰值,从而揭示了二次相位耦合现象的位置,另外由对称性能得到另外 11 个峰值位置,但与直接方法相比,间接方法的估计性能要差一点。

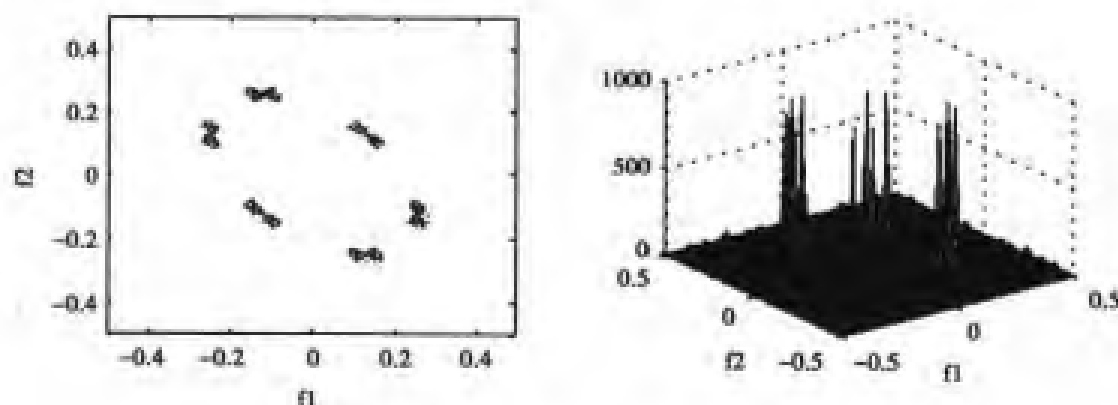


图 12.3.3 二次相位耦合信号的双谱估计(间接方法)

12.3.2 双相干系数的估计

双相干系数的估计式为:

$$b_{xyz} = \frac{B_{xyz}(f_1, f_2)}{\sqrt{S_{xx}(f_1 + f_2) S_{yy}(f_1) S_{zz}(f_2)}} \quad (12.3.3)$$

高阶统计分析工具箱中双相干系数的直接估计函数为 bicoher.m 和 bicoherx.m。

函数名:bicoher

语法:[bic,waxis]=bicoher(y, nfft, wind, nsamp, overlap)

说明:双相干系数的估计(直接方法)

输入参数:y 是数据向量或时间序列;nfft 是傅立叶变换的长度;wind 是时域窗函数;nsamp 是每一段的样本数;overlap 是重叠数,默认为 50。

输出参数:Bspec 返回估计的双相干系数,大小为 nfft×nfft,初始位置在矩阵中心;waxis

返回频率坐标,采样频率假设为 1.

参考函数:bicoherx, bispecd, bispecdx, bispeci

函数名:bicoherx

语法:[bic,waxis]=bicoherx(w,x,y,nfft,wind,nsamp,overlap)

说明:互双相干系数的估计(直接方法)

输入参数:w,x,y 是数据向量或时间序列;nfft 是傅立叶变换的长度;wind 是时域窗函数;nsamp 是每一段的样本数;overlap 是重叠数,默认为 50.

输出参数:Bspec 返回估计的互双相干系数,大小为 $nfft \times nfft$,初始位置在矩阵中心;waxis 返回频率坐标,采样频率假设为 1.

参考函数:bicoher, bispecd, bispecdx, bispeci

例程 12-6 估计二次相位耦合信号的双相干系数.

f12_6.m

```
%生成二次相位耦合信号
load qpc;
sig=zmat;
%估计双相干系数
%%设置参数
nfft=128;
[b1,waxis]=bicoher(sig,nfft);
%显示
subplot(2,2,1)
contour(waxis,waxis,abs(b1));
xlabel('f1');
ylabel('f2');
subplot(2,2,2)
mesh(waxis,waxis,abs(b1));
xlabel('f1');
ylabel('f2');
[b2,waxis]=bicoherx(sig,sig,sig,nfft);
%显示
subplot(2,2,3)
contour(waxis,waxis,abs(b2));
xlabel('f1');
ylabel('f2');
subplot(2,2,4)
mesh(waxis,waxis,abs(b2));
xlabel('f1');
ylabel('f2');
```

二次相位耦合信号的双相干系数估计结果如图 12.3.4 所示,比较可见两个不同函数估计的结果是相似的.

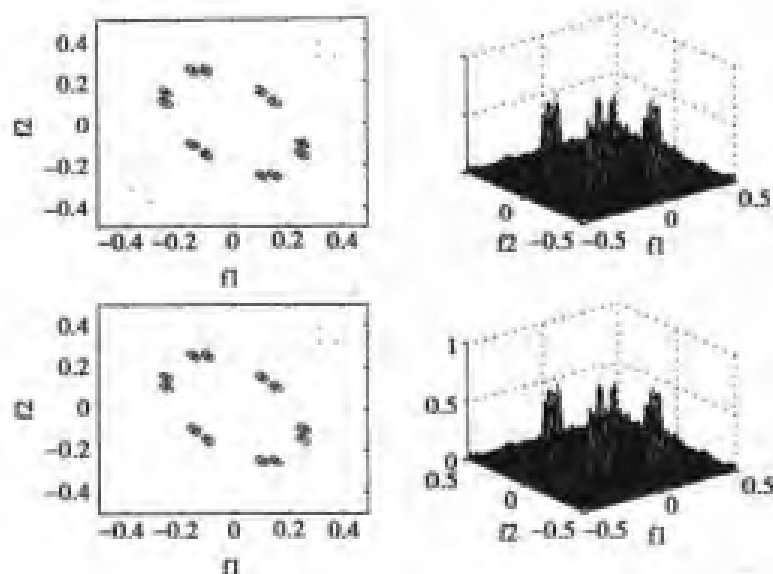


图 12.3.4 二次相位耦合信号的双相干系数估计结果

12.3.3 窗函数的设计

加窗是提高双谱估计质量的一个重要措施,选用合理的窗函数能降低估计方差和抑制噪声,但同时窗函数谱旁瓣的存在会产生能量的泄漏,出现假频,因此找到一个合适的二维窗是改善双谱估计性能亟待解决的一个实际问题。

双谱估计的二维窗函数最早是由 Sasaki、Sato 和 Yamashita 推导和讨论的,并证明了二维窗函数 $w(m, n)$ 必须满足以下四个约束条件:

(i) 满足三阶累积量的对称性质:

$$w(m, n) = w(n, m) = w(-m, n-m) = w(m-n, -n);$$

(ii) 若 (m, n) 位于累积量估计的支撑域外,则 $w(m, n) = 0$;

(iii) 原点处的滞后窗为 1, 即 $w(0, 0) = 1$;

(iv) $\forall (\omega_1, \omega_2), W(\omega_1, \omega_2) \geq 0$;

值得指出的是,满足上述四个条件的二维窗 $w(m, n)$ 可以由一维滞后窗函数 $d(m)$ 来构造:

$$w(m, n) = d(m)d(n)d(n-m). \quad (12.3.4)$$

因此在实际中,可以利用已有的一维窗函数构造出用于双谱估计的二维窗函数,下面以 Hanning-Poisson 组合窗为例,它的表达式为

$$w(n) = \frac{1}{2} \left[1 + \cos \left(\pi \cdot \frac{n}{M/2} \right) \right] e^{-\frac{|n|}{M/2}}, \quad n \in \left[-\frac{M}{2}, \frac{M}{2} - 1 \right]. \quad (12.3.5)$$

当 $a \geq 2$ 时,它是无旁瓣的.为了更好地控制窗的性质,通过对上式进行更新,重新定义了窗函数 $A_{(a,b)}(n)$,其思想是用无旁瓣但时域不受限的高斯函数与三角函数的幂相乘,通过控制参数 a 和 b ,可以使得旁瓣不存在,其表达式为:

$$A_{(a,b)}(n) = \left(1 - \frac{|n|}{M/2} \right)^a \cdot e^{-b \cdot \left(\frac{|n|}{M/2} \right)^2}, \quad n \in \left[-\frac{M}{2}, \frac{M}{2} - 1 \right]. \quad (12.3.6)$$

首先我们来观察 Hanning-Poisson 组合窗的时域和频域性质.

例程 12-7 Hanning-Poisson 组合窗的时域和频率图.

f12_7.m

```
M=1024;
t=1:M;
for i=-M/2:M/2-1;
    w(i+M/2+1)=hanning_poisson(M,i);
```

```
end
```

```
y=w;
```

```
fs=1;
```

```
nfft=256;
```

```
Y=fft(y,nfft);
```

```
P=Y.*conj(Y)/nfft;
```

```
P=20*log(P);
```

```
P=fftshift(P);
```

```
f=fs*(-128:127)/256;
```

```
%频域波形
```

```
subplot(2,2,1);
```

```
plot(f,P);
```

```
xlabel('归一化的频率');
```

```
ylabel('幅值(dB)');
```

```
%时域波形
```

```
subplot(2,2,2);
```

```
plot(2*t/M-1,y);
```

```
xlabel('归一化的时间');
```

```
ylabel('幅值');
```

以上程序中调用的 hanning_poisson 函数为

```
function y=hanning_poisson(M,n)
```

```
a=0.0001;
```

```
b=21.6;
```

```
y=(1-abs(n)/M)^a*exp(-b*(abs(n)/M)^2);
```

计算得到 Hanning-Poisson 组合窗的时域波形和功率谱如图 12.3.5 所示,由图可见,其频谱无旁瓣.

接下来,可以构造二维的 Hanning-Poisson 组合滞后窗:

$$w_{(a,b)}(m,n)=A_{(a,b)}(m)A_{(a,b)}(n)A_{(a,b)}(n-m) \quad (12.3.7)$$

最后,验证构造出的二维窗函数是否满足四个约束条件.

例程 12-8 观察二维 Hanning-Poisson 组合滞后窗的对称性.

f12_8.m

```
M=32;
```

```
k=M/2;
```

```
for m=-k:k
```

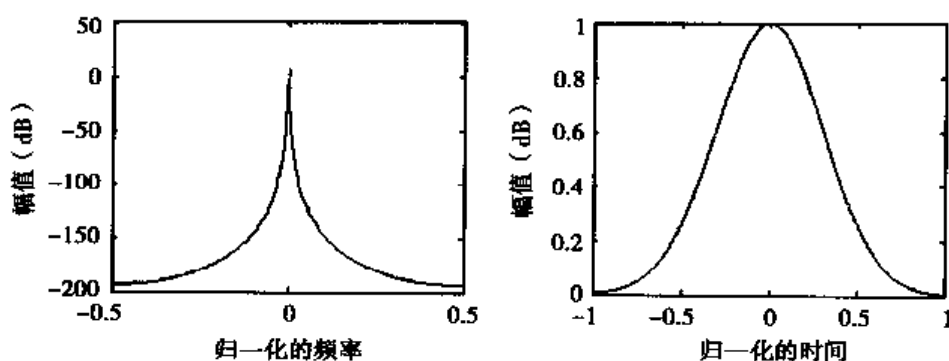


图 12.3.5 Hanning-Poisson 组合窗的时域和频率图

```

for n=-k:k
    yy(k+m+1,k+n+1)=two_hanning_poisson(M,m,n);
end
end
[x,y]=meshgrid(-k:k,-k:k);
figure(1);
subplot(2,2,1);
%三维图形
mesh(x,y,abs(yy));
xlabel('m');
ylabel('n');
zlabel('幅值');
%等高线图
subplot(2,2,2);
contour(x,y,abs(yy));
xlabel('m');
ylabel('n');
function y=two_hanning_poisson(M,m,n)
k=M/2;
y=hanning_poisson(k,m)*hanning_poisson(k,n)*hanning_poisson(k,n-m);

```

图 12.3.6 是其三维视图和等高线图,可以证明它满足上述四个约束条件. 同样通过控制参数 a 和 b 可以抑制二维 Hanning-Poisson 组合滞后窗的旁瓣.

12.3.4 $1\frac{1}{2}$ 维谱的估计

众所周知,高阶谱是由高阶累积量经过多维傅立叶变换得到的,计算量大并且耗时长,不利于在线应用. 为了使得高阶谱能够像功率谱那样应用 FFT,计算简单,许多学者一直都在努力寻找高阶谱的简化计算方法,使其在实际应用中发挥作用,但效果不尽如人意. 近些年,人们通过引入“降维”思想,将双谱投影到一维频率空间上,取得了一定的理论成果,这主要是通过

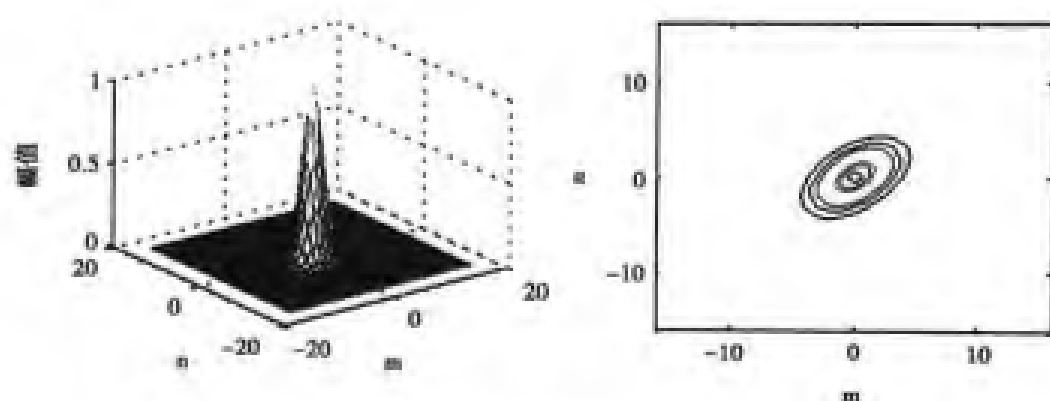


图 12.3.6 二维 Hanning-Poisson 组合滞后窗的空间形状

取对角切片的方式来实现的,即选择一组特殊的滞后 $\tau_1 = \cdots = \tau_{k-1} = \tau$,由此可以简化双谱。

零均值平稳随机过程 $x(n)$ 的三阶累积量和三阶矩相等,三阶累积量定义为:

$$c_{3x}(\tau_1, \tau_2) = E[x(n)x(n+\tau_1)x(n+\tau_2)]. \quad (12.3.8)$$

取 $\tau_1 = \tau_2$ 得到三阶累积量的主对角切片

$$c(\tau) = c_{3x}(\tau, \tau) = E[x(n)x(n+\tau)x(n+\tau)]. \quad (12.3.9)$$

对 $c(\tau)$ 进行一维傅立叶变换可以得到 $x(n)$ 的 $1\frac{1}{2}$ 维谱,即

$$SB(f) = \sum_{\tau=-\infty}^{\infty} c(\tau) e^{-j2\pi f\tau}. \quad (12.3.10)$$

$1\frac{1}{2}$ 维谱的定义首次是由 Giannakis 和 Mendel 提出的,它实际上是双谱在平面 $f_1 = f_2$ 上的投影。

下面通过实例说明 $1\frac{1}{2}$ 维谱的估计过程。

例程 12-9 估计下列仿真信号 $x(n)$ 的 $1\frac{1}{2}$ 维谱:

$$x(n) = \sum_{i=1}^6 A_i \cos(2\pi f_i n + \varphi_i),$$

其中 A_i ($i=1, \dots, 6$) 为确定的常数, φ_i ($i=1, \dots, 5$) 是 $[1, 2\pi]$ 上均匀分布的随机变量,取 $f_1 = 0.4909\text{Hz}$, $f_2 = 1.7671\text{Hz}$, $f_4 = 0.6381\text{Hz}$, $f_5 = 0.8345\text{Hz}$, $f_3 = f_1 + f_2$, $f_6 = f_4 + f_5$, $\varphi_6 = \varphi_4 + \varphi_5$ 。

fl2_9.m

%样点数

N=128*64;

%采样频率

fs=8;

%频率参数

f1=0.6381;

```

f2=0.8345;
f3=f1+f2;
f4=0.4909;
f5=1.7671;
f6=2;
%仿真信号
a=2*pi*rand(5);
x=cos(2*pi*f1/fs*(1:N)+a(1))+cos(2*pi*f2/fs*(1:N)+a(2))+cos(2*pi
*f3/fs*(1:N)+a(1)+a(2))+...
cos(2*pi*f4/fs*(1:N)+a(4))+cos(2*pi*f5/fs*(1:N)+a(5))+cos(2*pi
*f6/fs*(1:N)+a(4)+a(5));
%估计三阶累积量
K=64;
M=fix(N/K);
%分段,去均值
for i=0:K-1
    y=x(i*M+1:(i+1)*M);
    y=y-mean(y);
    xk(i+1,:)=y;
end
nfft=256;
%计算累积量
for t=-(nfft/2-1):nfft/2
    c(t+nfft/2)=third_slice_cumulant(xk,t);
end
%FFT 变换
Y=fft(c,nfft);
P=Y.*conj(Y)/nfft;
f=fs*(0:nfft/2-1)/nfft;
plot(f,P(1:nfft/2));
xlabel('频率');
ylabel('幅值');

```

上述程序中的 third_slice_cumulant 函数为

```

function y=third_slice_cumulant(x,t)
[K,M]=size(x);
s1=max([1,1-t]);
s2=min([M,M-t]);
for i=1:K
    templ=x(i,:);
    temp=sum(templ(s1:s2).*templ(s1+t:s2+t).*templ(s1+t:s2+t));

```

```

cc(i)=temp/M;
end
y=sum(cc)/K;

```

计算仿真信号 $x(n)$ 的 $1\frac{1}{2}$ 维谱如图 12.3.7 所示,由图可以看出, $1\frac{1}{2}$ 维谱是频率的一维函数,类似于功率谱,但是它只在 f_1 、 f_5 和 f_6 处有谱线存在,这一点与功率谱是不相同的,具体原因将在下一章进行说明.

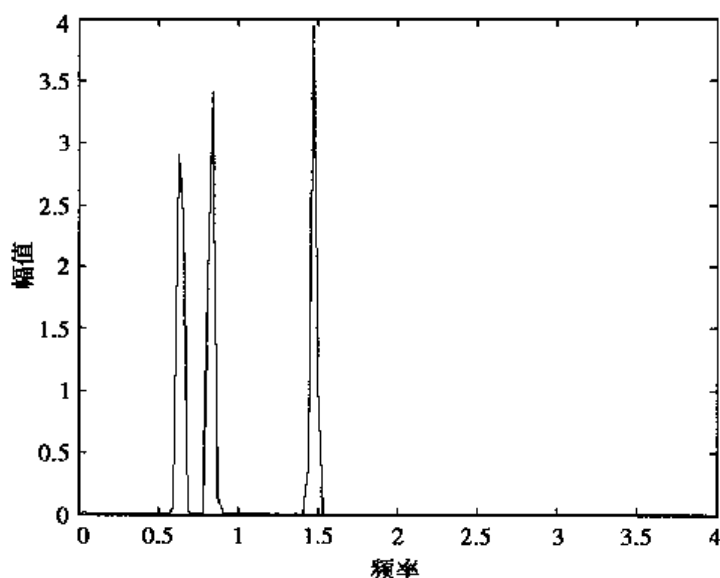


图 12.3.7 仿真信号的 $1\frac{1}{2}$ 维谱

12.3.5 参数化方法

以上介绍了双谱的非参数化估计方法,实际中参数化方法往往也是非常有用的.其基本思想是如果 $x(n)$ 依赖于一个有限的参数集 θ ,那么它的所有统计量可以用参数 θ 来表示.

参数 θ 和序列 $x(n)$ 之间的关系可以构成一个模型,常用的时间序列分析模型是自回归滑动平均模型 (ARMA),表达式为

$$x(n) = \sum_{k=1}^p a(k)x(n-k) + \sum_{k=0}^q b(k)u(n-k), \quad (12.3.11)$$

其中 $u(n)$ 假设为独立同分布的序列,方差为 σ^2 .

自回归多项式定义为

$$A(z) = \sum_{k=0}^p a(k)z^{-k}. \quad (12.3.12)$$

滑动平均多项式定义为

$$B(z) = \sum_{k=0}^q b(k)z^{-k}. \quad (12.3.13)$$

ARMA 过程的功率谱定义为

$$P_r(f) = \sigma^2 \frac{|B(z)|^2}{|A(z)|^2} \Big|_{z=e^{j2\pi f}}, \quad (12.3.14)$$

记 ARMA 过程的冲击响应函数为 $h(n)$, 其傅立叶变换为 $H(f)$, 则双谱的估计式为:

$$B(f_1, f_2) = \gamma_{3u} H(f_1) H(f_2) H^*(f_1 + f_2), \quad (12.3.15)$$

高阶统计分析工具箱中参数化估计累积量的函数为 cumtrue.m, 参数化估计双谱的函数为 bispect.m.

函数名: cumtrue

语法: cmat=cumtrue(ma, ar, norder, nlags, k)

说明: ARMA 模型累积量的理论值

输入参数: ma 是 MA 参数向量; ar 是 AR 参数向量; norder 是累积量的阶次, 为 2、3 或 4, 默认为 3; nlags 是计算累积量延迟的最大数目; 如果 norder=4, 则 k 表示累积量的第 3 个延迟, 默认为 0.

输出参数: cmat 返回计算的累积量向量或矩阵, 如果 norder=2, 则 cmat 是一个长度为 $2 * nlags + 1$ 的向量; 如果 norder=3, 则 cmat 是一个长度为 $(2 * nlags + 1) * (2 * nlags + 1)$ 的矩阵; 如果 norder=4, 则 cmat 是一个长度为 $(2 * nlags + 1) * (2 * nlags + 1)$ 的矩阵.

参考函数: bispect

函数名: bispect

语法: [Bspec, waxis]=bispect(ma, ar, nfft)

说明: ARMA 模型的理论双谱

输入参数: ma 是 MA 参数向量; ar 是 AR 参数向量, 默认为 [1]; nfft 是傅立叶变换的长度.

输出参数: Bspec 返回双谱; waxis 是频率坐标.

参考函数: cumtrue

下面通过实例说明双谱的参数化估计方法.

例程 12-10 估计 ARMA(2,1) 的三阶累积量.

f12_10.m

```
%ARMA 过程
ma=[1,-2];
ar=[1,-1.5,0.8];
%最大延迟量
maxlag=25;
%阶次
norder=3;
%估计三阶累积量
sig3cum=cumtrue(ma, ar, norder, maxlag);
%%显示三阶累积量
figure(1);
subplot(2,2,1);
contour(-maxlag:maxlag,-maxlag:maxlag,sig3cum);
xlabel('延迟量 k');
ylabel('延迟量 l');
subplot(2,2,2);
```



```
mesh(-maxlag:maxlag,-maxlag:maxlag,sig3cum);
xlabel('延迟量 k');
ylabel('延迟量 l');
```

利用参数化方法估计的三阶累积量的等高线图和三维图如图 12.3.8 所示,其结果与图 12.2.4 是类似的。

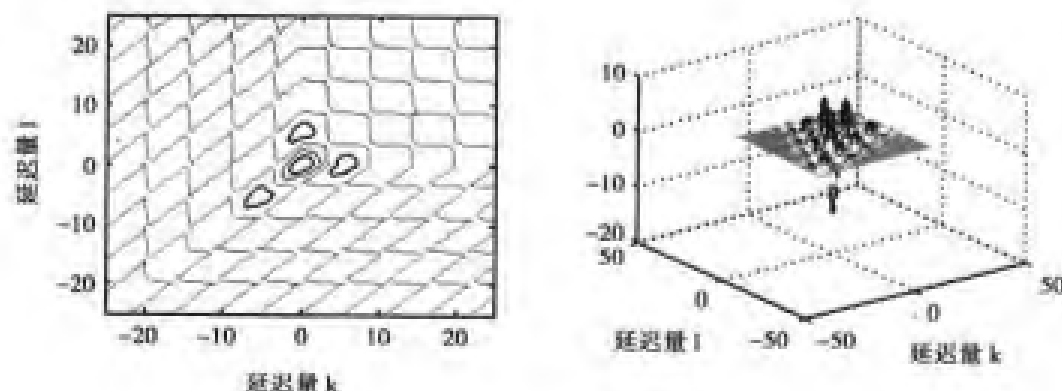


图 12.3.8 ARMA(2,1)过程的三阶累积量估计

进一步,利用参数化方法估计上述过程的双谱。

例程 12-11 估计 ARMA(2,1)的双谱。

f12_11.m

```
%ARMA 过程
ma=[1,-2];
ar=[1,-1.5,0.8];
%nfft
nfft=128;
%估计双谱
[Bspec,waxis]=bispect(ma,ar,nfft);
%%显示双谱
figure(1);
subplot(2,2,1);
contour(waxis,waxis,abs(Bspec));
xlabel('频率 f1');
ylabel('频率 f2');
subplot(2,2,2);
mesh(waxis,waxis,abs(Bspec));
xlabel('频率 f1');
ylabel('频率 f2');
```

计算得到的双谱的等高线图和三维图如图 12.3.9 所示。

提示:实际中利用参数化方法估计给定序列的累积量或高阶谱时,应该首先对其建立

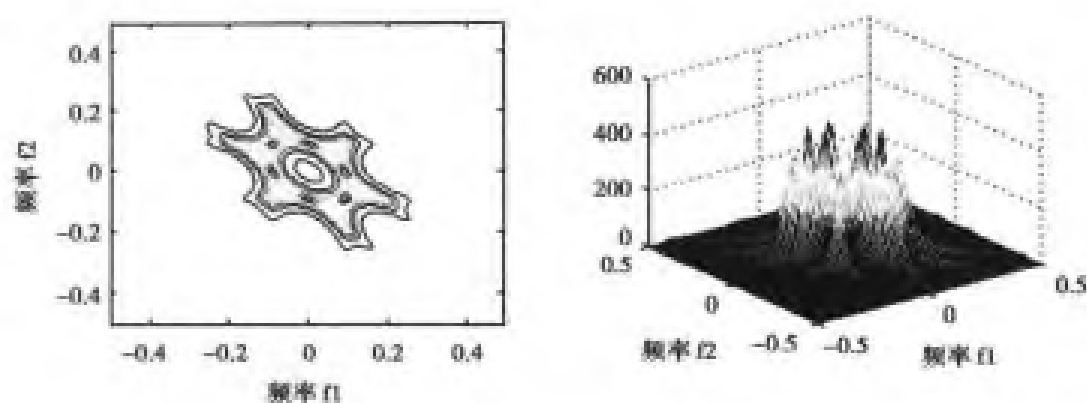


图 12.3.9 ARMA(2,1)过程的双谱估计

ARMA 模型,得到 AR 和 MA 参数后再利用高阶统计分析工具箱中的函数即可.关于 ARMA 模型的建立过程将在后面章节阐述.