



Translated by  Microsoft

鼠标悬停文本以查看原始文本。点击下面的按钮返回到页面的英文版本。

[Back to English](#)

dwt

单级一维离散小波变换

语法

```
[cA,cD] = dwt(x,wname)
[cA,cD] = dwt(x,LoD,HiD)
[cA,cD] = dwt( ___, 'mode',extmode)
```

描述

`[cA,cD] = dwt(x,wname)` 使用小波返回向量x的单级离散小波变换 (dwt)由wname指定。小波必须通过wavemngr识别。dwt返回 dwt 的近似系数向量cA和细节系数向量cD。

[例子](#)

注意

如果您的应用程序需要多级小波分解, 请考虑使用wavedec.

`[cA,cD] = dwt(x,LoD,HiD)` 使用缩放 (低通) 筛选器LoD级 dwt和小波 (高通) 滤波器HiD。筛选器的长度必须相同。

[例子](#)

`[cA,cD] = dwt(___, 'mode',extmode)`返回具有指定扩展模式extmode单级 dwt。有关详细信息, 请参阅 [dwtmode](#) .

例子

[崩溃所有](#)

使用小波名称的 dwt

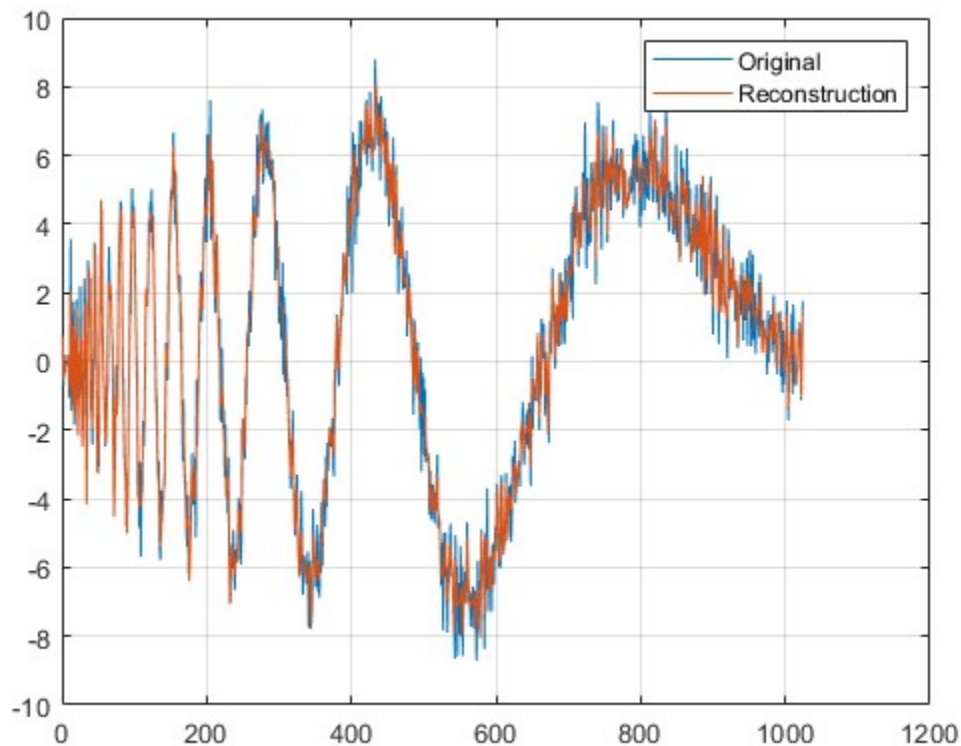
使用小波名称获取噪声多普勒信号的单级 dwt。

[在 matlab 中试用](#)

```
load noisdopp;
[cA,cD] = dwt(noisdopp,'sym4');
```

使用近似系数重建平滑版本的信号。绘制并与原始信号进行比较。

```
xrec = idwt(cA,zeros(size(cA)),'sym4');
plot(noisdopp)
hold on
grid on
plot(xrec)
legend('Original','Reconstruction')
```



使用小波和缩放滤波器的 dwt

使用小波 (高通) 和缩放 (低通) 滤波器获取噪声多普勒信号的单级 dwt。

在 matlab 中试用

```
load noisdopp;
[LoD,HiD] = wfilters('bior3.5','d');
[cA,cD] = dwt(noisdopp,LoD,HiD);
```

创建一个 dwt 滤波器库, 该滤波器库可应用于使用相同的小波的噪声多普勒信号。从过滤库中获取高通和低通滤波器。

```
len = length(noisdopp);
fb = dwtfilterbank('SignalLength',len,'Wavelet','bior3.5');
[lo,hi] = filters(fb);
```

对于bior3.5 lo和hi是12乘2矩阵。lo低通滤波器,hi是高通滤波器。lo列的 lo 和hi用于分析,第二列用于合成。将lo列的 lo 和hi分别与LoD和HiD进行比较。确认它们是相等的。

```
disp('Lowpass Analysis Filters')
```

Lowpass Analysis Filters

```
[lo(:,1) LoD']
```

```
ans = 12×2
```

```
-0.0138 -0.0138
 0.0414  0.0414
 0.0525  0.0525
-0.2679 -0.2679
-0.0718 -0.0718
 0.9667  0.9667
 0.9667  0.9667
-0.0718 -0.0718
-0.2679 -0.2679
 0.0525  0.0525
      :
```

```
disp('Highpass Analysis Filters')
```

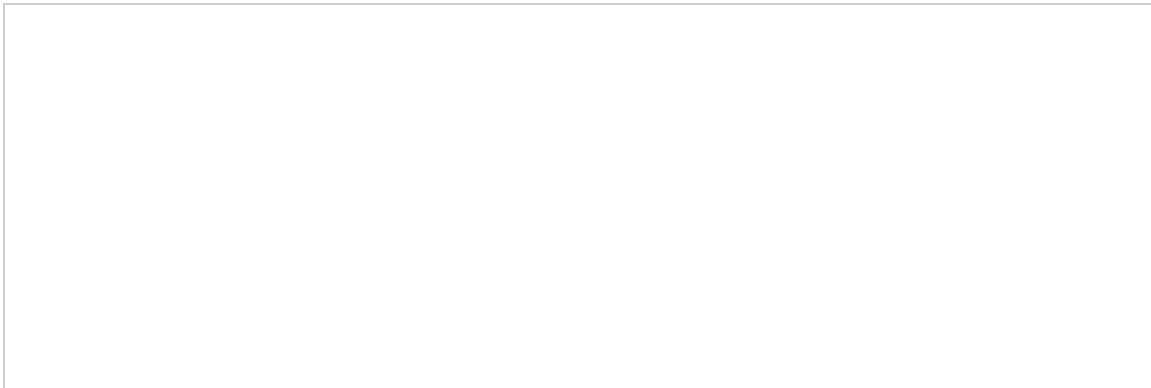
Highpass Analysis Filters

```
[hi(:,1) HiD']
```

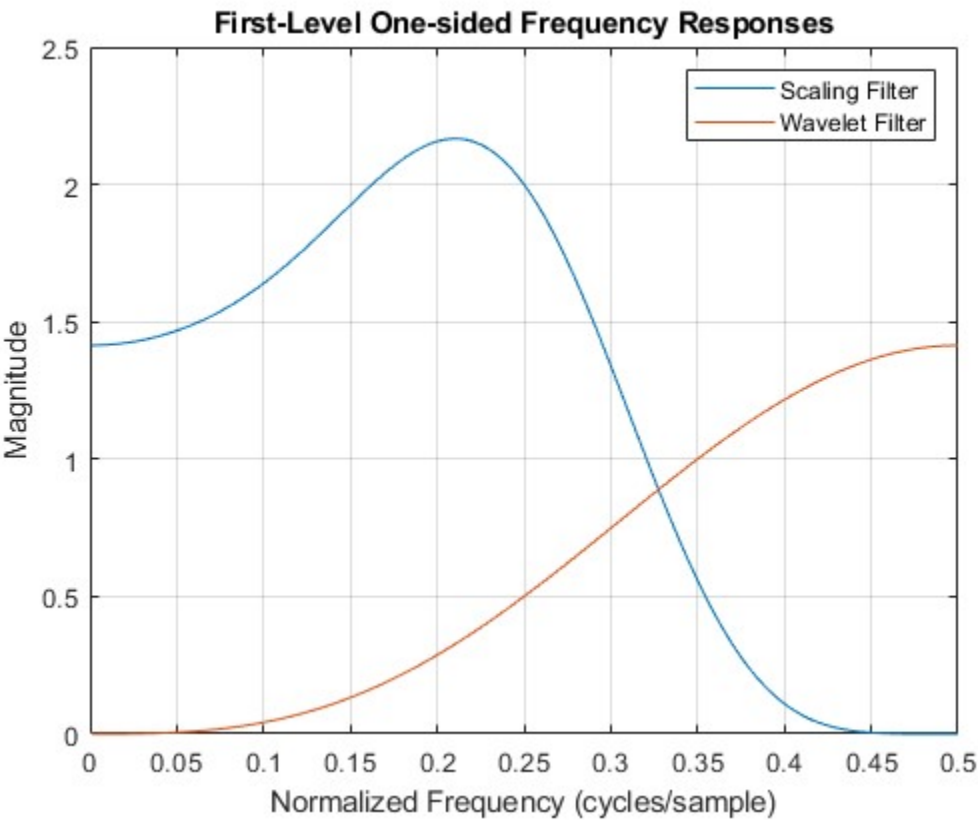
```
ans = 12×2
```

```
      0      0
      0      0
      0      0
      0      0
-0.1768 -0.1768
 0.5303  0.5303
-0.5303 -0.5303
 0.1768  0.1768
      0      0
      0      0
      :
```

绘制第一级小波滤波器和尺度滤波器的单侧幅度频率响应。



```
[psidft,f,phidft] = freqz(fb);
level = 1;
plot(f(len/2+1:end),abs(phidft(level,len/2+1:end)))
hold on
plot(f(len/2+1:end),abs(psidft(level,len/2+1:end)))
grid on
legend('Scaling Filter','Wavelet Filter')
title('First-Level One-sided Frequency Responses')
xlabel('Normalized Frequency (cycles/sample)')
ylabel('Magnitude')
```



输入参数

崩溃所有

✕ 输入数据
实值向量

输入数据, 指定为实值向量。
数据类型:double

✕ wname小波
字符向量字符串标量

分析小波用于计算单级 dwt, 指定为字符向量或字符串标量。小波必须通过 [wavemngr](#) 识别。分析小波来自以下小波族之一: daubechies、Coiflets、symlet、fejer-korovkin、离散 meyer、bi正交和反向双正交。有关每个系列中可用的小波, 请参阅 [wfilters](#) 。

示例: 'db4'

✓ **LoD 缩放过滤器**
偶数长度实值向量

缩放 (低通) 筛选器, 指定为偶数实值向量。LoD的长度必须与 [HiD](#)相同。有关详细信息, 请参阅 [wfilters](#) 。

数据类型:double

✓ **HiD 小波滤波器**
偶数长度实值向量

小波 (高通) 滤波器, 指定为偶数长度实值向量。HiD的长度必须与 [LoD](#)相同。有关详细信息, 请参阅 [wfilters](#) 。

数据类型:double

✓ **extmode-扩展模式**
'zpd' 'sp0' " 'spd' | ...

执行 dwt 时使用的扩展模式, 指定为以下模式之一:

mode	dwt 扩展模式
'zpd'	零扩展
'sp0'	订单的平滑扩展0
'spd'(或 'sp1')	订单的平滑扩展1
'sym'或'symh'	对称扩展 (半点): 边界值对称复制
'symw'	对称扩展 (整个点): 边界值对称复制
'asym'或'asymh'	对对称扩展 (半点): 边值反对称复制
'asymw'	对对称扩展 (整个点): 边值反对称复制
'ppd'	定期扩展 (1)
'per'	定期扩展 (2) 如果信号长度为奇数, wextend 向右侧添加一个等于最后一个值的额外示例, 并使用 'ppd' 模式执行扩展。否则, 'per' 到 'ppd'。此规则也适用于图像。

[dwtmode](#) 管理的全局变量指定默认扩展模式。有关扩展模式说明, 请参阅 [dwtmode](#) 。

示例:`[cA,cD] = dwt(x,'db4','mode','symw')`使用顺序 4 daubechies 极值相位小波和全点对称扩展返回x的单级 dwt。

输出参数

[崩溃所有](#)

✓ **cA近似系数**
实值向量

从小波分解中获得的近似系数, 作为向量返回。将输入信号 x 与缩放 x 滤波器 **LoD**合并, 然后是二进十进制, 得到近似系数。让 $sx = \text{size}(x)$ $lf =$ 分解筛选器的长度。

- 如果 dwt 扩展模式设置为周期,cA是长度的矢量 $\text{ceil}(sx/2 \text{ ceil}(sx/2))$.
- 对于其他扩展模式, cA是长度的矢量 $\text{floor}((sx+lf-1)/2 \text{ floor}((sx+lf-1)/2))$.

数据类型:double

✓ **cD详细系数**
实值向量

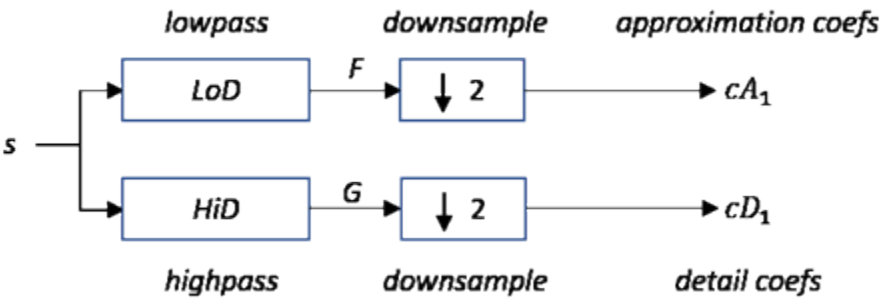
从小波分解中获得的详细系数, 作为向量返回。将输入信号 x 与小波 x 滤波器 **HiD**, 然后进行二进十进制, 得到细节系数。让 $sx = \text{size}(x)$ $lf =$ 分解筛选器的长度。

- 如果 dwt 扩展模式设置为周期, cD是长度的矢量 $\text{ceil}(sx/2 \text{ ceil}(sx/2))$.
- 对于其他扩展模式, cD是长度的矢量 $\text{floor}((sx+lf-1)/2 \text{ floor}((sx+lf-1)/2))$.

数据类型:double

算法

从长度为 n 的信号开始, 计算了两组系数: 近似系数 c_1 , 和细节系数 cd_1 .将其与缩放滤波器 **LoD**结合, 然后生成近似系数。同样, 利用小波滤波器 **HiD**进行旋转,然后是二进十进制, 得到了细节系数。



在哪里

- -使用过滤器 x 卷取

- [2]-下载示例 (保留偶数索引元素)

每个筛选器的长度等于 $2n$ 。如果 $n = \text{长度,信号}/\text{和}g\text{的长度}$ 为 $n + 2n - 1$, 系数 c_1 和 cd_1 是长度的地板 $\left(\frac{n-1}{2}\right) + n$ 。

为了处理基于卷积的算法所产生的信号端效应, `dwtmode` 的全局变量定义了所使用的信号扩展模式的类型。可能的选项包括零填充和对称扩展, 这是默认模式。

i 注意

对于相同的输入, dsp 系统工具箱™中的 `dwt` 函数和 `dwt` 块不会产生相同的结果。 `dwt` 块是为实时实现而设计的, 而 `wavelepoill`™软件则是为分析而设计的, 因此产品处理边界条件和过滤状态的方式不同。

要使 `dwt` 函数输出与 `dwt` 块输出匹配, 请通过在 matlab 中键入 `dwtmode('zpd')` 将函数边界条件设置为零填充®命令提示符。若要匹配使用 fir 筛选器实现的 `dwt` 块的延迟, 请将零添加到 `dwt` 函数的输入中。添加的零的数量必须等于筛选器长度的一半。

引用

[1] daubechies, i. 关于小波的十课。应用数学区域会议系列。宾夕法尼亚州费城: 工业和应用数学学会, 1992年。

[2] mallat, s. g. "多分辨率信号分解理论: 小波表示。基于模式分析和机器智能的 *ieee 交易*。第11卷, 第7期, 1989年7月, 第674-693 页。

[3] meyer, y. 小波和运算符。翻译: d. h. salinger。英国剑桥: 剑桥大学出版社, 1995年。

扩展功能

c/c ++ 代码生成

使用 matlab®编码器™生成 c 和 c++ 代码。

另请参见

`dwtfilterbank` `dwtmode` `idwt` `wavedec` `waveinfo`

在 r2006a 之前推出

How useful was this information?

