|  |
| --- |
|  |
| JXInst Spectrum Analysis |
| Class Library Design |
|  |
|  |
|  |

|  |
| --- |
|  |

**文档版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订说明 | 修订人签署 |
| 2017/02/14 | 1.0.0 | 初稿，频谱分析方法FindPeak、MeasurePower待定义 | 李远朝 |
| 2017/08/31 |  | 删除Commit这个动作，定义FindPeak、MeasurePowerInBand |  |

**目录**

1. 概要 1

1.1. 公共属性和方法概述 1

1.2. SpectrumTask最简单用法 2

1.3. SpectrumTask实现移动平均的功率谱计算 2

2. 公共属性 4

2.1. 基础参数 4

2.2. 频谱输出参数 4

2.3. 频谱单位参数 5

2.4. 频谱平均参数 5

2.5. 频谱输出信息(只读) 5

3. 公共方法 6

3.1. 频谱计算相关 6

3.2. 频谱分析相关 6

## 概要

### 公共属性和方法概述

聚星Spectrum Analysis类库提供实数和复数信号的频谱计算和分析测量功能。

Spectrum Analysis类库提供两组接口(类)：

1. 静态类Spectrum，提供基础快捷的频谱计算。
2. 可实例化的SpectrumTask类，提供高级频谱功能以支持RBW/频谱线数设置、获取所需波形点数、频谱平均（移动平均、多通道）、连续频谱等功能；

注：其中的Spectrum静态类应该通过调用SpectrumTask类来实现，即SpectrumTask类是核心库。

SpectrumTask类的公共属性和方法如下图所示。

公共方法

频谱计算：GetSpectrum(**+**), Reset, Commit

频谱分析测量：FindPeak、MeasurePower

公共属性

基础参数：InputDataType、SampleRate、WindowType、WindowPara(?)

频谱输出参数：*Output (SizeType/ NumberOfLines/ RBW/ CenterFrequency/ Bandwidth)*

频谱单位： *Unit (Type/ Impedance/ IsPSD)*

频谱平均参数：*Average (Mode/ WeightingType/ Size)*

频谱输出信息(只读)： *SpectralInformation (f0/ df / FFTSize/ FFTCount)*

### SpectrumTask最简单用法

// 生成正弦波，新建spectrum数组

double[] signal = new double[1000];

Generation.SineWave(ref signal, 1, 0, 100, 10000);

doublle[] powerSpectrum = new double[signal.Length / 2];

// 新建SpectrumTask，构造时设定输入数据类型和采样率；

var spectrum = new SpectrumTask(DataType.Real, 10000);

//设定窗函数、频谱线数和单位；

spectrum.WindowType =WindowType.Hanning;

spectrum.Output.NumberOfLines = powerSpectrum.Length;

spectrum.Unit.Type = SpectrumUnit.V2;

//计算功率谱并显示；

spectrum.GetSpectrum(signal, ref powerSpectrum);

easyChart1.Plot(powerSpectrum, spectrum.FreqStart, spectrum.FreqDelta);

### SpectrumTask实现移动平均的功率谱计算

// 生成正弦波，新建noise和spectrum数组

double[] signal = new double[1000];

Generation.SineWave(ref signal, 1, 0, 100, 10000);

double[] noise = new double[signal.Length];

doublle[] powerSpectrum = new double[signal.Length / 2];

// 新建SpectrumTask，构造时设定输入数据类型和采样率；

var spectrum = new SpectrumTask(DataType.Real, 10000);

//设定窗函数、频谱线数和单位；

spectrum.WindowType =WindowType.Hanning;

spectrum.Output.NumberOfLines = powerSpectrum.Length;

spectrum.Unit.Type = SpectrumUnit.V2;

//设定10次移动平均；

spectrum.Average.Mode = SpectrumAverageMode.RMSAverage;

spectrum.Average.WeightingType = AverageWeightingType.LinearMoving;

spectrum.Average.Size = 10;

//多次计算功率谱，并显示移动平均的频谱；

for (int i=0; i < 100, i++)

{

Generation.UniformWhiteNoise(ref noise, 0.1);

ArrayCalculation.Add(signal, noise, ref signal);

spectrum.GetSpectrum(signal, ref powerSpectrum);

easyChart1.Plot(powerSpectrum, spectrum.FreqStart, spectrum.FreqDelta);

}

## 公共属性

### 基础参数

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| InputDataType | enum | 输入数据类型，Real/Complex。默认为Real。 |
| SampleRate | double | 输入数据的采样率 |
| WindowType | enum | 频谱加窗类型 |
| WindowPara | double | 对于特定的窗类型(如Kaiser、Gaussian等)，可设定窗系数 |

### 频谱输出参数

频谱输出参数为SpectrumOutput类，用于定义关心的频段和所需谱线数，包含以下属性：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 说明 | |
| Type | enum | 定义如何确定频谱线数，默认为ByNumberOfLines。  ByNumberOfLines：用户通过NumberOfLines属性来设置所需的频谱线数，Commit()后可以获取实际频谱分辨率。  ByRBW：用户通过RBW属性来设置所需的频谱分辨率，Commit()之后可以获取实际的频谱线数。 | |
| NumberOfLines | int | 频谱线数 | |
| RBW | double | 频谱分辨率, 暂不可用1 | |
| CenterFrequency | double | 频谱的中心频率，默认为-1，暂不可用2 |
| Bandwidth | double | 频谱的带宽，默认为-1，暂不可用2 |

注1：暂不实现可配置的RBW特性，即只能设定谱线数。

注2：暂不实现可配置的CenterFrequency/Bandwidth特性，即对于Real输入类型，总是计算0~fs/2的频谱，而对于Complex输入类型，总是计算-fs/2 ~ fs/2的频谱。

### 频谱单位参数

频谱输出参数为SpectrumUnit类，用于定义频谱的单位，包含以下属性：

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| Type | enum | 幅度或功率谱的单位 |
| Impedance | double | 阻抗值，以欧姆为单位。在计算功率谱时使用。 |
| IsPSD | bool | 幅度或功率谱是否归一化为密度谱(Power Spectral Density)，默认为false。 |

### 频谱平均参数

频谱平均参数为SpectrumAverage类，用于定义频谱的平均方式，包含以下属性：

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| Mode | enum | 频谱平均方式。NoAverage/RMSAverage/PeakHold |
| WeightingType | enum | 当平均方式为RMSAverage时，设定平均的加权方式。  LinearMoving/LinearContinuous/Exponential |
| Size | double | 当平均方式为RMSAverage时，对于LinearMoving加权设定平均次数，对于Exponential加权设定指数衰减系数。 |

### 频谱输出信息(只读)

频谱平均参数为SpectrumInformation类，用于给出频谱计算相关信息，包含以下属性：

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| FreqStart | double | 频谱起始频率f0。 |
| FreqDelta | double | 频谱谱线的频率间隔df。 |
| FFTSize | int | 每次计算FFT时的长度，即每次应输入的时域波形长度。 |
| FFTCount | int | 当前已计算FFT的次数，该值会被Reset()方法清零。 |

## 公共方法

### 频谱计算相关

|  |  |
| --- | --- |
| 方法定义 | 说明 |
|  |  |
| void Reset() | 重置FFTCount属性值，若配置了频谱平均，则频谱平均历史也会被清空，即重新开始平均。 |
| void GetSpectrum(......) | 输入时域波形并获取频谱，提供多个重载1，以支持double/Complex输入，幅度或功率谱/相位谱输出。 |

注1：GetSpectrum多态方法一览：

void GetSpectrum (doube[] data, ref double[] spectrum);

void GetSpectrum (Complex[] data, ref double[] spectrum);

void GetSpectrum (double[] data, ref double[] amplitude, ref double[] phase);

void GetSpectrum (Complex[] data, ref double[] amplitude, ref double[] phase);

修改备注：已有的在Commit中赋值的属性，需要在属性的Set中update

### 频谱分析相关

|  |  |
| --- | --- |
| 方法定义 | 说明 |
| void FindPeak() | 查找频谱峰值。提供多个重载1，支持double输入，最大峰值、超过设定阈值的峰值输出 |
| void MeasurePowerInBand() | 计算带内功率2。 |

public struct Peak

{

/// <summary>

/// 峰值

/// </summary>

public double PeakValue;

/// <summary>

/// 频谱中峰值对应的加权频率，也就是实际频率

/// </summary>

public double PeakFrequency;

/// <summary>

/// 频谱中峰值对应的检索频率

/// </summary>

public double PeakIndexFreq;

}

注1：FindPeak多态方法一览：

Peak FindPeak(double[] spectrum);

Peak[] FindPeak(double[] spectrum，double threshold);

Peak FindPeak(double[] spectrum，double freqStart，double freqStop);

Peak[] FindPeak(double[] spectrum，double threshold, double freqStart，double freqStop);

注2：MeasurePowerInBand多态方法一览：

double MeasurePowerInBand(double centerFreq, double bandwidth)