振动指标计算及故障诊断方法

Version 1.0 Release 1

目录

[**一、 状态监测** **3**](#_Toc2875_WPSOffice_Level1)

[1.1. 振动位移 3](#_Toc27650_WPSOffice_Level2)

[1.2. 振动速度 3](#_Toc31015_WPSOffice_Level2)

[1.3. 振动加速度 3](#_Toc8524_WPSOffice_Level2)

[1.4. 平均值 3](#_Toc3922_WPSOffice_Level2)

[1.5. 峰值 4](#_Toc21071_WPSOffice_Level2)

[1.6. 峰峰值（极差） 4](#_Toc24792_WPSOffice_Level2)

[1.7. 整流平均值 4](#_Toc29356_WPSOffice_Level2)

[1.8. 有效值（均方根值） 4](#_Toc30994_WPSOffice_Level2)

[1.9. 方差、标准差 4](#_Toc5546_WPSOffice_Level2)

[1.10. 歪度（偏度） 5](#_Toc17975_WPSOffice_Level2)

[1.11. 峭度（峰度） 6](#_Toc8325_WPSOffice_Level2)

[1.12. 裕度 7](#_Toc4326_WPSOffice_Level2)

[1.13. 波形指标 7](#_Toc14421_WPSOffice_Level2)

[1.14. 峰值指标 7](#_Toc13577_WPSOffice_Level2)

[1.15. 脉冲指标 7](#_Toc4054_WPSOffice_Level2)

[**二、 故障诊断** **8**](#_Toc27650_WPSOffice_Level1)

[2.1. 频谱分析 8](#_Toc20192_WPSOffice_Level2)

[2.2. 故障率趋势分析 10](#_Toc7395_WPSOffice_Level2)

[2.3. 倒频谱分析 11](#_Toc25895_WPSOffice_Level2)

[2.4. SVD突变点识别](#_Toc14949_WPSOffice_Level2) [12](#_Toc14949_WPSOffice_Level2)

[2.5. 样本熵分析](#_Toc8634_WPSOffice_Level2) [13](#_Toc8634_WPSOffice_Level2)

[**三、 其他方法** **14**](#_Toc31015_WPSOffice_Level1)

[3.1. 模态分析法 14](#_Toc8954_WPSOffice_Level2)

[3.2. 随机减量技术 15](#_Toc29742_WPSOffice_Level2)

[**四、 python实现** **16**](#_Toc8524_WPSOffice_Level1)

1. 状态监测
2. 振动加速度

振动加速度被定义为振动速度的变化率，当一个机器的轴承座振动时，由于它连续不断地在前后运动中改变运动速度，所以它经受着力的加速作用。速度的变化率越大，也就是加速度值越大，施加在机器上的作用力也就越高。

连续信号中，振动加速度表示为振动信号的二阶导数：



而在数字信号下，振动加速度为振动速度的差分，即：



1. 平均值

平均值是常用的统计量，表示数据的平均水平。对于振动位移，振动速度和振动加速度来说，其平均值分别为：



1. 峰值

峰值为振动离平衡位置的最大偏离，平衡位置可能为平均值，可能为0，取决于研究对象参考标准。正峰值为偏离平衡位置正方向最大值，负峰值为偏离平衡位置负方向最大值，对于数值本身来说，正负峰值对应的正是数据的最大最小值，例如振动位移的正负峰值分别为：



而更值得注意的是**加速度的峰值**，常用于评价滚动轴承和齿轮的状态。

1. 峰峰值（极差）

峰峰值为正峰值与负峰值的差，也就是最大值与最小值的差，也称极差。观测最多的是振动**位移的峰峰值**，即正、负两方向间的最大振动距离：



以此来考察振动是否超出允许。

1. 整流平均值

整流来源于整流器，是将交流电转化为直流电的方式。整流平均值数学上也叫平均绝对值，是将数值大小取绝对值后再求平均值，例如位移的整流平均值为：



1. 有效值（均方根值）

有效值也叫均方根值，在数据统计分析中，将所有值平方求和，求其均值，再开平方，就得到均方根值。



在物理学中，我们常用均方根值来分析噪声。同样地我们可以计算速度有效值，即振动速度的均方根值，直接反映振动的能量。一台设备上不同位置测量的速度有效值中最大的一个称为该设备的振动烈度。

1. 方差、标准差

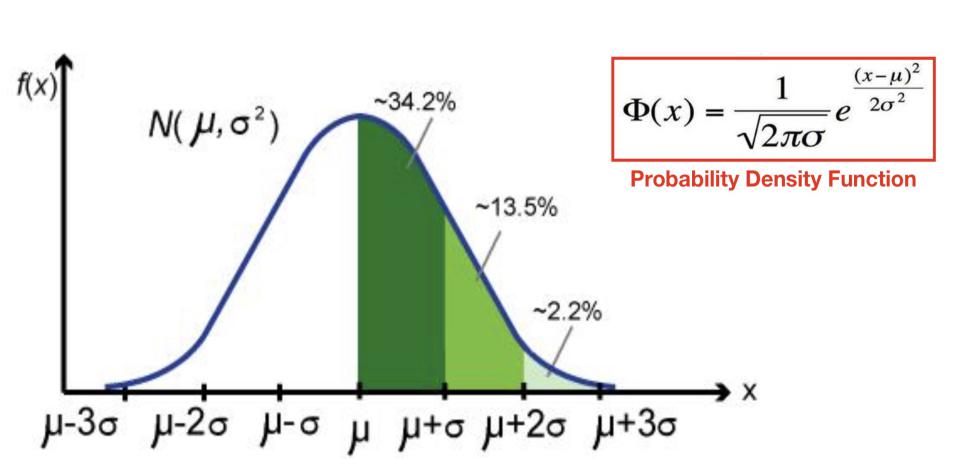
方差和标准差也是十分常用的统计量，是衡量一组数据离散程度的一个指标。在统计描述中，方差用来计算每一个变量（观察值）与总体均值之间的差异。为避免出现离均差总和为零，离均差平方和受样本含量的影响，统计学采用平均离均差平方和来描述变量的变异程度。总体方差计算公式：



其中*μ*是总体均值。而实际工作中，总体均数难以得到时，应用样本统计量代替总体参数，经校正后，样本方差计算公式：



标准差的平方就是方差，总体标准差*s*对应的是总体方差*s*2，样本标准差*σ*对应的是样本方差*σ*2。



图：均值与方差

从图可以看出方差越小，总体数据就越多靠近均值，数据也就越密集，所以运行状态也就越平稳。

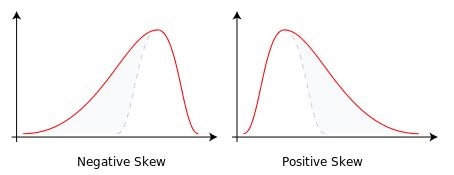
1. 歪度（偏度）

歪度（偏度）（skewness）也称为偏态、偏态系数，是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征：



歪度（偏度）是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征。歪度（偏度）是无量纲的量，取值通常在-3至3之间的值，它衡量了数据集的对称程度。歪度（偏度）系数越接近0，这说明数据集越对称，越远离0则表明数据集越不对称。

在数据序列呈对称分布（正态分布）的状态下，其均值、中位数和众数重合。且在这三个数的两侧，其它所有的数据完全以对称的方式左右分布。如果数据序列的分布不对称，则均值、中位数和众数必定分处不同的位置。这时，若以均值为参照点，则要么位于均值左侧的数据较多，称之为右偏；要么位于均值右侧的数据较多，称之为左偏；除此无它。



图：歪度示意

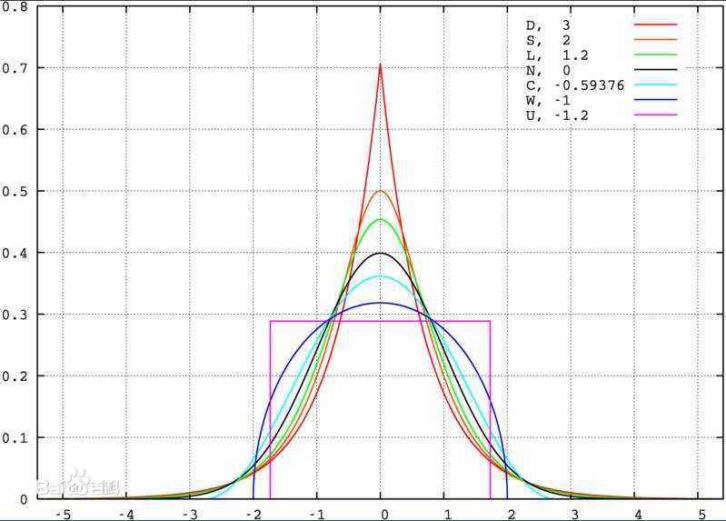
当你看到一个标准差的时候，你只能了解到这组数据的波动程度，却无法细致的了解到波动的是较小的数还是较大的数，而我们常常研究的正态分布是左右对称的。可以这么理解，标准差的大小是由数据的波动造成的，但是你并不知道是较大的数据有较大的波动还是较小的数据有较大的波动，比如-90,-50,-25,0,1,2,3,这就是较小的数有较大的波动。而-3,-2,-1,0,25,50,90就是较大的数有较大的波动。所以正歪度（偏度）即右偏分布有着较大值有较大的波动的特征，而负歪度（偏度）即左偏分布有着较小值有较大的波动的特征。

1. 峭度（峰度）

峭度（peakedness; kurtosis）又称峰度、峰态系数。表征概率密度分布曲线在平均值处峰值高低的特征数。直观看来，峭度反映了峰部的尖度。样本的峭度是和正态分布相比较而言统计量，如果峭度大于0，峰的形状比较尖，比正态分布峰要陡峭。反之亦然。



峭度是指次数分布曲线顶峰的尖平程度，是次数分布的又一重要特征。统计上，常以正态分布曲线为标准，来观察比较某一次数分布曲线的顶端尖顶或平顶以及尖平程度的大小。



图：峭度

根据变量值的集中与分散程度，峭度一般可表现为三种形态：尖顶峭度、平顶峭度和标准峭度。当变量值的次数在众数周围分布比较集中，使次数分布曲线比正态分布曲线顶峰更为隆起尖峭，称为尖顶峭度；当变数值的次数在众数周围分布较为分散，使次数分布曲线较正态分布曲线更为平缓，称为平顶峭度。可见，尖顶峭度或平顶峭度都是相对正态分布曲线的标准峭度而言的。

1. 裕度

裕度也称裕度指标，裕度常用来检测机械设备的磨损状况，计算方式为：



其中*xr*称之为方根幅值。

1. 波形指标

波形指标为有效值与均值绝对值的比，即：



中的平方和中的绝对值意味着值和形状因子在任何点都独立于波函数的正负号。因此，对于具有常规平均值0和完全相反方向的零一波，形状因子是相同的。

1. 峰值指标

峰值指标是[波形](https://en.wikipedia.org/wiki/Waveform)的参数，例如[交流电](https://en.wikipedia.org/wiki/Alternating_current)或声音，表示峰值与有效值的比:



换句话说，峰值指标表示峰值在波形中的极端程度。峰值指标1表示没有峰值，例如[直流](https://en.wikipedia.org/wiki/Direct_current)或[方波](https://en.wikipedia.org/wiki/Square_wave)。较高的峰值指标表示有尖峰，例如声波往往具有高峰值指标。

1. 脉冲指标

脉冲指标为峰值与平均值绝对值的比：



可以发现

峭度指标、裕度指标和脉冲指标对冲击脉冲型故障比较敏感。当早期发生故障时，大幅值的脉冲还不是很多，均方根值变化不大，上述参数已有增加。当故障逐步发展时，它们上升较快，但上升到一定程度后会逐步下降。

1. 故障诊断
2. 频谱分析

频谱分析是一种将复噪声号分解为较简单[信号](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7/32683" \t "https://baike.baidu.com/item/%E9%A2%91%E8%B0%B1%E5%88%86%E6%9E%90/_blank)的技术。许多物理信号均可以表示为许多不同频率简单信号的和。找出一个信号在不同频率下的信息（可能是[幅度](https://baike.baidu.com/item/%E5%B9%85%E5%BA%A6/5900395" \t "https://baike.baidu.com/item/%E9%A2%91%E8%B0%B1%E5%88%86%E6%9E%90/_blank)、[功率](https://baike.baidu.com/item/%E5%8A%9F%E7%8E%87/808705" \t "https://baike.baidu.com/item/%E9%A2%91%E8%B0%B1%E5%88%86%E6%9E%90/_blank)、[强度](https://baike.baidu.com/item/%E5%BC%BA%E5%BA%A6/1659506" \t "https://baike.baidu.com/item/%E9%A2%91%E8%B0%B1%E5%88%86%E6%9E%90/_blank)或[相位](https://baike.baidu.com/item/%E7%9B%B8%E4%BD%8D/2391710" \t "https://baike.baidu.com/item/%E9%A2%91%E8%B0%B1%E5%88%86%E6%9E%90/_blank)等）的作法就是频谱分析。频率分析的数学基础是傅里叶变换和快速傅里叶算法（FFT）。

数字信号作频谱分析利用的是离散傅里叶变换（DFT）。设*X*=[*x*1,*x*2,...,*xn*]*T*为一长度为*n*的向量（信号），则其离散傅里叶变换序列为：



其中，*i*为虚数单位。简单地，记，则有



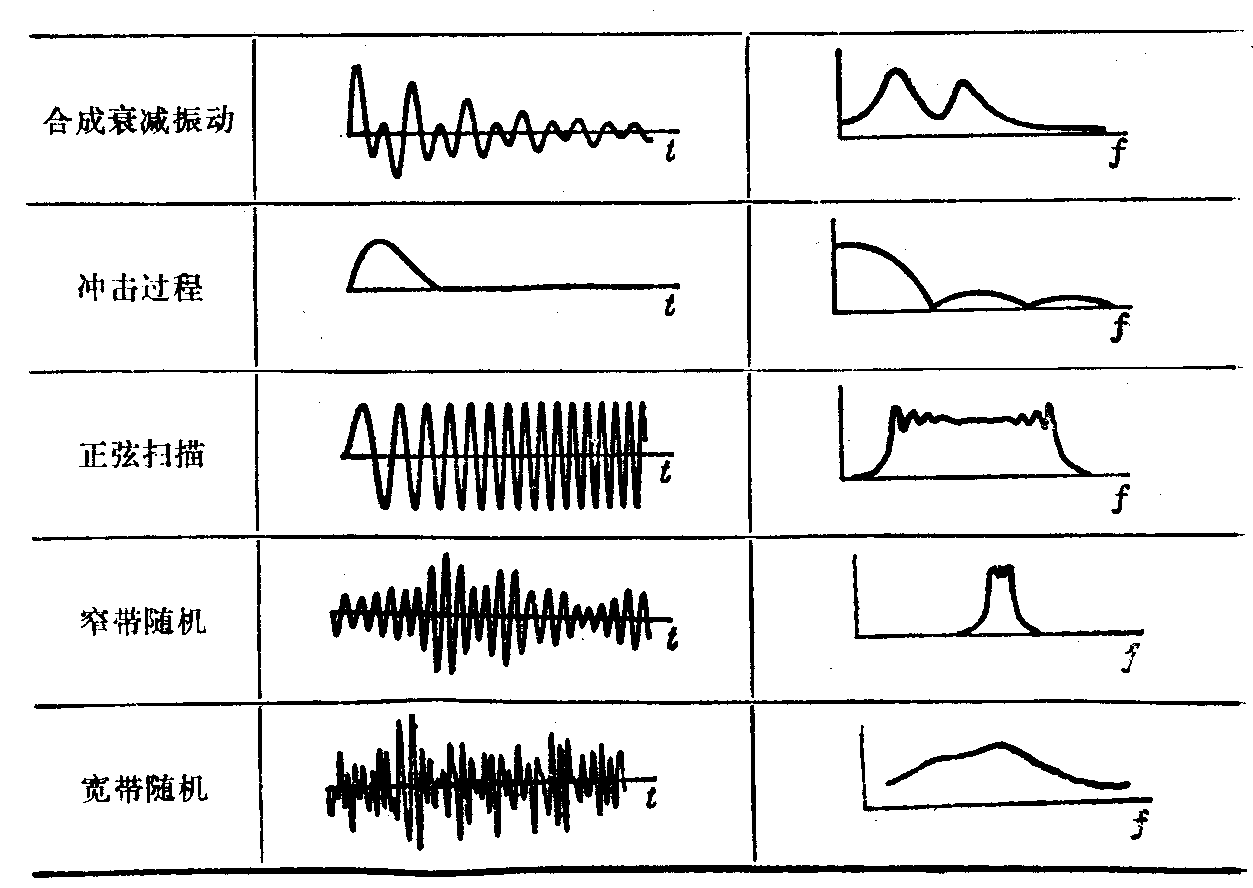
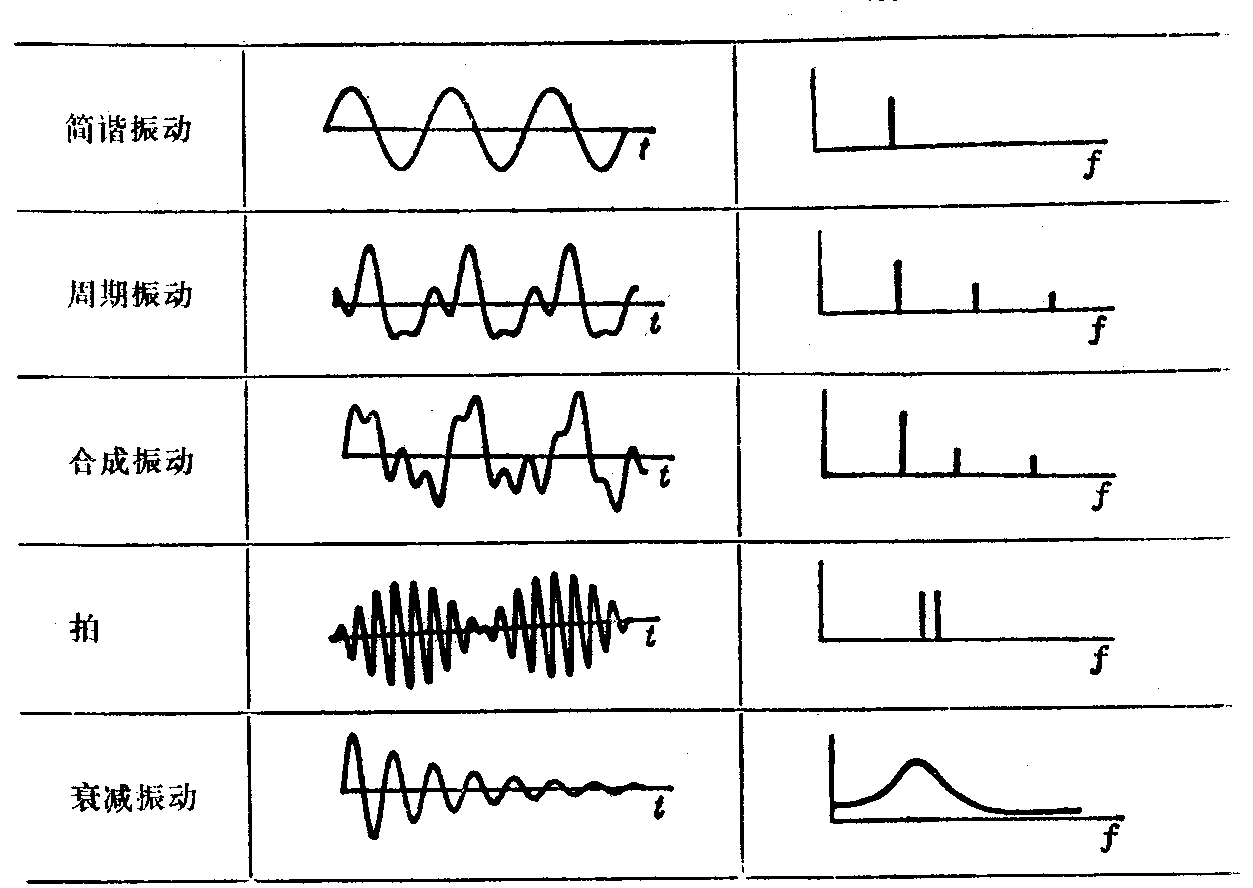
其中，



称之为傅里叶矩阵。

对一些常用的振动信号做傅里叶变换，可得到下图时频对照图：

图：常见振动时频分析图



名称 波 形 频 谱 名称 波 形 频 谱

数字信号处理的主要[数学工具](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E5%B7%A5%E5%85%B7/903000" \t "https://baike.baidu.com/item/%E7%AA%97%E5%87%BD%E6%95%B0/_blank)是傅里叶变换．而傅里叶变换是研究整个[时间域](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E5%9F%9F/691237" \t "https://baike.baidu.com/item/%E7%AA%97%E5%87%BD%E6%95%B0/_blank)和[频率域](https://baike.baidu.com/item/%E9%A2%91%E7%8E%87%E5%9F%9F/682476" \t "https://baike.baidu.com/item/%E7%AA%97%E5%87%BD%E6%95%B0/_blank)的关系。不过，当运用计算机实现工程[测试信号处理](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E4%BF%A1%E5%8F%B7%E5%A4%84%E7%90%86/5412339" \t "https://baike.baidu.com/item/%E7%AA%97%E5%87%BD%E6%95%B0/_blank)时，不可能对无限长的信号进行测量和运算，而是取其有限的时间片段进行分析。做法是从信号中截取一个时间片段，然后用截取的信号时间片段进行[周期延拓](https://baike.baidu.com/item/%E5%91%A8%E6%9C%9F%E5%BB%B6%E6%8B%93/5446570" \t "https://baike.baidu.com/item/%E7%AA%97%E5%87%BD%E6%95%B0/_blank)处理，得到虚拟的无限长的信号，然后就可以对信号进行傅里叶变换、相关分析等数学处理。无限长的信号被截断以后，其频谱发生了畸变，原来集中在f(0)处的能量被分散到两个较宽的频带中去了（这种现象称之为频谱能量泄漏）。

为了减少频谱能量泄漏，可采用不同的截取函数对信号进行截断，截断函数称为窗函数，简称为窗。对一个有限长的时域采样信号进行傅里叶变换实际上就是加了一个矩形窗，相当于用矩形窗把无限长的数据截断了，所以，我们也就可以根据需要把这个矩形窗换成其他窗函数，常用的有hanning窗、Gauss窗。

对于原始信号*X*=[*x*1,*x*2,...,*xn*]，构造与之相等长度的窗函数序列*W*=[*w*1,*w*2,...,*wn*]，而加窗的傅里叶变换相当于先将原始信号与窗函数序列作乘积之后再作傅里叶变换，即：



而分析的方式与傅里叶变换一样。

1. 故障率趋势分析

实践证明大多数设备的故障率是时间的函数，典型故障[曲线](https://baike.baidu.com/item/%E6%9B%B2%E7%BA%BF/12004395" \t "https://baike.baidu.com/item/%E6%B5%B4%E7%9B%86%E6%9B%B2%E7%BA%BF/_blank)称之为浴盆曲线(Bathtub curve,失效率曲线) ，曲线的形状呈两头高，中间低，具有明显的阶段性，可划分为三个阶段：早期故障期，偶然故障期，严重故障期。

偶然故障期

1. 严重故障期

早期故障期

1. 故障率
2. 时间

图：浴盆曲线

对于故障率来说早期故障期是磨合期，在整个应用周期中，虽然存在这种现象，但这个时期的数据数值并没有计算意义，也就是说，要做到预测性维护，则需要从正常使用的偶然故障期开始分析。

实际上，在偶然故障期内故障率是缓慢增长的，达到严重故障期之后故障率将迅速增长，我们需要寻找的正是偶然故障期与严重故障期的临界值，也就是故障率增长速度开始加快的时候。数学上，当故障率的加速度超过一定数值时，可视为骤变开始阶段，即将到达严重故障期，即：假定当前故障率为*rt*，前一时刻的故障率为*rt*-1，再前时刻故障率为*rt*-2，则当加速度



时（其中*ξ*为关于设备零件的阈值），则需要发出预警。

图：劣化曲线

停机

报警

1. 故障征兆
2. 时间
3. 劣化曲线
4. 报警值
5. 危险值
6. 倒频谱分析

对于高速大型旋转机械，其旋转状况是复杂的，尤其当设备出现不对中，轴承或齿轮的缺陷、油膜涡动、磨擦、质量不对称等现象时，则振动更为复杂，用一般频谱分析方法已经难于辩识(识别反映缺陷的频率分量)，而用倒频谱，则会增强识别能力。倒频谱分析一般分为两类，实倒频谱与复倒频谱。

实倒频谱即现在通常所说的幅值倒频谱，此倒频谱是对频谱的进一步谱分析得到的，设原始信号为*X*，傅里叶变换后序列为，即，功率谱为:



记，功率倒频谱*CX*即对功率谱作对数转换后再次进行傅里叶变换得到的功率谱，而幅值倒频谱为功率倒频谱的平方根：

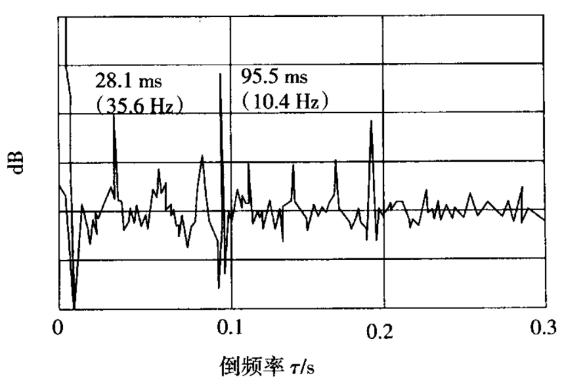
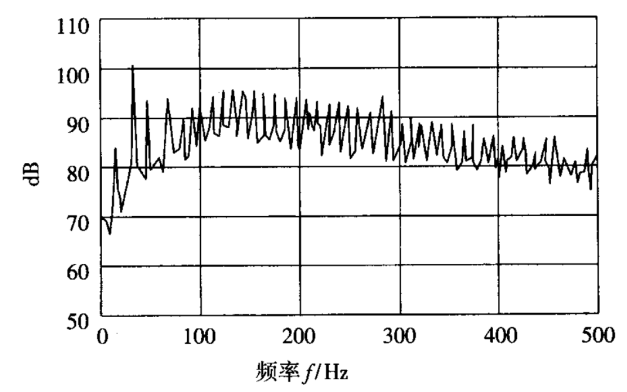


复倒频谱是另一类倒频谱，是从复谱得来，不损失相位信息，它与实倒频谱不同，复倒频谱获取倒频谱的过程是可逆的。设原始信号为*X*，傅里叶变换后序列为，则复倒频谱为



其中，*F*-1表示傅里叶逆变换。

如下图左为一有缺陷的齿轮在运行时测得的功率谱，它含有大量的边频带频谱分量。通过功率谱我们只能大致估计其边频间距约为10Hz，很难分辨确定出各种周期成分。但是通过做该齿轮的倒频谱下图右所示，我们能够精确的找到其倒频率*τ*=9.95ms，对应频率为1/*τ*=10.4Hz。这就很方便的找出了复杂频谱上的周期成分。另外，从齿轮倒频谱图中我们还可以找到频率为28.1ms的谱线，其对应的频率为35.6Hz ，这个频率是齿轮轴的旋转频率。



图：频谱与倒频谱

1. SVD突变点识别

由于振动信号各子波的起始时刻在原始信号上表现为突变，而SVD作为一种线性分解方法，恰好可以在某些分解层中将这些突变信息提取出来。具体步骤如下：

1. 由原始信号*X*构造Hankel矩阵



其中，*n*为原信号长度，即最大采样点数，*L*为束参数。

1. 对构造的矩阵*HX*进行奇异值分解得到



其中，*U*为(*N*-*L*)×(*N*-*L*)的正交矩阵，Λ为(*N*-*L*)×(*L*+1)的对角阵，主对角元素为*HX*的奇异值*σ*1,*σ*2,...,*σL*-1，*V*为(*L*+1)(*L*+1)的正交矩阵。对上式进行展开有



其中，*ui*∈*RN*-*L*，*vi*∈*RL*+1，*i*=1,2,...,*m*，*m*=min{*N*-*L*,*L*+1}。

1. 令



则



1. 将上式第一行和最后一列连起来得到



可证



从*P*1到*Pm*依次为原始信号的主要成分、突变信息和噪声。

1. 样本熵分析

复杂性是衡量系统无序程度的一个重要的非线性特征，对电机轴承振动信号的复杂度的度量，有利于从复杂的振动信号中挖掘出轴承的状态变化，有效地检测轴承早期故障，从而采取一定维护措施。基于熵的复杂性衡量算法具有计算方便、抗噪的功能强大等优点，更加适合检测出电机轴承状态的变化。

样本熵算法由Richman和Moorman联合提出，具有对时间序列长度依赖性较少，计算量小等优越性，已经广泛应用于机械故障诊断中。

对于给定的一段序列*X*=[*x*1,*x*2,...,*xn*]，样本熵的计算方法步骤如下：

1. 将原始数按顺序构成*m*维向量，即:



其中*i*=1,2,...,*n*-*m*+1。

1. 定义*Xi*和*Xj*之间的距离：



其中，*i*,*j*=1,2,...*n*-*m*+1。

1. 给定一个阈值*r*(*r*>0)，固定一个*i*，统计*dij*≤*r*的个数*zi*，称之为模板匹配数。这个数与总数*n*-*m*的比值为：



其中，1≤*j*≤*n*-*m*,*j*≠*i*，然后计算的平均值：



1. 将维数*m*增加到*m*+1，重复Step 1.到Step 3.得到；
2. 时间序列*X*的样本熵定义为：



实际计算中，*n*只能是有限值，因此样本熵的计算公式可以写为：



从样本熵的计算过程可以清楚地看出，样本熵的计算结果受到嵌入维数*m*以及相似容限*r*的取值的影响。

1. 其他方法
2. 模态分析法

[模态分析](https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/5108005" \t "https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/_blank)是研究结构动力特性一种方法，一般应用在工程振动领域。其中，模态是指机械结构的[固有](https://baike.baidu.com/item/%E5%9B%BA%E6%9C%89" \t "https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/_blank)[振动](https://baike.baidu.com/item/%E6%8C%AF%E5%8A%A8" \t "https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/_blank)特性，每一个模态都有特定的固有频率、[阻尼比](https://baike.baidu.com/item/%E9%98%BB%E5%B0%BC%E6%AF%94/4584802" \t "https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/_blank)和模态振型。分析这些模态参数的过程称为模态分析。按计算方法，模态分析可分为计算模态分析和试验模态分析。

[计算模态](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%A8%A1%E6%80%81/5017875" \t "https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/_blank)分析由[有限元](https://baike.baidu.com/item/%E6%9C%89%E9%99%90%E5%85%83/1761759" \t "https://baike.baidu.com/item/%E6%A8%A1%E6%80%81%E5%88%86%E6%9E%90/_blank)计算的方法取得，每一阶次对应一个模态，每个阶次都有自己特定的频率、阻尼、模态参数。试验模态分析通过试验将采集的系统输入与输出信号经过参数识别获得。

若系统产生频率为*f*的并联谐振现象，其节点电压、电流方程为：



其中，*Yf*为在频率*f*处的系统节点导纳矩阵；*Vf*和*If*分别为节点电压和阶段注入电流向量。节点导纳矩阵*Yf*可分解如下：



其中，为对角特征值矩阵，即，*L*=[*L*1,L2,...,*Ln*]、*T*=[*T*1,*T*2,...,*Tn*]分别为左右特征向量矩阵，且有*L*=*T*-1。

进一步，有



其中，命名为模态阻抗。

定义*Uf*=*TVf*为模态电压向量，*Jf*=*TIf*为模态电流向量，则有，即：



当系统发生并联谐振时，某些节点的注入电流将产生很高的节点电压，即若*λf*1=0或非常小，则很小的模态1注入电流*Jf*1将导致很大的模态电压*Uf*1，而其他模态电压不受影响。即在模态域中，根据特征值倒数易于识别出谐振的位置。

谐波谐振实际上只在特定的模式下发生，它与某个节点的注入电流无关也并非由其引起。可将最小的特征值称为谐振“关键模式”，对应的左、右特征向量则称为“关键特征向量”。

模态电流*Jf*1可表示为电流在第一个特征向量上的线性映射，即：



若*T*1*n*有最大值，则节点电流*Ifn*将对模态1注入电流具有最大的贡献度，即节点*n*最容易受到外界激励而产生模态1谐振。同理可得，模态电压与节点电压间存在关系*Vf*=*LUf*，则有：



若系统发生模态1谐振，则*Uf*1远大于其他模态电压，此时，上式可近似仅由[*L*11,*L*21,…,*Ln*1]*TUf*1来表示。如果*Ln*1的值最大，则节点*n*将会出现最大电压值，即节点*n*处最容易观测到模态1谐振。

综上可得结论：右特征向量矩阵*T*反映关键模式下节点的谐振可激励性，左特征向量矩阵*L*反映关键模式下节点的谐振可观测性，在具有最大激励性的节点注入相应信号可有效抑制谐波谐振现象。

实际上可将可激励性和可观测性结合成为一个新指标量。当系统发生模态1谐振时，模态阻抗1/*λf*1的值远大于其他模态阻抗值，则式可简化为：



上式中矩阵对角线元素显示了节点在关键模式下的可激励性和可观测性的结合，不妨称之为关键模式下节点的“参与因子”，参与因子反映了节点对所发生谐振的参与度以及贡献度，对某种模式具有最大可观测性的节点同时也具有最大可激励性。具有最大参与因子的节点则可被认为是系统谐波谐振的中心。

1. 随机减量技术

随机减量法是20世纪70年代国外首先发展起来的一种振动分析方法，它是一种对响应信号数据的预处理，基本思想是:将响应信号根据一定的规则划分为很多相等长度的部分，对所有划分得到信号经过叠加处理，则可得到一条原始信号对应的显自由衰减的振动信号，最后通过其他模态辨识方法来对模态参数进行识别。

一单自由度线性系统运动微分方程



其位移响应



其中*D*(*t*)是初始位移为1，初始速度为0的自由振动响应，*v*(*t*)是初始位移为0，初始速度为1的自由振动响应，*h*(*t*)为系统单位脉冲响应函数，*x*(0)、分别为系统的初始位移、初始速度。

选取一恰当的*x*(*ti*)使*A*≤*x*(*ti*)≤*H*去截取这个样本函数，其第一个检点对应的时刻为*ti*(*i*=1,2,...)，则有



称之为子样本函数。这样，可得许多子样本函数，子样本函数的全体构成子响应过程，记为*X*(*t*-*ti*)。将各子样本函数的起始时刻移至坐标原点，即*ti*=0。设*f*(*t*)是均值为零的平稳正态随机过程，对*X*(*t*)取数学期望，并注意到，可得



上式表明，子响应过程*X*(*t*)的期望是个初始位移为，初始速度为零的自由振动响应。实际测量时，因样本长度有限，数学期望以随机减量特征函数代替，即：



其中*n*为子样本个数，。

1. python实现

**import** numpy **as** np  
*# import scipy.stats as sts #计算歪度峭度可用  
# from matplotlib import pyplot as plt #画图*data\_input = [1, 3, 4, 5, -1, 3, -2, 1, 5, 7, 2] \* 10 *# 原始数据列表格式（例）*X = np.array(data\_input) *# 振动位移（向量格式）*delta\_t = 1 *# 时间间隔，默认等距*V = np.diff(X) / delta\_t *# 振动速度*A = np.diff(V) / delta\_t *# 振动加速度*x\_bar = np.mean(X) *# 位移平均值*v\_bar = np.mean(V) *# 速度平均值*a\_bar = np.mean(A) *# 加速度平均值*xp\_plus = np.max(X) *# 位移正峰值*xp\_minus = np.min(X) *# 位移负峰值*vp\_plus = np.max(V) *# 速度正峰值*vp\_minus = np.min(V) *# 速度负峰值*ap\_plus = np.max(A) *# 加速度正峰值*ap\_minus = np.min(A) *# 加速度负峰值*xpp = xp\_plus - xp\_minus *# 位移峰峰值*vpp = vp\_plus - vp\_minus *# 速度峰峰值*app = ap\_plus - ap\_minus *# 加速度峰峰值*x\_av = np.mean(np.abs(X)) *# 位移整流平均值*v\_av = np.mean(np.abs(V)) *# 速度整流平均值*a\_av = np.mean(np.abs(A)) *# 加速度整流平均值*x\_rms = np.sqrt(sum(X \*\* 2) / len(X)) *# 位移有效值*v\_rms = np.sqrt(sum(V \*\* 2) / len(V)) *# 速度有效值*a\_rms = np.sqrt(sum(A \*\* 2) / len(A)) *# 加速度有效值*var\_x = np.var(X) *# 位移方差*std\_x = np.std(X) *# 位移标准差*var\_v = np.var(V) *# 速度方差*std\_v = np.std(V) *# 速度标准差*var\_a = np.var(A) *# 加速度方差*std\_a = np.std(A) *# 加速度标准差*Skew\_x = sum(((X - x\_bar) / std\_x) \*\* 3) / len(X) *# 位移歪度（偏度）  
# Sk=sts.skew(X)*Kurt\_x = sum(((X - x\_bar) / std\_x) \*\* 4) / len(X) - 3 *# 位移峭度  
# ku=sts.kurtosis(X)*x\_r = sum(np.sqrt(np.abs(X))) / len(X) *# 方根幅值*L = max(abs(xp\_plus), abs(xp\_minus)) / x\_r *# 裕度*Sf = x\_rms / abs(x\_bar) *# 波形指标（波形因子）*Cf = max(abs(xp\_plus), abs(xp\_minus)) / x\_rms *# 峰值指标（峰值因子）*If = max(abs(xp\_plus), abs(xp\_minus)) / abs(x\_bar) *# 脉冲指标（脉冲因子）*X\_hat = np.fft.fft(X) *# 快速傅立叶变换*Gx = np.abs(X\_hat) \*\* 2 *# 功率谱  
  
##加窗傅立叶变换（hanning窗）##*hann = np.hanning(len(X))  
Xhann\_hat = np.fft.fft(hann \* X) *# 加窗傅立叶变换*Ghann = np.abs(Xhann\_hat) \*\* 2 *# 功率谱  
  
##倒频谱分析##*lgGx = np.log10(Gx)  
*# lgGx[np.isinf(lgGx)]=0*Cr = np.abs(np.fft.fft(lgGx)) *# 幅值倒频谱（实倒频谱）*lgX\_hat = np.log10(X\_hat)  
*# lgX\_hat[np.isinf(lgX\_hat)]=0*Cc = np.fft.ifft(np.abs(lgX\_hat)) *# 复倒频谱  
  
##SVD分析##*Lnum = 30 *# 束参数*Hankel\_X = []  
**for** ii **in** range(len(X) - Lnum + 1):  
 row\_cut = X[ii:ii + Lnum]  
 Hankel\_X.append(row\_cut)  
Hankel\_X = np.array(Hankel\_X) *# 构造Hankel矩阵*[U, Lamb, V] = np.linalg.svd(Hankel\_X) *# SVD分解*P\_all = []  
m = min(len(X) - Lnum, Lnum + 1)  
**for** jj **in** range(m - 1):  
 HH = Lamb[jj] \* np.dot(np.array([U[:, jj]]).T, np.array([V[jj]]))  
 Pp = HH[0].tolist() + HH[1:, -1].tolist()  
 P\_all.append(Pp)  
  
  
*##样本熵##***def** calculate\_Bmr(X, m\_dim, r\_dis):  
 All\_X = []  
 **for** ii **in** range(len(X) - m\_dim + 1):  
 Xi = X[ii:ii + m\_dim]  
 All\_X.append(Xi) *# 向量Xi集合，相当于Hankel矩阵* All\_Bmir = []  
 **for** kk **in** range(len(All\_X)):  
 dis\_k = []  
 **for** ll **in** range(len(All\_X)):  
 d\_kl = np.max(np.abs(All\_X[kk] - All\_X[ll]))  
 dis\_k.append(d\_kl)  
 dis\_k = np.array(dis\_k) *# 记录距离* count\_zi = sum(np.array(dis\_k <= r\_dis, dtype=int)) - 1 *# Zi模板匹配数* Bmir = count\_zi / (len(X) - m\_dim) *#* All\_Bmir.append(Bmir) *# 所有Bmir* Bmr = sum(All\_Bmir) / len(All\_Bmir)  
 **return** Bmr  
  
  
m\_dim = 4 *# 嵌入维数*r\_dis = 3 *# 相似容忍*Bmr = calculate\_Bmr(X, m\_dim, r\_dis)  
Bm1r = calculate\_Bmr(X, m\_dim + 1, r\_dis)  
Emr = -np.log(Bm1r / Bmr) *# 样本熵*  
*# plt.plot(np.abs(X))  
# plt.show()*