振动与噪声测试帮助

1 时域分析

1.1 有效值

有效值衡量信号的平均能量或功率。在交流电信号分析中尤为重要，代表信号的“有效”幅度。假设信号为：$x_1, x_2, \cdots, x_n$，其有效值可计算为：

$$\sqrt{\frac{x_1^2 + x_2^2 + \cdots x_n^2}{n}} \tag{1}$$

1.2 平均值

假设信号为：$x_1, x_2, \cdots, x_n$，其有效值可计算为：

$$\frac{x_1 + x_2 + \cdots + x_n}{n} \tag{2}$$

1.3 最大幅值

假设信号为：$x_1, x_2, \cdots, x_n$，其最大幅值可计算为：

$$\max\left(|x_1|, |x_2|, \cdots, |x_n|\right) \tag{3}$$

1.4 平均幅值

假设信号为：$x_1, x_2, \cdots, x_n$，其平均幅值可计算为：

$$\frac{|x_1| + |x_2| + \cdots + |x_n|}{n} \tag{4}$$

1.5 方根幅值

方根幅值通常用于电信号处理中，表示信号的幅度大小。它是信号幅度的一种评估方法，即信号强度的均方根值。

在电信号处理中，方根幅值常用于衡量变化幅度较大的信号或周期性信号的振幅大小。它表示信号在给定时间段内的平均幅度。将信号的所有样本值求绝对值，然后取平均值，最后再开平方根，就得到了方根幅值。这个值给出了信号在整个时间段内的典型幅度。

方根幅值的物理意义在于提供了一个用于比较信号强度的标准。通过计算方根幅值，可以量化信号的整体强度，有助于分析信号的特性、比较不同信号之间的强度差异以及评估信号在传输或处理过程中的变化。

假设信号为：$x_1, x_2, \cdots, x_n$，其方根幅值可计算为：

$$\sqrt{\frac{|x_1|+|x_2|+\cdots+|x_n|}{n}} \tag{4}$$

1.6 方差

方差:

$$s^2=\frac{1}{n}[(x_1-\bar{x})^2+(x_2-\bar{x})^2+\ldots(x_n-\bar{x})^2]$$

1.7 标准差

标准差:

$$s=\sqrt{\frac{1}{n}[(x_1-\bar{x})^2+(x_2-\bar{x})^2+\ldots(x_n-\bar{x})^2]}$$

1.8 总声压级

```
%计算有效声压
% 根据定义计算有效声压，pa = sqrt((x(1)^2+x(2)^2+...+x(M)^2)/M)
% 单位为Pa
pp = 0;
M=length(sig);
for i = 1:M
   pp = pp + sig(i)^2;%下面公式2-5
end
pa = sqrt(pp/M);

%%计算声压级
% 声压级值spl=20*log10(pa/p0),单位为dB
% 基准声压p0，单位为Pa
p0 = 2*10^-5;
spl = 20*log10(pa/p0);%下面公式2-6
```

1.9 峰峰值

```
ppv=max(sig)-min(sig);
```

2 频谱分析

幅频曲线

```
function [f,Y,fs]=pingpu(time,data)
fs=1/(time(2)-time(1));
N=length(time); %样点个数
df=fs/(N-1);%分辨率
f=(0:N-1)*df;%其中每点的频率，第一个点对应的频率为0
Y = fft(data(1:N))/(N/2);%真实的幅值
Y=20*log10(abs(Y)/2e-5);

f=f(1:N/2);
Y=Y(1:N/2);
```

3 功率谱分析

3.1 自功率谱分析

```
function [psdestx,Fxx] =psd_analysis(noise,N,Fs)

[psdestx,Fxx] = periodogram(noise,rectwin(length(noise)),N,Fs);
psdestx=pow2db(psdestx);
```

3.2 互功率谱分析

```
function [Pxy2,arg2,f]=cpsd_analysis(x,y,Fs)

[Pxy, f] = cpsd(x,y,[],[],[],Fs);
Pxy2=pow2db(abs(Pxy));
arg2=angle(Pxy);
```

4 相关分析

4.1 自相关分析

```
// 计算自相关函数
/*
N 序列长度
n 偏移位置
data 数据
*/
double AutoCorrelation(double* data, int n, int N){
    double    r= 0.0;
    if (n>=0){
        for(int i = n; i <N;i++ ) {
            r +=data[i] * data[i-n];
        }
    } else {
        for(int i = 0; i <N+n;i++ ) {
            r +=data[i] * data[i-n];
        }
    }
    return  r ;
}

void autoCorrelationAnalysis(double* data, int N, double dt, double* xlist, double* ylist){
    for (int n=-N/2;n<N/2;n++){
        *xlist=n*dt;
        *ylist=AutoCorrelation(data,n,N);
        xlist++;
        ylist++;
    }
}
```

4.2 互相关分析

```
double CrossCorrelation(double* data1, double* data2, int n, int N){
    double    r= 0.0;
    if (n>=0){
        for(int i = n; i <N;i++ ) {
            r +=data1[i] * data2[i-n];
        }
    } else {
        for(int i = 0; i <N+n;i++ ) {
            r +=data1[i] * data2[i-n];
        }
    }
    return  r ;
}


void crossCorrelationAnalysis(double* data1, double* data2, int N, double dt, double* xlist, double* ylist){
    for (int n=-N/2;n<N/2;n++){
        *xlist=n*dt;
        *ylist=CrossCorrelation(data1, data2,n,N);
        xlist++;
        ylist++;
    }
}
```

5 1/3 倍频程分析

```
function [Ya_1,Ya_2,f,fc,nfft,nc]=onethird(t,x)
%% 傅里叶变换
nfft = length(t);
fs=1/(t(2)-t(1));

Y1 = fft(x,nfft);     % 傅里叶变换

f1 = (0:nfft-1)/(nfft-1)*fs;      % 频率向量

f = f1(1:nfft/2);       % 根据对称性取半

Y = Y1(1:nfft/2)*2/nfft;   % 根据对称性取半

YE = abs(Y); % 频域中的能量

ji_Ya = 2*10^(-5);  % 声压级基底

%% 计算频谱声压级
Ya_1 = 20 * log10(YE/ji_Ya);  % 计算频谱声压级

%% 计算总声压级
% 找出10-8k
YE_z = YE(find(f>=10&f<8000));
Z_Ya = 20 * log10(sqrt(sum(YE_z.^2))/ji_Ya)-3

%% 三分之一倍频程
% 定义三分之一倍频程的中心频率fc
fc = [20 25 31.5 40 50 63 80 100 125 160 200 ...
    250 315 400 500 630 800 1000 1250 1600 2000 ...
    2500 3150 4000 5000 6300 8000 10000 12500 16000];
% 下限频率
fl = round(fc/(2^(1/6)));
% 上限频率
fu = round(fc*(2^(1/6)));

fu(end) = f(end);   % 修复fu，末尾变为16000

%%
```

```
%%
% 频率向量f中有L/2个数据，对应的频率是(0: L/2-1)/(L/2-1)*fs/2;
nl = round(fl*2/fs*(nfft/2-1) + 1); % 下限频率对应的频率向量的序号
nu = round(fu*2/fs*(nfft/2-1) + 1); % 上限频率对应的频率向量的序号
nc = length(fc); % 中心频率的长度

YE_C=zeros(1,nc);
for i = 1:nc
    nn = zeros(1,nfft)+1i*zeros(1,nfft);
    nn(nl(i):nu(i)) = Y1(nl(i):nu(i));
    nn(end-nu(i)+1:end-nl(i)+1) = Y1(end-nu(i)+1:end-nl(i)+1);
    cc = ifft(nn);
    YE_C(i) = sqrt(var(real(cc(1:nfft))));     % 求取1/3倍频程
%     YE_C(i) = sqrt(sum(YE(nl(i):nu(i)).^2)/2);    % 求取第i个中心频率的能量：频带的
end

Ya_2 = 20 * log10(YE_C/ji_Ya); % 计算中心频率的声压级
```

6 倒谱

```
function z=daopu_analysis(y)


% nn=1:nfft/2;
% ff=(nn-1)*fs/nfft;                % 计算频率刻度

Y=log(abs(fft(y)));                                % 按式(3-1-8)取实数部分
% subplot 211; plot(ff,Y(nn),'k'); hold on;      % 画出信号的频谱图
z=ifft(Y);                                         % 按式(3-1-8)求取倒谱
```