

A METHOD FOR POWER SPECTRAL DENSITY SYNTHESIS Revision B

By Tom Irvine

Email: tomirvine@aol.com

November 28, 2000

INTRODUCTION

The power spectral density function is described in Reference 1. Certain situations arise whereby a specified power spectral density must be converted to an acceleration time history for test or analytical purposes. Conventionally, the underlying time history has a random character. The purpose of this report is to give a synthesis method based on white noise.

PHASE ANGLE NOTE

The power spectral density calculation discards phase angles. Thus, a given power spectral density function does not have a unique corresponding time history.

METHOD OUTLINE

The synthesis steps are given in Table 1.

Step	Description
1	Generate a white noise time history. The initial overall level is arbitrary. The corresponding histogram should have a Gaussian or normal distribution. The sample rate should be ten times the highest frequency of interest. A sample program which performs this step is given in Appendix A. Further information is given in Reference 2.
2	Lowpass filter the time history. The cutoff frequency should be slightly above the maximum frequency of interest. This step is optional. Filtering algorithms are discussed in Reference 3.
3	Divide the data into manageable segments. The segment length is constrained by the maximum allowable input into the available Fast Fourier transform program. Note that only one segment may be needed if the computer software and hardware have sufficient memory.
4	Take the Fast Fourier transform of each segment, using the method in Reference 4. Use a rectangular window.
5	Scale the magnitude of each Fourier transform component in each segment so that it will meet the power spectral density specification.

Table 1. Synthesis Steps (continued)	
Step	Description
6	Take the inverse Fourier transform of each Fourier transform segment.
7	Reassemble the segments from the inverse Fourier transform. The resulting time history is the synthesized time history.
8	Take a histogram or probability density function of the time history to verify a normal distribution. Also check kurtosis, as explained in Appendix B.
9	Take the power spectral density of the time history to verify compliance with the specification.

SYNTHESIS EXAMPLE

Specification

The specification was taken as the component acceptance power spectral density in MIL-STD-1540C. The breakpoints are given in Table 2.

Table 2. MIL-STD-1540C Acceptance Level, 6.1 GRMS Overall	
Frequency (Hz)	PSD (G ² /Hz)
20	0.0053
150	0.04
600	0.04
2000	0.0036

Step 1

A white noise time history was generated. Note that this time history is independent of the specification, except for duration and sample rate.

Steps 2 through 6

The white noise signal was lowpass filtered and then divided into segments. Each segment was manipulated in terms of Fourier spectra.

Step 7

The resulting time history is shown in Figure 1.

Step 8

The kurtosis value is 3.02, which is consistent with random vibration. Kurtosis is explained in Appendix B.

The probability density function is given in Figure 2. It has the shape of a normal distribution. Note that a probability density function is calculated by dividing the histogram curve by the total number of points.

Step 9

The power spectral density of the synthesized time history is shown in Figure 3. The specification level is also shown in Figure 3. The agreement is excellent.

ALTERNATE METHOD

A series of sine functions may be used to synthesize a time history which satisfies a power spectral density. This method does not produce a random time history, however.

REFERENCES

1. T. Irvine, An Introduction to Spectral Functions, Vibrationdata.com Publications, 1998.
2. T. Irvine, Integration of the Normal Distribution Curve, Vibrationdata.com Publications, 1998.
3. T. Irvine, An Introduction to the Filtering of Digital Signals, Vibrationdata.com Publications, 1998.
4. T. Irvine, The Fast Fourier Transform (FFT), Vibrationdata.com Publications, 1998.

SYNTHESIZED TIME HISTORY

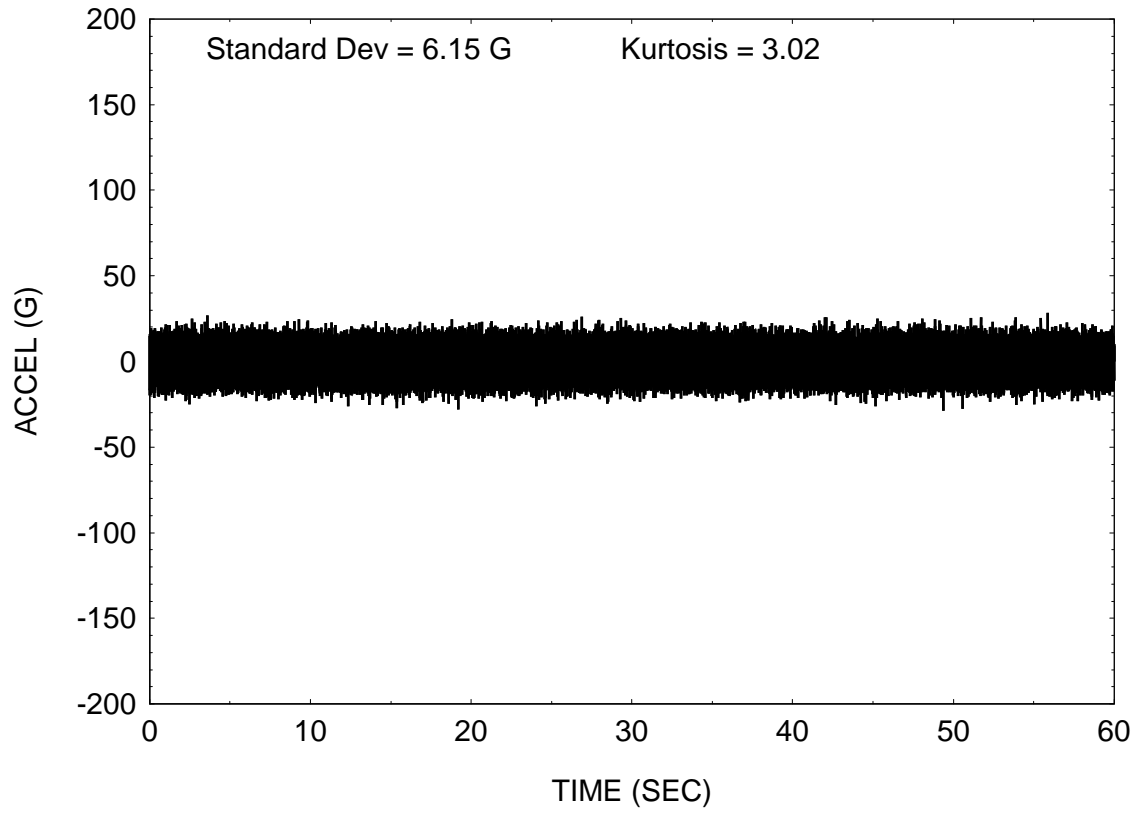


Figure 1.

PROBABILITY DENSITY FUNCTION OF SYNTHESIZED TIME HISTORY 60 SEC DURATION

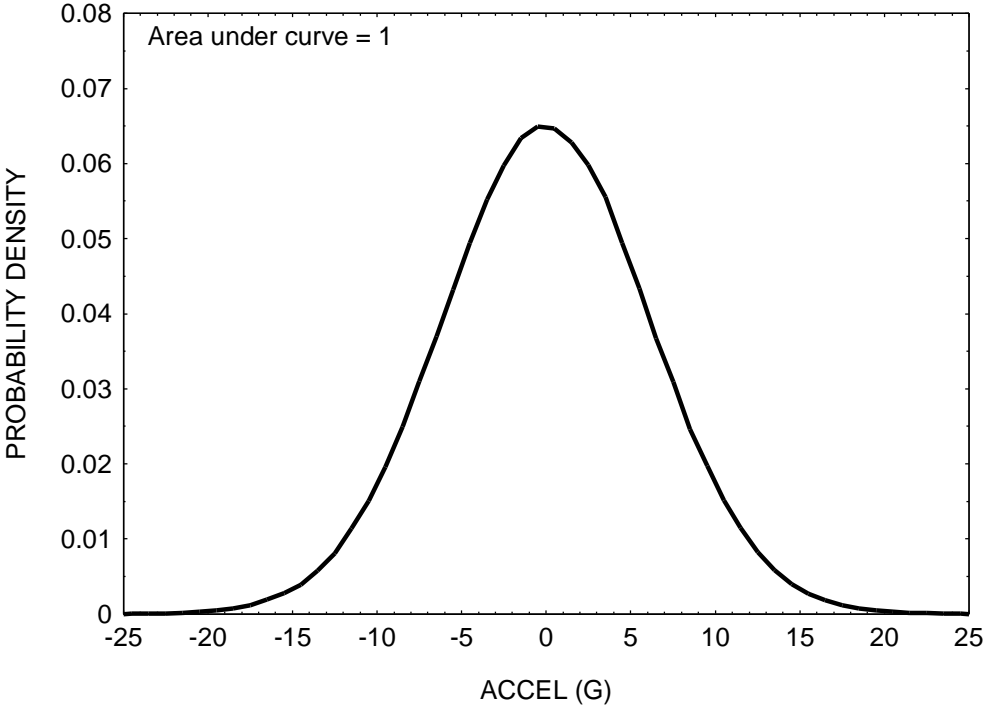


Figure 2.

POWER SPECTRAL DENSITY OF SYNTHESIZED TIME HISTORY 60 SEC DURATION

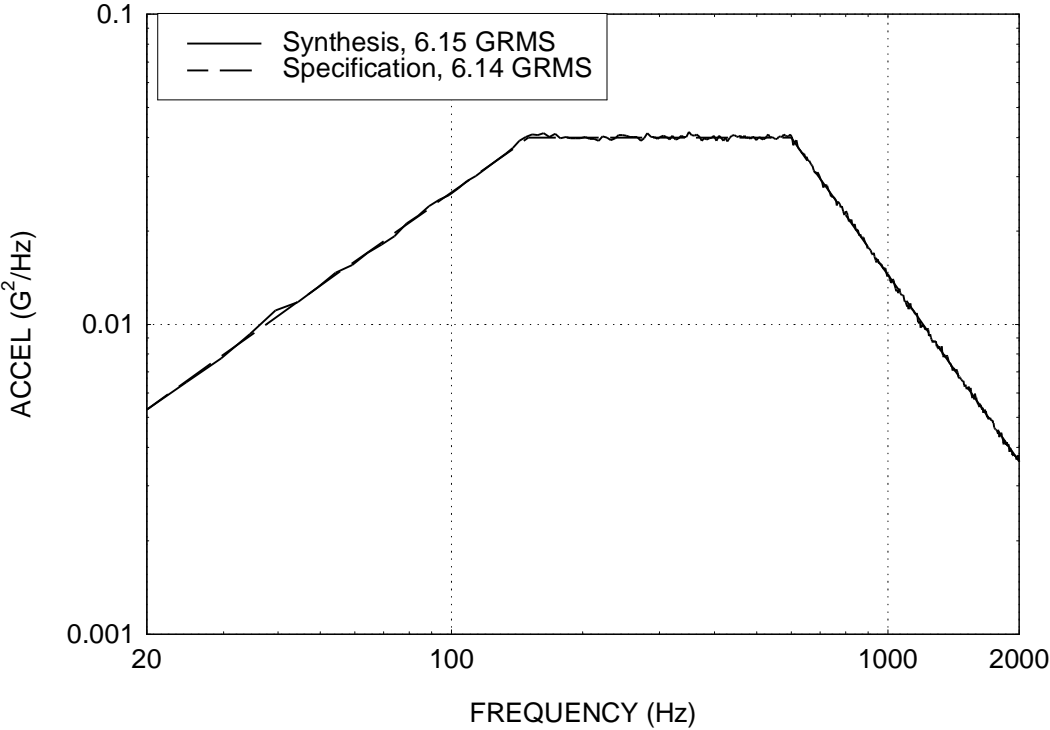


Figure 3.

APPENDIX A

WHITE NOISE PROGRAM

Method

The white noise program uses a random number function to generate a number between 0 and 1. This number then represents a probability area under the normal distribution curve. The corresponding sigma value is then determined from the probability area. This sigma value is taken as the amplitude. A second random number is then generated to determine whether the amplitude is positive or negative.

Source Code

```
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

#define MAX 2000000

void gauss(void);

long i,j,np;

double a[MAX],aq,ave,delta,dt,ms,sdg,sr,ss,stddev,t,tmax,var,x,y,z;
double b[1000],sum[1000];

const double pi=atan2(0.,-1.);
const double e=1./sqrt(2.*pi);
const double max=32767.;

char filename[2][16];
FILE *pFile[2];

void main()
{

printf("\n ");
printf("\n white.cpp \n");
printf("\n by Tom Irvine");
printf("\n Email: tomirvine@aol.com ");
printf("\n ");
printf("\n This program generates a white noise time history. \n");

strcpy(filename[0],"a.out");
pFile[0]=fopen(filename[0],"w");
```

```

printf("\n Enter sampling rate (sample/seconds): \n");
scanf("%lf",&sr);

dt=1./sr;

printf("\n Enter duration (sec): \n");
scanf("%lf",&tmax);

printf("\n Enter standard deviation: \n");
scanf("%lf",&sdg);

np=long(tmax/dt);

if(np > MAX){np=MAX;}

printf("\n np= %ld",np);

if( np > 2 )
{
// The gauss function determines the probability for
// each Nsigma value.

    gauss();

    ave = 0.;
    ms = 0.;

    for(i=0; i<np; i++)
    {
        x=rand()/max;
        a[i]=0.;

        for(j=1; j<= 599; j++)
        {

// x is a probability value.  0 < x < 1

                if( x >= sum[j] && x <= sum[j+1] )
                {

// a = Nsigma value which corresponds to probability x.

                        a[i]=b[j];
                        break;
                }
        }

// 50% of the amplitude points are multiplied by -1.

        y=rand()/max;

        if(y <= 0.5){a[i]*=-1;}

```



```

        ave+=a[i];
        ms+=pow(a[i],2.);
    }

    ave/=double(np);
    ms/=double(np);

    var=ms-pow(ave,2.);
    stddev=sqrt(fabs(var));

//    print the output file

    for(i=0; i<np; i++)
    {
        t=dt*i;
        a[i]=(a[i]-ave)*(sdg/stddev);
        fprintf(pFile[0],"%14.7e %14.7e \n",t,a[i]);
    }

    printf("\n\n The output file is:  a.out ");
    printf("\n\n This file has two columns:  time(sec) and amplitude.");
    printf("\n Please call this file into your own plotting program.\n\n");

}
fclose(pFile[0]);

printf("\n Press any key to exit.");
getch();

exit(1);

}

void gauss(void)
{
    strcpy(filename[1],"gauss.out");
    pFile[1]=fopen(filename[1],"w");

/*
    b[j] is a standard deviation.  The maximum b[j] is six sigma.

    sum[j] is the area under the Gaussian curve from 0 to b[j].
*/

    double ab;

    for( j=1;j<=600;j++)
    {
        b[j]=j/100.;
        ab=-b[j];
        ss=0.;
        delta=(b[j]-ab)/1000.;

        for( i=0; i<=1000; i++)
        {

```

```
        z=(delta*i)+ab;
        ss+=exp( ( -pow(z,2.) )/2.);
    }

    sum[j]=ss*e*delta;
    aq=1.-sum[j];

    fprintf(pFile[1]," %14.7e %14.7e %14.7e \n",b[j],sum[j],aq);
}
fclose(pFile[1]);
}
```

APPENDIX B

KURTOSIS THEORY

Kurtosis is a measurement of the “peakedness” of the data. A sine wave, for example, has a kurtosis value of 1.5. A pure Gaussian random signal has a kurtosis value of 3.0. A higher kurtosis value may indicate the presence of higher sigma peaks on the time history than would be expected from a Gaussian distribution. These peaks could, for example, represent a transient event. On the other hand, they may simply represent random vibration which has a probability density function which departs from the Gaussian ideal.

The kurtosis equation is:

$$\text{kurtosis} = \frac{\sum (x - \bar{X})^4}{n\mathbf{s}^4} ,$$

where n = number of samples,

\mathbf{s} = standard deviation,

\bar{X} = mean value.

Note that the kurtosis value appears dimensionless.