#### A Comparison of the Consecutive Peak and Rainflow Cycle Counting Methods for Cumulative Damage Calculation

By Tom Irvine Email: tom@vibrationdata.com

October 23, 2014

## **Introduction**

The purpose of this paper is to compare two different methods for counting fatigue cycles for response time histories. The results from either method can then be fed into a Miners-type cumulative damage summation.

The first simply counts each consecutive local absolute peak as a half-cycle. This method essentially uses the Rayleigh distribution of peaks for the case of the narrowband response of a spring-mass oscillator to broadband random base input or applied force excitation.

The second uses the rainflow counting method from Reference 1. This method establishes cycles based on absolute local peaks which are not necessarily consecutive. It effectively ranks the differences between the permutations of possible peak pairings and then extracts cycles accordingly.

This *Rainflow* method is considered to be more precise than the consecutive peak counting method.

The Consecutive Peak method is faster and more conservative.

The purpose of this paper is to assess this conservatism via numerical examples.

The results will show that the Consecutive Peak method can be used as a "quick look" estimation of the cycle count. The counting can then be considered complete if the resulting damage shows positive margin with respect to the failure threshold. Otherwise, the Rainflow counting method can be used if "sharpening the pencil" is needed.

In addition, the Consecutive Peak method can be used as an upper limit check of the Rainflow results, for cases where Rainflow is the primary method.

# Background

The Rainflow calculation for the second method is conceptually very simple. In reality, this method requires complex, subtle bookkeeping. The programming implementation must be very precise.

There are anecdotal reports that various rainflow counting software programs give significantly different results. Some of this could simply be due to using range instead of amplitude as the convention.

Note that:

Range = (peak-valley) Amplitude = (peak-valley)/2

But this possible explanation does not seem to cover all of the informally reported discrepancies.

Another concern is that Reference 1 presents four variants of the rainflow cycle counting method. Any difference between these variations is amplified when the fatigue exponent is included in the Miners calculation.

Furthermore, the Rainflow calculation can require long computation times for lengthy random response time histories. This requires intermediate amplitude array elements to be deleted throughout the counting. The C/C++ language handles this fairly efficiently, but interpreted languages like Matlab and Python can be very slow in this regard.

In contrast the Consecutive Peak counting method is very fast in all common programming languages.

## Numerical Experiment

Six signals were generated to simulate response time histories. The signals vary as to realism, but were chosen to represent some extreme cases. The amplitude is simply referred to as *unit*.

Three sinusoidal signals were generated, each with duration of 180 seconds. The phase angle was set to zero for each sine tone.

Sine 1 had parameters (1 unit, 100 Hz).

Sine 2 had parameters (1 unit, 190 Hz) added to Sine 1.

Sine 3 had parameters (1 unit, 360 Hz) added to Sine 2.

Three random signals were generated, each with duration of 60 seconds.

A band-limited white noise signal, *Random* k=3, was generated with a Gaussian distribution and a kurtosis=3. The signal had a standard deviation of 1 unit RMS. The sample rate was 20 KHz, with lowpass filtering at 2 KHz.

The previous white noise signal was applied as a base input to a spring-mass system with a natural frequency = 400 Hz and Q=10, as if it were an acceleration time history. The *NB* acceleration response was narrowband random with a Gaussian distribution. The response can be considered to represent a generic amplitude for the purpose of this study.

A band-limited white noise signal was generated and then modified so that it had a non-Gaussian distribution with a kurtosis=5. The signal had a standard deviation of 1 unit RMS. The sample rate was

20 KHz, with lowpass filtering at 2 KHz. This *Random* k=5 response is highly unlikely due to central limit theorem, but is nevertheless useful for a comparison case.

Each of the six methods was then applied to the Consecutive Peak and Rainflow counting methods. The relative damage was then calculated for two fatigue exponents b=4 & 9 using the method in Appendix A.

The damage unit is: (unit^b), where b is the fatigue exponent.

Table 1. Damage Result Comparison						
	b=4		b=9		b=4	b=9
Туре	Rainflow	Consecutive Peak	Rainflow	Consecutive Peak	Ratio	Ratio
Sine 1	18000	18000	18000	18000	1.0	1.0
Sine 2	1.43e+05	1.64e+05	2.2e+06	3.55e+06	1.1	1.6
Sine 3	6.13e+05	6.65e+05	6.31e+07	7.43e+07	1.1	1.2
NB	8.84e+04	1.07e+05	2.00e+07	2.61e+07	1.2	1.3
Random, k=3	4.73e+04	5.81e+04	5.80e+06	7.74e+06	1.2	1.3
Random, k=5	8.61e+05	1.06e+06	8.68e+08	1.12e+09	1.2	1.3

Ratio = (Consecutive Peak)/Rainflow

#### **Conclusions**

The Consecutive Peak method gave results which were consistently greater than or equal to the Rainflow technique.

The Consecutive Peak results were at most 30% higher than the Rainflow damage excluding the Sine 2, b=9 case.

The Consecutive Peak method thus appears to be a suitable upper limit estimation method for fatigue cycles and the resulting damage.

#### Postscript

Vibration fatigue calculations are "ballpark" calculations given uncertainties in S-N curves, stress concentration factors and other variables. Perhaps the best that can be expected is to calculate the accumulated fatigue to the correct "order-of-magnitude."

# <u>References</u>

1. ASTM E 1049-85 (2005) Rainflow Counting Method, 1987.

## APPENDIX A

Relative Fatigue Damage

The fatigue damage D is

$$D = \sum_{i=1}^{m} A_i^b n_i \tag{A-1}$$

where

- A<sub>i</sub> is the acceleration amplitude from the response cycle analysis
- n<sub>i</sub> is the corresponding number of cycles
- b is the fatigue exponent