SAND2006-7240C

Removing Undesired Periodic Data from Random Vibration Data*

Curtis F. Nelson and Thomas G. Carne
Members of the Technical Staff
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185

Abstract

When measuring the structural dynamic response of test objects, the desired data is sometimes combined with some type of undesired periodic data. This can occur due to N-per-revolution excitation in systems with rotating components or when dither excitation is used. The response due to these (typically unmeasured) periodic excitations causes spikes in system frequency response functions (FRFs) and poor coherence.

This paper describes a technique to remove these periodic components from the measured data. The data must be measured as a continuous time history which is initially processed as a single, long record. Given an initial guess for the periodic signal's fundamental frequency, an automated search will identify the actual fundamental frequency to very high accuracy. Then the fundamental and a user-specified number of harmonics are removed from the acquired data to create new time histories. These resulting time histories can then be processed using standard signal processing techniques.

An example of this technique will be presented from a test where a vehicle is dithered with a fixed-frequency, sinusoidal force to linearize the behavior of the shock absorbers, while measuring the acceleration responses due to a random force applied elsewhere on the vehicle.

Introduction and Motivation

We have been performing structural dynamics testing on an automobile where its four wheels/tires have been removed and replaced by square metal plates. In order to softly support the vehicle, each of these metal plates was attached to the top of an air spring sitting on the floor of the laboratory. In the testing, a shaker was used to apply continuous, random, force excitations to the metal plates and the resulting acceleration responses were measured on these plates and at various locations in and on the vehicle.

For the small force levels used, the vehicle's shock absorbers usually act as rigid members because the applied forces are not large enough to overcome the initial friction. In other cases, the shocks "break free" for portions of the excitation, but then remain "locked up" the rest of the time. Neither case is desirable because the measured acceleration-over-force FRFs are dependent on applied force levels and are not repeatable.

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

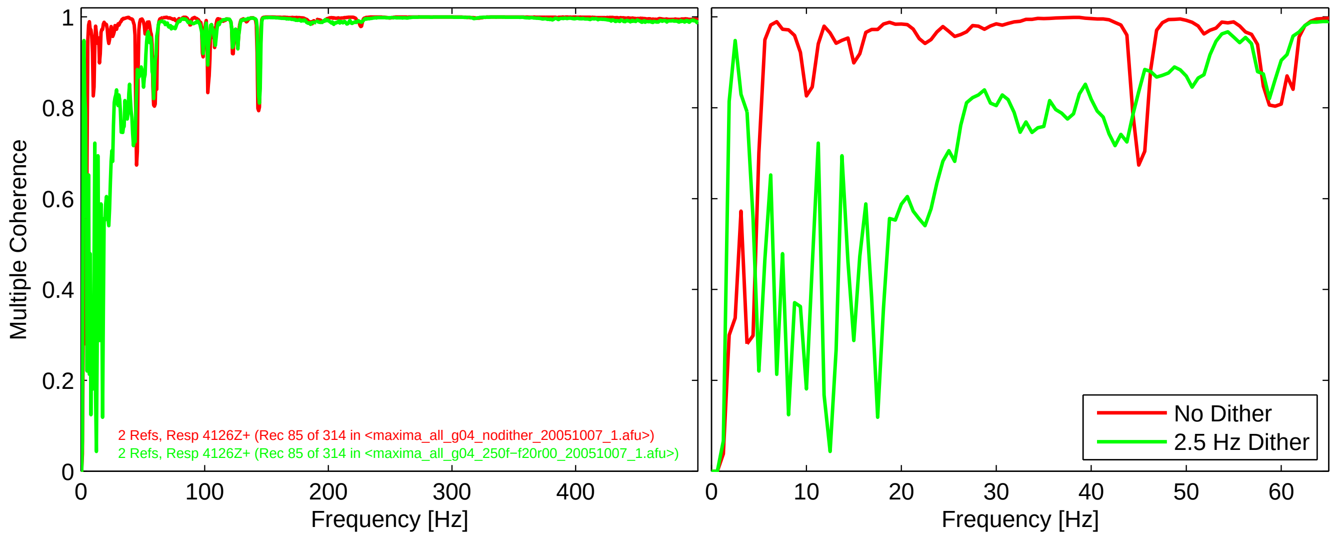

Figure 1: Coherence of test system with and without 2.5 Hz dither (the low-frequency region has been expanded in the plot on the right).

To address this problem, we "dither" the vehicle to ensure that the shocks are always broken free. This is done by placing an additional shaker under the vehicle and then exciting the vehicle with a single-frequency, sinusoidal force in order "break free" and achieve relative motion across all the shocks. To minimize the required dither force, the dither frequency was chosen to excite the rigid body modes of the vehicle bouncing on its suspension (typically below 5 Hz). Now the measured accelerations include the response due to the desired random excitation input plus the response due to the dither input. Because the dither frequency is well below the first elastic mode of the vehicle, we don't expect the dither to affect the measured FRFs in our frequency range of interest.

However when we acquire data, the resulting FRFs show many sharp peaks at multiples of up to 10 times the dither frequency. Also, there are sharp drops in coherence at these same multiples of the dither frequency. Even though the dither frequency is well below the frequency range of interest, these peaks at multiples of the dither frequency are now getting up into the range of interest. As an example, data was acquired for a continuous random force of 17.8 N (4 lbf) RMS applied to the front, left metal plate while 2.5 Hz, 89.0 N (20 lbf) RMS dithering was applied under the front of the vehicle. Figure 1 shows the coherence for the driving-point FRF on the metal plate with and without dither applied. While the coherence is close to 1.0 for most of the frequency range, at frequencies below 50 Hz it is dramatically worse when the dither is applied. Because the FRF is based only on the random force, we expect the coherence to be poor at 2.5 Hz while dithering because the measured acceleration at 2.5 Hz is due to both the random and dither forces — we essentially have an unmeasured input and this will reduce the coherence. However, the reduced coherences at all the other frequencies up to 50 Hz must be due to another factor.

To learn why all these harmonics were appearing, an accelerometer was added to the top and bottom of each shock. The typical motion of a shock is shown in Figure 2. The figure shows that the random motion of the shock is combined with the periodic motion due to the dithering. If one looks at the motion of the top and bottom of the shock after averaging over many dither cycles, the effect of dithering on the plate motion becomes apparent. The top of the shock, which is directly attached to the body of the car, has an average acceleration that looks like a 2.5 Hz sinusoid; just like the applied dither force. However, the average motion of the bottom of the shock (attached to the wheel spindle and metal plate) is still periodic, but has picked up a large amount of third and higher harmonics of the dither frequency. The shock absorber, while broken free, is definitely not acting in a linear fashion. When the body side of the shock is excited with a low-frequency sinusoid, the opposite end (which is attached to the metal plate on which we are measuring FRFs) sees motion with many harmonics that extend up into the frequency range of interest. The coherence drops because these harmonics due to the dithering appear in the acceleration responses measured on the metal plate without there being an associated coherent input force at the same frequency (there is still random input force at all frequencies, including the dither harmonic frequencies).

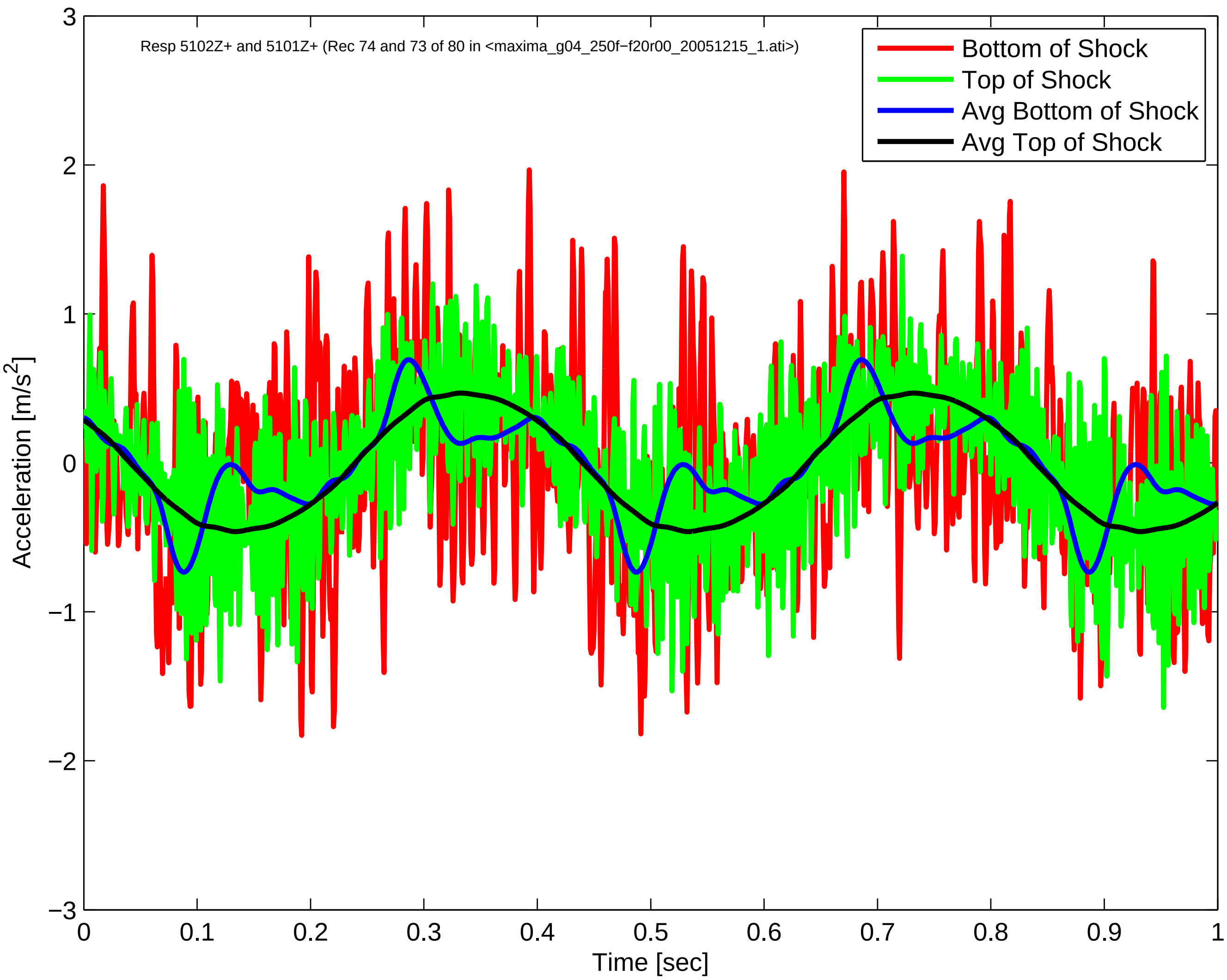

Figure 2: Measured acceleration at the top and bottom of a shock absorber when both random and dither forces are applied to the vehicle system.

In order to improve the FRFs and coherences, we decided to try to remove the periodic dither responses from our measured data. However, this is not possible if we only save our acquired data as averaged FRFs. For Figure 1, the data was saved as FRFs averaged over 120 non-overlapping frames with a frequency resolution of 0.625 Hz. This means that the harmonics of the 2.5 Hz dither occur every 4 frequency lines. In the range over which the dither harmonics are present, the desired random data cannot be known because it is hidden under a series of spikes that has a peak every 4 frequency lines.

The solution to this problem is to save time-history data instead of averaged FRF data. In fact, a recent article by Brandt, *et al.* [1] advocates always recording time-history data. The authors argue that the usual practice of only saving averaged spectral data and discarding the time data is a remnant held over from the past, when computer memory and hard disk storage were extemely limited compared to what is available today. By saving the raw time-history data, it becomes possible to later analyze the data with different frequency resolutions or windowing, to investigate glitches or overranges, or (the subject of this paper) to remove the presence of periodic noise that is present in the data.

In fact, improvements in computer capabilities are also what makes the technique described in this paper possible. While removing periodic noise from random data has been done before, for example see Tucker [3], the technique described in this paper is very fast and can be implemented in only tens of computer code lines using the computer hardware and technical computer languages/tools available today. In past years, this technique would not have been practical because of the amount of memory and computations required.

Mathematical Development

We would like to fit a time-history data record, y, with N harmonics of a sinusoid with fundamental frequency ω. This can be written as

$$y \approx c_0 + c_1 \sin(\omega t + \theta_1) + c_2 \sin(2\omega t + \theta_2) + \cdots + c_N \sin(N\omega t + \theta_N).$$

Since $c_k \sin(k\omega t + \theta_k) = c_k[\cos(k\omega t)\sin\theta_k + \sin(k\omega t)\cos\theta_k] = a_k \cos(k\omega t) + b_k \sin(k\omega t)$, the equation can be rewritten as

$$y \approx c_0 + [a_1 \cos(\omega t) + b_1 \sin(\omega t)] + [a_2 \cos(2\omega t) + b_2 \sin(2\omega t)] + \cdots + [a_N \cos(N\omega t) + b_N \sin(N\omega t)].$$

In either case there are $(2N+1)$ unknowns that must be calculated. In the first equation, half the unknowns are amplitudes and half are phases, whereas all the unknowns are amplitudes in the second equation. Using the second equation makes it easier for an optimizer to converge on the $(2N+1)$ unknowns because all the coefficients have similar sensitivities (an amplitude and phase have different sensitivity) and are not periodic (phases are periodic every 2π). The second equation can be written in matrix form as

$$\{y\} = \left\{ \begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_{\mathrm{nPts}} \end{array} \right\} \approx \left[\begin{array}{ccccccc} 1 & \cos(\omega t_1) & \cdots & \cos(N\omega t_1) & \sin(\omega t_1) & \cdots & \sin(N\omega t_1) \\ 1 & \cos(\omega t_2) & \cdots & \cos(N\omega t_2) & \sin(\omega t_2) & \cdots & \sin(N\omega t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos(\omega t_{\mathrm{nPts}}) & \cdots & \cos(N\omega t_{\mathrm{nPts}}) & \sin(\omega t_{\mathrm{nPts}}) & \cdots & \sin(N\omega t_{\mathrm{nPts}}) \end{array} \right] \left\{ \begin{array}{c} c_0 \\ a_1 \\ \vdots \\ a_N \\ b_1 \\ \vdots \\ b_N \end{array} \right\}, \qquad (1)$$

where y is a $\mathrm{nPts} \times 1$ column vector (there are nPts samples in the time history), the sinusoidal harmonics are a $\mathrm{nPts} \times (1+2N)$ matrix, and the coefficients are a $(1+2N) \times 1$ column vector. The sampling frequency used to acquire the time-history data is used to generate the time values, t_i, for the sinusoidal harmonics. If we know the fundamental frequency, ω, then the harmonics matrix and its pseudo-inverse (denoted by a $+$ superscript) can be calculated. Now the $(2N+1)$ coefficients can be calculated in a least-squares sense by multiplying the pseudo-inverse of the harmonics matrix by the time history.

$$\left\{ \begin{array}{c} c_0 \\ a_1 \\ \vdots \\ a_N \\ b_1 \\ \vdots \\ b_N \end{array} \right\} = \left[\begin{array}{ccccccc} 1 & \cos(\omega t_1) & \cdots & \cos(N\omega t_1) & \sin(\omega t_1) & \cdots & \sin(N\omega t_1) \\ 1 & \cos(\omega t_2) & \cdots & \cos(N\omega t_2) & \sin(\omega t_2) & \cdots & \sin(N\omega t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos(\omega t_{\mathrm{nPts}}) & \cdots & \cos(N\omega t_{\mathrm{nPts}}) & \sin(\omega t_{\mathrm{nPts}}) & \cdots & \sin(N\omega t_{\mathrm{nPts}}) \end{array} \right]^+ \left\{ \begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_{\mathrm{nPts}} \end{array} \right\}. \qquad (2)$$

By slightly rewriting Eqn. 1, the best fit to y using the N harmonics is

$$\{y_{\mathrm{periodic}}\} = \left[\begin{array}{ccccccc} 1 & \cos(\omega t_1) & \cdots & \cos(N\omega t_1) & \sin(\omega t_1) & \cdots & \sin(N\omega t_1) \\ 1 & \cos(\omega t_2) & \cdots & \cos(N\omega t_2) & \sin(\omega t_2) & \cdots & \sin(N\omega t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos(\omega t_{\mathrm{nPts}}) & \cdots & \cos(N\omega t_{\mathrm{nPts}}) & \sin(\omega t_{\mathrm{nPts}}) & \cdots & \sin(N\omega t_{\mathrm{nPts}}) \end{array} \right] \left\{ \begin{array}{c} c_0 \\ a_1 \\ \vdots \\ a_N \\ b_1 \\ \vdots \\ b_N \end{array} \right\}, \qquad (3)$$

and the time history with the harmonics removed is

$$\{y_{\text{random}}\} = \{y\} - \{y_{\text{periodic}}\}. \tag{4}$$

Implementation of Technique

This technique has been implemented as a MATLAB [2] code. To demonstrate how easy this technique is to implement, complete MATLAB code listings are included at the end of this paper. In general, the harmonics removal process is implemented as follows:

- Measure and save your data as time-history records. Each time-history record contains the response due to the random input plus the stationary input from the dither shaker. It is VERY important that the time history MUST be continuous (*i.e.*, there can be no gaps in the time record) for this technique to work. Many data acquisition software packages have the capability to automatically discard records when an overrange is detected — this option must NOT be used. Instead, the user must accept all records, so it is important to properly set acquisition voltage ranges before running the test.
- Find the fundamental dither frequency using an automated optimization on a time-history record in which the stationary signal shows up strongly (we use the measured dither force for this). Because the fundamental frequency must be calculated extremely accurately,[1] using a discrete Fourier transform to find the closest frequency line is not good enough, nor is using the frequency setting displayed on your function generator. However, either one of these methods would provide a good initial guess for the fundamental frequency.

  After the time-history data has been initially read into MATLAB, the user has to make an initial guess for the fundamental frequency. In our application, we automatically generate this initial guess from the dither frequency that is applied to the vehicle. The time and data values for the time-history record that will be used to find the fundamental frequency is then saved to a MATLAB MAT-file — this is necessary because the optimization routine needs to call a function over and over again and there is no simple way to save the time-history data between function calls; thus it is faster to read the data each time from a MAT-file. An optimization routine[2] then varies the fundamental frequency until the squared error between the time history and the N harmonics is minimized. To speed up the optimizations, the fundamental frequency is calculated only using a small number of harmonics (typically use 3 harmonics, which gives 7 unknown coefficients).
- Use the fundamental frequency with Eqn. 2 to perform a least-squares calculation of the $(2N + 1)$ coefficients of the periodic fit for each time-history data record. In the sample code included with this paper, each time history is fit with 17 harmonics, which gives 35 unknown coefficients.
- Use Eqn. 3 to calculate the periodic fit for each time-history data record.
- Use Eqn. 4 to remove the response due to the stationary input from the measured data, leaving only the response due to the random input.
- Now perform the desired processing on the resulting random time-history data.

Example Application of Technique

As an example of using this technique, some additional data will be shown for the automobile test and conditions previously described. This data was acquired at 1280 Hz for a period of 192 sec (total of 245,760 samples per

[1] An error of Δf Hz in the fundamental frequency will give a phase error of $\Delta\theta = 360T\Delta f$ degrees at the end of a T second sampling period. For the example in this paper, the fundamental frequency must be known to within 0.0000145 Hz in order for the phase to be off less than 1° at the end of the 192 second data record.

[2] Used the Nelder-Mead simplex (direct search) method as implemented by the `fminsearch()` function in MATLAB; this is a multidimensional, unconstrained nonlinear minimization.

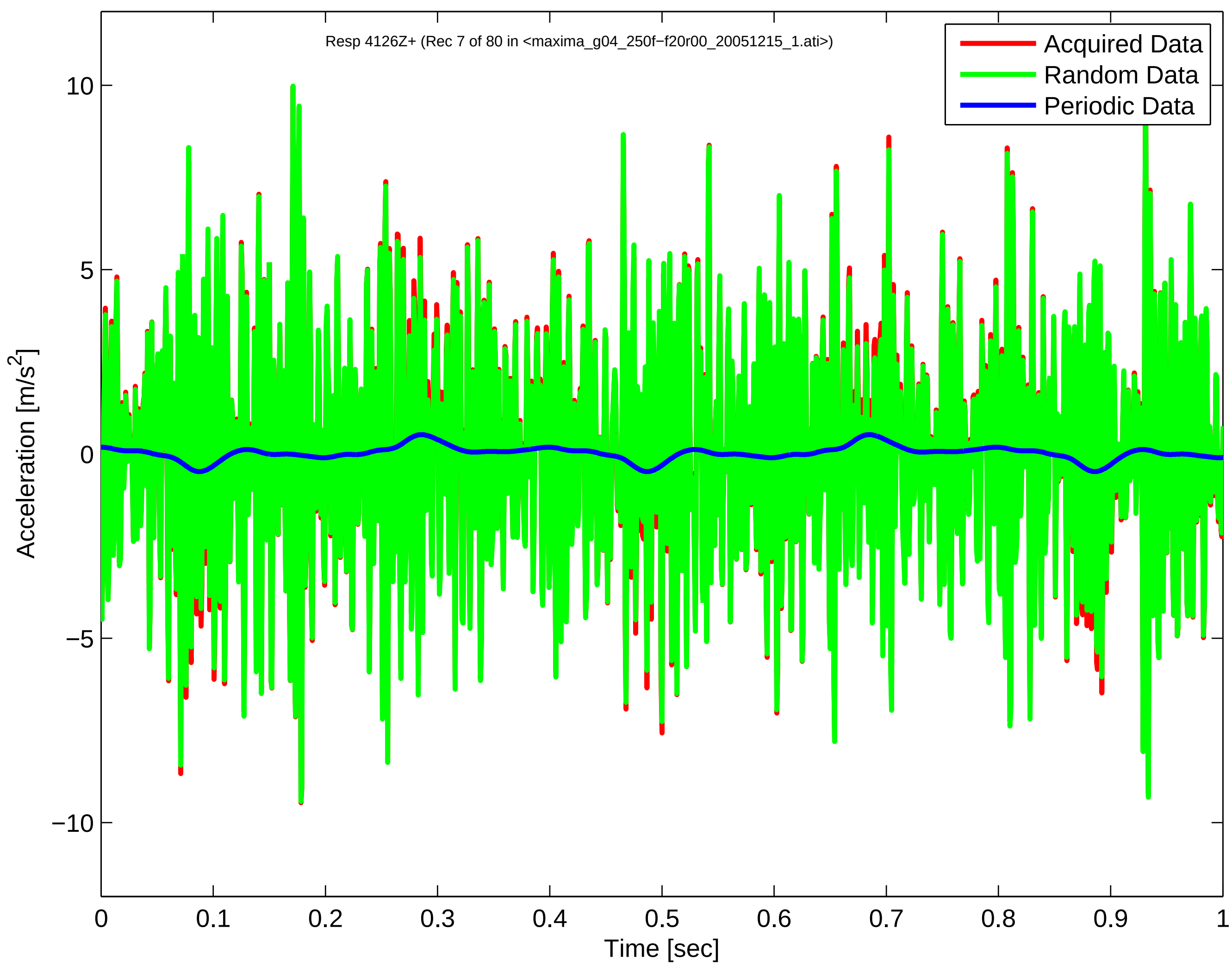

Figure 3: Comparison of a typical acceleration time-history showing its acquired, periodic, and random components.

acquisition channel). While the dither frequency was set to 2.50 Hz on the signal generator[3], the actual dither frequency was calculated to be 2.5052675 Hz with its 17^{th} harmonic at 42.590 Hz. Equations 3 and 4 were then used to identify and remove the stationary, periodic components from all the acquired time-history data records, leaving time-history records that correspond only to the random input.

Figure 3 shows the first second of the driving-point acceleration response on the metal plate. The periodic acceleration is only a small fraction of the measured data, as the acquired acceleration and the random acceleration time histories appear almost identical to the eye. Figure 4 shows the same acquired data in the frequency domain, where the time history was processed as a single, 245,760-sample record (0.00521 Hz resolution). In the Fourier transforms, the spikes due to the dither and its harmonics dominate the acquired data in the low-frequency range that has been plotted. For this data, the 2.5 Hz fundamental and its odd harmonics at 7.5 Hz, 12.5 Hz, etc. are more prominent than the even harmonics. However, because the Fourier transform of the periodic response is a series of extremely narrow spikes, these spikes are only a very small portion of the energy in the signal. For the acquired data shown in Figs. 3 and 4, the random data accounts for 99.5% of the mean-squared acceleration and the periodic data is only 0.5%.

Now we can post-process the time-history records due to the random input to estimate FRFs in the usual fashion

[3]The signal generator used for the testing has a digital readout that allowed the frequency to be set with a resolution of 0.01 Hz near 2.5 Hz.

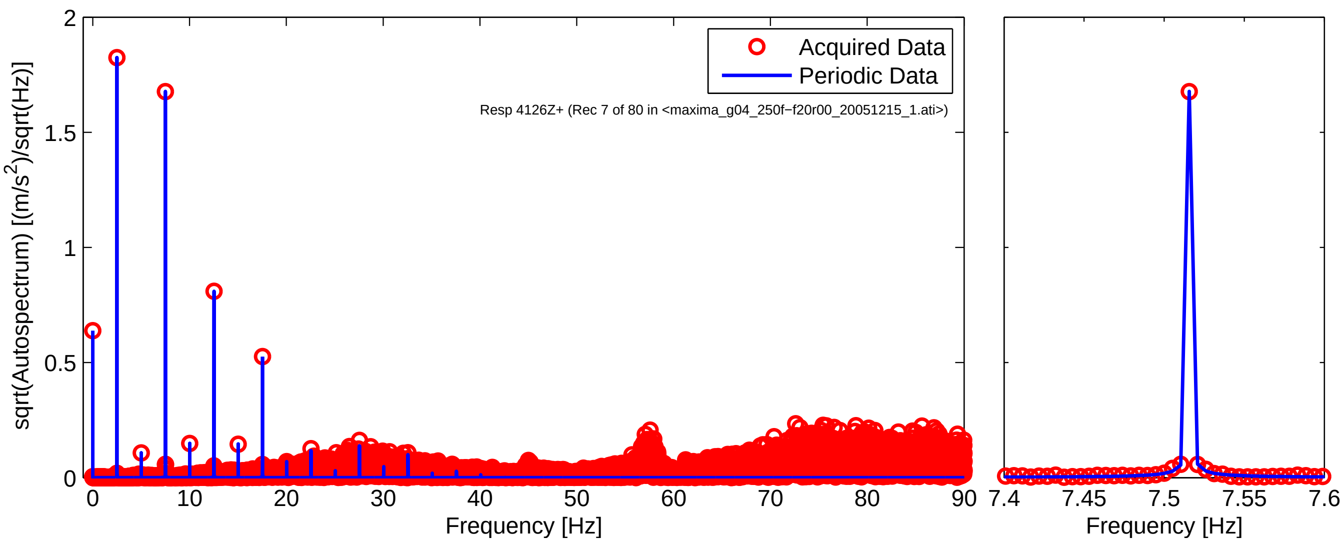

Figure 4: Fourier transforms of the acquired acceleration time history and the fit periodic response (only the low-frequency region is shown).

where each record is broken into many segments which are averaged to improve the estimates. In Figure 5, the time history has been broken up into 239 half-overlapping segments of 2048 samples (0.625 Hz resolution) which were windowed using a Hann window to calculate the driving-point FRF and coherence. At low frequencies, the removal of the periodic data from the acquired data has eliminated the extraneous spikes from the FRF and improved the coherence. There is no effect at high frequencies which is expected because we only removed harmonics up to 42.6 Hz. In fact, we now see that there is a small peak in the FRF at 5 Hz which was previously hidden by the dither harmonics.

While the dither removal has improved the coherence estimates at low frequencies, they still are not close to 1.0. This is probably because the nonlinear shocks are still present in the system and motion across them, caused by the random force, will still generate motion at harmonics of the relative motion.

Conclusion

In this paper, we have developed and demonstrated a simple-to-understand and simple-to-implement technique for identifying a stationary, periodic signal in what is otherwise random vibration data. This periodic data can be removed from acquired time-history data, resulting in random data which can then be post-processed in the usual fashion. To use this technique, the acquired data must be saved as time-history data instead of averaged frequency-domain data.

The technique can also identify periodic motion in the presence of random noise. This works very well when compared to low-pass filtering, as it produces the true stationary, periodic signal instead of one with cycle-to-cycle variation. In addition, the technique also avoids the initial transient at the start of the time-history record that will occur unless the data is windowed to force it to start at zero.

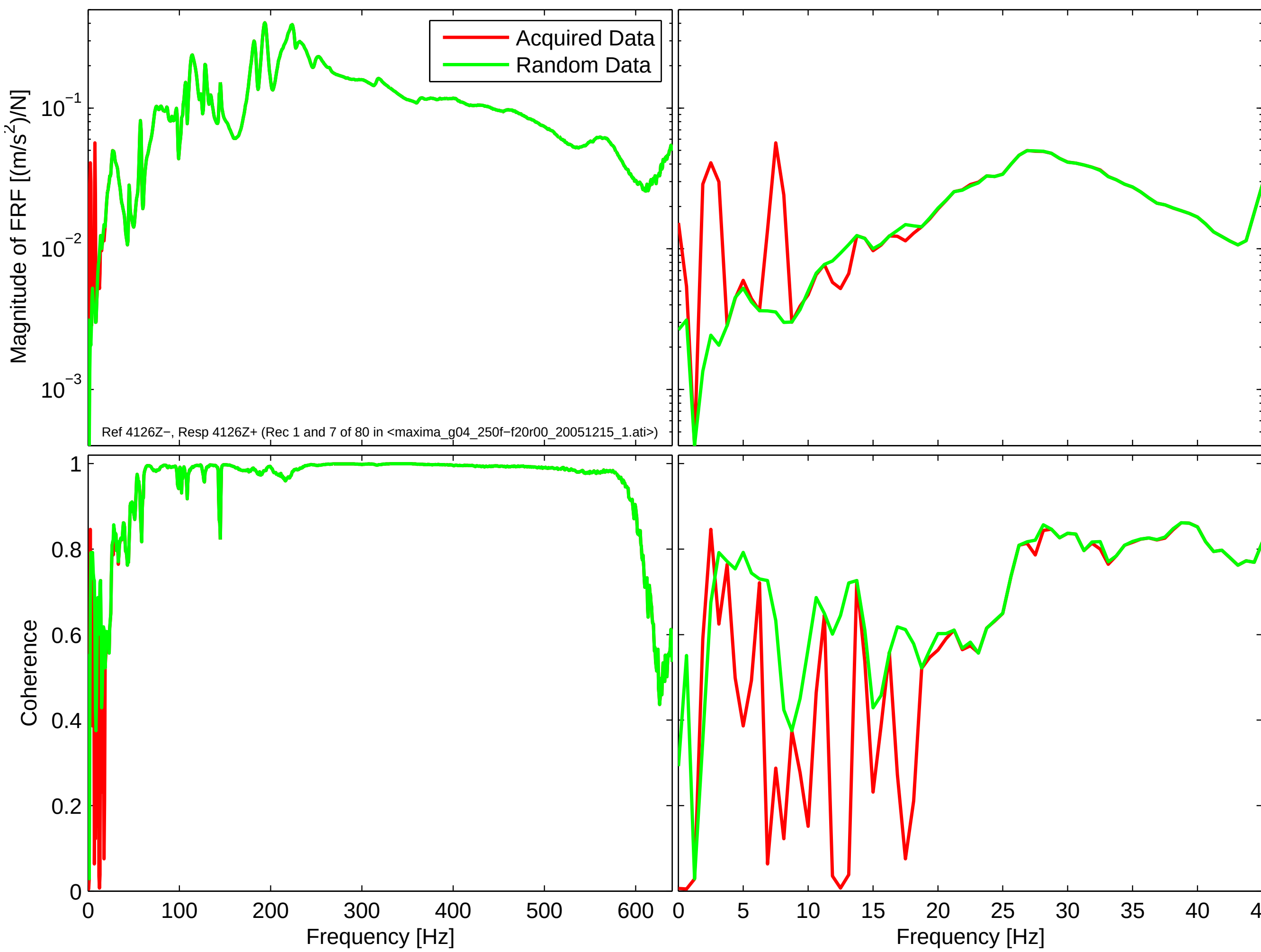

Figure 5: Driving-point FRF and coherence of acquired and random acceleration response signals (the low-frequency region has been expanded in the plots on the right).

References

[1] Brandt, A., Ahlin, K., and Lagö, T., "Noise and Vibration Measurement System Basics," *Sound and Vibration*, Vol. 40, No. 4, April 2006, pp. 9–11 (ISSN 1541-0161).

[2] MATLAB (MATrix LABoratory) is a high-level language and interactive environment for technical computing; it is a commercial software package by The MathWorks, Inc. (Natick, MA).

[3] Tucker, M.D., "An Adaptive Filtering Technique for Removing Periodic Noise from Modal Response Data," *Proceedings of the 6th International Modal Analysis Conference* (held Feb. 1988), pp. 1420–1426.

MATLAB Code

Sample MATLAB source codes (M-files) implementing the technique described in this paper are included here. While these code listings take two pages to print, the actual processing requires less than 30 lines of MATLAB code. The M-files were tested in MATLAB version 7.1, but should run fine in other versions.

Main M-File (remove_dither_harmonics.m)

```
% Fit and then remove dither harmonics from time-history data.
clear all
nRec = 5;            % number of time-history data records
iDitherForce = 3;    % time-history record that contains dither force
iRec = 4;            % time-history record for which to display results
nPts = 100000;       % number of samples in each time-history data record
SampFreqHz = 1000.0; % rate at which time-history samples are acquired [Hz]
ActDitherHz = 2.504; % actual fundamental frequency [Hz] of dither input
DitherHz0 = 2.5;     % initial guess for dither freq [Hz]
NH = 17;             % number of harmonics to use (not counting 0 Hz)

%----- create time-history data to demonstrate technique ---------------
fprintf(1, 'Actual dither frequency is %g Hz.\n', ActDitherHz)
time = [0:1:nPts-1]/SampFreqHz;
time = time'; % transpose time vector so it is a column vector
data_acq = zeros(nPts,nRec); % preallocate memory to speed up code
for ii=1:nRec
    data_acq(:,ii) = rand(nPts,1) - 0.5; % each time history is a column vector
    for kk=1:NH % add periodic data to the random data
        data_acq(:,ii) = data_acq(:,ii) +...
          (rand*cos(2*pi*kk*ActDitherHz*time)/kk) +...
          (rand*sin(2*pi*kk*ActDitherHz*time)/kk);
    end
end

%----- find the fundamental dither frequency --------------------------
fprintf(1, '  Initial guess for dither frequency is %.2f Hz.\n', DitherHz0)
fprintf(1, 'Use dither force (record %d) to find dither freq.\n', iDitherForce)
data = data_acq(:,iDitherForce);
save temp time data
DitherHz = fminsearch('periodic_fit_err', DitherHz0);
delete temp.mat
fprintf(1, 'Nominal dither frequency = %.2f Hz; actual value = %.7f Hz.\n',...
  DitherHz0, DitherHz)

%----- fit all the time-history records with NH harmonics --------------
fprintf(1, 'Remove %d harmonics of %.7f Hz dither from data.\n', NH, DitherHz);
AA = zeros(nPts,1+(2*NH)); % preallocate memory to speed up code
AA(:,1) = 1.0;
for kk=1:NH
    AA(:,kk+1) = cos(2*pi*kk*DitherHz*time);
    AA(:,kk+1+NH)= sin(2*pi*kk*DitherHz*time);
end
invA = pinv(AA);

coeffs = zeros(1+(2*NH),nRec); % preallocate memory to speed up code
data_periodic = zeros(nPts,nRec); % preallocate memory to speed up code
data_random = zeros(nPts,nRec); % preallocate memory to speed up code
for kk=1:nRec
    coeffs(:,kk) = invA*data_acq(:,kk);
    data_periodic(:,kk) = AA*coeffs(:,kk);
    data_random(:,kk) = data_acq(:,kk) - data_periodic(:,kk);
end
```

```
fprintf(1, 'Data processing complete.\n');

%----- do something with the results that were just calculated ---------
fprintf(1, 'Dither harmonics for record %d:\n', iRec);
fprintf(1, '\t\tcos()\t\tsin()\t\tMag\t\tFreq\n');
fprintf(1, '0\t\t----\t\t----\t%9.3f\t%.3f Hz\n', coeffs(1,iRec), 0*DitherHz);
for kk = 1:NH
    fprintf(1, '%d\t%9.3f\t%9.3f\t%9.3f\t%.3f Hz\n', kk, coeffs(kk+1,iRec),...
      coeffs(kk+1+NH, iRec), abs(coeffs(kk+1,iRec) + i*coeffs(kk+1+NH,iRec)),...
      kk*DitherHz);
end

plot(time, data_acq(:,iRec), 'g',...
  time, data_random(:,iRec), 'b',...
  time, data_periodic(:,iRec), 'r')
legend(['Original Data (record ' num2str(iRec) ')'],...
  ['Data without ' num2str(DitherHz) ' Hz Harmonics'],...
  ['Fit using ' num2str(NH) ' Harmonics']);
xlabel('Time [sec]')
ylabel('Data Value')
xlim([0 2.1/DitherHz]) % zoom in to show just over 2 cycles of the fundamental
% end of file
```

M-File Used by Optimizer to Find Fundamental Frequency (periodic_fit_err.m)

```
function err2=periodic_fit_err(FundHz)
% Calculate squared error in fitting a time history using a periodic signal with
%   fundamental frequency 'FundHz' and its first 'NH' harmonics.
% The time-history data is read from file 'temp.mat'.
%-----------------------------------------------------------------------
% Sample usage:
% >> save temp time data % save time() and data() variables to temp.mat
% >> FundFreq0 = 2.5; % initial guess for fundamental frequency [Hz]
% >> FundFreq = fminsearch('periodic_fit_err', FundFreq0)
%-----------------------------------------------------------------------
load temp.mat % read time() and data(); both are nPts x 1 column vectors
nPts = length(data);
NH = 3; % number of harmonics to use (not counting f = 0 Hz)

AA = zeros(nPts,1+(2*NH)); % preallocate memory to speed up code
AA(:,1) = 1.0;
for kk=1:NH
    AA(:,kk+1) = cos(2*pi*kk*FundHz*time);
    AA(:,kk+1+NH)= sin(2*pi*kk*FundHz*time);
end
invA = pinv(AA);

coeffs = invA*data; % coefficients for harmonics in the fit
data_fit = AA*coeffs; % synthesize data using the harmonics
err = data - data_fit;
err2 = err'*err; % (1 x nPts)*(nPts x 1) = (1 x 1)
fprintf('%f Hz gives (for NH=%d) squared error = %g\n', FundHz, NH, err2);
return
```