Using Digital Signal Processing (DSP) to Significantly Improve the Interpretation of ABAQUS/Explicit Results

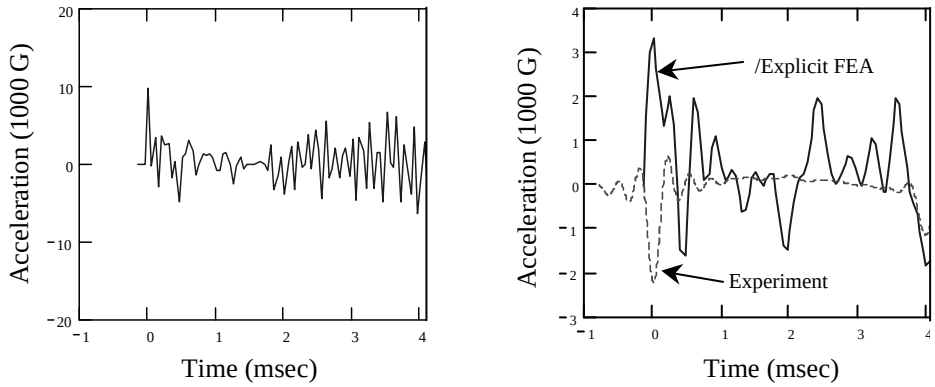
Ted Diehl, Doug Carroll, and Ben Nagaraj
Motorola, Plantation
8000 W. Sunrise Blvd.
Fort Lauderdale, FL 33322

Abstract

A long-standing issue with Explicit Dynamics is the method's propensity to calculate noisy solutions caused by unwanted high-frequency ringing. The relatively recent application of this technique to modeling drop and impact of personal electronic devices such as laptop computers, two-way radios, portable phones, and pagers has only exacerbated the problem. Quantities such as acceleration and contact force have been the most susceptible to noise, and to a lesser extent, velocity, strain, and stress. Traditional FEA post-processing approaches to reduce solution noise have been based on various curve smoothing/averaging methods and/or digital filtering (often applied improperly). These approaches have had limited success for the general class of elastically-dominated impact problems. As a result, the FEA community has struggled to develop meaningful correlation between transient quantities from simulations and experiments. The primary cause has been the improper recording of FEA data, which frequently results in corruption by aliasing. This paper presents proper techniques to avoid aliasing of the data output from transient solutions. Also presented are methods to accurately reduce (decimate) transient data sets to more manageable sizes, and to controllably remove solution noise utilizing digital filters. The success of these DSP techniques will be demonstrated on a very challenging "portable phone display" impact problem, correlating analytical and experimental results.

1.0 Introduction

Impact (and drop) analysis of personal electronic devices such as laptop computers and portable phones differs dramatically from the more established analysis of car crashes. Where as a car crash is dominated by plasticity, the impact behavior of these electronic devices is dominated by elasticity. Without sufficient inelastic deformation to dissipate energy, the resulting solution variables for an elastically-dominated impact problem can contain significant amounts of high-frequency energy, often leading to extremely noisy results. In the most severe cases, especially for contact force and acceleration, plots may contain numerous peak values that look to be completely unrealistic. Worse yet, re-plotting results using only a slightly different time interval for plotting commonly yields a completely different result. Additionally, attempts to validate FEA predictions with measured experimental data, such as accelerometer accelerations, typically yields very poor correlation (see Figure 1). All of these difficulties create confusion and uncertainty with respect to the fidelity of these impact simulations.



(a) Raw /Explicit FEA data output using a 20 kHz sampling rate (0.05 msec time interval)

(b) Filtered FEA data from (a) compared with experimental acceleration. Both signals are filtered w/ 3 kHz LP filter.

Figure 1: Example of poor correlation between an explicit dynamics model and an experimental measurement for an elastically-dominated impact problem.

The root cause of much of the problem is a violation of some fundamental rules of DSP. The most serious of which is the improper recording (sampling) of raw FEA solution output data. Due to the very small time increments utilized in the explicit dynamics solution method, it is common practice to simply request output at some time interval that is much greater than the actual solution's time increment. For noisy variables such as acceleration, this approach frequently results in corruption by *aliasing*.

Like many powerful weapons, DSP can be an extremely effective tool. However, when used improperly, it can yield bizarre, confusing, and even disastrous results. Historically, applications of DSP in modal analysis (especially on the experimental side) are well established. In this case, a majority of the analysis is performed in the frequency domain. Analysis of transient impact is different in that much of the assessment must be done in the time domain. Questions like "What is the maximum displacement or stress due to the impact?" are only answered in the time domain. Applying DSP to acquire and process *experimentally measured* impact data is also established within certain industry groups.[1] However, its application to explicit dynamics modeling, especially for personal electronic devices, is less understood and often misapplied within the FEA community.

To understand why the problem persists, it is enlightening to understand the background of a typical analyst trying to perform such explicit dynamic simulations. Most mechanical engineers doing FEA have had little, if any, training in the field of DSP. Generally, analysts have simply progressed from linear statics, through modal dynamics and nonlinear statics, to nonlinear transient

1. It is the authors' experience that shock measurement techniques are often poorly understood in many commercial companies.

explicit dynamics. As such, they bring along with them methodologies and assumptions for interpreting and processing their results which might have worked fine for their prior projects. But now these approaches seem to fail quite frequently when they work on highly transient impact problems. Even analysts who have experience modeling car crashes with explicit dynamics are found to make fundamental DSP related errors when they attempt to transfer their knowledge to solve impact analyses of personal electronics. Moreover, many of the FEA vendors that supply these Explicit codes do not provide proper tools and training to correctly deal with these problems. This paper provides a brief overview of key issues in DSP, as related to explicit dynamics, and then concludes with several examples to demonstrate its power.

2.0 Overview of DSP

Due to space limitations, only a brief description of DSP will be given. The reader is directed to IES, Ifeachor (1993), Madisetti (1998), Mathsoft (1998), Math Works (1998), Oppenheim (1975) and Stearns (1988) for more details on DSP theory. In addition, all the DSP related calculations presented here are performed using *Diehl's DSP Extensions*, an easy-to-use extension pack that works with the program Mathcad. The extension pack is specially designed for transient impact analysis and can be downloaded at http://mathcad.adeptscience.co.uk/dsp/.

Digital signal processing is concerned with the aspects of representing signals in a digital (discrete) form. The field encompasses analog to digital conversion as well as various transformation methodologies to study and process discrete data signals in both the time domain and frequency domain. The two most important concepts from DSP that are used in processing explicit dynamic results are sampling, including decimation, and lowpass filtering. The following sections will provide a practical overview of each concept and then conclude with a summary of the proper process that should be used with transient impact data.

2.1 Digital Signals, Sampling and Aliasing

Signals contain amplitude, time, frequency, and phase information, all of which are interrelated. Signals can be either continuous or discrete (digital). All variables in an explicit analysis (acceleration, displacement, stress, strain, etc.) are digital signals because they are computed at discrete points in time. A typical portable-phone drop analysis might simulate 5 msec of physical time using an average time increment of 0.1 μsec (to satisfy integration stability requirements). Each variable in the simulation is a digital signal containing approximately 50,000 data points. The storage requirements for XY plotting only a few variables with that many points is quite large. Even the act of XY plotting 50,000 data points for one curve (signal) may seem ridiculous. Out of necessity, the common approach used in the FEA community is to output the results at much larger time increments, reducing the data size down to say a couple hundred points per variable. The act of reducing this data is called *decimation* and it is integrally related to the *Sampling Theorem* of DSP.

Sampling is often defined as the acquisition of a continuous signal at discrete time intervals. Sampling is also the recording of a digital signal at discrete time intervals (generally different than the time intervals of the original digital signal). For clarity and to avoid any potential confusion, it is necessary to define the meaning of the following terms:

- *Sampling (or Sampled)* - The act of evaluating, measuring, recording, or storing a continuous or digital signal at discrete time intervals.
- *Original Signal* - The signal *before* it is sampled.
- *Sampled Signal* - The resulting signal *after* sampling.

Common parameters include the time interval, Δt_s, the sampling rate, ω_s, the Nyquist frequency, ω_{Nyq}, and the maximum frequency content in a signal, ω_{max}. They are interrelated as[1]

$$\omega_s = \frac{2\pi}{\Delta t_s} \tag{1}$$

$$\omega_{Nyq} = 0.5\omega_s \tag{2}$$

To Avoid Aliasing: $$\omega_s > 2\omega_{max} \quad \rightarrow \quad \omega_{max} < \omega_{Nyq} \tag{3}$$

Equation 3, comes from the fact that at least two sample values per cycle are required to describe the frequency content of a sinusoid. These equations also tell us that given a signal that has a sample rate of ω_s, the highest frequency content it can describe is $\omega_{max} = 0.5\omega_s$. Similarly, given a signal with a maximum frequency content of ω_{max}, the sample rate required to avoid aliasing must be greater than or equal to $2\omega_{max}$.

Figure 2 depicts the results of sampling a 1.0 kHz sine wave with different sample rates. Evaluation of the results are shown in the time domain and frequency domain. To demonstrate the worst case scenarios, the sampling vectors for cases (a) and (b) were shifted slightly in time. All the frequency domain results were computed using a Discrete Fourier Transform with a Hanning window (with 100 msec length) that was normalized to correct for amplitude loss.

Figure 2 plainly shows the physical impact of Equations 1 - 3. When the sample rate is much higher than the actual frequency content (Figure 2a), the time domain and frequency domain representations of the sampled signal match closely to the theoretical result (original signal). The sample rate of Figure 2b is relatively coarse but still satisfies the Sampling Theorem. However, the time domain representation suffers, predicting only 50 % of the positive peak amplitude. Notice though, that the frequency value is still accurately represented in both domains. Results for Figures 2c - d clearly show the result of aliasing. In this case, the sample rate does not satisfy Equation 3. The original 1.0 kHz signal, when sampled, becomes a perfect 0.1 kHz sinusoid.

1. All formulae presented utilize *circular* frequency, ω, with fundamental units of rad/sec. Most engineers commonly use the unit of Hz to describe frequency, where 1 Hz = 2π rad/sec. While Hz is really only a different set of units (similar to radians & degrees or inches & meters), we will nonetheless, adhere to the common notation of using "*f*" to denote frequency values when using the unit of Hz.

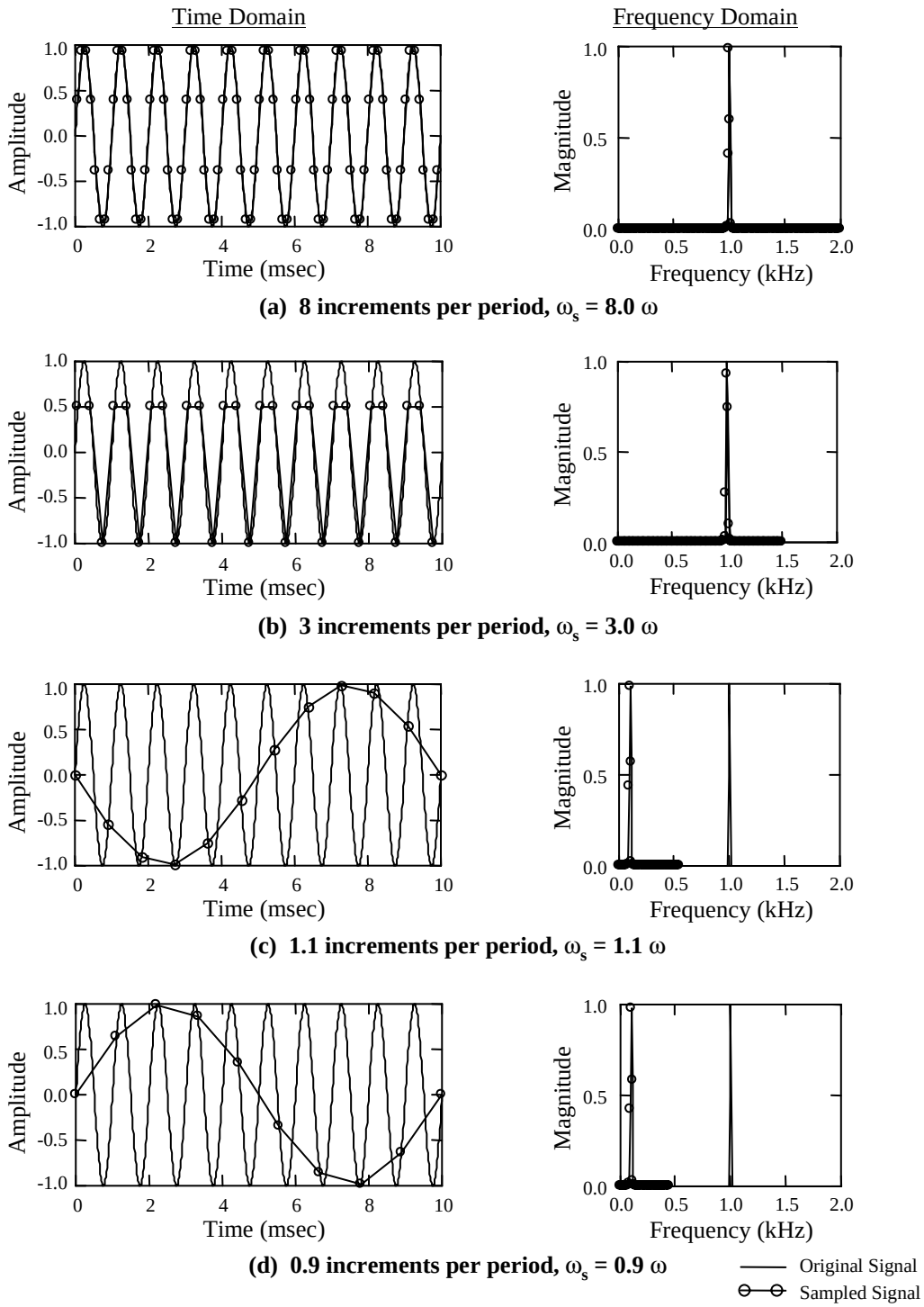
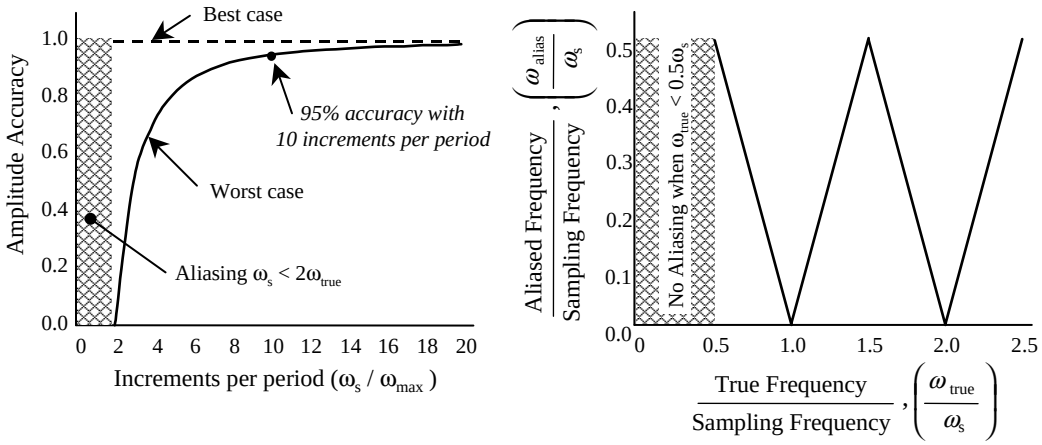


Figure 2: Sampling a 1.0 kHz sine wave with different sample rates.



(a) Sampling rate influence on accuracy of peak amplitude for a sinusoid in the time domain

(b) Sampling rate influence on alias distortion for a sinusoid

Figure 3: Summary of errors induced by sampling

Moreover, only a slight change in sampling frequency makes the sampled signal's amplitude flip sign (180 deg. out of phase). It is important to notice that the magnitude of the amplitude in cases (c) and (d) are quite accurate, but the frequency is said to have been *aliased*.

The results depicted in Figure 2 contain two distinct types of errors caused by sampling. Provided that the sampling theorem is satisfied, amplitude errors in the time domain are caused when the sample rate is only a few times larger than the actual frequency content of the original signal.[1] Figure 3a generically characterizes this error. The "worst case" curve is obtained by the following approach. For a given $\omega_{s,}$ evaluate $\sin(\omega t + \phi)$ over the range $0 \le \phi \le 2\pi$, recording the peak amplitude computed for each ϕ. The worst case value is the lowest peak recorded. The graph shows that for time-domain analysis, a good rule of thumb is that the sample rate should be 10 times the highest frequency of interest in order to obtain accurate peak amplitudes in the time domain. Most FEA analysts are aware of (and are concerned about) this type of error.

The second type of error, however, is much more damaging and dangerous. The aliasing errors shown in Figure 2c - d are more dangerous because the results can look so good. Aliased results can look like physically reasonable signals – smooth curves with low frequency content. ***In fact, with out prior knowledge of the original signal, it is impossible to tell whether the 0.1 kHz signals in Figure 2c - d were originally 0.1 kHz signals or aliases from some originally higher frequency signal.*** *It is these aliasing errors that are at the root of much of the misinterpretation of explicit dynamics models.*

1. There are DSP-consistent interpolation methods, sometimes called *up-sampling*, that can be used to improve the estimate of peak amplitudes in the time domain for coarsely sampled data.

Whenever Equation 3 is not satisfied, each frequency component in the original signal, ω_{true}, will be represented as ω_{alias} according to (see IES, pp. 136):

$$\omega_{alias} = n\omega_s \pm \omega_{true} \qquad \text{if } \omega_s < 2\omega_{true} \tag{4}$$

where

$$n = \begin{cases} \text{floor}(\omega_{true}/\omega_s) + 1 & \text{if } (\omega_{true}/\omega_s - \text{floor}(\omega_{true}/\omega_s)) > 0.5 \\ \text{floor}(\omega_{true}/\omega_s) & \text{otherwise} \end{cases} \tag{5}$$

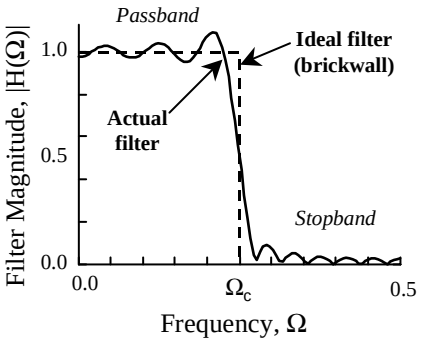
and the function floor(x) returns the greatest integer $\leq x$. Using the above equation, we can see that sampling a 1.0 kHz signal with either a 1.1 kHz or 0.9 kHz sample rate will create a 0.1 kHz alias. Figure 3b generalizes this result and shows how insufficient sampling rates can yield low frequency aliasing. For example, if a sampling rate of 30.0 kHz is used on a signal that has a significant frequency component of say, 67.5 kHz, the ratio of true frequency to sampling rate would be 2.25. In this case, the graph shows that the resulting sampled signal would be aliased to $0.25 f_s$ = 7.5 kHz (significantly different than 67.5 kHz). Another important characteristic to notice about aliasing is that as the value of the true frequency in the original signal increases relative to a given sample rate, the resulting aliased frequency continuously ranges back and forth from 0 to $0.5\omega_s$. Thus, when sampling an original signal (that contains very high frequency content) at a much lower sampling rate, it is impossible to predict (without exact knowledge of the original signal) what aliased frequency values will occur.

2.2 Digital Filtering

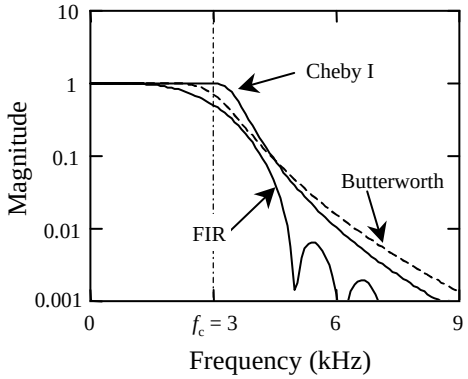
The signals that we have looked at up to this point have been simple sinusoids with one frequency. Real-world signals are composed of a spectrum of frequencies that can be separated into physically meaningful frequency content and unwanted frequency content, generally called noise. One of the most useful capabilities of digital filters is their ability to remove undesired frequency content from a discrete time-domain signal without ever taking the signal into the frequency domain.

Four of the most common filter types are: lowpass (LP), highpass (HP), bandpass (BP) and bandstop (BS). Lowpass filters are the most common type used with explicit dynamics and will be discussed here. They attenuate frequency components above a specified cut-off frequency while allowing the low frequency content of a signal to “pass” through with minimal attenuation. Figure 4a shows the common characteristics of a lowpass filter. The results are plotted as a function of normalized frequency Ω and normalized cut-off frequency Ω_c.

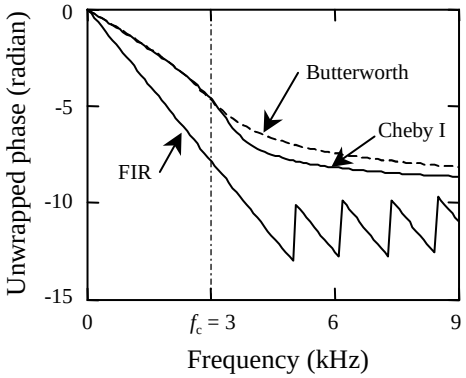
$$\Omega = \frac{\omega}{\omega_s} \qquad \Omega_c = \frac{\omega_c}{\omega_s} \tag{6}$$



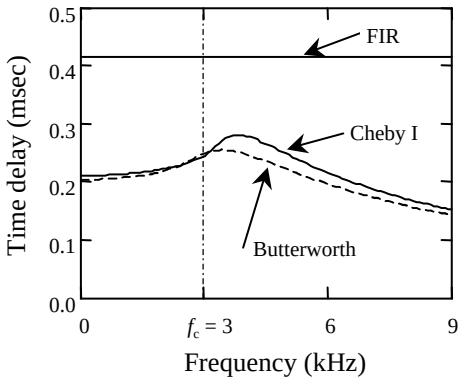
(a) Generic lowpass filter characteristics



(b) Magnitude characteristics of FIR and IIR Lowpass filters



(c) Phase characteristics of FIR and IIR Lowpass filters



(d) Time delay characteristics of FIR and IIR Lowpass filters

Filter Parameters for (b) - (d)

All:	Sampling freq, f_s = 240 kHz, Cutoff freq, f_c = 3 kHz, $\rightarrow \Omega_c = 0.0125$
	Filter Passes: 1
Butterworth:	Order = 6
Cheby I:	Order = 6, Passband ripple = 0.25%
FIR:	Taps (number of coefficients) = 201, Window type = Hanning

Figure 4: Overview of digital filters

Digital filters typically come in two forms, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). Both can be characterized in the frequency domain by the generic *z*-domain transfer function *H(z)* as

$$H(z) = \frac{B_0 + B_1 z^{-1} + B_2 z^{-2} + \ldots + B_L z^{-L}}{1 + A_1 z^{-1} + A_2 z^{-2} + \ldots + A_L z^{-L}} \tag{7}$$

where L is the filter order, vectors A and B are filter coefficients, and z is a complex variable related to frequency. Via the *z*-transform[1], we can implement digital filters entirely in the time-domain. Given a discrete time domain data sequence x_i (digital signal) and filter coefficient vectors A and B, the filtered time domain response y_i is simply computed as (Ifeachor, 1993, pp. 143)

$$y_i = \sum_{j=0}^{L} B_j x_{i-j} \underbrace{- \sum_{j=1}^{L} A_j y_{i-j}}_{\textit{Feedback Terms}} \tag{8}$$

Equation 8 demonstrates that digital filters are seen to be moving weighted averagers. Commonly used moving-average smoothing schemes are actually a subset of digital filters. However, well designed digital filters are much more capable and have very specific coefficients that depend on several specified parameters. FIR filters are defined only with B coefficients, they have no feedback terms (all $A_j = 0$). Looking at Equation 8 it is clear that if a digital impulse is fed into a FIR filter, the output will have a non-zero response for the same number of increments as there are terms in the filter coefficient vector B. Hence, the term Finite Impulse Response (FIR). IIR filters have both sets of coefficients A and B. The feedback terms computed from the A coefficients make it such that an IIR filter will, in general, have an Infinite Impulse Response (the response will last forever). Many classical analog filters such as Butterworth and Chebyshev Type I and II are digitally approximated very well by the IIR form.

The ideal filter in Figure 4a is perfectly sharp about the cut-off frequency. Its passband response is exactly 1.0, it has no transition band, and its stopband response is exactly 0.0. The impulse response of the ideal brickwall filter is a function of the sinc function ($\text{sinc}(\theta) = \sin(\theta)/\theta$). Taking a *finite* number of terms from the sinc function is a common method to develop a lowpass FIR filter. This type of filter usually needs a window to improve the filter response. For the remainder of this paper, comments relative to FIR filters imply a sinc-based FIR filter, unless stated otherwise.

Figure 4b - d compare two classic IIR filters and a sinc-based FIR filter by plotting commonly used frequency response measures of magnitude, unwrapped phase response, and time delay.[2] All of these quantities are computed by evaluating the filter's gain over a range of frequencies. The magnitude plot of Figure 4b is the most common. It depicts how each frequency component of a

1. Use of the *z*-transform with digital signals is similar to using Laplace transforms with continuous systems.
2. ABAQUS/Post is unable to compute or plot any of these commonly used quantities.

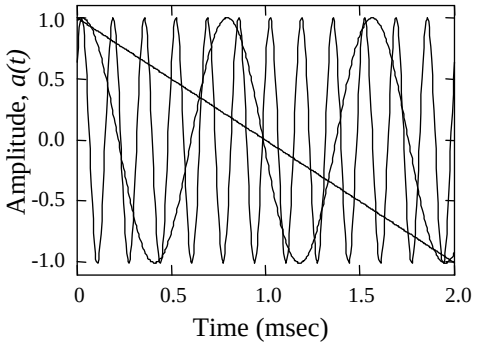
signal will be attenuated when it is passed through the filter. For example, a 1.0 kHz frequency component will pass through all the filters shown in Figure 4b with negligible attenuation, whereas a 6.0 kHz component will experience significant attenuation (0.001 for the FIR filter and approximately 0.01 for the two IIR filters). Figure 4c shows the *unwrapped* phase response, which corrects for the discontinuities of 2π that are commonly seen in a phase plot.[1] A more meaningful representation of phase information for our transient time-domain analysis is in terms of time delay (Figure 4d), computed as the unwrapped phase divided by frequency.[2] The time delay shows how much delay each frequency component in a signal will have as it passes through the filter. The symmetric FIR filter shown has a linear phase response which results in a constant time delay. The two IIR filters have nonlinear phase responses and nonlinear time delays.

For time domain analysis, time delays are undesirable because the resulting filtered signal is distorted in time. The most general method to remove nonlinear phase distortion is a double pass filter technique (Math Works, 1998, pp. 6-134). In this method, the digital signal is passed through the filter, the resulting signal is reversed and passed through the filter again, and finally this latter signal is reversed again to achieve a zero-phase filtered signal (with absolutely no time delay). During this process, the resultant magnitude response of the filter is squared relative to the commonly reported single-pass magnitude response (single-pass shown in Figure 4b). Other undesirable distortions can be created by filter start-up and ending effects. Filter start-up effects are caused by the common assumption that the signal to be filtered has zero amplitude for all time before the beginning of the signal. Filtering signals that violate this assumption can cause transient distortions because of the inherent non-smooth transition between the assumed zero amplitude region before the signal's beginning and the actual signal content. Similar distortions can occur at the end of the signal. This type of error can be minimized by artificially projecting the original signal back in time by a finite amount. A second source of filter start-up distortion is a step-discontinuity created when a signal's initial value is nonzero. This commonly occurs in a drop analysis when filtering a velocity or when filtering stress/strain from a preloaded component. This type of distortion is easily corrected by precharging the filter with the filter's DC response. The filter algorithms used in *Diehl's DSP Extensions* offer several methods to minimize all these filter-induced distortions.

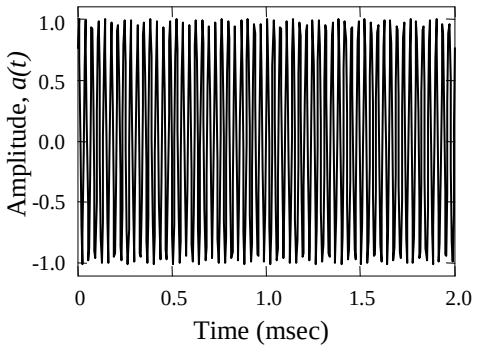
Figure 5 shows the time-domain results of some simple signals before and after lowpass filtering. The test signals depicted in (a) are a sloped line and three sine waves with frequencies of 1.3 kHz, 6.0 kHz, and 29.0 kHz (note that the sine waves are defined such that they begin with non-zero amplitude). The curves depicted in (b) - (e) show the ideal solution, results from ABAQUS/Post, and our "proposed method." ABAQUS/Post has implemented a sine-Butterworth IIR filter with no filter distortion compensation. The sine-Butterworth filter is a slight variation on

1. The apparent 2π discontinuities seen in the FIR filter after the cut-off frequency are caused by a different phenomenon that the unwrap algorithm does not correct for.

2. Signal delay is a complex subject. The reader is directed to Brillouin (1960) for an in-depth discussion.

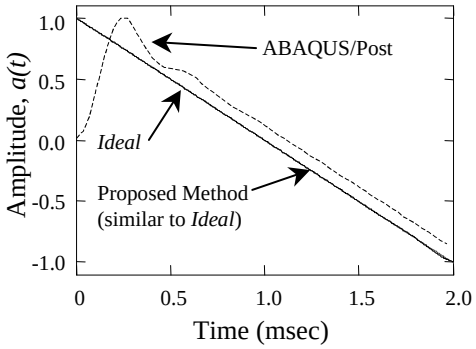


Sloped line (f = 0) and sine waves with f = 1.3 kHz and f = 6.0 kHz.

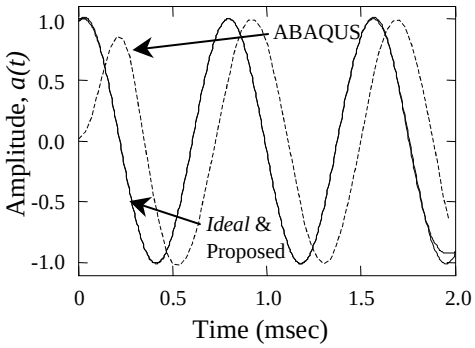


Sine wave with f = 29.0 kHz

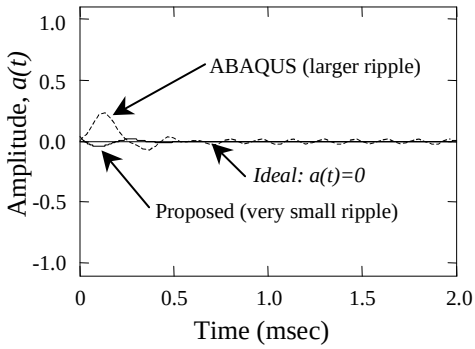
(a) Original signals for filter testing



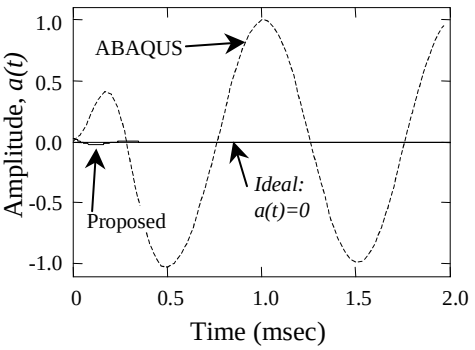
(b) Filtered results: Sloped line



(c) Filtered results: f = 1.3 kHz



(d) Filtered results: f = 6.0 kHz



(e) Filtered results: f = 29.0 kHz

Figure 5: Comparing ABAQUS/Post lowpass filtering and proposed method using some digital test signals. Sample rate of all signals is 240 kHz and cut-off frequency is 3.0 kHz.

the more common Butterworth filter. The "proposed method" shown is a Butterworth filter[1] using double-pass nonlinear phase compensation, filter precharging, and filter start-up minimization. Both the ABAQUS and "proposed" were 6^{th} order filters. Figures 5a - d demonstrate the deficiencies of the ABAQUS filter implementation, end-distortions and time delay are clearly evident. The same figures show the relatively distortion-free results from the "proposed method."

Figure 5e shows a serious deficiency with the ABAQUS lowpass filter implementation, it produced an aliased result! ABAQUS/Post has a default resampling algorithm that will resample (decimate or interpolate) the original signal to a sample rate that is 10 times the specified cut-off frequency of the filter.[2] This resampling is done prior to applying the lowpass filter. In our test case, the 29.0 kHz sine wave, which originally had a sample rate of 240 kHz, was decimated to a sample rate of 30 kHz (10 times 3 kHz). Based on Equation 4, the 30 kHz sample rate caused the 29.0 kHz signal to be aliased to a 1.0 kHz signal. This resampled signal then passed through the 3 kHz filter "untouched", except for the filter distortions already discussed. These simple cases show that the ABAQUS filter implementation not only creates undesirable distortions but that it is very susceptible to inducing aliasing errors.

2.3 Summary of Proper Sampling and Decimation With Antialias Filters

The previous sections have demonstrated the dangers of improper sampling techniques on a variety of simple signals. The fundamental problem when dealing with transient FEA data is that we generally do not know ahead of time the maximum frequency content of the various solution variables. If we did, then we would know a safe sampling rate. Lacking knowledge of the maximum frequency content, the following template will always insure aliased-free data.

1. For every variable of interest, output the result at *every time increment.* This is done using *HISTORY OUTPUT, TIME INTERVAL = 0.0 (placed in the /Explicit deck).
   The data will be stored in the *.sel file. Yes, the file might get quite large!

2. In ABAQUS/Post, write the data out from the *.sel file to an ascii file for processing using a program with appropriate DSP capabilities such as *Diehl's DSP Extensions.*[3]

3. Because the time increment in an explicit dynamics analysis changes throughout the solution, *regularize* the data to a constant time increment (DSP algorithms require this). To avoid aliasing during regularization, use the following procedure. Define a new time vector (sampling vector) with a time increment equal to the *smallest time increment* from the entire /Explicit solution. Interpolate all the raw FEA results onto this new time vector (spline interpolation is preferred, Lagrange polynomial or linear interpolation should also be adequate).

---

1. Similar results are easily achieved with other common filters such as Cheby I or a sinc-based FIR.

2. The ABAQUS/Post manual, version 5.8, incorrectly states the resampling to be 5 times the cut-off freq.

3. Other programs with appropriate DSP capabilities may also be used. but they must be able to properly handle very small normalized cutoff frequencies (values on the order of $\Omega_c = 0.001$). For example, MATLAB's Signal Processing Toolbox (V5.1) experienced a variety of numerical difficulties for $\Omega_c < 0.01$.

4. Reduce the regularized data sets down to 10 times the highest frequency of interest. To decimate the data safely, it must first be lowpassed filtered to sufficiently attenuate all the frequency content that is above the Nyquist frequency of the desired (reduced) sample rate. This is known as *antialias filtering.* If the regularized sample rate is ω_{orig}, and the reduced sample rate is $\omega_{desired}$, then all the data must first be filtered such that all frequency content above 0.5 $\omega_{desired}$ is sufficiently attenuated. Note, the cut-off frequency of the antialias filter must be sufficiently less than the ideal cut-off frequency (0.5 $\omega_{desired}$) to account for the transition band of the filter. Once this frequency content is "removed," the data sets can be safely decimated to the new sample rate of $\omega_{desired}$. To avoid filter distortions, special precautions as discussed in Section 2.2 must be utilized. *Diehl's DSP Extensions* does this automatically.

5. Plot the data as desired or apply additional filtering as desired.

The recipe described in steps 1 - 5 will ensure that aliasing errors do not appear in your data. However, the application of this methodology must be done prudently, in that significant amounts of data will need to be stored initially. Note, after the decimation procedure in step 4, the file size is greatly reduced. It is recommended that the user stores only a few output variables for this type of detailed analysis, say maybe 10 or so.

A few additional practical things to remember. Responses such as acceleration and contact force will be highly noisy, especially in solid elements. This type of data often has large amplitude, high frequency noise components. All data of this type that is to be evaluated should be post-processed in the manner described above. On the other end of the spectrum is displacement data. By its nature, high frequency displacement components have very low amplitude and therefore pose a relatively low risk of alias corruption. Stress, strain, and velocity are in between these two cases.

Unfortunately, application of the process outlined above is not generally feasible for animated contours, as too much initial data must be stored. A reasonable approach around this limitation is to output data at a few representative nodes (or elements) within the contour by the method outlined above. Using this selected data set, a safe sampling rate that can be determined which will avoid aliasing. Then the entire data set can simply be stored (output) from the solution using this safe sampling rate. Remember, you must be sure that there is no significant frequency content in the solution variables of interest greater than one half of your output frequency. If there is, aliasing will occur which you will not be able to detect.

3.0 Faux Impact Acceleration Example

The following artificial impact problem is provided so that the reader can further evaluate and study the issues using a closed-form example. Consider simulating the drop of a personal electronic device, such as a phone, onto a hard concrete floor. Computed accelerations (and all other output variables) can be viewed to contain physical frequency content and undesired frequency content, usually termed solution noise. Below we assume some simple functions that are somewhat characteristic of impact accelerations.

$$\underline{\text{Physical}} \qquad a^{p}(t) = a_o^{p} \cdot \sin(\omega^{p} t) \cdot e^{\frac{-t}{\tau^{p}}}$$

$$\underline{\text{Noise}} \qquad a^{n}(t) = a_o^{n} \cdot \sin(\omega^{n} t) \left(1 - e^{\frac{-t}{\tau^{n_1}}} \right) \begin{cases} 1 & \text{if } t \le \tau^{n_2} \\ e^{\frac{-t-\tau^{n_2}}{\tau^{n_3}}} & \text{if } t > \tau^{n_2} \end{cases} \tag{9}$$

$$\underline{\text{Total}} \qquad a(t) = a^{p}(t) + a^{n}(t)$$

Where:

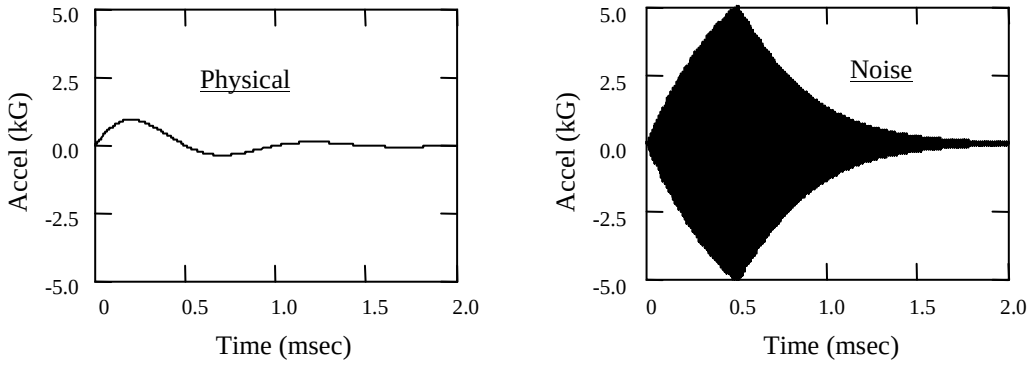
$a_o^{p} = 1.5\text{kG}$ $\quad f^{p} = 1.0\text{kHz}$ $\quad \tau^{p} = 0.5\text{msec}$

$a_o^{n} = 8.5\text{kG}$ $\quad f^{n} = 105.8\text{kHz}$ $\quad \tau^{n_1} = 0.5\text{msec}$ $\quad \tau^{n_2} = 0.5\text{msec}$ $\quad \tau^{n_3} = 0.3\text{msec}$

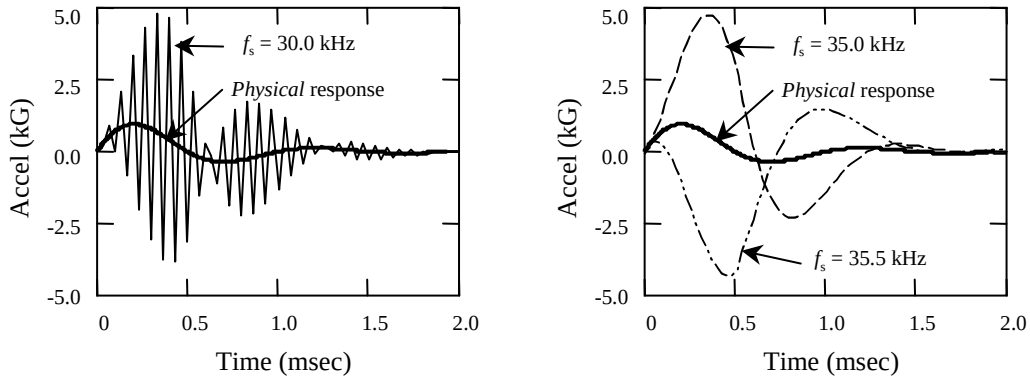
Figure 6a depicts the physical and noise components of the signal. The physical portion of Equation 9 is simply a sinusoid that decays to zero over time. The noise function is also sinusoidal but is enveloped by two exponential functions, one to mimic the growing ringing that often occurs at impact and one that eventually decays the ringing back toward zero. As shown, the noise has very high frequency content[1] and large amplitude compared to the physical portion. This is a typical trait of acceleration ringing in elastically-dominated explicit dynamic models. The combined signal, which is not plotted, looks very much like the noise component, because the noise is so dominant. In Figure 6b the combined signal is sampled with three very similar sampling rates. As seen in the graphs, the results vary significantly. The 30 kHz sample rate looks rather noisy, but the 35.0 kHz and 35.5 kHz sample rates both yield smooth, yet extremely different curves. In (c), these results are passed through a 3.0 kHz lowpass filter. After filtering, the noisy 30 kHz sample rate becomes smooth and now matches the physical response, but the 35.0 kHz and 35.5 kHz sample rates remained unchanged - still significantly different than the benchmark. Lastly, (d) shows that when proper sampling methods are used that incorporate antialias filtering, all the sample rates yield good results. For this example, the antialias lowpass filtering just happened to remove all the high-frequency noise. In general, an antialias filter will remove sufficiently high-frequency content to avoid aliasing, but it may still pass undesirable noise (composed of lower frequency content).

Let's further clarify what happened with the results in Figure 6b - c. First, all three sample rates yielded aliased results in (b). The 30 kHz sampling aliased the 105.8 kHz noise to 14.2 kHz. The 35.0 kHz and 35.5 kHz rates aliased the noise to -0.8 kHz and 0.7 kHz (the negative frequency implies a 180 deg phase shift which causes amplitudes of sine waves to change sign. See Ziemer,

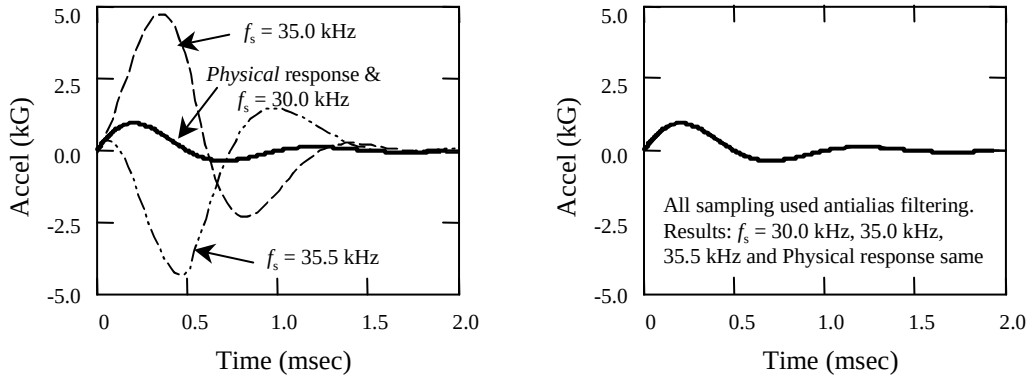
1. The frequency content is so high, that when the signal plotted in the time domain over the time scale of 2.0 msec, the resulting plot looks like a solid image.



(a) Physical and noise portions of the signal (Benchmark)



(b) Sampled results without antialias filtering - No additional LP filtering applied



(c) Results from (b) passed through a 3.0 kHz, 6th Order, Butterworth LP filter.

(d) Sampled results using antialias filtering. (No additional LP filtering applied)

Figure 6: Results from faux impact example

1983, pp. 119.). When a 3.0 kHz LP filter was then applied (Figure 6c), the 14.2 kHz alias of the 1st sample rate was removed but the aliases in the other two signals remained since they were below the cut-off frequency of the filter.

This simple example provides a controllable demonstration of how aliasing can cause significant confusion. **It is also important to note this example clearly demonstrates that just because a result looks smooth (before or after filtering), there is no way of knowing if aliasing has occurred unless the appropriate precautions were taken before sampling.**

4.0 Ball Bearing Impact on a Portable Phone Lens

Figure 7 depicts a very challenging ball bearing impact problem. A plastic housing with plastic display lens is supported at 4 bosses and subjected to the impact of a 130 g, 31.75 mm diameter steel ball which is dropped from 500 mm. The quantities of interest are the len's transient responses of acceleration and displacement under the point of impact (back side of the lens). The objective of this example is to correlate both experimental and FEA results.

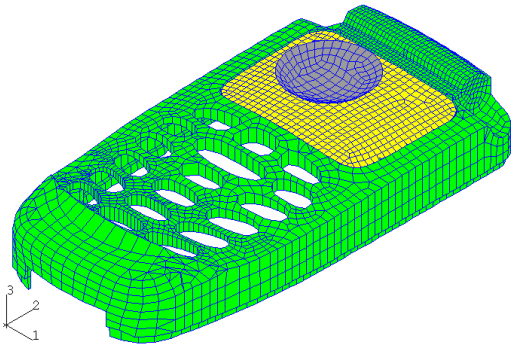
The explicit FEA model is composed of shell elements for the housing and solid elements for the variable thickness lens (three elements through the thickness). The plastic material is modeled using only Hooke's law (no plasticity or viscous effects). The experimental measurement of the acceleration utilized an Endevco 2255B-01 Isotron accelerometer connected to an Endevco model 133 Signal Conditioner with all the Conditioner's HP & LP filters turned off. The lightweight accelerometer has a resonance of 300 kHz and an internal 2-pole Butterworth lowpass analog filter with an approximate cut-off frequency of 28 kHz. The experimental acceleration was captured, alias free, at a sample rate of 250 kHz and is displayed in Figure 7c. The /Explicit prediction of acceleration from a single node on the bottom side of the lens, directly under the point of impact, is displayed in Figure 7d. This data has already been regularized per steps 1- 3 of Section 2.3.[1] Figures 7d - e show these results plotted in the frequency domain.[2] The question we need to answer is "Does the simulation and experiment correlate?" Based on the data presented in Figure 7, we would have to say "no." Let's try to further analyze the data to see if things improve.

A common approach to improve correlation is to apply a lowpass filter to remove noise that is likely present in both the experiment and the simulation. A common cut-off frequency used for this type of structure might be between 1 kHz and 10 kHz. For our case, we will choose 5 kHz. Using the rule of thumb of 10x, we decide that our desired sampling frequency should be 50 kHz. Figure 8 presents what happens if we just sample the raw FEA data at 50 kHz without protecting against aliasing. The experimental acceleration (sampled at 250 kHz) and the FEA acceleration (sampled at 50 kHz) displayed in Figure 8a definitely do not correlate. Figure 8b presents the results after lowpass filtering with a 5 kHz Butterworth filter. To match filter responses for the two

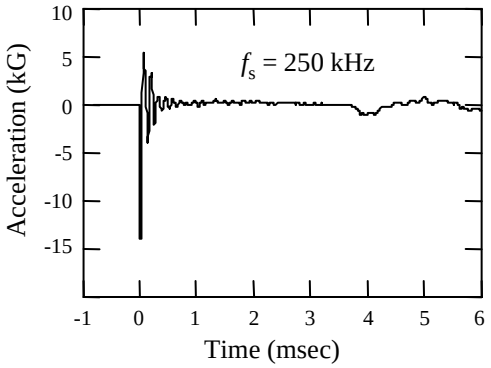
1. The regularized and raw explicit data are very similar and would be indistinguishable on this plot.

2. The frequency spectrum of a time-domain signal was computed with *Diehl's DSP Extensions*. This capability does not exist within any of the ABAQUS software.

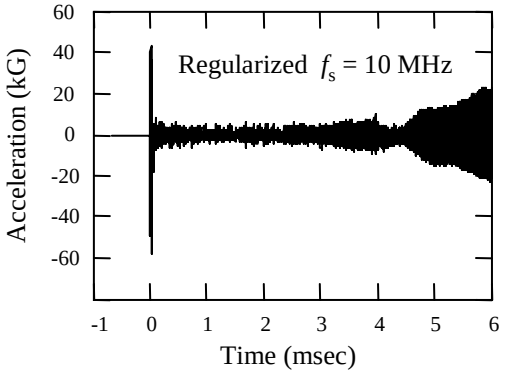
(a) Experimental setup



(b) Explicit dynamics model

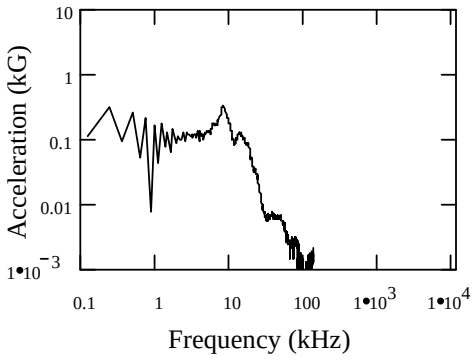


(c) Raw experimental acceleration

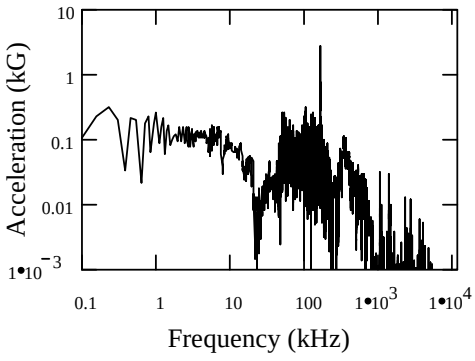


(d) Raw acceleration from FEA model

Note: vertical scale for (c) and (d) are different.

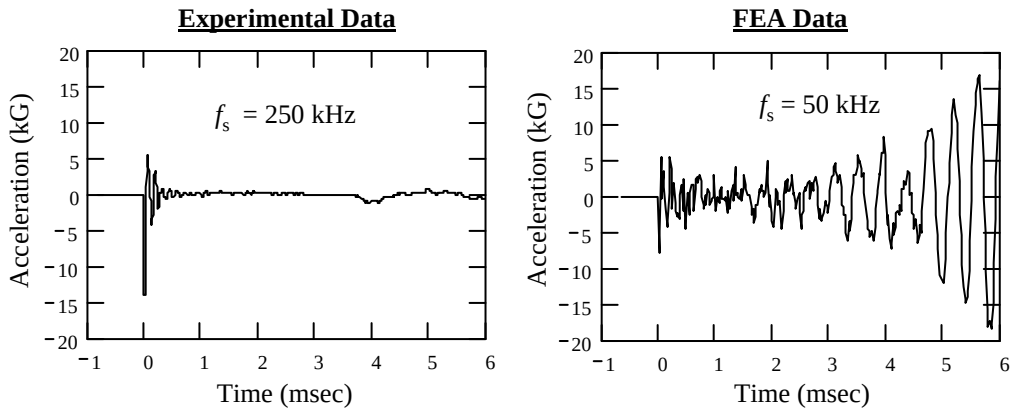


(e) Frequency content of experimental accel.

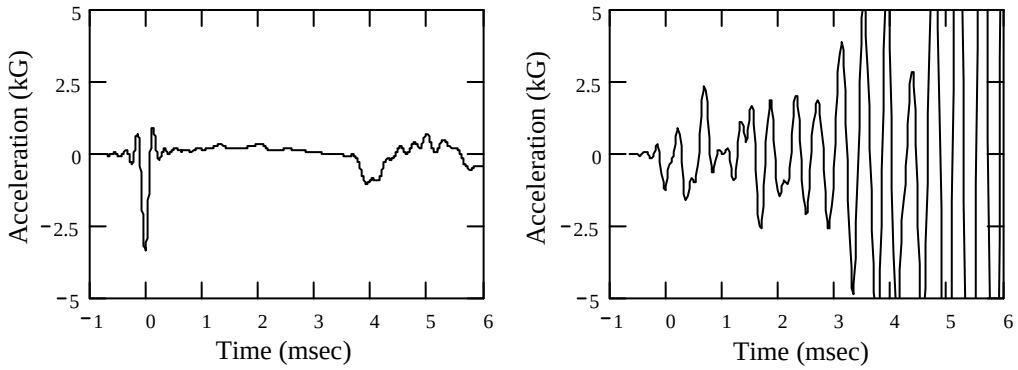


(f) Frequency content of FEA accel.

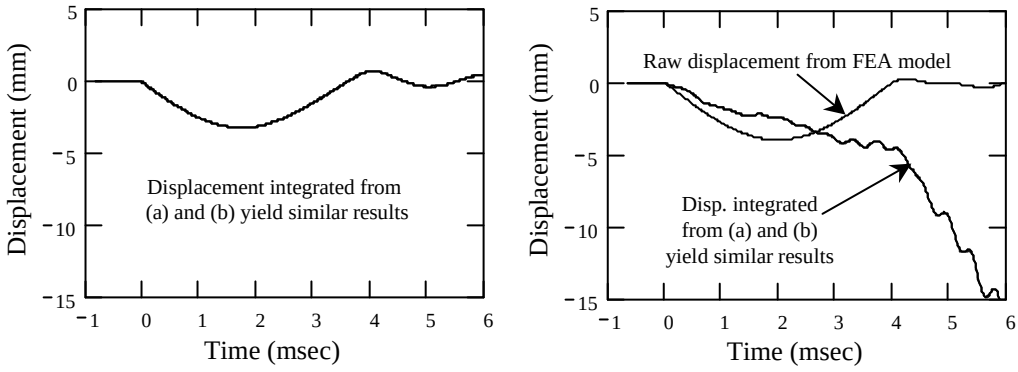
Figure 7: Ball bearing impact example, experiment and /Explicit FEA model.



(a) Comparison of acceleration data before 5 kHz lowpass filtering



(b) Comparison of acceleration data after 5 kHz lowpass Butterworth filtering



(c) Comparison of displacement data

Figure 8: Processing data without protecting against aliasing. FEA data is simply requested from solution at a 50 kHz sample rate.

data sets, the experimental data utilized an 8^{th}-order Butterworth and the simulation data used a 7^{th}-order Butterworth.[1] Even after filtering, the two acceleration data sets look completely unrelated. Lastly, we compare displacement data (Figure 8c). Displacements for the experiment are computed by double integrating the experimental acceleration signal. The FEA code computes displacements directly. Interestingly enough, the raw FEA displacement curve and the integrated experimental displacement curve correlate quite well. However, acceleration and displacement are directly related and we observed that the accelerations were completely different! Also plotted in Figure 8c are "integrated FEA displacements" computed by double integration of the FEA *acceleration* curves from Figure 8a -b. Both of these integrated results are very different from the raw displacements computed directly by ABAQUS. How could that be? The answer is that all the FEA acceleration data in Figure 8 is aliased! It is aliased because we sampled the acceleration data without first removing the high frequency content above 25 kHz (half of 50 kHz). Figure 7f (regularized based on sampling every solution increment) shows that the original acceleration signal contains significant frequency content up to approximately 1 MHz. This extremely high frequency content is caused by, among other things, the individual element vibrational modes.

Figure 9 presents data that is properly decimated to 50 kHz by the process described in Section 2.3. In addition, the experimental data has also been decimated down to a sample rate of 50 kHz. The results in Figure 9a, before the 5 kHz LP filter, look very promising. After filtering, the correlation between the simulation and experimental accelerations is excellent (Figure 9b). Now, the integrated displacements for the simulation match the raw FEA displacement data (Figure 9c). An import thing to note is that the success of obtaining correlation was dependent on proper sampling technique, not filter form (IIR or FIR). Equally good results are obtained with a Cheby I IIR or sinc-based FIR filter provided that the "proposed filtering method" described previously is used and that the filter parameters are defined such that the magnitude of the filter responses are similar. Lastly, one filter parameter that will influence correlation is the filter cut-off frequency. In this example, if a 2.0 kHz lowpass filter is used, too much frequency content is removed from the acceleration signals. In this case, the integrated displacements would begin to contain errors because frequency content that governs the displacement has been removed.

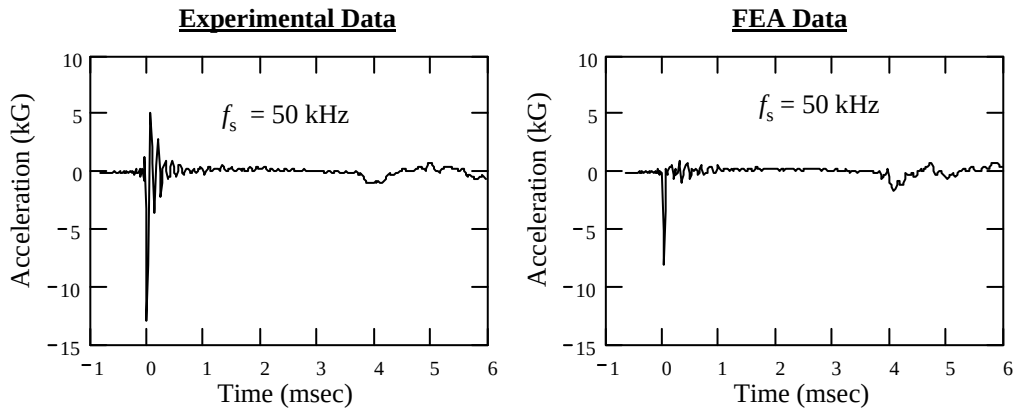
5.0 Conclusions

It is imperative that highly transient Explicit Dynamic solution data is properly handled, especially for elastically-dominated impact problems. Proper processing can make the difference between realistic and nonsensical conclusions. The key DSP issues related to this application are:

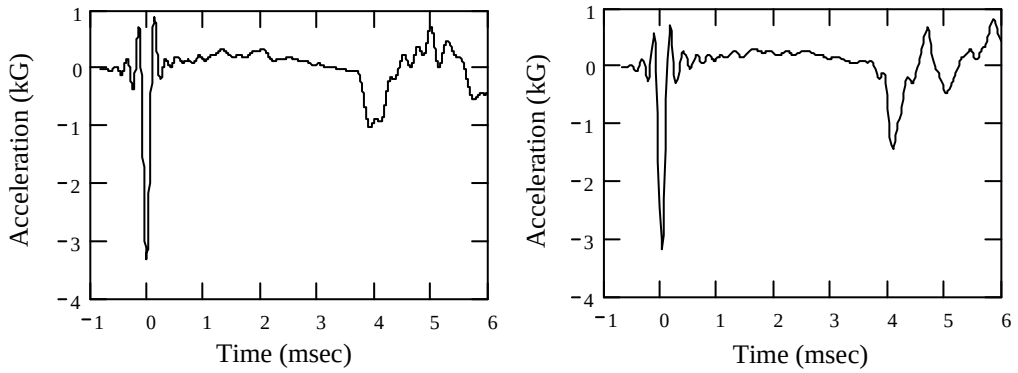
1. In general, it is impossible to determine if a given sampled signal has aliased frequency content without detailed knowledge of the original signal prior to sampling. Even smooth, low-frequency signals can be corrupted by aliasing.

---

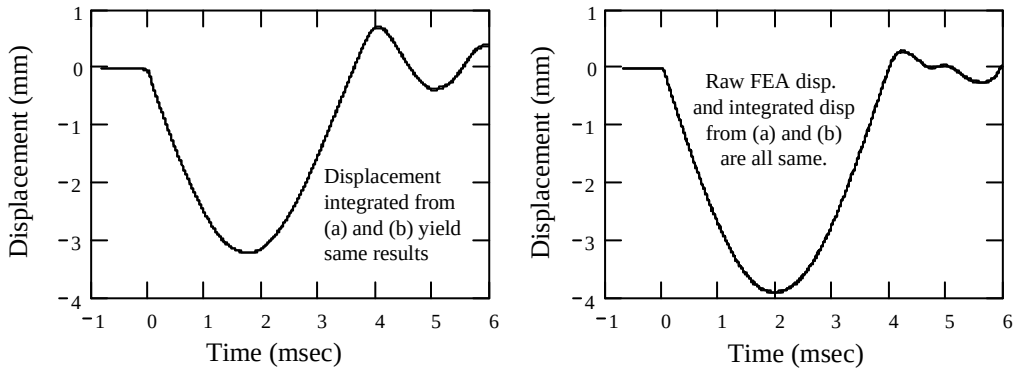
1. Remember, filter responses are a function of the *normalized* cut-off frequency (the cut-off frequency divided by the sampling rate). To match filter responses as close as possible, different filter orders are used because the two data sets had different sampling rates.



(a) Comparison of acceleration data before 5 kHz lowpass filtering



(b) Comparison of acceleration data after 5 kHz lowpass Butterworth filtering



(c) Comparison of displacement data

Figure 9: Processing data with antialias filtering. FEA data is properly decimated down to a 50 kHz sample rate.

2. Quantities such as acceleration and contact force are the most susceptible to alias errors, and to a lesser extent, velocity, strain, and stress.

3. The proper method to process transient data and avoid aliasing errors is to output the desired FEA solution variables at every solution increment, regularize this data using the minimum time increment from the solution, and then decimate the data (incorporating an antialias lowpass filter) to a manageable sample rate. Applying this approach to large data sets, such as animations of contours, is not feasible. For large data sets, this approach should be applied to a small selected set to determine a sufficient sample rate that would avoid aliasing for the entire set.

4. HKS needs to significantly improve ABAQUS' ability to process highly transient data. The current implementation of outputting data uses improper decimation methodology which easily induces aliasing. The digital filter implementation induces several distortions and is very susceptible to causing additional aliasing errors because of its unprotected resampling algorithm. Lastly, no features are available to compute or plot spectrum from time-domain data and filter response quantities such as magnitude, phase and time delay.

Acknowledgments

The authors wish to sincerely thank colleagues Dr. Dave Yeager and Dr. Jason McIntosh for their numerous discussions of and insight into the world of DSP. Additional appreciation is owed to Dr. McIntosh for his efficient C++ coding of many of the algorithms used in this work.

References

Brillouin, L., *Wave Propagation and Group Velocity*, Academic Press, 1960.

Diehl, T., *Diehl's DSP Extensions*, 1999. Available for download at http://mathcad.adeptscience.co.uk/dsp/

IES, *Handbook for Dynamic Data Acquisition and Analysis*, Institute of Environmental Sciences, ISBN 1-877862-47-9, no copyright data listed.

Ifeachor, E., Jervis, B., *Digital Signal Processing: A Practical Approach*, Addison-Wesley, 1993.

Madisetti, V., Williams, D., *The Digital Signal Processing Handbook*, CRC Press, 1998.

Mathsoft, *Mathcad 7 User's Guide*, Mathsoft Inc., 1997.

Mathsoft, *Mathcad Signal Processing Function Pack*, Mathsoft Inc., 1998.

Math Works, *Signal Processing Toolbox For Use with MATLAB: User's Guide*, The Math Works Inc., 1998.

Oppenheim, A., Schafer, R., *Digital Signal Processing*, Prentice-Hall, 1975.

Stearns, S., David, R., *Signal Processing Algorithms*, Prentice-Hall, 1988.

Ziemer, R., Tranter, W., Fannin, D., *Signals and Systems: Continuous and Discrete*, Macmillan Publishing, 1983.