U.S. Department
of Transportation

**National Highway
Traffic Safety
Administration**

DOT HS
Draft Report

January 2006

The SimFil Program v.2: Simultaneous Filtering

Stuart G. Mentzer
Information Systems and Services, Inc.
8601 Georgia Ave., Suite 708
Silver Spring, MD 20910

Prepared for:

National Highway Traffic Safety Administration
400 7th Street, S.W.
Washington, DC 20590

Under Contract No. DTNH22–01–C–07317

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT HS		

4. Title and Subtitle	5. Report Date
THE SIMFIL PROGRAM V.2: Simultaneous Filtering	January 2006
	6. Performing Organization Code
	8. Performing Organization Report No.
7. Author(s) Stuart G. Mentzer	

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Information Systems and Services, Inc. 8601 Georgia Ave., Suite 708 Silver Spring, MD 20910	11. Contract or Grant No. DTNH22–01–C–07317
	13. Type of Report and Period Covered Draft Report October 2001 – January 2006
12. Sponsoring Agency Name and Address National Highway Traffic Safety Administration 400 7th Street, S.W. Washington, DC 20590	14. Sponsoring Agency Code

15. Supplementary Notes

Office of Crashworthiness Research/NHTSA, NRD-11

16. Abstract

This report presents a low-pass motion filtering technique that simultaneously provides accurately filtered acceleration, velocity, and displacement signals corresponding to a given motion. This specialized filter was developed for sensitive vehicle crash modeling applications where applying standard low-pass digital filters to one of the three signals causes unacceptable distortions in one or both of the other two corresponding signals. This filter has been implemented in the SimFil program v.2.

SimFil separates out a baseline motion and performs a mirroring transformation to obtain a residual motion for which frequency domain filtering is equivalent in all three signal domains. A simultaneous DFT representation is used that is constrained to assure that the signal tail values in all three signal domains are maintained by the filter.

The techniques used in SimFil are readily extendable to other applications that require simultaneously filtering a signal and some of its integrals and/or derivatives.

SimFil v.2 adds enhanced tail smoothing and other capabilities to SimFil v.1, which was documented in the earlier report DOT HS 807 731 (available to the public from the National Technical Information Service). The source code for the SimFil program is available from the National Highway Traffic Safety Administration, Office of Crashworthiness Research (NRD-11).

17. Key Words	18. Distribution Statement
Digital filter, low-pass filter, discrete Fourier transform, least squares, constrained optimization.	This document is available to the public from the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	25	

Contents

1 Introduction

Digital low-pass filters are designed to exclude higher frequencies from a one-dimensional discrete-time signal. Standard filter designs do not specifically consider the effect of filtering on the integrals and derivatives of a signal. Despite the fact that integration is a linear process, experience shows that filtering a signal can cause significant distortion of its integrals. This report explains the cause of such integral distortion and describes an approach to filtering motion data, where the filtered motion is required to be a reasonable low-pass representation in the acceleration, velocity, and displacement signal domains. Although this document describes filtering of motion signals SimFil's *simultaneous filtering* method is applicable to filtering any three "adjacent" domains of a time series signal and can be extended to handle other simultaneous filtering requirements.

The motivation for developing this filtering approach comes from sensitive vehicle crash modeling applications at the National Highway Traffic Safety Administration (NHTSA). The modeling process is sensitive to small errors and inconsistencies in the acceleration, velocity, and displacement signals. Standard time domain filters were found to introduce unacceptable distortion in displacements computed from filtered accelerometer signals and in the tail (time-zero and final time) values of all three signals.

The SimFil program was developed as a stand-alone filter and its algorithms have also been incorporated into NHTSA crash modeling software. The key to the SimFil approach is the observation that removing an appropriate baseline function and performing a simple mirroring transformation leaves a residual motion for which frequency domain filtering in all three signal domains is equivalent, and for which tail value control and an analytical continuous-time motion are readily obtained.

This report presents the mathematical basis for SimFil and some of the key implementation aspects. An attempt has been made to indicate where alternative, possibly more efficient, algorithms can be used when less stringent performance criteria are acceptable.

SimFil v.2 is integrated into the NHTSA computing environment. In addition to filtering linear motions, SimFil can handle the full range of NHTSA time series, such as force-impulse and angular motion data. Linear motion terminology is used in this report for simplicity of presentation.

SimFil v.2 extends the original SimFil v.1 program [2] with enhanced tail smoothing capabilities and an updated user interface. The source code for SimFil is available from the NHTSA Research & Development Office of Crashworthiness Research.

1.1 Integral Distortion

Standard numerical integration formulas correspond to linear time-invariant (LTI) systems thus they are equivalent to convolution filters in the unbounded time domain. If the measured acceleration is a_n, then the trapezoid rule velocity is

$$\begin{aligned} v_n &= v_{n-1} + \tfrac{\Delta}{2}\,(a_n + a_{n-1}) \\ &= \tfrac{\Delta}{2}\,a_n + \Delta \sum_{i=1}^{\infty} a_{n-i} \\ &= h^{\mathrm{tr}}(n) * a(n) \end{aligned}$$

where Δ is the time step, $h^{\mathrm{tr}}(n)$ is the impulse response of the trapezoid rule, and $*$ is the convolution operator. Since $h^{\mathrm{tr}}_k = 0$ for $k < 0$, $h^{\mathrm{tr}}_0 = \Delta/2$, and $h^{\mathrm{tr}}_k = \Delta$ for $k \geq 1$, this recursive first order integration process has an infinite impulse response (IIR).

The properties of LTI systems [4] reveal that convolution is commutative. This seems to imply that filtering and integration can be applied in either order, yet in practice integration is performed over a finite time period with initial conditions, so it is not a true convolution. Performing trapezoid rule integration with an initial condition yields

$$v_n = v_0 + \tfrac{\Delta}{2}\,a_n + \Delta \sum_{i=1}^{n-1} a_{n-i} + \tfrac{\Delta}{2}\,a_0.$$

Let an arbitrary filter operation with impulse response $h(n)$ be defined by

$$\tilde{y}_n = h(n) * y(n) = \sum_k h_k\, y_{n-k}.$$

Then filtering the acceleration followed by integration yields the signal

$$\begin{aligned} v'_n &= v_0 + \tfrac{\Delta}{2}\,\tilde{a}_n + \Delta \sum_{i=1}^{n-1} \tilde{a}_{n-i} + \tfrac{\Delta}{2}\,\tilde{a}_0 \\ &= v_0 + \tfrac{\Delta}{2} \sum_k h_k\, a_{n-k} + \Delta \sum_{i=1}^{n-1} \sum_k h_k\, a_{n-i-k} + \tfrac{\Delta}{2} \sum_k h_k\, a_{-k} \\ &= v_0 + \sum_k h_k\,(v_{n-k} - v_{-k}) \\ &= v_0 + \tilde{v}_n - \tilde{v}_0. \end{aligned}$$

Thus if the velocity at time-zero is changed by filtering, integration and filtering do not commute. If this "biased" velocity is integrated to obtain a displacement, the same type of

analysis shows that

$$\begin{aligned} d'_n &= d_0 + \tfrac{\Delta}{2}\, v'_n + \Delta \sum_{i=1}^{n-1} v'_{n-i} + \tfrac{\Delta}{2}\, v'_0 \\ &= d_0 + \tfrac{\Delta}{2}\, \tilde{v}_n + \Delta \sum_{i=1}^{n-1} \tilde{v}_{n-i} + \tfrac{\Delta}{2}\, \tilde{v}_0 + n\Delta\,(v_0 - \tilde{v}_0) \\ &= d_0 + \tilde{d}_n - \tilde{d}_0 + n\Delta\,(v_0 - \tilde{v}_0)\,. \end{aligned}$$

The velocity bias has become a linear displacement drift over time, and integrating the filtered velocity contributes a bias to the displacement, just as occurred in the velocity above.

These results suggest a simple solution to the integral distortion problem for time domain filters. Namely, use the filtered initial conditions, $\tilde{v}_0$ and $\tilde{d}_0$, for the integration constants v_0 and d_0. (This only requires values of the integrals that are used in the convolution sum for the time-zero filtered value, which depends on the nonzero elements of $h(n)$.) Note that the time domain filter also requires some signal values beyond the time range desired for the output, and consistent values must be used when filtering its integrals.

It is worth noting that all standard numerical differentiation formulas correspond to true finite impulse response (FIR) operations, and thus differentiation and filtering commute. This suggests that the integration distortion problem could alternatively be avoided by first integrating to the highest order signal of interest, and then generating any filtered signals needed by filtering and differentiating (in any order) this signal. Unfortunately, numerical differentiation introduces roundoff noise so this is not an ideal alternative.

Although these techniques are recommended for eliminating integral distortion in otherwise satisfactory time domain filters, the criteria for motion filtering in sensitive modeling applications can not be met by filters of this class. They do not maintain the signal tail values at low cutoff frequencies, and they can not provide an analytical representation for highly accurate, consistent interpolation of the filtered motion in all three domains.

1.2 Motion Representation and Tail Values

A frequency domain approach to motion filtering can provide a continuous-time, analytical motion representation from which filtered acceleration, velocity, and displacement signals can be generated without the errors associated with numerical integration, differentiation, or interpolation. By applying careful transformations to the motion the time-zero and final time tail values can also be preserved. For these reasons, SimFil is based on a discrete Fourier transform (DFT) representation. The only drawback to this approach is its greater computational cost.

The transformations used in SimFil can be motivated by first considering the problem of using a DFT representation to filter a single time series signal. A DFT implicitly treats the signal as periodic, so any difference between the time-zero and final tail values appears as a discontinuity causing Gibb's oscillations in the filtered result. Because of this effect it is common to remove a line through the tails of a signal before applying a DFT-based filter to the residual signal, and finally add the line back to obtain the filtered result. The periodic version of the residual signal may still have a slope discontinuity (which can also cause noticeable oscillations). As described later in this report, a transformation that extends the residual signal by adding a mirrored, inverted copy of itself can be used to eliminate the slope discontinuity.

While these transformations suffice for filtering one signal, more complex methods are needed for the motion filtering application. In general the separately computed DFT representations of the acceleration, velocity, and displacement signals will not be analytically related as integrals/derivatives. In the next section a baseline motion (having more than the two parameters of a tail line) is developed. Separating out this baseline motion before applying the mirroring transformation is shown to leave a residual motion for which frequency domain filtering is essentially equivalent in all three signal domains. These transformations also facilitate a method of maintaining the motion signal tail values in the filtered result.

The complete filtering process is described in detail in the sections that follow.

2 Signal Transformations

As the cutoff frequency of a standard low-pass filter approaches zero the filtered signal approaches the average value of the signal. For some applications the initial and final values of the signal are considered accurate or have defined values. In the case of motion filtering, the initial velocity and displacement values corresponding to an acceleration signal may be specified, and the initial acceleration in crash test data is generally near zero.

The approach in SimFil is to maintain all six tail values of the acceleration, velocity, and displacement while providing a good low-pass approximation in each of the three signal domains. SimFil achieves this by removing a low frequency baseline function from the motion, filtering the residual motion without changing the zeroed tail values, and finally restoring the baseline function. The baseline function is a low frequency representation of the motion that matches all six tail values. As the filtering cutoff frequency approaches zero the filtered motion approaches the baseline motion.

Let the raw motion, after any necessary integration or differentiation, be given by

$$d_n, \quad v_n, \quad a_n, \qquad n = 0, \ldots, N$$

with the six tail values $\{d_0, d_N, v_0, v_N, a_0, a_N\}$.

The baseline function used in SimFil (shown in all three domains) is

$$\begin{aligned}
\bar{a}(t) &= a_0 + a_s t + \beta_1 \sin\left(\frac{\pi t}{T}\right) + \beta_2 \sin\left(\frac{2\pi t}{T}\right) \\
\bar{v}(t) &= v_0 + a_0 t + a_s \frac{t^2}{2} - \frac{T}{\pi}\beta_1 \cos\left(\frac{\pi t}{T}\right) - \frac{T}{2\pi}\beta_2 \cos\left(\frac{2\pi t}{T}\right) + \frac{T}{\pi}\left(\beta_1 + \frac{\beta_2}{2}\right) \\
\bar{d}(t) &= d_0 + v_0 t + a_0 \frac{t^2}{2} + a_s \frac{t^3}{6} - \left(\frac{T}{\pi}\right)^2 \beta_1 \sin\left(\frac{\pi t}{T}\right) - \left(\frac{T}{2\pi}\right)^2 \beta_2 \sin\left(\frac{2\pi t}{T}\right) + \frac{T}{\pi}\left(\beta_1 + \frac{\beta_2}{2}\right) t
\end{aligned} \tag{2.1}$$

where $T = N\Delta$ is the time span of the motion data, and where

$$\begin{aligned}
a_s &= \tfrac{1}{T}(a_N - a_0) \\
\beta_1 &= \tfrac{\pi}{2T}(v_N - v_0) - \tfrac{\pi}{4}(a_N + a_0) \\
\beta_2 &= \tfrac{2\pi}{T^2}(d_N - d_0) - \tfrac{\pi}{T}(v_N + v_0) + \tfrac{\pi}{6}(a_N - a_0).
\end{aligned}$$

This baseline function depends on the six parameters $\{d_0, v_0, a_0, a_s, \beta_1, \beta_2\}$ and passes through the six tail values. The sinusoidal terms correspond to the lowest two frequencies that will

be used in the DFT representation. Other well-behaved six parameter functions could serve equally well. (A quintic polynomial would probably be a poor choice because of the tendency of higher degree polynomials to oscillate.) Generally, removing the baseline function will not significantly alter other than the lowest frequencies in the motion spectrum.

After subtracting the baseline function the *zeroed signals* (having zero tail values) are defined by

$$\hat{a}_n = a_n - \bar{a}_n, \quad \hat{v}_n = v_n - \bar{v}_n, \quad \text{and} \quad \hat{d}_n = d_n - \bar{d}_n$$

where the subscript n corresponds to time $n\Delta$ for the continuous baseline function. The zeroed signals have no tail discontinuities when considered as circularly wrapped, periodic functions (as occurs implicitly when taking the DFT). Nevertheless, there are some derivative discontinuities and asymmetries at the tails that would allow the tail values to change as a result of directly filtering the zeroed motion.

Campbell's transformation [6] can be applied before filtering to help achieve the desired tail behavior. This involves appending an inverted, mirrored copy of a signal onto itself. Transforming the acceleration corresponds to transforming the displacement in the same manner, and to appending a mirrored (but not inverted) copy of the velocity, as shown in Figure 2.1.

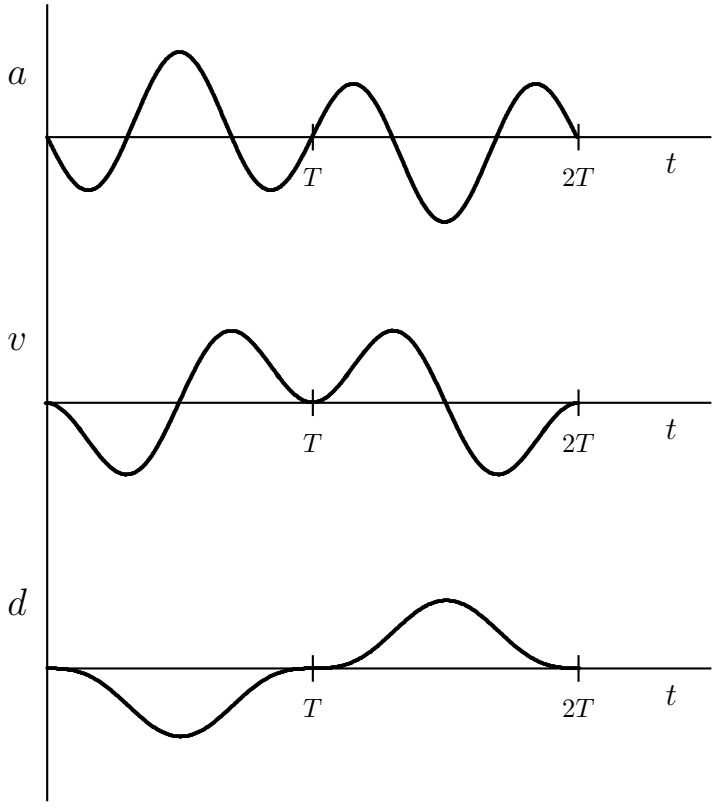

Figure 2.1: Campbell's transformation of zeroed motion.

Formally, the definitions of the zeroed signals are extended to include

$$\hat{a}_{2N-n} = -\hat{a}_n, \quad \hat{v}_{2N-n} = \hat{v}_n, \quad \text{and} \quad \hat{d}_{2N-n} = -\hat{d}_n, \quad n = 0, \ldots, N-1.$$

This zeroed, extended motion will be called the *residual motion*.

The periodic version of the residual motion has at least first derivative continuity in all three signal domains. Also note that the acceleration and displacement are odd functions and the velocity is an even function. If a time domain filter with a symmetric impulse response is applied to an odd periodic function the tail values remain unchanged. So Campbell's transformation has made it possible to retain four of the six tail values for any cutoff frequency, and produces signals with a high degree of tail continuity. The filtering process in SimFil is applied to this periodic, residual motion.

If the numerical errors of a time domain representation are acceptable, a symmetric FIR filter could be applied to the residual signal along with the methods of Section 1.1 to prevent integral distortion instead of the frequency domain filter developed in the remainder of this report. In this case, filtering the acceleration requires only updating the initial velocity since the filtered initial displacement will be unchanged. The velocity tail values would *not* be preserved in this approach.

3 DFT-Based Filter

Given the residual motion, a frequency domain representation is used for the filtering process. The periodic, residual acceleration and displacement are odd functions, so they have pure imaginary sine series as their discrete Fourier transforms (DFT's). Similarly, the velocity is an even function, so it has a pure real cosine transform. The tails of all three residual signals are also zero. These observations suggest that the following windowed sine and cosine series would be appropriate representations for the filtered residual signals

$$\begin{aligned} \tilde{a}(t) &= \sum_{k=1}^{K} \grave{A}_k \sin\left(\tfrac{\pi k t}{T}\right) \\ \tilde{v}(t) &= \sum_{k=1}^{K} \grave{V}_k \cos\left(\tfrac{\pi k t}{T}\right) \\ \tilde{d}(t) &= \sum_{k=1}^{K} \grave{D}_k \sin\left(\tfrac{\pi k t}{T}\right) \end{aligned} \tag{3.1}$$

where the windowed coefficients $\grave{A}_k$, $\grave{V}_k$, and $\grave{D}_k$ are assumed to be zero for $k > K$.

These representations have the correct even and odd senses and the zero acceleration and displacement tails. Zero velocity tails are not automatically assured by this representation, but, as shown below, they can be achieved by constraining the choice of coefficients. No $\grave{A}_0$ or $\grave{D}_0$ coefficient is included because the corresponding sine term is identically zero. A nonzero $\grave{V}_0$ term in the velocity would introduce an undesired linear term of $\grave{V}_0\, t$ in displacement.

For the filtered signals of (3.1) to represent the same motion they must be analytically related as continuous-time integrals and derivatives, which requires

$$\grave{V}_k = -\left(\frac{T}{\pi}\right)\frac{\grave{A}_k}{k}\,, \qquad \text{and} \qquad \grave{D}_k = \left(\frac{T}{\pi}\right)\frac{\grave{V}_k}{k} = -\left(\frac{T}{\pi}\right)^2\frac{\grave{A}_k}{k^2}. \tag{3.2}$$

Coefficients that meet these criteria will be called *consistent*.

The remaining task is to select a set of consistent coefficients that provide an acceptable filtered representation of the motion. The DFT's of the residual motion signals are a natural choice for the coefficients. Taking DFT's of each residual signal independently provides these three candidate sets of unwindowed coefficients

$$\begin{aligned} A_k &= \left(\frac{2}{N}\right) \sum_{n=0}^{N-1} \hat{a}_n \sin\left(\frac{\pi k n}{N}\right) \\ V_k &= \left(\frac{2}{N}\right) \sum_{n=0}^{N-1} \hat{v}_n \cos\left(\frac{\pi k n}{N}\right) \\ D_k &= \left(\frac{2}{N}\right) \sum_{n=0}^{N-1} \hat{d}_n \sin\left(\frac{\pi k n}{N}\right) \end{aligned} \tag{3.3}$$

for $1 \leq k \leq N-1$. (The sums can actually start at $n = 1$ since the initial values are all zero.) Note that Campbell's transformation implicitly determines the form of the DFT's.

Any one of these sets of coefficients could be then be windowed by multiplying by windowing factors, such as in $\grave{A}_k = H_k A_k$, and then the consistency requirements of (3.2) applied to determine the coefficients for the other two signal domains. (Actually these two steps can be done in either order since windowing preserves consistency.) The question remains to decide which of these three choices provides the best filter.

In fact, it is not difficult to show that, for a given continuous-time motion, these three sets of coefficients are asymptotically consistent as N goes to infinity for residual motion signals generated with any convergent integration and/or differentiation formulas (with infinite precision real arithmetic). Moreover, choosing one set of coefficients is equivalent to using its DFT representation as the integration and/or differentiation tool, which is certainly a valid, if not fast, method.

Since in practice only one of the three signals is actually measured, it seems that a good filtered representation of the residual motion can be obtained by taking the independent DFT of the residual measured signal, windowing, and applying the consistency requirements. This avoids dependence on the numerically computed residual signals in the other two domains (although the tail values in the other two domains are still used to compute the baseline function). This approach does indeed provide a good motion filter, but maintaining the zero tails of the residual velocity will require a more involved approach as described below.

3.1 Simultaneous DFT

A more general approach to choosing the DFT representation will facilitate the inclusion of velocity tail constraints. A consistent *simultaneous DFT* representation can be explicitly formed as a weighted least squares approximation to all three signals simultaneously. (A standard DFT representation with any subset of the frequency terms is a least squares approximation of one signal.)

With errors in the three domains weighted by w_a, w_v, and w_d, respectively, the simultaneous DFT (without velocity tail constraints), expressed in terms of the acceleration domain

coefficients, is

$$A_k = \left(\frac{2}{N}\right) \frac{\sum_{n=0}^{N-1} \left[w_a \hat{a}_n \sin\left(\frac{\pi k n}{N}\right) - w_v \left(\frac{T}{\pi k}\right) \hat{v}_n \cos\left(\frac{\pi k n}{N}\right) - w_d \left(\frac{T}{\pi k}\right)^2 \hat{d}_n \sin\left(\frac{\pi k n}{N}\right) \right]}{w_a + w_v \left(\frac{T}{\pi k}\right)^2 + w_d \left(\frac{T}{\pi k}\right)^4} \tag{3.4}$$

The corresponding V_k and D_k coefficients are readily computed by using the consistency requirements (3.2).

This is a generalization of the independent DFT's since they can be reproduced by simply setting two of the weights to zero. The asymptotic consistency of the independent DFT's implies that the unconstrained simultaneous DFT is asymptotically independent of the weights. The real value of the simultaneous DFT comes when the velocity tail constraints are added and this weight independence is lost. In that case the weighting can be used to assure that the filtered motion is still a good smoothed approximation in all three signal domains.

3.2 Velocity Tail Constraints

Although any of the DFT representations given above can be expected to approximately match the zero tail values of the residual velocity, it is reasonable to explicitly constrain the velocity tails. In practice the DFT coefficients are windowed to achieve a given frequency response, so it is this windowed representation that must be constrained.

If the windowed, constrained velocity coefficients are $\grave{V}_k$, the velocity representation of (3.1) and the requirements $\tilde{v}(0) = \hat{v}_0 = 0$ and $\tilde{v}(T) = \hat{v}_N = 0$ can readily be shown to yield the constraints

$$\sum_{\substack{k=1 \\ k \text{ odd}}}^{K} \grave{V}_k = 0 \quad \text{and} \quad \sum_{\substack{k=1 \\ k \text{ even}}}^{K} \grave{V}_k = 0 \qquad \stackrel{(3.2)}{\Longrightarrow} \qquad \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} \frac{\grave{A}_k}{k} = 0 \quad \text{and} \quad \sum_{\substack{k=1 \\ k \text{ even}}}^{K} \frac{\grave{A}_k}{k} = 0.$$

Solving for the constrained DFT is then a least squares problem with two equality constraints. Standard Lagrange multiplier techniques [1] give the solution for the constrained simultaneous DFT as

$$\grave{A}_k = H_k A_k - \lambda_k \left(\frac{H_k^2}{k W_k}\right) \tag{3.5}$$

where H_k is the filter windowing factor, A_k is the unconstrained simultaneous DFT (3.4),

$$W_k = w_a + w_v \left(\frac{T}{\pi k}\right)^2 + w_d \left(\frac{T}{\pi k}\right)^4,$$

and

$$\lambda_k = \begin{cases} \sum_{\substack{j=1 \\ j \text{ odd}}}^{K} \frac{H_j A_j}{j} \Big/ \sum_{\substack{j=1 \\ j \text{ odd}}}^{K} \frac{H_j^2}{j^2 W_j} & , \quad k \text{ odd} \\ \sum_{\substack{j=1 \\ j \text{ even}}}^{K} \frac{H_j A_j}{j} \Big/ \sum_{\substack{j=1 \\ j \text{ even}}}^{K} \frac{H_j^2}{j^2 W_j} & , \quad k \text{ even.} \end{cases}$$

The effort of computing the $\grave{A}_k$ is almost entirely in computing the unconstrained coefficients, A_k. So A_k can be stored, independent of the choice of filtering window, and $\grave{A}_k$ computed quickly for a specified window (as long as all the needed coefficients were saved).

Although the first term in (3.5), $H_k A_k$, is asymptotically independent of the weights, the second term is not. Thus the weights can be selected to balance the emphasis on closeness of fit among the three signal domains. The choice of weights also determines how the changes to the DFT are distributed over the frequency range.

In practice, these two constraints rarely cause a large absolute change from the unconstrained filtered representation, so the choice of weights is generally not critical. It is important to note that the changes to the DFT are additive, not multiplicative, thus the effect on the overall frequency response can be significant at small magnitude coefficients even though the absolute change to the DFT is small.

3.3 Filtered Signals

Once a consistent set of windowed coefficients $\grave{A}_k$, $\grave{V}_k$, and $\grave{D}_k$ has been selected, the overall filtered motion signals are given by

$$\begin{aligned} \grave{a}(t) &= \sum_{k=1}^{K} \grave{A}_k \sin\left(\frac{\pi k t}{T}\right) + \bar{a}(t) \\ \grave{v}(t) &= \sum_{k=1}^{K} \grave{V}_k \cos\left(\frac{\pi k t}{T}\right) + \bar{v}(t) \\ \grave{d}(t) &= \sum_{k=1}^{K} \grave{D}_k \sin\left(\frac{\pi k t}{T}\right) + \bar{d}(t) \end{aligned} \tag{3.6}$$

where $\bar{a}(t)$, $\bar{v}(t)$, and $\bar{d}(t)$ are the baseline functions (see Section 2).

Because of the signal transformations and velocity tail constraints, the overall frequency response depends somewhat on the motion. The only general classification that applies to this filtering process is linearity.

4 SimFil Implementation

A number of algorithmic details are important to the SimFil implementation of the simultaneous DFT filter presented above. These include the integration and differentiation algorithms, tail smoothing, the choice of DFT computation domain, and techniques to enhance the efficiency of the DFT computations.

4.1 Signal Data Types

SimFil supports the full range of NHTSA time series signal data types, including linear and angular motion and force/impulse signals. Input signals can be from all of the common domains: acceleration, velocity, displacement, force, impulse, and so forth.

When working with angular motion signals it is important to understand that when more than one rotational motion axis is present the angular position, or orientation, is not a vector and cannot be represented by the integral of the angular velocity. A number of 3D rotational position representations are in common use including Euler or Cardan/Bryant angles, rotation matrices, and unit quaternions. It is possible to apply SimFil-based filtering to 3D rotational motion (this has been done in SISAME-3D [3]) but SimFil only considers signals on one axis at a time and so rotations are treated as simple integrals of angular velocity.

4.2 Integration and Differentiation

SimFil accepts an acceleration, velocity, or displacement signal to define a motion. The tail values in all three signal domains are needed for the baseline function and these are obtained by numerical time domain methods. These numerically derived signals are also used in the tail smoothing process (see below) and if the user requests the subsampling or "No Filtering" modes (using SimFil as a numerical integration/differentiation tool and/or to subsample the motion signals). As described below, the frequency domain computations are based only on the residual measured (input) signal to minimize the effects of numerical errors.

The numerical integration and differentiation methods used are given below. A combination of standard low order and cubic spline based methods are used to minimize sensitivity to

noisy test data while using integration and differentiation formulas that are true inverses wherever possible.

The sampling time step is Δ. The initial velocity, v_0, is assumed specified with measured acceleration and displacement signals, and the initial displacement, d_0, is assumed specified with measured acceleration and velocity signals (SimFil uses $d_0 = 0$ by convention in the NHTSA environment). Although SimFil only produces filtered output from time-zero to a user-specified final time, portions of the signals outside this range are used in performing tail smoothing (see below), so they are computed over the full time span of the measured signal (integration formulas for negative indices are not shown explicitly but are readily derived). Conversion factors between the units used in the different signal domains are ignored here, as they are throughout this report. The subscripts l and r are used below for the left and right signal endpoints, respectively, when these values require special treatment.

Acceleration Input

$$\begin{aligned} v_n &= v_{n-1} + \tfrac{\Delta}{2}(a_n + a_{n-1}) = v_0 + \tfrac{\Delta}{2}\left[a_n + 2a_{n-1} + \cdots + 2a_1 + a_0\right] \\ d_n &= d_{n-1} + \Delta v_{n-1} + \tfrac{\Delta^2}{6}(a_n + 2a_{n-1}) = d_0 + n\Delta v_0 + \tfrac{\Delta^2}{6}\left[a_n + 6\sum_{i=1}^{n-1} i a_{n-i} + (3n-1)a_0\right] \end{aligned}$$

Velocity Input

$$\begin{aligned} a_n &= \tfrac{1}{2\Delta}(v_{n+1} - v_{n-1}) \\ a_l &= \tfrac{1}{6\Delta}(-7v_l + 6v_{l+1} + 3v_{l+2} - 2v_{l+3}) \\ a_r &= \tfrac{1}{6\Delta}(7v_r - 6v_{r-1} - 3v_{r-2} + 2v_{r-3}) \\ d_n &= d_{n-2} + \tfrac{\Delta}{3}(v_n + 4v_{n-1} + v_{n-2}) \end{aligned}$$

Displacement Input

$$\begin{aligned} v_n &= \tfrac{1}{2\Delta}(d_{n+1} - d_{n-1}) \\ v_l &= \tfrac{1}{6\Delta}(-7d_l + 6d_{l+1} + 3d_{l+2} - 2d_{l+3}) \\ v_r &= \tfrac{1}{6\Delta}(7d_r - 6d_{r-1} - 3d_{r-2} + 2d_{r-3}) \\ a_n &= \tfrac{1}{2\Delta}(v_{n+1} - v_{n-1}) \end{aligned}$$

$$
\begin{aligned}
a_l &= \tfrac{1}{6\Delta}(-7v_l + 6v_{l+1} + 3v_{l+2} - 2v_{l+3}) \\
a_r &= \tfrac{1}{6\Delta}(7v_r - 6v_{r-1} - 3v_{r-2} + 2v_{r-3})
\end{aligned}
$$

4.3 Tail Smoothing

The simultaneous DFT filter used by SimFil maintains the tail values in all three signal domains. Measured and differentiated signals may exhibit significant noise so that the raw tail values are not appropriate at lower cutoff frequencies. SimFil provides optional tail smoothing to obtain more reasonable tail values.

Tail smoothing is optionally performed for each tail in a two phase process based on the selected tail smoothing frequency. First, the desired smoothed tail values are determined by a weighted average corresponding to an ideal low-pass time domain filter windowed by a raised cosine function. The ideal low-pass impulse response is

$$h_n^I = \frac{\sin(\omega_s n)}{\pi n}$$

where $\omega_s = 2\pi f_s \Delta$ and f_s is set to twice the user-specified smoothing frequency. The cosine (Hanning) window is given by

$$w_n = \tfrac{1}{2}\left(1 + \cos\left(\frac{\omega_s n}{\rho}\right)\right) \quad , \qquad -N_w \le n \le N_w \quad , \qquad N_w = \frac{\pi\rho}{\omega_s}$$

where ρ determines the length of the windowed filter ($\rho = 2$ is used in SimFil), and $w_n = 0$ outside the range $-N_w \le n \le N_w$. The windowed filter impulse response is then

$$h_n = \frac{1}{\sigma} w_n h_n^I = \frac{1}{2\sigma}\left(1 + \cos\left(\frac{\omega_s n}{\rho}\right)\right)\frac{\sin(\omega_s n)}{\pi n}$$

for $-N_w \le n \le N_w$, where $\sigma = \sum_{k=-N_w}^{N_w} w_k h_k^I$ normalizes the response to a total weight of one.

If the full range of tail data is available the smoothed tail values are computed as

$$\grave{a}_0 = \sum_{k=-N_w}^{N_w} h_k a_k \qquad \text{and} \qquad \grave{a}_N = \sum_{k=-N_w}^{N_w} h_k a_{N-k}$$

and similarly for smoothed velocity and displacement tails. This process generates tail values that approximate those obtained by filtering the extended acceleration with a well-behaved filter to a cutoff frequency equal to the specified smoothing frequency.

If the full range of data on both sides of a tail is not available SimFil corrects the truncated impulse response such that tail values of straight line signals would be preserved (the full symmetric impulse response has this property). This improves the smoothed tail value selection. For a tail data range of $N_L \le n \le N_R$ the corrected impulse response used by SimFil is of the form $h_n + \gamma_0 + \gamma_1|n|$. The parameters are determined by normalization to a total weight of one

$$\sum_{n=N_L}^{N_R} (h_n + \gamma_0 + \gamma_1|n|) = 1$$

and by the straight line (zero moment) condition

$$\sum_{n=N_L}^{N_R} (h_n + \gamma_0 + \gamma_1 |n|)\, n = 0\,.$$

The second phase of the tail smoothing process is to adjust the motion to pass through the selected tails without introducing discontinuities and without altering the central portion of the signals. This is accomplished by finding an appropriate smoothing function at each tail and, linearly in time, blending the smoothing function into the actual tail. Looking at the time-zero tail, let $\grave{a}_0$, $\grave{v}_0$, and $\grave{d}_0$ be the selected (smoothed or raw) tail values, and let $s(t)$ be the smoothing function in the displacement domain. The smoothed time-zero tail has the representation

$$\begin{aligned}
\breve{d}(t) &= \tfrac{1}{p}\big[td(t) + (p-t)s(t)\big] \\
\breve{v}(t) &= \tfrac{1}{p}\big[d(t) + tv(t) - s(t) + (p-t)s'(t)\big] \\
\breve{a}(t) &= \tfrac{1}{p}\big[2v(t) + ta(t) - 2s'(t) + (p-t)s''(t)\big] \\
\breve{a}'(t) &= \tfrac{1}{p}\big[3a(t) + ta'(t) - 3s''(t) + (p-t)s'''(t)\big]
\end{aligned}$$

for $0 \le t \le p$, where p is the smoothing time span, chosen in SimFil to equal $3N_w\Delta$.

The time-zero tail smoothing function is determined by the six conditions

$$\breve{d}(0) = \grave{d}_0,\;\; \breve{v}(0) = \grave{v}_0,\;\; \breve{a}(0) = \grave{a}_0,\;\; \breve{v}(p) = v(p),\;\; \breve{a}(p) = a(p),\;\; \breve{a}'(p) = a'(p)$$

where the additional continuity condition $\breve{d}(p) = d(p)$ is assured by the definition of $\breve{d}(t)$. These conditions require

$$s(0) = \grave{d}_0, \quad s'(0) = \grave{v}_0 - \frac{d(0) - \grave{d}_0}{p}, \quad s''(0) = \grave{a}_0 - \frac{2(v(0) - s'(0))}{p}$$

$$s(p) = d(p), \quad s'(p) = v(p), \quad s''(p) = a(p)$$

which says that $s(t)$ much match the six tail values of the signals over the smoothing time span. The baseline function presented in Section 2 provides a well-behaved function of this type, so SimFil uses this same functional form for $s(t)$. The corresponding function for the end tail can be derived directly, or the above results can be applied in a time reversed fashion to the end tail with all velocities negated.

The smoothing process provides a gradual smoothing towards the tails over the smoothing time span without affecting the central portion of the motion. Velocity and displacement tail smoothing can introduce some sinusoidal tail baseline function content to meet this requirement, but this is usually eliminated if the filtering frequency is no greater than the the tail smoothing frequency. There is some loss of frequency content in the tail regions and the approach taken here is an admittedly heuristic solution to a practical problem. The best choice of smoothing frequency depends on the specific application.

4.4 DFT Signal Domain

The constrained, windowed DFT coefficients $\grave{A}_k$ are computed from the unconstrained, unwindowed coefficients A_k. Since the unconstrained simultaneous DFT is asymptotically independent of the weights (and nearly so in practice), SimFil uses the residual measured signal to determine the A_k. This significantly reduces the computational burden.

Regardless of the measured signal domain, the velocity tail constraints are implemented to minimize error in the velocity domain, which is equivalent to setting $W_k = 1/k^2$. If the acceleration domain were used, the lowest frequency components would be altered most by the constraints, which would be beneficial in reducing the impact on the frequency response but could cause significant change to the displacements. If the displacement domain were used the constraints would introduce more high frequency content. Of course a weighted combination of the three domain errors could be the best choice for a particular application.

4.5 Filter Window

The frequency window applied to the DFT coefficients is a modified cosine function designed to achieve -3 dB at the cutoff frequency. The window coefficients, H_k, are sampled from this function, but the response at interpolated frequencies varies with the number of points in the measured signal. Although the overall frequency response of SimFil depends to some degree on the input data, the response is typically close to that of the window (with zero phase response) except at the lowest frequencies.

For a cutoff frequency of f_c, the filter window response is given by

$$H(f) = \begin{cases} 1 & , \quad f \le \frac{1}{2} f_c \\ \frac{1}{2}\left[1 + \cos\left(\left(\frac{f}{2f_c} - \frac{1}{4}\right)^{\alpha} \pi\right)\right] & , \quad \frac{1}{2} f_c < f < \frac{5}{2} f_c \\ 0 & , \quad f \ge \frac{5}{2} f_c \end{cases}$$

where

$$\alpha = \frac{\log(\pi) - \log\left(\arccos\left(2 \cdot 10^{-.15} - 1\right)\right)}{\log(4)}.$$

The correspondence with the sampled window coefficients is $H_k = H(k/2T)$. As shown in Figure 4.1, the SimFil frequency window falls within the envelope specified in the Society of Automotive Engineers' J211 recommended practice [5].

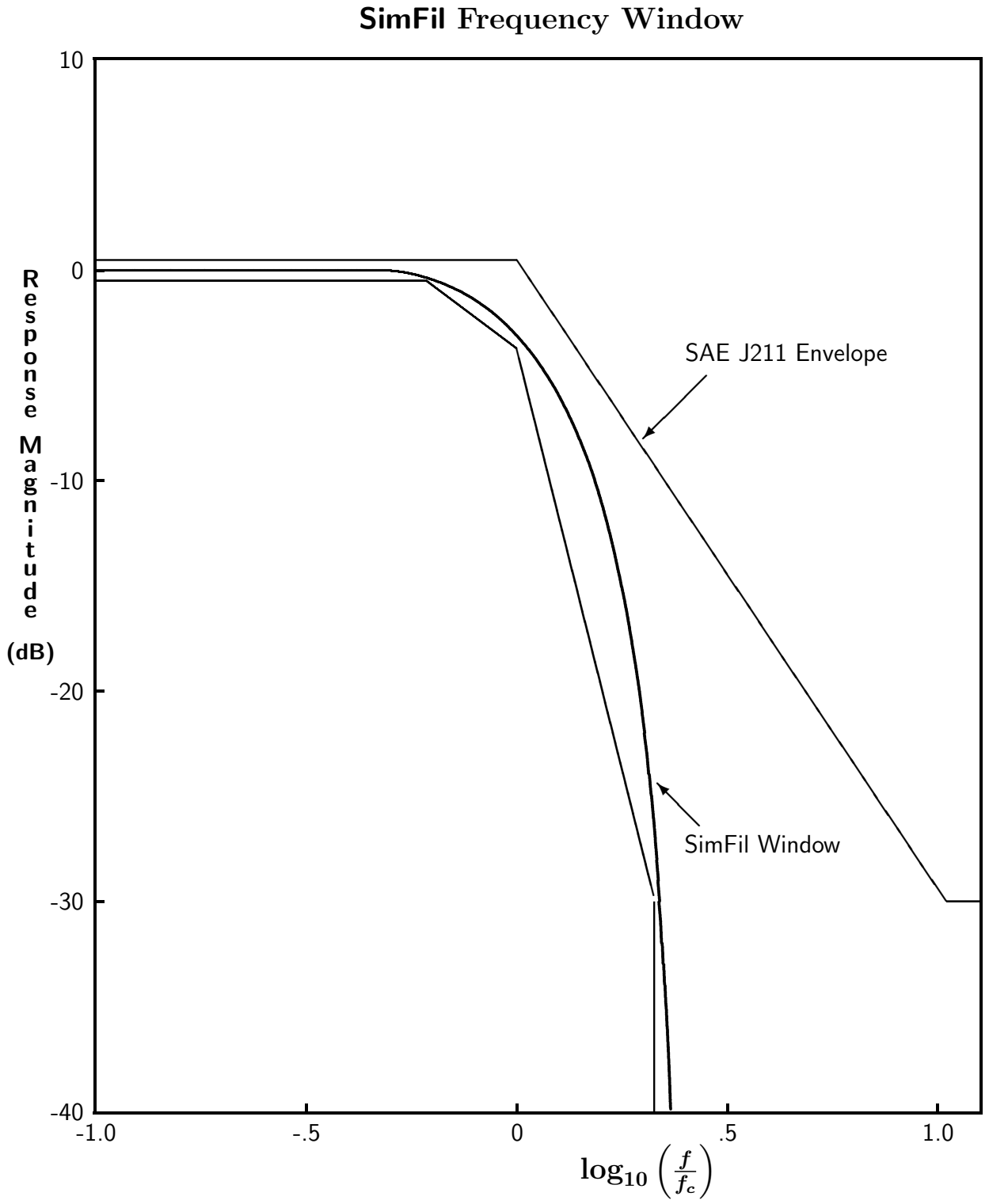

Figure 4.1: SimFil frequency window.

4.6 DFT Computation

SimFil is implemented with an efficient direct DFT algorithm using a single time fold that takes $O(NK)$ time for K coefficients. This algorithm is sufficiently fast for typical low frequency SimFil applications. An FFT algorithm could be substituted, but if padding is required then either the initial or end tails become interior points and they are no longer precisely preserved during filtering. (An FFT can be implemented that takes only $O(N \log K)$ time for K coefficients by performing only $\log K$ of the FFT time or frequency folds and completing the computation with the direct method.)

The SimFil DFT computation exploits the redundancy that is present for any N by performing a single time fold (or decimation in frequency). If a suitable filter time step is used then a time fold is also used for the inverse DFT. These folds reduce the effort by roughly a factor of two.

Pre-tabulated sine and cosine arrays and an efficient indexing scheme are also used in the DFT and, if the filtered time step allows, the inverse DFT to avoid the expense of computing the same trigonometric values many times. SimFil allows any combination of the acceleration, velocity, and displacement filtered outputs, and the reconstruction routine used is tailored to the combination selected for efficiency.

When multiple filtered outputs are requested for the same input signal, DFT time span, and tail smoothing, the unconstrained DFT from a previous SimFil operation is reused as long as the maximum frequency required is no greater than that contained in the DFT. Thus a non-increasing sequence of frequencies assures that only one DFT needs to be computed.

4.7 SimFil Computation Overview

The actual filtering process used by SimFil is summarized below.

1. Numerically integrate and/or differentiate the measured motion signal provided to obtain the tail values in all three signal domains.
2. Perform any user-specified tail smoothing.
3. Compute the baseline function (2.1) from the tail values.
4. Remove the baseline function from the measured signal and (implicitly) apply Campbell's transformation to obtain the residual measured signal.
5. Compute the unwindowed, unconstrained independent DFT of the residual measure signal (3.3).

6. If necessary, compute the corresponding consistent acceleration DFT coefficients, A_k (3.2).

7. Compute the windowed, constrained coefficients $\grave{A}_k$ using velocity domain error weighting $W_k = 1/k^2$ (3.5).

8. Compute the consistent windowed, constrained velocity and displacement coefficients $\grave{V}_k$ and $\grave{D}_k$ (3.2).

9. Compute the desired filtered motion signals (3.6).

References

[1] Luenberger, D.G. (1973). *Introduction to Linear and Nonlinear Programming.* Addison–Wesley, Reading, Mass.

[2] Mentzer, S.G. (1991). *The SIMFIL Program: Simultaneous Filtering.* U.S. Department of Transportation Report No. DOT HS 807 731.

[3] Mentzer, S.G. (2003). *The SISAME-3D Program: Structural Crash Model Extraction and Simulation.* U.S. Department of Transportation Draft Report.

[4] Oppenheim, A.V. (1989). *Discrete-Time Signal Processing.* Prentice–Hall, Englewood Cliffs, New Jersey.

[5] Society of Automotive Engineers (1988). *SAE Recommended Practice SAE J211 OCT88.*

[6] Sterns, S.D. (1975). *Digital Signal Analysis.* Hayden, Rochelle Park, New Jersey.