Measurement 58 (2014) 511–519

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Integrating time signals in frequency domain – Comparison with time domain integration

Anders Brandt [a,*], Rune Brincker [b]

[a] Dept. of Technology and Innovation, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark
[b] Aarhus School of Engineering, Aarhus University, Ny Munkegade 120, Building 1521, DK-8000 Aarhus C, Denmark

ARTICLE INFO

Article history:
Received 13 December 2013
Received in revised form 19 May 2014
Accepted 5 September 2014
Available online 16 September 2014

Keywords:
Integration
Signal processing
Frequency domain
Windowed overlap-add
Time series

ABSTRACT

Integrating sampled time signals is a common task in signal processing. In this paper we investigate the performance of two straightforward integration methods: (i) integration in the frequency domain by a discrete Fourier transform (DFT), division by $j\omega$ followed by inverse DFT (IDFT) back to the time domain, and (ii) a method using a weighted overlap-add (WOLA) technique which is developed in the paper. These two methods are compared with two time domain methods: (a) the trapezoidal rule, and (b) an optimized IIR filter. It is shown that the intuitive method of a straightforward DFT/IDFT is a very good method which is recommended for data lengths exceeding 16 K samples, provided data are short enough to allow a single DFT. The IIR filter integration is shown to have very similar accuracy and can also be recommended. The WOLA integration method is shown to perform well in most cases for steady-state conditions. For cases with short transients it should, however, be avoided. A signal integrated by the WOLA method is further shown to be incoherent with the signal before integration. This suggests that the WOLA method should be avoided in cases where coherence between the signals before and after integration is important. It is also demonstrated by a simulation example that integration by the trapezoidal rule should be avoided, as it gives biased results, particularly for higher frequencies.

1. Introduction

Integrating a sampled time signal is a common task in signal processing, for example in vibration engineering applications. It is common in vibration engineering to convert a measured acceleration signal into velocity or displacement. This may be important when time data from sensors producing different output units are analyzed, for example combining accelerometers and laser Doppler vibrometers, or accelerometers and geophone sensors, or when combining accelerometers with strain gauges whose output is proportional to displacement. In mechanical engineering, it is also common to integrate signals in various analytical analysis methods, e.g. the restoring force method [1].

Despite the frequent need of time domain integration, not much has been published on best practices or best methods to be used. Standard textbooks on signal processing [2,3] do not discuss means to integrate signals optimally, although some of the methods for designing infinite impulse response (IIR) filters could be used, albeit with quite some knowledge of signal processing. As shown in a recent overview of methods [4], many methods commonly used because they are intuitive and/or thought to be good, actually perform very badly compared to what can be achieved with little extra effort using more sophisticated, but less known, methods.

* Corresponding author.
E-mail addresses: abra@iti.sdu.dk (A. Brandt), rub@iha.dk (R. Brincker).

http://dx.doi.org/10.1016/j.measurement.2014.09.004
0263-2241/

In [4] it was demonstrated that the trapezoidal rule, a common method for integration which is generally considered to be an accurate integration method, should be avoided for integrating time signals. This is because the integration error for this method is very large except for low frequencies (relative to the sampling frequency). The same was concluded in [5], where it was also shown that the fourth-order Runge–Kutta method, and indeed all methods of Newton–Cotes type [6], are inferior to signal processing based methods for integration of measured signals. It was particularly pointed out in [5], that the Newton–Cotes methods are very prone to errors due to disturbing noise for high-order integrators, because they have sharp resonances in their amplitude characteristics.

Finite impulse response (FIR) filters are in general a bad choice for integration, since a pole at zero cannot be realized by such filters. If the integration operation can be restricted to a limited frequency band, however, such filters have been successfully designed [7]. In [5], a method for designing optimal IIR filters for integration of time signals was presented. The IIR filters can be designed with integer sample time delays, so that the output (integrated) signal is synchronized with the signal before integration, or with other parallel channels of data. Filter parameters for integration with a relative amplitude error of less than -120 dB were presented in the paper.

Digital filters are good for real-time applications, and can also be used in offline data analysis. There are some issues with stability with the filters from [5], although these issues can be addressed by highpass-filtering the signal prior to integrating it. This, in any case, usually should be done with measured vibration signals, to avoid the low-frequency content of the signal dominating the integrated result [4].

The topic of integrating time signals is thus very important, yet very little research has been presented on the performance of different integration methods and no recommendations on best practices have been found by the present authors. Indeed, inferior methods are commonly recommended; a fact most likely due to the lack of existing recommendations. In the present paper, two methods for integrating time domain signals will therefore be investigated and compared to previously presented methods. First, the intuitive method of performing the integration in the frequency domain using one long discrete Fourier transform (DFT), followed by division by $j\omega$, and an inverse DFT will be investigated. This method can be used in post processing only. Second, a method based on the short-time Fourier transform (STFT) and a windowed overlap-add (WOLA) technique will be developed and investigated. This method can be implemented for real-time processing as well as for post processing.

2. Theory

2.1. Integration by a long DFT

An intuitive method for integrating a signal is by computing its DFT (computed, of course, by the fast Fourier transform FFT), then multiplying the DFT by the frequency response function (FRF) $H(f) = 1/j\omega$, and obtaining the time response of the integrated signal by computing the inverse Fourier transform of that product. This method requires that the signal is short enough so that a DFT (FFT) of the entire signal is possible without reaching memory limitations. With the increasing memory capacity of modern computers, this is rarely a limitation in, for example, (offline) vibration analysis tasks.

It should be noted, that in order to avoid aliasing caused by the cyclic convolution property of the inverse DFT in such a procedure, the time signal should be zero padded prior to computing its DFT (see, e.g. [4]). If we assume a sampled input signal $x(n),\ n = 0, 1, \ldots, L-1$, sampled by the sampling frequency f_s, we compute the 2L size DFT

$$X(k) = \sum_{n=0}^{2L-1} x_z(n) e^{-j2\pi kn/(2L)}, \qquad k = 0, 1, \ldots, L-1 \tag{1}$$

where $x_z(n)$ is the zero padded time signal with L values $x(n)$ followed by L zeros. Note that we only compute the positive frequencies in Eq. (1). We then compute the integration operator (frequency response) by

$$H_i(k) = \begin{cases} 1/j\omega_k, & k = 1, \ldots, L \\ 0, & k = 0 \end{cases} \tag{2}$$

since we cannot divide by zero at DC. The frequencies in Eq. (2) are the DFT frequencies $\omega_k = 2\pi k f_s/(2L)$. We now compute the product $Y(k) = X(k)H(k)$ which corresponds to integration in the frequency domain. Before we can compute the inverse transform, the spectrum $Y(k)$ should be extended with even real part and odd imaginary part for the negative frequencies, by defining

$$Y(L+k) = Y^*(L-k), \quad k = 1, 2, \ldots, L-1 \tag{3}$$

where * represents complex conjugate. Finally, the integral $y(n)$ of the signal $x(n)$ is computed by the inverse DFT

$$y(n) = \frac{1}{2L} \sum_{k=0}^{2L-1} Y(k) e^{j2\pi nk/(2L)}, \quad n = 0, 1, \ldots, L-1 \tag{4}$$

which is the first half of the entire DFT, since the remaining values will be zero due to the zero padding.

It should be noted that the long DFT in Eq. (1) can be implemented so that it does not require the same memory as a full $2L$-sized DFT, because the multiplications by zero do not need to be performed. In MATLAB, for example, this is implemented in the FFT command, by a parameter specifying the FFT length, without actually padding zeros to the signal $x(n)$. In addition, it should be noted that it is good practice to remove the mean of the signal $x(n)$ prior to integration, and also to detrend (i.e. removing the mean and slope of) the output signal $y(n)$ in Eq. (4) to correct for errors that can occur due to leakage in the DFT/IDFT process.

We can thus outline the method of integration by direct DFT (FFT) by the following steps.

1. Remove the mean of the signal to be integrated, $x(n),\quad n = 0, 1, \ldots L-1$.
2. Compute the first $L+1$ values of the FFT of $x(n)$ using zero padding as in Eq. (1).