Using integro-differential operators on low cost underwater autonomous vehicles

Paulo M. Oliveira* and Victor Lobo**

*CINAV - Portuguese Navy Research Center

**CINAV and ISEGI - Instituto Superior de Estatística e Gestão de Informação

Abstract—When operating low cost autonomous vehicles, we are often faced with the need to apply integro-differential operators to numerical sequences. The need may arise in many different contexts, from vehicle navigation to data collection/estimation, and is strongly reinforced by constraints to the number of deployable sensors. Integration and, particularly, differentiation of discrete data sequences are, however, error prone operations. Even in the absence of noise, the traditional approaches introduce distortions and artifacts in the output data, mostly due to the mismatch between their underlying polynomial model and the spectral contents of the collected data. This article presents an alternative way to apply integro-differential operators to discrete data streams. The operators are applied in a strictly band-limited way, in both time and frequency, and no extraneous artifacts are introduced in the data. No assumptions or models are used, other than assuming that the original data stream was correctly sampled. As such, the procedure can be safely applied to data streams sampled at rates close to Nyquist, without the usual performance degradation.

Keywords—numerical integration, numerical differentiation, differentiating filters, integrating filters.

1 Introduction

The need to obtain reliable data streams of non-observed variables by differentiating/integrating observed data sequences is unavoidable. It occurs in the case of virtually all underwater navigation systems, where the required information must be obtained from, e.g., inertial, gyro, pressure, doppler, or acoustic range discrete measurements; it occurs in a multitude of other specific vehicle control systems, where, for example, body rotation rates are required; it also occurs in many of the data collection/estimation tasks these vehicles must perform (e.g. [1], [2], [3]).

Integration and, particularly, differentiation of discrete data streams are, however, operations which may introduce severe distortions and artifacts in the resulting data. Being intrinsically continuous in nature, these operators do not map kindly to the digital world. The traditional, mainstream approaches to numerical integration or differentiation of data sequences are mostly based on equivalent operations performed on interpolating functions, usually (even though not always) polynomial in nature (e.g. [4], [5]). The resulting difference or quadrature formulae are, therefore, typically designed for exactness when applied to polynomials of (up to) a certain order. However, in the engineering field, signals are seldom polynomial. The eigenfunctions of LTI (linear time-invariant) systems are sinusoids, not polynomials; hence, the spectral view of signals is harmonic in nature, and the correctness of our operators when viewed in the frequency domain is of paramount importance for virtually all engineering purposes. This mismatch between the polynomial model supporting the typical differentiating/integrating formulae, and the harmonic model supporting real life signals and systems (and the very concept of sampling itself) is the cause of many of the difficulties of the traditional approaches, when applied to real sampled signals. As illustrative examples, let us consider two of the most ubiquitous traditional formulae for integration and differentiation of discrete sequences: the Central-Difference formula (1) for differentiation, and the Simpson's rule (2) for integration.

$$s^{'}(n) = \frac{s(n+1) - s(n-1)}{2} \tag{1}$$

$$\int_{n-1}^{n+1} s(\tau) d\tau \approx \frac{1}{3}(s_{n-1} + 4s_n + s_{n+1}). \tag{2}$$

In these formulas, $s(t)$ is the signal whose sampling resulted in sequence $s(n)$, and $s'(n)$ is the estimate of its sampled derivative at the same sampling points. When using (1), the error is proportional to the value of the third derivative of $s(t)$ at some point within the period being considered; this formula will, thus, be exact when applied to polynomials of order $n \leq 2$; the error of (2) is proportional to the fourth derivative of the function being integrated, and this formula is, therefore, exact when applied to polynomials of order $n \leq 3$. However, the eigencomponents of real life signals (sinusoids) have derivatives of all orders and, therefore, the errors will not, in general, be zero. Even worst, these derivatives have amplitudes which depend on the frequency of the sinusoid being differentiated/integrated. This means that the behavior of theses formulas will vary along the frequency spectrum and will, therefore, become a strong source of frequency distortion, when applied to signals with more than a single spectral component (all real signals, therefore, with no exception). To better appreciate this distorting behavior, we can look at (1) and (2) as being non-recursive digital filters, whose amplitude frequency responses are easily seen to be: $|H(\theta)| = |sin(\theta)|$, for (1), and $|H(\theta)| = |(cos(\theta) + 2)/(3sin(\theta))|$, for the recursive implementation of (2), θ being the digital angular frequency ($-\pi < \theta \leq \pi$). These amplitude responses can be seen in Figure 1. In this figure, we also represent the amplitude transfer functions of both an ideal differentiator ($|H(\theta)| = |\theta|$) and an ideal integrator ($|H(\theta)| = \frac{1}{|\theta|}$). Only the positive frequency axis is represented.

978-0-933957-40-4 ©2013 MTS

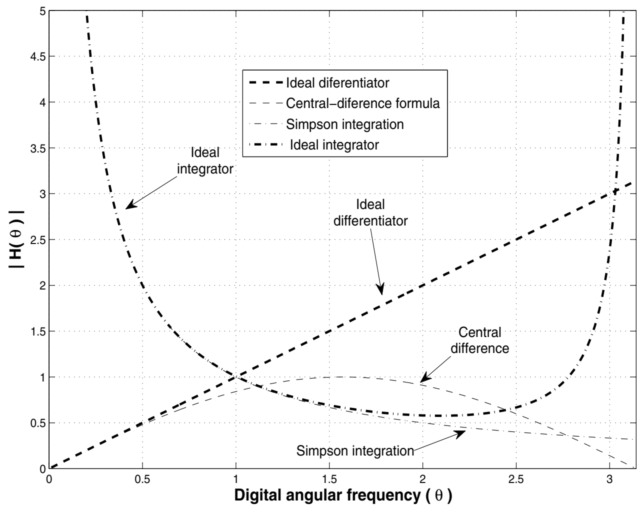

Fig. 1. Frequency response. Ideal vs (1) and (2).

As can be seen from this figure, both (1) and (2) have a frequency response far from the response of the operator they try to emulate. Both formulas will always (except in the trivial case $\theta = 0$) underestimate the derivative/integral, and (what is much worst) the severity of this error is highly dependent on frequency. They will do a fair emulation of the ideal operator's response at low frequencies (up to approximately $\pi/4$ in the case of (1), and $\pi/2$ in the case of (2)), but both fall apart if and when we try to apply them to signals with frequency contents closer to the Nyquist rate ($\theta = \pi$). We can, therefore, expect them to perform well when applied to highly oversampled signals, but they will always be a source of high levels of frequency distortion when applied to broadband signals, due to the varying amplitude response at the different constituent frequencies. This problem of frequency distortion becomes particularly serious and worrisome if we need to process real data which has been sampled at rates close to Nyquist. In many applications, this type of inaccuracy can simply not be afforded.

Up to a point, this difficulty may be controlled by simply increasing the sample rate of the sequence to which the differentiation/integration operator will be applied, well beyond what would strictly be required from non-aliasing considerations and, therefore, guaranteeing that the signal is well within the low digital frequency zone. However, both physical (size, power, storage and computational constraints) and economical considerations often limit the achievable sampling rates, particularly on small, low cost, autonomous devices. Hence, the use of traditional polynomial based approaches will typically introduce non-negligible levels of distortion in the output data stream, contributing to additional errors and lack of precision in an already delicate type of operation.

The errors and artifacts introduced by these traditional discrete implementations of integro-differential operators are not, by far, the only (or even the main) reason for the currently still insufficient performance of low cost underwater navigation systems. However, in the "it all adds up" type of game which ultimately determines the capability of these systems to support prolonged underwater operations, being capable of performing more precise integro-differential operations does contribute to more capable low cost systems.

2 The Spectral Approach

This mismatch between polynomial based formulae and sinusoidal based signals has led to many proposal of alternative approaches to the numerical differentiation/integration of data sequences, based on sinusoidal approximations of functions and sequences (e.g. [5] to [10]). We can classify these approaches on two broad categories:

i) *Direct methods*. Direct construction of formulas designed for exactness when dealing with sinusoids of a (or several) given frequency. The procedure is well described in [5]. By designing for exactness on frequencies spanning the full frequency axis, one might expect to achieve good overall amplitude frequency response characteristics. This approach's main disadvantage is its sensitivity to aliasing.

ii) *Differentiating/integrating filters*. These approaches are based on the design and use of filters with desirable characteristics. Both finite-impulse response (FIR) and infinite-impulse response (IIR) filters have been proposed, and their design methodologies vary widely. The main limitation of this approach lies on the fact that imposing compliance with the chosen design criteria will leave the filter's response uncontrolled in some other respects (e.g. a small number of coefficients in a FIR filter will affect sharpness; an increase in sharpness affects aliasing sensitivity; the option for IIR has an impact on phase response; etc);

As will be seen, the approach to be discussed in this paper possesses elements of both these two approaches. It has elements of a direct method, insofar as it starts by specifying the desired frequency response at a set of frequencies; it has elements of the filtering approach, opting for FIR filtering and its superior phase response characteristics; it also has elements not belonging to any of these categories.

The proposed approach is based on one of the most useful properties of the Fourier Transform: the fact that differentiation/integration in one domain maps to multiplication/division in the other. This property has been extensively used in the solution of differential equations, but its use as a differentiation/integration numerical technique has not been deeply explored. Direct application of Fourier transforms for this purpose does present some difficulties and discouraging results, which may well constitute the reason why this seemingly simple technique has not received proper attention. With proper care, these difficulties can be surpassed, and the Fourier Transform can constitute the basis for a simple, fast, reliable, stable and more accurate way of obtaining numerical derivatives or integrals of real life signals [9].

2.1 DFT differentiation

Using a Fourier Transform to differentiate a signal is a conceptually simple procedure: i) Fourier Transform the signal; multiply the resulting spectrum by $j2\pi f$; iii) transform it back into the time domain. This procedure is justified by the identity

$$\frac{d\,s(t)}{dt} = \int (j2\pi f) S(f) e^{j\,2\pi f t}\, df, \tag{3}$$

where $S(f)$ is the Fourier Transform of $s(t)$. Its discrete version for sequences of N points is:

$$s'(n) = j\frac{2\pi}{N}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} kS(k)e^{j\,2\pi\frac{k}{N}n}, \qquad (4)$$

$S(k)$ being the Discrete Fourier Transform (DFT) of $s(n)$. In the DFT domain, differentiation can be, thus, simply replaced by the multiplication of the sequence's spectrum by a ramp $d(k)$, defined as[1]:

$$d(k) = \begin{cases} j2\pi k/N, & -\frac{N}{2}+1 \leq k < \frac{N}{2} \\ 0 & k = \frac{N}{2} \end{cases}, \qquad (5)$$

After inverse DFT transform (IDFT), this procedure is precisely equivalent to the use of a N-point differentiating formula obtained by the direct method, if it imposes exactness on (N/2) equally spaced digital frequencies (including DC). This was to be expected, since the differentiation operation is being performed exactly, at sample points (through multiplication by $d(k)$).

This DFT based approach is, in theory, exact for any properly sampled signal, and should provide better results than the traditional polynomial formulas - such as (1) - when dealing with non-polynomial signals. The caveat lies in the fact that there will always be aliasing present in any finite length sample sequence. In fact, direct application of (4) produces very discouraging results. As an example, consider the simple sequence

$$s(n) = \cos(5\pi n/128), \quad \text{for} \quad 0 \leq \text{n} \leq 127 \qquad (6)$$

Differentiating this sequence with (4), we obtain the sequence plotted in the top plot of Figure 2, which is a very bad, unusable, approximation to the expected inverted sine.

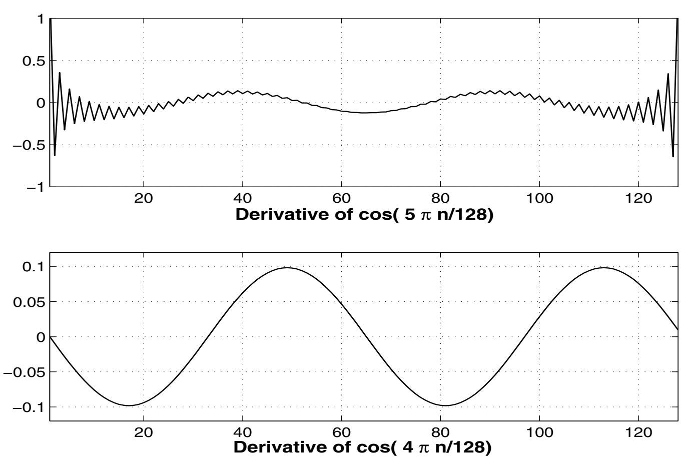

Fig. 2. Derivative of a (co)sinusoidal segment with a non-integer (top) and integer (bottom) number of cycles.

The traditional solution to avoid this type of behavior would be to smooth out the requirements on the differentiating filter, by amputating its amplitude response, and tapering the transfer function $d(k)$, in an attempt to obtain a differentiating filter with a better time response (e.g. [6], [8]). The specific type of tapering varies considerably, but the taper functions are typically designed to preserve good frequency response characteristics at the low end of the spectrum, even if at the expense of the response at higher digital frequencies. In a sense, these approaches are moving away from our objective: the ability to differentiate wideband sequences, and/or sequences sampled at rates close to Nyquist.

We will thus follow a different line of approach. Instead of focusing on how to "improve" the differentiating filter, we will focus on the data to be differentiated, and determine what type of pre-processing can be done on the data to alleviate the problem. The objective is to eliminate the ripple without having to compromise in the length or shape of the differentiating filter and, hence, on its closeness to a full band differentiator.

In the digital DFT world, both time and frequency sequences are implicitly periodic. As is easily seen, the periodic replication of (6) will be discontinuous, due to the different values at end points. Its derivative will, thus, also be discontinuous at end points. This implies that its spectral coefficients will converge only as $1/k$ (e.g. [5]). This slow rate of convergence will, in turn, imply the existence of a considerable amount of aliasing in its DFT representation. The severe ringing in time should, thus, have been expected.

Let us now consider an equivalent sequence, but this time containing two full periods of the cosine: $s(n) = \cos(4\pi n/128), \quad \text{for} \quad 0 \leq \text{n} \leq 127$. Having an integer number of cycles means that the periodic replication of the sequence will be continuous, the same being true also for its derivative. Hence, we should expect a much faster convergence of the Fourier coefficients and, therefore, less aliasing in the derivative. The result of differentiating this sequence with (4) can be seen in the bottom plot of Figure 2. In fact, in this particular case, the periodic replication of the sequence has continuous derivatives of all orders, which accounts for the perfect results obtained with the DFT differentiation procedure. The previous examples suggest that the key to obtain good quality results from the differentiation with (4) is to guarantee that the sequence to be differentiated is, at least, continuous with continuous first derivative, by adequately pre-processing the data before the differentiating procedure.

2.1.1 Discontinuities at end-points

Arithmetic elimination.

One simple way of eliminating the discontinuities of the sequence at end points is to subtract, to the observed data sequence, a straight line segment $l(n)$, with the same end-point values [5]. That is:

$$l(n) = s(0) + \frac{s(N-1) - s(0)}{(N-1)}n, \qquad 0 \leq n \leq N-1 \quad (7)$$

We can now differentiate the resulting sequence $\hat{s}(n) = s(n) - l(n)$, with improved results, due to the absence of end-point discontinuities. After differentiating, we will need to add the slope of the subtracted segment $((s(N-1) - s(0)/(N-1))$ to the result, to obtain the derivative of $s(n)$. However, arithmetic elimination will only deal with the discontinuities of the function itself. Nothing is being done to prevent discontinuities of $s'(t)$, or higher order derivatives. We should thus expect some improvement by using arithmetic elimination, but it

1. Without loss of generality, we will always consider N to be even. Hence, for spectral hermitian symmetry, we impose $d(N/2) = 0$

would be unreal to expect the problem to have been fully addressed.

Data tapering

The problem of end-point discontinuities in the periodic replication of the sequence is not substantially different from the one so often encountered in spectral estimation, where the problem is dealt with by tapering the data sequence with an appropriate window. This suggests that using a multiplicative window prior to the differentiation procedure might help eliminate the end-point discontinuities. As discussed, to avoid aliasing, we should guarantee, at least, both the continuity of $s(t)w(t)$, and of its derivative[2]:

$$\frac{d}{dt}\left[s(t)w(t)\right] = s'(t)w(t) + s(t)w'(t). \quad (8)$$

We thus conclude that the chosen window must obey the following constraints [9]:

i) To obtain a continuous function, $w(t)$ should be zero valued at end-points;

ii) To avoid discontinuities of the first derivative at end-points due to the first term in (8) ($s'(t)w(t)$), we should use a continuous window, zero valued at end-points.

iii) To avoid discontinuities of the first derivative at end-points due to the second term of (8) ($s(t)w'(t)$), the chosen window should have a continuous first derivative - zero valued at end-points - implying a side-lobe fall rate of, at least, 60 dB/decade.

The problem is, of course, that multiplying the sequence by an appropriate window will severely distort the derivative we are aiming to obtain. For any window other than the rectangular, obtaining the correct derivative implies, therefore, a recovery procedure:

$$s'(n) = \frac{IDFT(d(k)\cdot S(k)) - w'(n)s(n)}{w(n)}. \quad (9)$$

If the used window tapers down to zero at end-points, the derivative at the first sample should not be computed, since (9) would just go to infinity[3].

These two constraints on the choice of a window (continuous and with continuous first derivative) should eliminate most of the aliasing in the output of the differentiating procedure. Using windows with continuous derivatives of higher orders (e.g. Blackman window [11]) will not, in general, lead to considerable improvements, since the faster convergence advantage may be offset by the numerical errors resulting from the use of (9) on their heavily tapered extremities. A Hann window [11] is, therefore, an example of an appropriate choice.

To exemplify the effect of the two above techniques (arithmetic elimination and window tapering), let us again consider differentiating the sequence $s(n) = \cos(5\pi n/128)$ with (4), but this time using the discussed pre-processing techniques. The results can be seen in Figure 3, for the cases of i) arithmetic elimination only; ii) arithmetic elimination *and* tapering with a Hann window; iii) arithmetic elimination *and* tapering with a Bohman window. By comparing the top plot of this figure with the top plot of Figure 2, we can see how tremendously effective the discussed pre-processing techniques are. The bottom plot of this figure is a detailed view of the the first 6 points of the derivatives. The true value of the derivative is also represented, for *ground truth* comparison. As can be seen, the pre-processing is highly effective, and achieves very high rates of convergence to the true value of the derivative. In just four samples, the estimates using tapering converged to the true value. Note that, even though it is not visible in the figure, the convergence rate of the Bohman window was indeed higher than the one provided by the Hann window; however, it produced higher errors than the Hann window for $n \leq 2$, as expected from the above discussion concerning the choice of a window.

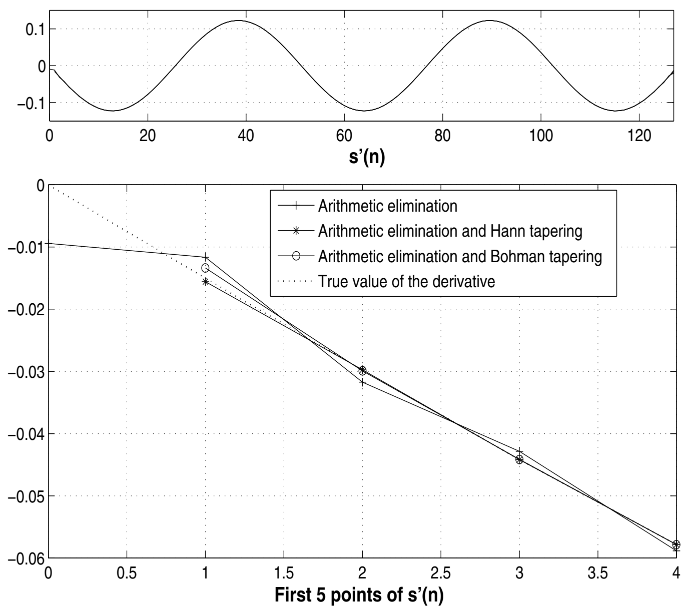

Fig. 3. Derivatives of $cos(5\pi n/128)$.

2.1.2 Discontinuities at inner points

If a signal is correctly sampled, we should never have to face the existence of discontinuities within the sequence. But there is always the possibility of very sharp behavior of the derivative at inner points. The resulting aliasing may still imply strong oscillations. If we can determine the time of occurrence of these inner point discontinuities (or near discontinuities), it would, in principle, be possible to eliminate them, by splitting the sequence to be differentiated at that point, and treating the discontinuities as end-point discontinuities. In general, however, we may not know the points were these discontinuities lie, and we must find a different solution to the problem.

A more practical approach is to leave the inner discontinuities as they are, but improve the rate of convergence of the approximating Fourier series. This can be done by multiplying the Fourier coefficients by Lanczos factors [5]. This procedure can, however, be recognized as just one instance of the use

2. Continuous time notation is being used here, for clarity.

3. No such precaution is needed at the last point, since the discrete versions of windows which taper to zero have a single zero sample, at the first (or last) sample; the corresponding zero on the other extremity is already a part of the next periodic replica. See [11], for additional details on the appropriate definition of discrete windows.

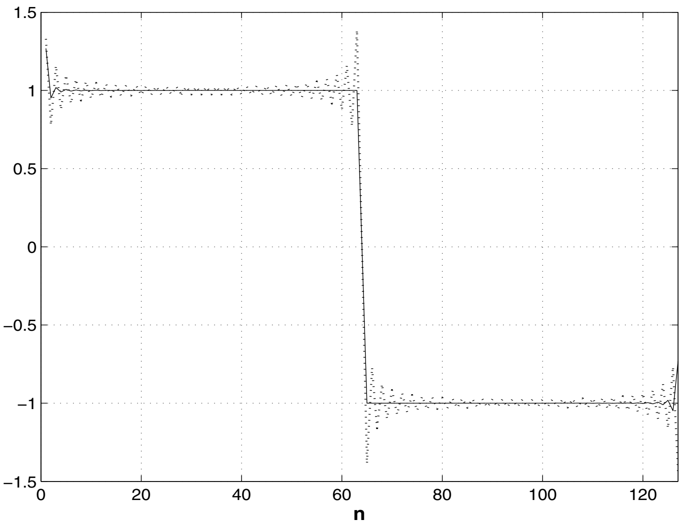

Fig. 4. Effectiveness of frequency windowing.

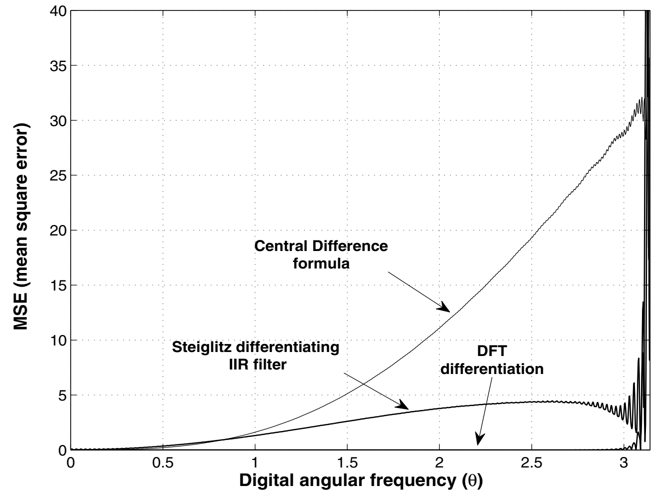

Fig. 5. Single sinusoidal component. MSE of the derivative estimation, as a function of frequency.

(in the frequency domain) of the Riemann window [11]. This immediately suggests that other windows, with more desirable features (mainly in the side-lobe decay rate), might be useful. Using a frequency window $W(k)$ implies an additional step in the algorithm:

$$s'(n) = IDFT\{DFT\{\hat{s}'(n)\} \cdot W_f(k)\}$$

In this equation, $\hat{s}'(n)$ on the right hand side designates the output of (9); $s'(n)$ is the final estimate of the derivative (to which the slope of $l(n)$ may still need to be added, if arithmetic elimination was used).

The choice of the frequency window to be used is subject to the very same type of constraints one considers in choosing a time window for spectral estimation. It is always, however, a distorting, low-pass procedure. It should only be used when required, since the resulting low pass effect will be defeating our original purpose of being capable of adequate differentiation up to the Nyquist rate.

As an illustrative example, let us consider a sequence whose derivative is discontinuous at midpoint:

$$s(n) = \begin{cases} n\,, & 0 \le n \le 64 \\ 128 - n\,, & 65 \le n \le 127 \end{cases}\,, \qquad (10)$$

The dashed line of Figure 4 represents the derivative of (10) obtained with arithmetic elimination and a Hann time window. The solid line of the same figure represents the same derivative, but obtained with the additional use of a Lanczos window in frequency. As can be seen, the middle-segment discontinuity of the derivative does originate strong oscillations (amplified at end-points by the recovery procedure (9)), but frequency windowing effectively reduces them.

2.1.3 The DFT differentiation algorithm

The full differentiation procedure can thus be summarized as follows:

$$\begin{aligned}
1.\quad \hat{s}(n) &= s(n) - [s(0) + \frac{s(N-1) - s(0)}{(N-1)} \cdot n] \\
2.\quad \hat{S}(k) &= DFT\{\hat{s}(n) \cdot w(n)\} \cdot d(k) \\
3.\quad \hat{s}'(n) &= \frac{IDFT\{\hat{S}(k)\} - w'(n) \cdot s(n)}{w(n)} \\
3a.\quad \hat{s}'(n) &= IDFT\{DFT\{\hat{s}'(n)\} \cdot W_f(k)\} \\
4.\quad s'(n) &= \hat{s}'(n) + \frac{s(N-1) - s(0)}{N-1}
\end{aligned}$$

Step 3a is optional, as previously discussed, and should only be used it the original sequence contains discontinuities (or first derivative discontinuities), which cause aliasing (and, therefore, ringing in the derivative estimate). This should not happen for properly sampled signals, but no finite length signal will ever be properly sampled.

Even though all the synthetic examples presented so far have used highly oversampled signals, for visual purposes, we note that the obtained results are valid for the (near) totality of the frequency axis, up to (almost) the Nyquist rate. To appreciate that, let us look at the mean square error of the result of differentiating a pure sinusoidal component, $s(n) = cos(\theta n + \pi/3.44)$ as a function of its frequency (Figure 5). For comparison, we also show the estimation error, as a function of frequency, of both (1), and one of the known good representatives of the class of IIR (Infinite Impulse Response) differentiating filters[4]. As can be seen, the DFT approach has lower MSE error at virtually all frequencies, except at the very top of the band, where all approaches, in fact, break down.

4. In this computation, we discarded the first and last $0.05N$ points of the resulting derivatives, due to the relatively high settling time of the Steiglitz filter; this also eliminated the areas of lower accuracy of the DFT procedure.

2.2 DFT integration

Using a DFT to perform integration is simpler than using it for differentiation. The underlying principle is exactly the same. i) Fourier Transform the signal; *divide* the resulting spectrum by $j2\pi f$; iii) transform it back into the time domain. This procedure is justified by the identity

$$\int s(\tau)d\tau = \int \frac{1}{(j2\pi f)} S(f) e^{j\,2\pi f t}\, df \qquad (11)$$

where $S(f)$ is, again, the Fourier Transform of $s(t)$. Hence, and up to a constant, we can compute the primitive of a sequence by simply dividing its spectrum $S(k)$ by $j2\pi k/N$. This division cannot be performed at $k = 0$, but $k = 0$ is a special point, amenable to a trivial exception handling procedure: since $S(0)$ is the mean value of the sequence being integrated ($\bar{s}$), its contribution to the integral is simply the ramp $n\bar{s}$). Hence, the difficulty can be circumvented by simply not considering the point $k = 0$ in the spectral integration, and adding, a posteriori, the term $n\bar{s}$ to the result of the integration procedure. The same problem will be faced at $k = N/2$ (if N is even), since, for spectral symmetry, the frequency dividing factor must again be zero at $\theta = \pi$. We can safely discard the bin at $\theta = \pi$, however, since it will have a null net contribution to the value of the integral. The final, modified ramp $p(k)$, to be used in the integration procedure thus becomes:

$$p(k) = \begin{cases} j2\pi k/N, & -\frac{N}{2}+1 \le k < \frac{N}{2},\ \ k \neq 0 \\ 1 & k = \{0, \frac{N}{2}\} \end{cases}, \qquad (12)$$

If $s(n)$ is a continuous sequence, its primitive will also be continuous, and we will not face the problem of severe aliasing that, in the differentiation case, forced us to use time windows. The full differentiation procedure can thus be summarized as follows (designating the integrated sequence by $\lambda(n)$, and its spectrum by $\Lambda(k)$):

1. $S(k) = DFT(s(n))$
2. $S(0) = 0$ (exception handling for mean)
3. $S(\frac{N}{2}) = 0$ (iif N is even)
4. $\Lambda(k) = S(k)\frac{1}{p(k)}$
5. $\lambda(n) = IDFT(\Lambda(k)) + n\bar{s}$

To appreciate the type of result provided by this algorithm, consider Figures 6 and 7. In Figure 6, we can see the result of integrating sequence (6), a sequence whose periodic replication has discontinuities (and which, thus, created initial difficulties to the differentiation procedure). The true value of the primitive function is also represented on this plot, even though it is visually coincident with the integration results. As can be seen, the integration procedure did produce a very good estimate of the true primitive sequence, even for an original sequence whose periodic replica is discontinuous. We can now look at the mean square error of the integration of a single sinusoidal component, $s(n) = cos(\theta n + \pi/3.44)$, as a function of its frequency, to appreciate the wideband behavior of the method. This is represented in Figure 7. For comparison, we also represented the mean square error of the Simpson and Trapezoidal quadrature rules, when applied to the same signal.

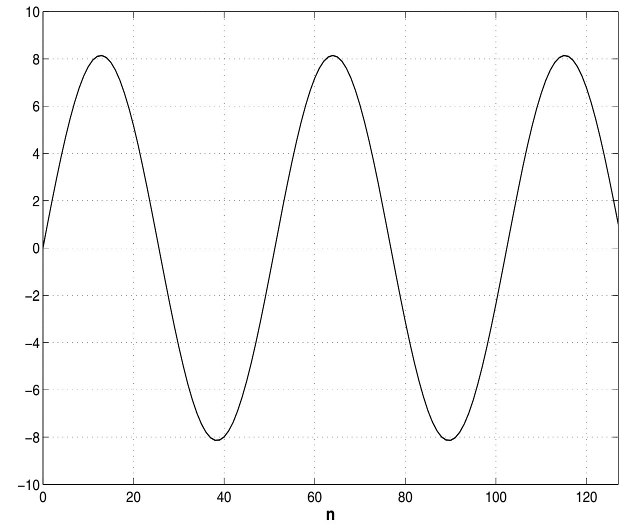

Fig. 6. Primitive of $s(n) = cos(5 * pi * n/N)$, $0 \le n \le 127$

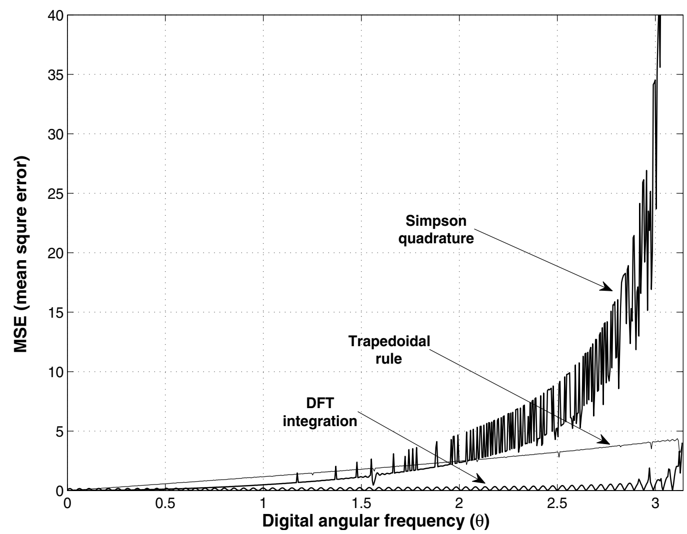

Fig. 7. Single sinusoidal component. MSE error of primitive estimation, as a function of frequency.

The good behavior of the DFT estimate almost up to the Nyquist rate is apparent in this figure. It is also interesting to see that the Simpson method starts to perform worst than the trapezoidal at approximately $2/3$ of Nyquist. This is not unexpected, single the less local behavior of Simpson (encompassing two sampling periods at a time) is prone to suffer more from an insufficient sampling rate than the more localized (one sampling period at a time) trapezoidal rule. A final note must be made, concerning the fundamentally different natures of quadrature rules such as Simpson or the Trapezoidal, and the DFT integration method. In the later, the primitive function is obtained at once for all the integration interval. This contrasts with the behavior of quadrature rules, which must be run recursively to produce the primitive sequence, since, at each iteration, they only provide estimates for the total area in the interval being encompassed by the rule.

2.2.1 Behavior in noise

Even though the analysis of the behavior of the DFT differentiation/integration algorithms in the presence of noise will not be presented in this paper, some comments should nevertheless be made. Any filter or procedure that approaches a perfect differentiator must necessarily possess a very high noise amplification factor (as an ideal differentiator will). When dealing with oversampled signals, the low-pass effect intrinsic to traditional differentiation formulae such as (1) and (2) (see Figure 1), confers them some white noise suppression capability and, therefore, better SNR ratios at output than the ideal operators (and therefore, better than a DFT differentiator), without excessively affecting the signal. However, even in cases where this attenuation of high-frequency noise is advantageous, having the low-pass filtering effect implicitly and irrevocably mixed up with the differentiation/integration operation is not an advantage. It is better to have the numerical differentiator perform as close as possible to the ideal desired operator and, if needed, introduce an additional (or prior) step of noise reduction. Having the ability to freely choose the type of noise reduction to be used (be it statistical, or mere frequency based filtering) much more than compensates the hassle of having to perform a separate step in the overall processing algorithm. Differentiation and noise filtering are different operations, and it seems preferable to keep them that way.

We note that using a frequency window in the DFT differentiation procedure (Step 3a in the procedure) will also perform noise reduction by low pass filtering. When noise is the dominant problem, a Tukey window [11] is a good choice, since, if properly parameterized, it allows us to (almost) completely avoid distortion when dealing with oversampled signals, while still effective at the intended noise reduction.

2.2.2 Real data example

In this section, we will use the data obtained from an Xsens MTi-28 A53 G35 AHRS (Attitude and Heading Reference System). The objective is to show the improved capability of the DFT differentiation algorithm to deal with wideband real data. To do it, we will use the Roll and Roll-rate informations. Even though the Roll-rate can not not be considered *ground truth* for the derivative of the Roll information (the Roll data is at the output of an EKF Extended Kalman Filter which also takes into account the accelerometers information to correct for drift), it is useful for the purpose. The procedure will be as follows: we will differentiate the Roll information, and evaluate its "error" when compared with the Roll-rate information, for both a DFT differentiator (arithmetic elimination and time Hann window) and the Central Difference formula (1). The Roll information will then be decimated, and the "error" determined again, successively, until the moment when further decimation is not possible without losing spectral information. As decimation proceeds, the signal's spectral contents will be spreading throughout the frequency axis and, hence, we can appreciate the performance of both differentiators as the signal becomes more wideband, and with frequency contents approaching the Nyquist rate. The results can be seen in Figure 8. In this figure, each subplot on the left shows the power spectrum of one of the decimated Roll instances (to convey the notion of the amount of oversampling still existing in the data), and the corresponding Decimation Factor DF ($DF = 1$ means no decimation); on the right, we can see the evolution of the error of each differentiator, as the Decimation Factor increases and, thus, the signal becomes increasingly wideband. The DFT differentiation procedures's consistency as the signal becomes increasingly wideband is apparent, and contrasts with the behavior of 1.

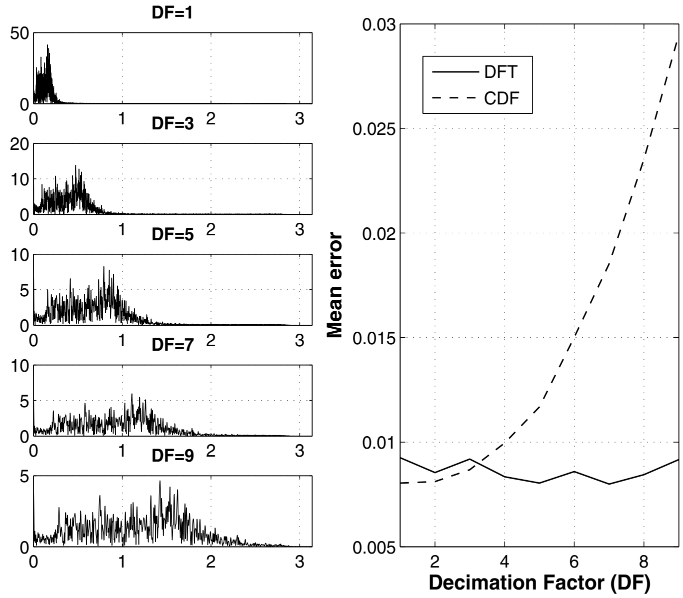

Fig. 8. Mean error of differentiation, as the occupied frequency band increases

2.3 Conclusions

The traditional, polynomial approaches to differentiate/integrate data sequences are not capable of adequately coping with wideband signals, or signals sampled at rates close to Nyquist. However, it is possible to devise a spectral based approach, capable of properly dealing with such signals. The method is based one of the most useful properties of the Fourier Transform: the fact that differentiation/integration in one domain maps to multiplication/division in the other. Direct application of the method poses some difficulties and produces unacceptable results, but it is possible to pre-process the data in order to circumvent theses difficulties, and hence, produce good estimates without compromising the quality of the differentiating/integrating filter. Three techniques have been presented: arithmetic elimination, tapering the data with appropriately chosen time windows, tapering the spectrum with a frequency windows, or any combination of these. The criteria for choosing adequate windows was also discussed. As a whole, the method proved to be effective, in that not only does it provide low estimation errors, but is capable of doing so consistently, throughout the frequency axis, almost up to the Nyquist rate.

References

[1] Savage, P. G., Strapdown Inertial Navigation System Integration Algorithm Design Part 2 - Velocity and Position Algorithms, AIAA Journal

Of Guidance, Control, And Dynamics, Vol. 21, No. 2, March-April 1998, pp. 208-221.

[2] Szklany, Steven A., A Feasibility Assessment of a New Navigation System for Unmanned Underwater Vehicles with Adaptive Gain Sliding Mode Differentiation, MSc Thesis, Rochester University, New York, 2012.

[3] Jouffroy, J. and Reger, J., An algebraic perspective to single-transponder underwater navigation, Proceedings of the 2006 IEEE International Conference on Control Applications Munich, Germany, October 4-6, 2006.

[4] Hildebrand, F. B., Introduction to Numerical Analysis, Second Edition, Dover Publications, Inc., New York, 1987.

[5] Hamming, Richard. W., Numerical methods for Scientists and Engineers, Second Edition, Dover Publications, Inc., New York, 1986.

[6] Kaiser, J. F. "Digital Filters". In *System Analysis by Digital Computer*, pp. 218 - 285. J. Wiley,New York- London- Sydney, 1966.

[7] Lanczos, C.: "Applied Analysis". Prentice-Hall, New Jersey, 1956

[8] Hamming, Richard. W., "Digital Filters, Second Edition", Prentice-Hall, New Jersey, 1983.

[9] Oliveira, Paulo M, and Barroso, Victor "Numerical Differentiation of Discrete-time Sequences", 4th Conference on Telecommunications, Aveiro, June 2003.

[10] Steiglitz, K. Computer-Aided Design of Recursive Digital Filters, IEEE Trans. on Audio Electroacoustics, Vol.AU-18, pp. 123129, June 1970.

[11] Harris, Frederic J., On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proc. IEEE, Vol. 66, No 1, January 1978.