Property 1: Let \( a \in \mathbb{P}^n \) and separate \( a \) into its even and odd parts, \( a_e \) and \( a_o \),

\[
a(s) = a_e + a_o = a_e(s^2) + s a_o(s^2)
\]

(35)

where the notation \( a_e(s^2) \) and \( a_o(s^2) \) is used to enhance the fact that \( a_e \) and \( a_o \) contain only even powers. Then, \( a \) is Hurwitz if and only if there exists \( \lambda_1, \lambda_2, \ldots, \lambda_n \) and \( c \) in \( \mathbb{R} \) satisfying

\[
a_e(-\omega^2) = (\lambda_1 - \omega^2)(\lambda_2 - \omega^2) \cdots (\lambda_n - \omega^2)
\]

(36)

\[
\tilde{a}_o(-\omega^2) = c(\xi_1 + \omega^2)(\xi_2 + \omega^2) \cdots (\xi_n + \omega^2 - 1)
\]

(37)

where \( c > 0 \) and \( 0 < \lambda_1 < \xi_1 < \lambda_2 < \xi_2 < \cdots < \lambda_n \).

Lemma 1: Consider a Hurwitz polynomial \( q \) and let \( q_e \) and \( q_o \) denote its even and odd parts. Then there exists an even function \( u \) and an odd function \( v \) satisfying

\[
q_e(s)u(s) + q_o(s)v(s) = 1.
\]

(38)

Proof: Equation (38) is a Bezout identity and its satisfaction is equivalent to the statement that \( q_e(s) \) and \( q_o(s) \) are coprime. To show this, we reason by contradiction. Suppose \( q_e \) and \( q_o \) are not coprime. In this case we must have,

\[
q_e(s) = a(s)f(s)
\]

(39)

\[
q_o(s) = b(s)f(s)
\]

(40)

for some nontrivial polynomial \( f(s) \). In this case, one of the following must be true.

\[
f \text{ even} \Rightarrow a \text{ even and } b \text{ odd}
\]

(41)

\[
f \text{ odd} \Rightarrow a \text{ odd and } b \text{ even}
\]

(42)

We assume without loss of generality that (41) holds and \( q \) is even. In this case, \( q_e \) and \( q_o \) can be rewritten as follows,

\[
q_e(s^2) = a(s^2)f(s^2) = (a_1 + a_2s^2 + a_4s^4 + \cdots)
\]

\[
\times (f_0 + f_2s^2 + f_4s^4 + \cdots)
\]

\[
q_o(s^2) = b(s^2)f(s^2) = (b_1 + b_3s^2 + b_5s^4 + \cdots)
\]

\[
\times (f_0 + f_2s^2 + f_4s^4 + \cdots) = s\tilde{q}(s^2).
\]

Thus, if \( \{\zeta_1, \zeta_2, \cdots, \zeta_m\} \) are the roots of \( f(-\omega^2) \), we have

\[
q_e(-\omega^2) = a(-\omega^2)(\zeta_1 - \omega^2)(\zeta_2 - \omega^2) \cdots (\zeta_m - \omega^2)
\]

\[
q_o(-\omega^2) = s\tilde{q}(s^2)(\zeta_1 - \omega^2)(\zeta_2 - \omega^2) \cdots (\zeta_m - \omega^2)
\]

It follows that \( q \) is not Hurwitz, since the \( m \) roots of \( q_e(-\omega^2) \) and \( q_o(-\omega^2) \) contained in \( f(-\omega^2) \) do not satisfy Property 1. This contradicts the assumptions. To complete the proof of Lemma 1 there remains to show that \( u \) and \( v \) are respectively even and odd. This is a straightforward consequence of the Euclidean algorithm (see, for example, [14]), by which \( u \) and \( v \) can be determined, and the even property of \( mu + nv = 1 \).

REFERENCES


technique, it automatically adjusts for possible errors. We show that actually any error appearing in the DFT is attenuated as it propagates over time, and is completely eliminated after a number of iterations equal to the length of the DFT.

II. THE LMS SPECTRUM ANALYZER

The LMS spectrum analyzer is represented in Fig. 1. The signal to be Fourier transformed is used as the desired output \( d_k \) of a linear adaptive filter. The input to the filter at time \( k \) is the complex phasor

\[
X_k = \frac{1}{N} \left[ e^{j2\pi(k-1)/N} \cdots e^{j2\pi(N-1)/N} \right]^T.
\]

where \( T \) denotes the transpose. The series of phasors \( X_0, X_1, \cdots X_{N-1}, X_N, \cdots \) satisfies two important properties: first the series is periodic with period \( N \), second \( \{X_0, X_1, \cdots X_{N-1}\} \) form an orthonormal basis in the \( N \)-dimensional space.

The filter weight vector \( W_k \) is adapted with the complex LMS algorithm [3]

\[
W_{k+1} = W_k + 2\mu e_k X_k.
\]

where \( \mu \) is the learning rate, \( e_k \) is the complex conjugate of \( e_k \), and \( e_k \) is the error signal defined as \( e_k = d_k - X_k^H W_k \). Widrow et al. showed [2] that by setting the learning rate to 1/2, by iterating over \( k \) from the initial conditions \( W_0 = 0 \), and by using the above-mentioned properties of the input phasors, one finds for the weight vector

\[
W_k = \sum_{m=-N}^{k-1} d_m X_m.
\]

With the same notation, the DFT of the data samples \( d_{k-N}, \cdots, d_{k-1} \), can be expressed as

\[
DFT_k = \sum_{m=-N}^{k-1} d_m X_m - k.
\]

The phasor \( X_{m-k} \) is related to \( X_m \) by the formula

\[
X_{m-k} = P^k X_m,
\]

where the diagonal matrix \( P \) is defined as \( P = \text{diag}\{e^{j2\pi(N-1)/N}, \cdots, e^{j2\pi(N-1)/N}\} \). Comparing (3) and (4), one finds

\[
DFT_k = \sum_{m=-N}^{k-1} d_m X_m = P^k W_k.
\]

At each instant \( k \), the weight vector of the LMS filter is proportional to the DFT of the past \( N \) data samples.

It should be noted that the behavior of the LMS algorithm in the LMS spectrum analyzer is somewhat "special" in the sense that the LMS filter does not converge asymptotically and with misadjustment noise to its optimal solution as it usually does. Rather, it provides at each iteration the exact desired solution.

III. NONADAPTIVE RECURSIVE IMPLEMENTATIONS OF THE DFT

By iteratively updating \( W_k \), the LMS spectrum analyzer evaluates the sliding-DFT recursively. A more straightforward but nonadaptive recursive implementation of the sliding-DFT [4] can be obtained by comparing the DFT at times \( k \) and \( k+1 \) and by observing that

\[
DFT_{k+1} = P^{k+1} \sum_{m=k+1-N}^{k} d_m X_m
\]

which is identical to (6) since the DFT and the weight vector at time \( k \) differ only by a multiplicative factor \( P^k \). Although the nonadaptive sliding-DFT and the LMS spectrum analyzer perform very similar operations, their behaviors in limited precision arithmetic differ drastically.

IV. PROPAGATION OF ROUNDOFF ERRORS IN THE SLIDING-DFT

In software and hardware implementations, limited precision causes roundoff errors that propagate from iteration to iteration. The LMS spectrum analyzer, on the contrary, has a "built-in" error cancellation mechanism. Consider a situation where the weight vector of the LMS filter is free from errors up to time \( k-1 \). At time \( k \), a noise vector \( e_k \) is deliberately introduced in \( W_k \). Let \( \bar{W}_k = W_k - \bar{e}_k \) be the perturbed weight vector. The LMS error signal at time \( k \) is given by

\[
\bar{e}_k = d_k - X_k^H \bar{W}_k = d_k - X_k^H W_k + X_k^H \bar{e}_k.
\]

Assuming that the learning rate \( \mu \) is equal to 1/2, the weight vector at time \( k+1 \) is given by

\[
\bar{W}_{k+1} = \bar{W}_k + \bar{e}_k X_k
\]

where \( I \) is the \( N \times N \) identity matrix. Similarly, the weight vector can be evaluated at times \( k+2, k+3, \ldots \), and in general, for any time \( k + j \), one has

\[
\bar{W}_{k+j} = \bar{W}_{k+1} - (I - X_k^H X_k^H) \bar{e}_k.
\]

It can be verified that the order in which the matrix multiplies are effected is irrelevant. This justifies the otherwise ambiguous notation \( \prod_{m=0}^{k+j} \).
The multiplication of the error by the matrix sin(2πf/k) and its 32-point DFT. We wrote a C program implementing the nonadaptive sliding-DFT and the LMS spectrum analyzer algorithms. To demonstrate the effect of limited precision, we rounded off to 7 bits the mantissas of the limited precision, we rounded off to 7 bits the mantissas of the floating point results of all arithmetical operations. For comparison, we also coded the exact DFT of x(k). Fig. 2 represents the sum of

\[ \sum_{i=0}^{N-1} |DFT_i|^2 \]

d as a function of time, for \( f_1 = 0.03 \) and \( f_2 = 1.1 \). The DFT given by the LMS spectrum analyzer practically coincides with the exact DFT. The nonrecursive sliding-DFT follows the exact DFT for a while but it eventually diverges.

V. CONCLUSION

While the nonadaptive sliding-DFT allows roundoff errors to accumulate over time, the LMS spectrum analyzer uses its adaptation loop to automatically eliminate errors in a number of iterations equal to the length of the DFT. It does not require significantly more operations per iteration than the nonadaptive sliding-DFT. These results naturally extend to other transforms once implemented adaptively (see [5] for a generalization of the LMS spectrum analyzer to other orthonormal transforms). For these reasons, we recommend the use of the LMS spectrum analyzer for any application where a sliding orthonormal transform must be performed over long trains of data.

REFERENCES


A Class of Second-Order Integrators

and Low-Pass Differentiators

Mohamad Adnan Al-Alaoui

Abstract—A novel class of stable, minimum phase, second-order, lowpass IIR digital differentiators is developed. It is obtained by inverting the transfer functions of a class of second-order integrators, stabilizing the resulting transfer functions, and compensating their magnitudes. The class of second-order integrators is obtained by interpolating the traditional Simpson and trapezoidal integrators. The resulting integrators have a perfect –90° phase over the Nyquist interval and could better approximate the ideal magnitude response than either of the two traditional integrators. The low order and high accuracy of the filters make them attractive for real-time applications.

I. INTRODUCTION

The basic concept came from observing that the ideal integrator response lies between the responses of the traditional trapezoidal and...