XVI IMEKO World Congress
Measurement - Supports Science - Improves Technology - Protects Environment ... and Provides Employment - Now and in the Future
Vienna, AUSTRIA, 2000, September 25-28

MODELLING OF PREDICTIVE SIGMA-DELTA CONVERSION

F. Sandu and W. Szabo

Electronics and Computers Department
Electrical Engineering Faculty, Transilvania University
Bulevardul Eroilor, nr. 29 – Brasov – 2200 – Romania

Abstract: The idea of acquiring not the proper samples at the input of an ADC but the difference ("Delta" and its sign "Sigma") between the new sample and the previous one (the derivative of the signal, in finite-difference sense) led to an important decrease of samples' dynamic, provided the over-sampling frequency goes towards the slew-rate of the conversion circuits divided by the last significant bit. Considering the previous sample as the simplest (1-st order) estimate of the new sample, the authors modeled a more efficient way (in terms of speed and memory requirements): the acquisition of the difference between the new sample and a better estimate of it (higher order estimate, adapted to signal's statistic).

The Levinson-Durbin algorithm is justified and implemented into an Itakura lattice. Computational similarities with the Schur-Cohn tests of stability are underlined. The authors implemented this computation scheme in a LabView "virtual ΣΔ ADC".

Modelling by Mason graphs of the Itakura finite-impulse-response (FIR) of the predictive ΣΔ-ADC is followed by its symmetric flow-graph for the infinite-impulse-response (IIR) DAC.

Keywords: linear prediction, sigma-delta converters, flow-graphs

1 INTRODUCTION

The idea of "differential" acquisition or modulation of signals is not very recent but was brought to actuality by the progress in fast switching (that allows important over-sampling). From the frequency domain point of view, the derivative of the signal (in finite-difference sense) although expressed with less bits/sample, keeps the overall bandwidth (even emphasizing high frequencies), so there isn't a real spectral compression to be expected from a ΣΔ ADC. Neither is the bit-rate decrease, as signals are to be reconstructed in the simplest way, by integration (genuine counting, as the numeric integrator is an accumulator) from the bit stream. 1 bit arithmetic (including the previous attempts to attach information to the binary rhythm) is still a challenge to innovation in ΣΔ domain.

2 PREDICTIVE Σ-Δ ADC

The linear prediction of x_n as a linear combination of previous M samples, $x_{(n-m)}$) is $x_n = \sum_{m=1}^{M} a_m \cdot x_{(n-m)} + v_n$, computed with a_m coefficients matched to the signal so as the difference v_n should be only a white noise, totally non-correlated with the signal x_n [3].

Computing the first M+1 inter-correlations of the previous predictive equation with { $x_n + v_n$ } gives, noting with r(k) the auto-correlation $r_{xx}(k)$, the linear system :

$$r(k) = \sum_{m=1}^{M} a_m \cdot r(k-m) + \sigma^2 \cdot \delta(k) \quad , \ k = 0 , \dots , M . \tag{1}$$

We must mention that white noise has a constant spectral power density, σ^2 , representing, according to Wiener-Hincin theorem, the spectrum of its auto-correlation, having, accordingly, the temporal $r_{vv}(k) = \sigma^2 \delta(k)$. This isolated term in the first equation of system (1) comes from the non-correlation between the signal and the noise.

Writing in a matrix form the last M equations of system (1), it results,

$$R_M \cdot \begin{bmatrix} a_1 \\ \dots \\ a_M \end{bmatrix} = \begin{bmatrix} r(1) \\ \dots \\ r(M) \end{bmatrix} \Leftrightarrow \begin{bmatrix} a_1 \\ \dots \\ a_M \end{bmatrix} = R_M^{-1} \cdot \begin{bmatrix} r(1) \\ \dots \\ r(M) \end{bmatrix} \qquad (2)$$

So, based on the auto-correlation values, a "template" (statistically representative for the whole class of $\{ x_n \}$, not only for one particular sequence), can be obtained as the optimal coefficients of the M^{th} order estimation. The computations based on the inversion of R_M belong to the Yule-Walker algorithms; there are different methods for the efficient inversion of R_M, beginning with the gaussian algorithm with sequential reductions, or considering its Töplitz structure (each parallel to the main diagonal has identical elements) and its symmetry relative to the same diagonal, due to parity of the auto-correlation. There are also algorithms to recursively compute the set $\{ a_m \}$ of the $(M+1)^{th}$ order estimation based on the set $\{ a_m \}$ of the previous order, M. The most famous is the Levinson-Durbin algorithm, that is justified here in an operative and compact way by an arrangement resulting immediately from (2), passing to the next order of estimation :

$$\underbrace{\left[\begin{array}{ccc:c} & & & r(M) \\ & R_M & & r(M-1) \\ & & & \dots \\ & & & r(1) \\ \hdashline r(M) & \dots & r(1) & r(0) \end{array}\right]}_{R_{M+1}} \cdot \left[\begin{array}{c} \overbrace{a_{1,[M]} - a_{M+1,[M+1]} \cdot a_{M,[M]}}^{a_{1,[M+1]}\ ,\dots \text{etc.}} \\ a_{2,[M]} - a_{M+1,[M+1]} \cdot a_{M-1,[M]} \\ \dots \\ a_{M,[M]} - a_{M+1,[M+1]} \bullet a_{1,[M]} \\ \hdashline a_{M+1,[M+1]} \end{array}\right] = \left[\begin{array}{c} R_M \cdot \begin{bmatrix} a_{1,[M]} \\ a_{2,[M]} \\ \dots \\ a_{M,[M]} \end{bmatrix} \\ \hdashline r(M+1) \end{array}\right] \qquad (3)$$

Indeed, for each line k (= 1,..,M) of R_{M+1}, last column's element, r(M+1-k), *multiplied by* $a_{M+1,[M+1]}$, *compensates* { line k of R_M, (identical with line (M+1-k) in reversed order, as R_M is Töplitz) }, *multiplied by* { column of correction terms $a_{M+1,[M+1]} \cdot a_{M+1-k,[M]}$ (that are subtracted from $a_{k,[M]}$ to obtain $a_{k,[M+1]}$) }, multiplication that, according to M^{th} order equation system, the previous recurrence step), delivers just r(M+1-k) ($\cdot\, a_{M+1,[M+1]}$).

The coefficients of the (M+1) estimation are those of the column vector factor, $a_{k,[M+1]} = a_{k,[M]} - a_{M+1,[M+1]} \cdot a_{M-k,[M]}$ for k = 1, … , M , with the "pivot" coefficient $a_{M+1,[M+1]}$ that results from the product of the last line of R_{M+1} with the column of $a_{k,[M+1]}$,

$$r(M+1) = \sum_{m=1}^{M} a_{m,[M]} \cdot r(M+1-m) - (r(0) - \sigma_{[M]}^2 - r(0)) \cdot a_{M+1,[M+1]} \qquad (4)$$

$$\Rightarrow a_{M+1,[M+1]} = \frac{1}{\sigma_{[M]}^2} \cdot [r(M+1) - \sum_{m=1}^{M} a_{m,[M]} \cdot r(M+1-m)] \qquad (5)$$

The last computation of iteration [M+1] gives

$$\sigma_{[M+1]}^2 = \sigma_{[M]}^2 \cdot (1 - a_{M+1,[M+1]}^2) \quad (= r(0) - \sum_{m=1}^{M+1} a_{m,[M+1]} \cdot r(m) =$$

$$= \underbrace{r(0) - \sum_{m=1}^{M} a_{m,[M]} \cdot r(m)}_{\sigma_{[M]}^2} - a_{M+1,[M+1]} \cdot \underbrace{[- \sum_{m=1}^{M} a_{m,[M]} \cdot r(M+1-m) + r(M+1)]}_{\sigma_{[M]}^2 \cdot a_{M+1,[M+1]}}) \qquad (6)$$

Another important obvious property is that the regression coefficients $\{ b_m \}$ are obtained out of a linear system based on the same values of auto-correlation, then, with the same formula (2), $\{ a_m \} \equiv \{ b_m \}$ for any M.

The DELTA value, that would be acquired by an adaptive ΣΔ-ADC will be computed as the "prediction" error $v_{(P),\, n,\, [M]}$, the difference between x_n and its approximant value,

$\overline{x}_{n,[M]} = \sum_{m=1}^{M} a_{m,[M]} \cdot x_{(n-m)}$. The recursive computations can be oriented on the difference to the superior prediction order (using the link between $a_{m,[M+1]}$ and $a_{m,[M]}$) :

$$v_{(P),\,n,\,[M+1]} - v_{(P),\,n,\,[M]} = x_n - \sum_{m=1}^{M+1} a_{m,[M+1]} \cdot x_{(n-m)} - [\, x_n - \sum_{m=1}^{M} a_{m,[M]} \cdot x_{(n-m)}] =$$

$$= a_{M+1,[M+1]} \cdot \underbrace{[\sum_{m=1}^{M} a_{M-m+1,[M]} \cdot x_{(n-m)} - x_{(n-M-1)}\,]}_{\text{regression error } v_{(R),(n-M-1),[M]}} \tag{7}$$

In the same way it can be put into evidence the difference to the superior regression order :

$$v_{(R),(n-(M+1)),[M+1]} - v_{(R),(n-1-M),[M]} = a_{M+1,[M+1]} \cdot \underbrace{(\, x_n - \sum_{m=1}^{M} a_m \cdot x_{n-m}\,)}_{\text{prediction error } v_{(P),n,[M]}} \tag{8}$$

The computations evolve in a lattice FIR, implemented in an Itakura structure [4] that can be successfully applied to predictive ΣΔ ADC (fig.1). A similar recurrence can be applied for Schur-Cohn tests of stability (it is necessary to obtain all $|\, a_{m,[M]} \,| < 1$).

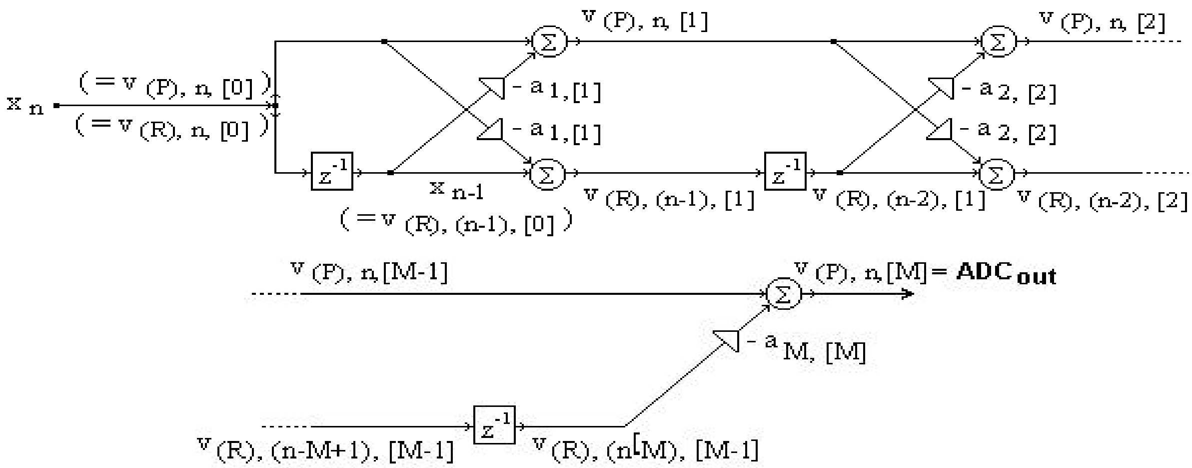

Figure 1. Predictive ΣΔ ADC ; the coefficients are computed on a previous acquisition window and (stored with the ΣΔ sequence), or pre-existent in a template based on signal's statistic.

3 THE VIRTUAL SD ADC

The authors implemented the Levinson-Durbin computation scheme in a National Instruments-LabView ™ virtual instrument ("VI") having the panel of fig. 2 (as an example, it was loaded, from the file test.prn, the pattern generated with the simple MathSoft - MathCAD ™ procedure of fig.3)

To avoid edge errors, cyclic autocorrelation (with a period of 1024) was implemented in the first sequence (0) of the VI diagram (fig. 4). Autocorrelation coefficients are passed to the Levinson-Durbin sequence (1), presented in fig.5. Most of array processing blocks are of "Index Array" type (returning the element of array at the specified index) and of "Array Subset" type (returns a portion of array starting at a specified index and containing a specified number of elements). Recursive computation is implemented by "Sfhit Registers" that bring the results of the previous iteration (M) of the "For Loop" to the next one (M+1). Initialization is done via "Select" switches. The first iteration step, M =1, is considered (together with $\sigma_{[\,0]}{}^2 = r(0)$, for M=0) as a base for higher order iterations:

$$a_{1,[1]} = \frac{1}{\sigma_{[0]}{}^2} \cdot r(1) \left(= \frac{r(1)}{r(0)} \right) ; \quad \sigma_{[1]}{}^2 = \sigma_{[0]}{}^2 \cdot (1 - a_{1,[1]}{}^2) \left(= r(0) - \frac{r(1)^2}{r(0)} \right) \tag{9}$$

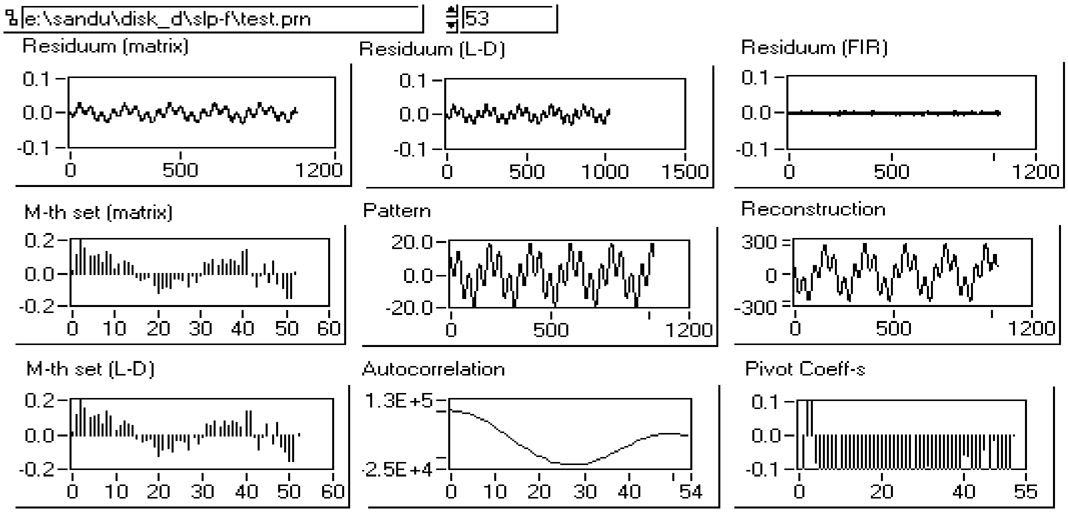

Figure 2. Panel of the ΣΔ ADC – VI. The test pattern was brought repetitively (by Continuos Run) from a file. The signal could be easily acquired by a Data Acquisition Board (the authors used National Instruments AT-MIO16E10 ; if the input is periodical, it is used an exterior sampling clock synchronized by PLL in a frequency multiplication - by 1024 - configuration)

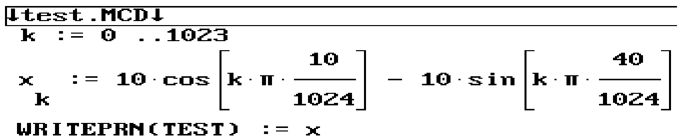

Figure 3. Pattern generation with a MathSoft - MathCAD procedure (oversampling about 25 times).

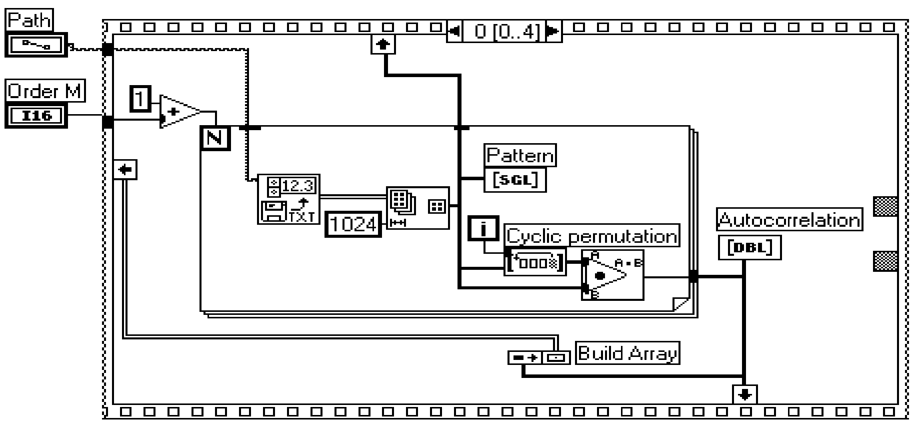

Figure 4. Sequence (0) of the VI. Cyclic autocorrelation coefficients are passed to sequence (1).

The versatile VI controls enabled the authors to study the optimum compression of samples' dynamic, that occurs for an order M with a minimum a $_{M, [M]}$ (local or global – for M = 1, 2, 3, …) out of the pivot-coefficients. Indeed, if the set {$a_{k,[M]}$} doesn't, practically, need improvement ($\Leftarrow a_{k,[M+1]} = = a_{k,[M]} - a_{M+1,[M+1]} \cdot a_{M-k,[M]} \approx a_{k,[M]}$), it is because $\sum_{m=1}^{M} a_m \cdot x_{(n-m)} \approx x_n$ so the residuum, $v_n = x_n - \sum_{m=1}^{M} a_m \cdot x_{(n-m)}$ is small. After such an optimal M is determined (pre-processing of a first frame of 1024 samples of the input signal), it will be kept in the acquisition of the following frames.

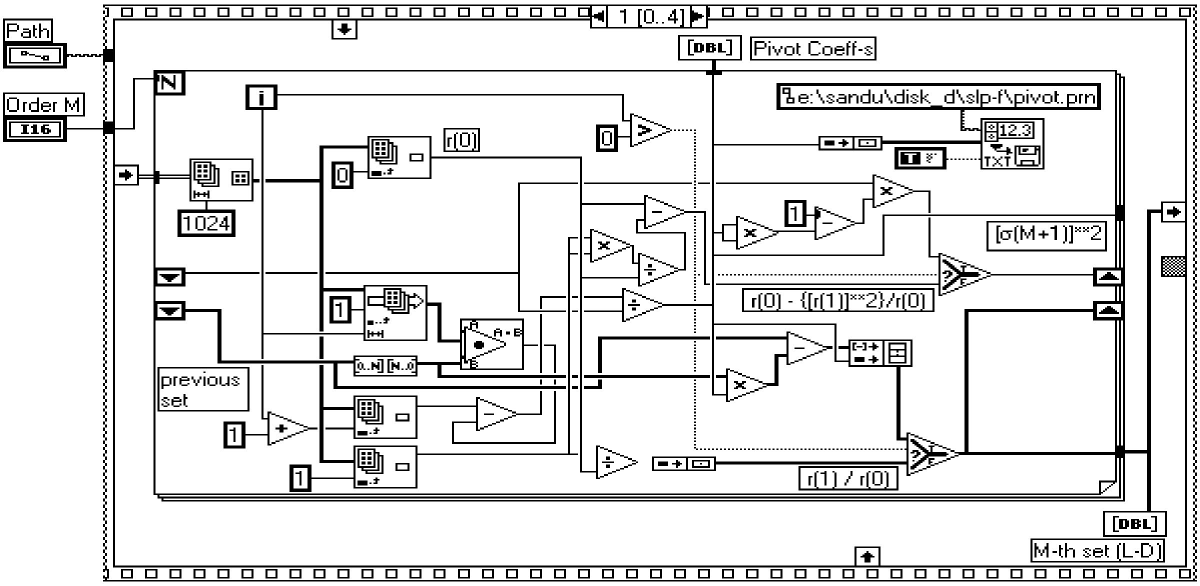

Figure 5. Sequence (1) of the VI (the Levinson-Durbin, "L-D", iteration) outputs the M-th set $a_{k,[M]}$.

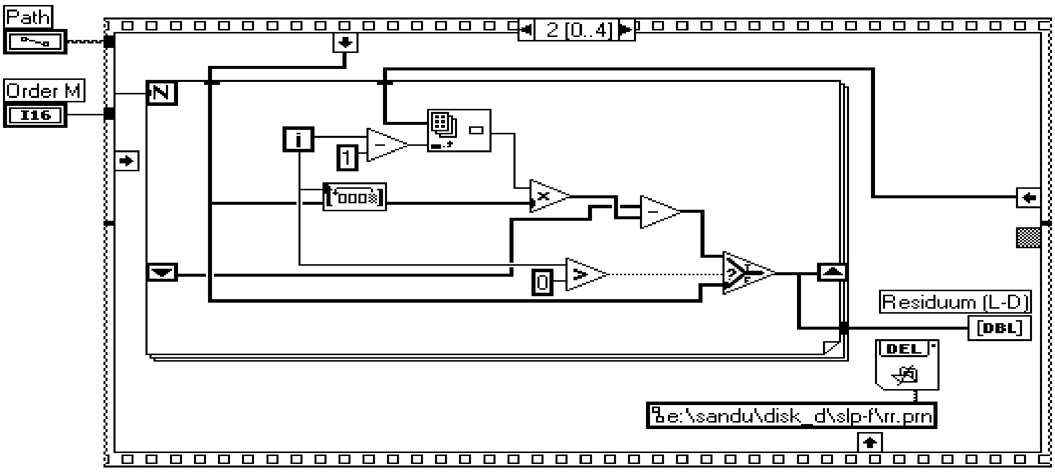

Figure 6. Sequence (2). Residuum $v_n = x_n - \sum_{m=1}^{M} a_m \cdot x_{(n-m)}$ is computed reverting samples' order.

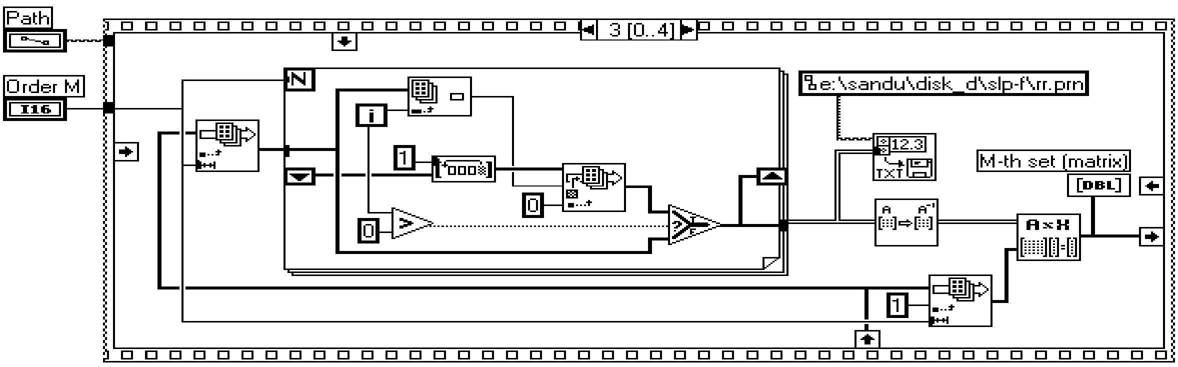

Figure 7. Sequence (3) – classic matrix computation, checking the (identical) results of sequence (1).

The last sequence of this complete VI is presented in fig. 8. *Reconstruction can be done by the M coefficients* $a_{k,[M]}$ implemented into a IIR filter. Reconstruction is accurate if the IIR input is the (exact) residuum (acquired as well, with a reduced dynamic, *going towards a bit stream*) and with a good approximation *if the only recorded are the M coefficients* $a_{k,[M]}$. Input of the IIR reconstruction filter (or of the ΣΔ DAC presented below) *can be any "1 Bit White Noise"* [implemented here, with a good approximation, as signum(effective white noise)], provided that IIR filter is set to initialize its internal states to the final ones from its previous call (ideal for continuous filtering of input frames).

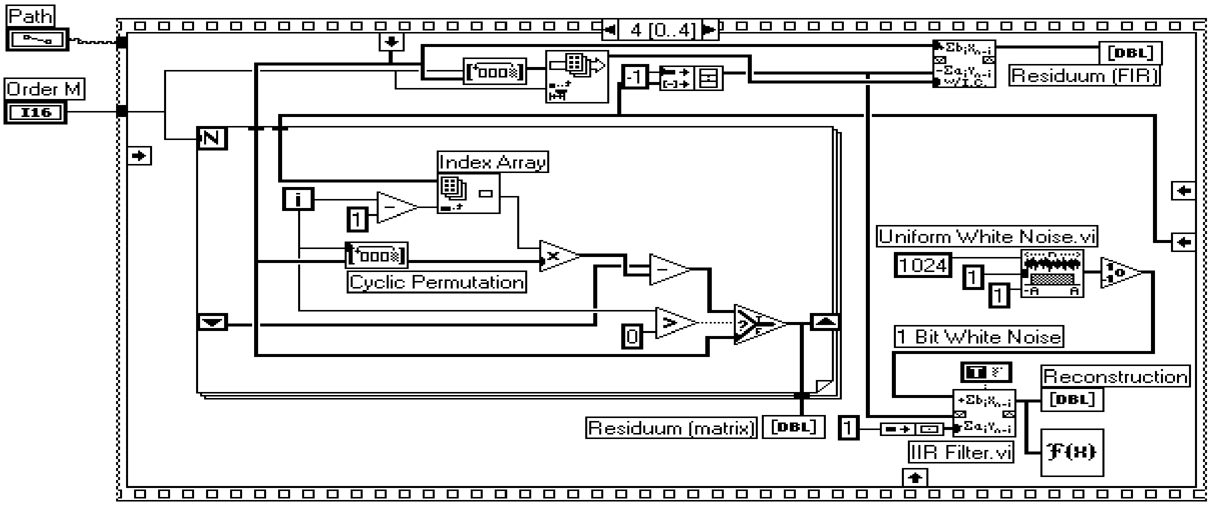

Figure 8. VI's final sequence. Computing the exact matrix residuum and the approximate one (FIR filter with coeff-s $a_{k,[M]}$). Reconstruction of (harmonic-redundant) signals only by $a_{k,[M]}$ set.

4 THE SD DAC

Fig. 9 presents a symmetrical structure that implements the *dual* ΣΔ DAC :

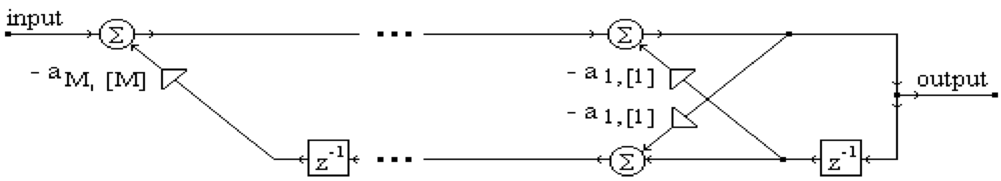

Figure 9. The dual DAC associated to the ADC in fig.1. It was obtained a recursive Itakura lattice.

Modeling with *Mason graphs* provides, for these *dual* ΣΔ - ADC and DAC circuits, a deeper perspective. Both dual graphs look practically alike, if transmittance 1 is considered for all segments different from z^{-1} or $a_{k,[k]}$. Global transmission factors are $H_{ADC}(z) = 1 / H_{DAC}(z)$. Indeed, the "fraction" of H_{ADC} has denominator 1 (ADC has a non-recursive lattice structure, without any loops in the graph). The numerator of H_{DAC} is 1 as well (the graph has only one direct path input → output, by simple unitary (superior) transmittances, "s.u.s.t.". Starting from the evidence that DAC's graph is identical with ADC's one, in which only the sense of "s.u.s.t." is inverted, it is obvious that:

- the loops in DAC's graph have the aspect of trapeze (with the greater base of "s.u.s.t."-s).
- in ADC's graph, excluding "s.u.s.t."-s out of the direct input → output paths, the separate, disjoint, groups that are left (to constitute the proper factors of the transmittances), can be found also in the graph of the DAC, where the reverted "s.u.s.t."-s close loops that constitute groups of simple, double, triple, ... loops, each two disjoint, then *numerator of* H_{ADC} *equals denominator of* H_{DAC} !

REFERENCES

[1] E. Ifeachor, B. Jervis, *Digital Signal Processing*, Addison-Wesley, 1993
[2] J. Proakis, C. Rader, F. Ling, C. Nikias, *Advanced Digital Signal Processing* Macmillan, 1992
[3] S. Mitra, J. Kaiser, *Handbook for Digital Signal Processing*, John Wiley, 1993
[4] F. Op 't Eynde, W. Sansen, *Analog Interfaces for DSP Systems*, Kluver, 1993

AUTHORS: Assoc. Univ. Prof. Dr. F. SANDU, Univ. Prof. Dr. W. SZABO, "Transilvania" University, Bulevardul Eroilor, nr. 29 – Brasov – 2200 – Romania, Phone ++40 68 474718, Fax ++40 68 475751, E-mail: sandu@vega.unitbv.ro and szabo@vega.unitbv.ro