Synthesis of minimal cost nonlinear feedback shift registers

D. Linardatos and N. Kalouptsidis *

Abstract

This paper presents a block type algorithm which determines the "minimal" nonlinear feedback shift register (NLFSR) that generates a given sequence. Minimality is defined with respect to a total ordering between structural vectors that takes into account the implementation cost.

1 Introduction

In this paper a block type algorithm is developed which determines the minimal NLFSR that generates a given finite sequence with elements in a field F. The proposed algorithm combines a total ordering, a linear dependence test and a minimality test. The total ordering facilitates the evaluation of the NLFSR. It is necessitated by the fact that each NLFSR is characterized by a structural multiindex representing the order of nonlinearities, the main signal products (called primary signals) and the maximum delays. The coefficients of a NLFSR producing the given signal live in the null space of a suitable matrix. Hence, each NLFSR amounts to expressing a column of the above matrix as a linear combination of preceding columns. A special feature of this matrix is that the lengths of its columns are not the same. This complication requires a more careful analysis of linear dependence. The linear dependence test achieves the above goal via manipulation of column subblocks and a proper adaptation of the FIA algorithm [2]. The third ingredient of the algorithm matches the linear dependence test with the total ordering. In this manner it eventually determines the NLFSR that is minimal with respect to the above ordering.

*University of Athens, Department of Informatics and Telecommunications, Panepistimiopolis, Ilissia, 157 84 Athens, Greece, e-mail: linardat@di.uoa.gr, kalou@di.uoa.gr

2 Regular representation

Suppose that the given sequence $x(n), 1 \leq n \leq N$ is generated by the following nonlinear source:

$$\begin{aligned} x(n) = & \sum_{i=1}^{L} c_i x(n-i) + \\ & + \sum_{i_1=0}^{p_{22}} \sum_{i=1}^{L_{i_1}} c_{i,i_1} x(n-i)x(n-i-i_1) + \cdots + \\ & + \sum_{i_1=0}^{p_{k2}} \cdots \sum_{i_{k-1}=0}^{p_{kk}-\sum_{l=1}^{k-2} i_l} \sum_{i=1}^{L_{i_1,\cdots,i_{k-1}}} c_{i,i_1,\cdots,i_{k-1}} x(n-i) \cdot \\ & \cdot \prod_{j=1}^{k-1} x(n-i-\sum_{w=1}^{j} i_w), \quad n > M \end{aligned} \tag{1}$$

In the so-called regular representation (1) which defines a recursively computable expression, k denotes the maximum order (degree) of nonlinearity [4]. The linear part is characterized by its order L and the corresponding coefficients $c_i, 1 \leq i \leq L$. The second order part is parameterized by the orders p_{22} and L_{i_1} and the quadratic coefficients $c_{i,i_1}, 1 \leq i \leq L_{i_1}, 0 \leq i_1 \leq p_{22}$. The higher order terms are similarly defined with $p_{m2} \leq p_{m3} \leq \cdots \leq p_{mm}, 3 \leq m \leq k$. The parameters $L, L_{i_1}, \cdots, p_{22}, p_{32}, \cdots$ are nonnegative integers. The coefficients $c_i, c_{i,i_1}, \cdots, c_{i,i_1,\cdots,i_{k-1}}$ and the resulting signal values reside in a field F and additions and multiplications are the field operations. Finally, parameter M is related to the length of initial conditions of the NLFSR.

The structural parameters of (1) can be incorporated into the vector

$$u = (k, r(k), S(k, r(k))) \tag{2}$$

where $r(k) = (p_{22} \mid p_{32} \;\; p_{33} \mid \cdots \mid p_{k2} \;\; \cdots)$ and $S(k, r(k)) = (L \mid L_0 \;\; \cdots \;\; L_{p_{22}} \mid L_{0,0} \cdots)$.

Eq. (1) reveals that the output signal $x(n)$ is obtained by a linear combination of shifted copies of a set of so-called primary signals [3]. The number of primary signals associated with a specific nonlinearity of order m is determined by the parameters

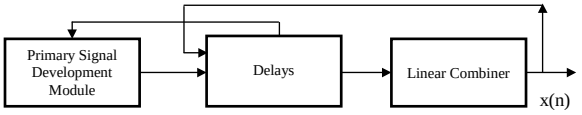

Figure 1: The architecture of NLFSR implementation

$\{p_{ml}\}, l = 2, \cdots, m$, while the shifts involved in a specific primary signal determined by the indices $i_1, \cdots, i_{m-1}$ is indicated by $L_{i_1,\cdots,i_{m-1}}$. By convention $L_{i_1,\cdots,i_{m-1}} = 0$ declares the absence of the corresponding primary signal.

The regular representation can be directly and canonically mapped onto an architecture where three main structural modules are involved, as indicated in Fig. 1. The primary signal development module generates the primary signals. It takes as inputs the delays of signal $x(n)$ and requires the use of multipliers for the generation of the primary signals. The delay module includes a delay structure for each primary signal that spans all pertinent signal products involved in eq. (1). The third module includes the scalors associated with the coefficients $c_i, c_{i,i_1}, \cdots$ and the appropriate adders that are combined to generate $x(n)$.

Eq. (1) is nonlinear with respect to $x(n)$ but is linear with respect to the unknown coefficients $\{c_{i,i_1,\cdots,i_{m-1}}\}$. Collecting all signal samples $x(n)$, $M < n \leq N$ and setting $c_0 = -1$, we write eq. (1) in matrix form as

$$T_1(u) \cdot C = 0 \tag{3}$$

where the $(N-M) \times Q$ matrix $T_1(u)$ and the $Q \times 1$ column vector C are defined as:

$$T_1(u) = (X_{M-L}^{N-M} \;\; X_{M-L+1}^{N-M} \;\; \cdots \;\; X_{M}^{N-M} \;\; X_{M-L_{0,0}}^{N-M} \;\; \cdots)$$

$$C = (c_L \;\; c_{L-1} \;\; \cdots \;\; c_0 \;\; c_{L_{0,0}} \;\; \cdots)^T$$

and

$$X_P^R = (x(P+1) \;\; x(P+2) \;\; \cdots \;\; x(P+R))^T, \tag{4}$$

$$X_{P,i_1,\cdots,i_{m-1}}^R = \begin{pmatrix} x(P+1) \cdot \prod_{w=1}^{m-1} x(P+1-\sum_{j=1}^{w} i_j) \\ x(P+2) \cdot \prod_{w=1}^{m-1} x(P+2-\sum_{j=1}^{w} i_j) \\ \vdots \\ x(P+R) \cdot \prod_{w=1}^{m-1} x(P+R-\sum_{j=1}^{w} i_j) \end{pmatrix}$$

where $2 \leq m \leq k$ and $P \geq 0$ determines the first entry. Clearly u controls the size of $T_1(u)$. Due to the shift invariance of eq. (4) we have for any $0 \leq K \leq R$, $X_P^R = (X_P^{K\,T} \;\; x(P+K+1) \;\; \cdots \;\; x(P+R))^T$, where X_P^K provides the upper subblock of X_P^R of length K. Vector $X_{P,i_1,\cdots,i_{m-1}}^R$ is partitioned in a similar fashion.

It is useful to harmonize the column lengths by arbitrarily extending $T_1(u)$ to

$$T(u) = (X_{M-L}^N \;\; X_{M-L+1}^N \;\; \cdots \;\; X_M^N \;\; X_{M-L_{0,0}}^N \;\; \cdots)$$

where the signal values $x(n)$ that fall outside the given range $1 \leq n \leq N$ are arbitrarily assigned and are referred to as "don't care" entries.

3 Total ordering of regular forms

Let $V(u)$ be a real function $V(u) : \mathcal{A} \to \mathcal{R}$ that represents the implementation cost associated with the architecture corresponding to u, where $\mathcal{A}$ includes all u that satisfy problem requirements. $V(u)$ is termed the ordering function. Let

$$\mathcal{B}(u) = (1 \;\; \sum_{i_1} g_s(L_{i_1}) \;\; \cdots \;\; \sum_{i_1} \cdots \sum_{i_{k-1}} g_s(L_{i_1,\cdots,i_{k-1}}))^T$$

$$\mathcal{Q}(u) = (L+1 \;\; \sum_{i_1} L_{i_1} \;\; \cdots \;\; \sum_{i_1} \cdots \sum_{i_{k-1}} L_{i_1,\cdots,i_{k-1}})^T$$

where $g_s(n)$ equals zero for $n \leq 0$ and 1 otherwise. If $g_s(L_{i_1,\cdots,i_{m-1}}) = 0$ the corresponding primary signal is absent. The i-th component of $\mathcal{B}(u)$ provides the number of primary signals associated with the order i. Similarly the entries of $\mathcal{Q}(u)$ describe the delays of the primary signals.

An ordering denoted by $<_V$ is defined as follows: Let $u_1 = (k_1, r_1(k_1), S_1(k_1, r_1(k_1)))$, $u_2 = (k_2, r_2(k_2), S_2(k_2, r_2(k_2))) \in \mathcal{A}$. Then

$u_1 <_V u_2$ if and only if one of the following holds: (5)

1. $V(u_1) < V(u_2)$
2. $V(u_1) = V(u_2)$ and $k_1 < k_2$
3. $V(u_1) = V(u_2)$, $k_1 = k_2$ and $\mathcal{B}(u_1) <_T \mathcal{B}(u_2)$
4. $V(u_1) = V(u_2)$, $k_1 = k_2$, $\mathcal{B}(u_1) = \mathcal{B}(u_2)$ and $r_1(k_1) <_T r_2(k_2)$
5. $V(u_1) = V(u_2)$, $k_1 = k_2$, $\mathcal{B}(u_1) = \mathcal{B}(u_2)$, $r_1(k_1) = r_2(k_2)$ and $\mathcal{Q}(u_1) <_T \mathcal{Q}(u_2)$
6. $V(u_1) = V(u_2)$, $k_1 = k_2$, $\mathcal{B}(u_1) = \mathcal{B}(u_2)$, $r_1(k_1) = r_2(k_2)$, $\mathcal{Q}(u_1) = \mathcal{Q}(u_2)$ and $S(k_1, r_1(k_1)) <_T S(k_2, r_2(k_2))$

where the total degree ordering between two $m \times 1$ vectors p and q is defined by :

$$\begin{pmatrix} p_1 & \cdots & p_i & \cdots & p_m \end{pmatrix} <_T \begin{pmatrix} q_1 & \cdots & q_i & \cdots & q_m \end{pmatrix}$$

$$\text{iff either } \sum_{1\le i\le m} p_i < \sum_{1\le i\le m} q_i \text{ or } \left(\sum_{1\le i\le m} p_i = \sum_{1\le i\le m} q_i \right)$$

$$\wedge(\exists i)\left((1 \le i \le m) \wedge (p_i < q_i, p_{i+1} = q_{i+1}, \cdots, p_m = q_m)\right)$$

The ordering $<_V$ is a total ordering and is mainly based on the cost function $V(u)$. For equal cost implementations the ordering $<_V$ is based on the degree of nonlinearity. If equality sustains the ordering $<_V$ relies on the number of primary signals or the number of delays. $V(u)$ can be assessed by summing the cost of scalors, adders, multipliers and delay units and has the property that adding extra feedback taps or inserting extra primary signals to the register increases its cost.

The problem under consideration is formulated as follows: Given a sequence $\{x(n)\}$ of length N in the field F, a set of constraints and a cost function V, we search for $u = (k, r(k), S(k, r(k))) \in \mathcal{A}$ and $C \neq 0$ such that $T_1(u) \cdot C = 0$, C is recursively computable and for all $u^{'} \in \mathcal{A}$ with $u^{'} <_V u$ there exists no recursively computable $C^{'} \neq 0$ such that $T_1(u^{'}) \cdot C^{'} = 0$.

A first approach in dealing with the problem is exhaustive search. Indeed, by the use of eq. (5) let us order the structural vectors as $u_1 <_V u_2 <_V \cdots <_V u_i <_V u_{i+1} <_V \cdots$. Then exhaustive search successively constructs $T_1(u_i)$ and checks for recursively computable $C \neq 0$ such that $T_1(u_i) \cdot C = 0$. This scheme involves an enormous number of calculations since the procedure for linear dependence is repeated for each $T_1(u_i)$.

An alternative is the insertion of all $T(u_j), u_j <_V u_i$ in a matrix $T_{max}(u_i)$ whose linear dependence eventually reflects the linear dependence of the minimal $T_1(u_i)$. However, the resulting matrix contains a lot of "don't care" elements and the issue of linear dependence needs proper modification.

4 The proposed algorithm

The organization of the proposed algorithmic scheme is illustrated in Fig. 2. It consists of the preprocessing module and the main module. The preprocessing module builds the total ordering of the structural vectors contained in $\mathcal{A}$. This module does not depend on the given sequence and provides the framework for the main module. It can be implemented in an off-line fashion.

The main module includes three submodules. The first submodule builds column by column a matrix T_{max} that traces the ordering of structural vectors. The second submodule identifies a linear combiner (if any) of subcolumns of T_{max} that forms a candidate for a minimal solution. The third submodule checks if the linear combiner corresponds to the minimal solution.

More specifically, the first submodule follows the total ordering and updates the matrix $T_{max}(u_i)$ by adding the column $X^N_{I_i}$ (if any) of $T(u_i)$ not included in $T_{max}(u_{i-1})$. By definition of total ordering and the architecture based on primary signals, it is obvious that every matrix $T(u_i), i > 1$ contains at most one column not contained in any $T(u_w), w < i$. This column may correspond either to a new primary signal not contained in any $T(u_w)$, $w < i$ or to the next shift of an existing primary signal. Thus all except possibly one column are contained in a preceding matrix $T_{max}(u_{i-1})$. Then the addresses of all columns of $T(u_i)$ included in $T_{max}(u_i)$ are determined. This task is performed by the column addressing box of Fig. 2.

The second submodule estimates the discrepancy detector f_s and the maximum length l_s of the upper subblock of column $X^N_{I_i}$ that is linearly dependent on the corresponding subblocks of the previous columns of $T_{max}(u_i)$, under the constraint that if a previous subblock contains a "don't care" entry, the corresponding tap is zero. The discrepancy detector of a column checks if the maximum linear combiner is applicable to the next row. This requires that all "don't care" entries in this row correspond to zero coefficients of the linear combiner. The discrepancy detector takes the value -1 (if combiner is not applicable) and 1. If all columns of $T(u_i)$ are included in $T_{max}(u_{i-1})$, their characteristics f_s and l_s have already been determined in a previous step. Estimation of l_s and f_s is realized via a modified version of FIA [2] (EFIA) taking into consideration that the "don't care" entries are located successively at the bottom end of the columns.

The features l_s and f_s critically affect subsequent algorithmic steps. If for $X^N_{I_i}$ holds that $f_s = 1$ and $l_s < N_i$, then the algorithm proceeds to the first submodule. ($X^{N_i}_{I_i}$ denotes the upper subblock of $X^N_{I_i}$ that belongs to $T_1(u_i)$). Otherwise the third

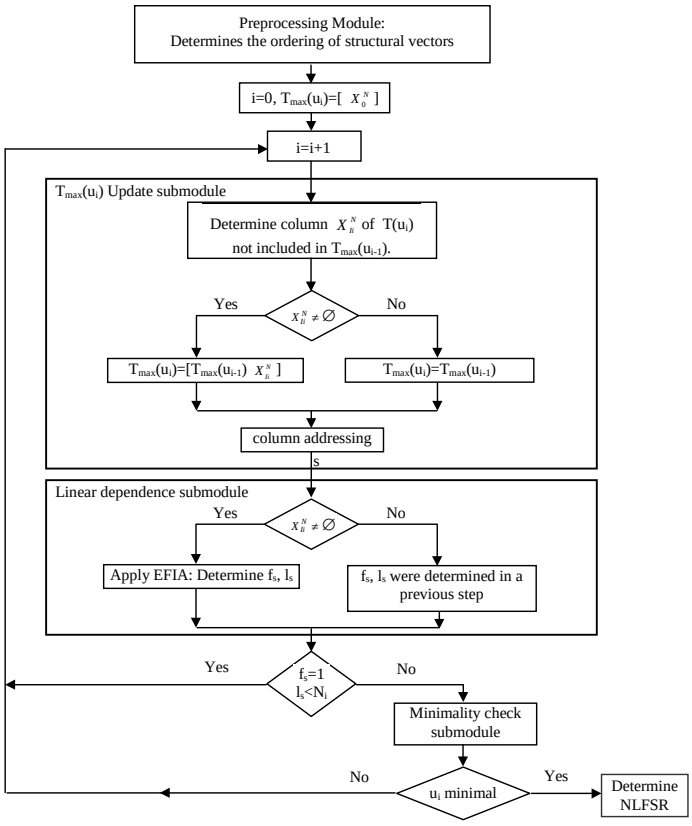

Figure 2: The structure of the proposed algorithm

submodule checks if the obtained maximum linear combiner corresponds to the desired minimal solution. If this is not the case the algorithm proceeds to the first submodule. If that is the case then u_i is the minimal solution, since as the algorithm proceeds sequentially according to the total ordering, at step i all solutions $u <_V u_i$ have been rejected. Scaling and shifting leads to the minimal cost NLFSR.

Complexity analysis

The exhaustive search involves the application of the linear dependence test on a set of matrices (matrix by matrix), while the proposed algorithm mainly involves the application of the linear dependence test on a single matrix (column by column) until the minimal NLFSR is reached. Suppose that u_b is the structural vector of the minimal system generating the terms of the given sequence. Then, the proposed algorithm applies the linear dependence test on a $N \times q$ matrix, $q \leq b+1$ columns. If the exhaustive search is applied the solution is obtained through the application of FIA on b matrices of size $N \times q_i$, $i = 1, \cdots, b$. The reduction is obvious since $\sum_{i=1}^{b} q_i > q$.

Example

Consider the finite field $GF(2^4)$ generated by $p(x) = 1+x+x^4$, a primitive element a and the sequence a^7, a^5 ,0, a^{10}, a^7, a^4, a^3. Application of the algorithm for structural vectors $u \in \mathcal{A}$ with $k \leq 4$, $L \leq 10$ and $L_{i_1,\cdots,i_{m-1}} \leq 10, 2 \leq m \leq 4$, provides the following minimal NLFSR

$$x(n) = a^2 \left(x(n-1)\right)^2 + a^5 x(n-2), \quad n > 2$$

which indeed generates the given sequence. The same sequence is generated by the linear equation

$$x(n) = a^{10}x(n-1)+ax(n-2)+a^7x(n-3)+a^9x(n-4),$$

$n \geq 5$. This linear register would be obtained by the algorithm if the latter was allowed to continue further. ■

5 Conclusions

The proposed algorithm can be viewed as extension of the Berlekamp-Massey algorithm [1] and the FIA scheme [2]. The updates are based on a discrepancy detection mechanism but the search is carried over a multidimensional set ordered by the total degree order plus a cost function reflecting the implementation cost. Although the proposed algorithm is designed to produce the minimal architecture, it can also be employed to determine alternate architectures that generate the given sequence. Furthermore the proposed algorithm applies to sequences in arbitrary fields.

References

[1] E.R. Berlekamp, Algebraic Coding Theory, New York:McGraw-Hill, 1968.

[2] G.L. Feng and K.K. Tzeng, "A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes", IEEE Trans. on Inf. Theory, vol. 37, 1991, pp. 1274-1287.

[3] G-O. Glentis, P. Koukoulas and N. Kalouptsidis, "Efficient algorithms for Volterra system identification", IEEE Trans. on Signal Processing, vol. 47, 1999, pp. 3042-3057.

[4] N. Kalouptsidis, "Signal Processing Systems: Theory and Design", John Wiley & Sons, Inc., 1997.