# CYCLE ANALYSIS FOR DETERMINISTIC FINITE STATE AUTOMATA

**Johann Reger** [1]

*Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg*
*Cauerstraße 7, D–91058 Erlangen, Germany*
*fax: +49(0)9131/85-28715*
*e-mail: johann.reger@rzmail.uni-erlangen.de*

Abstract: A method for the determination of all cycles of deterministic finite state automata is presented. Using a finite field description *GF(2)* these automata are shown to correspond to multilinear state equations. Basic results from feedback shift register theory are recalled and developed towards a complete analysis tool for affine-linear automata. The multilinear case is reduced to the affine-linear case by application of a linearizing constant state feedback and a linear state space embedding. Some brief examples illustrate the basic ideas. 

Keywords: Discrete event systems, finite automata, finite fields, shift registers

## 1. INTRODUCTION

Based on a state space model the problem of determining all cycles of deterministic finite state automata was investigated recently (Franke, 1994; Sonnenberg, 1999). In these approaches three major difficulties occur. On the one hand the modeling generally lacks sufficient existence criteria for cycles in the automata. Even for linear systems just necessary criteria are available. On the other hand cycles of certain length cannot be specified in number without enumerating the state space. But the main problem is the complexity: solving for certain cycle states is (unnecessarily) NP-complete, since solving a linear diophantine system of equations for *boolean solutions only* (e. g. cyclic states) belongs to the class of NP-complete problems. There is no polynomial algorithm that constructs boolean vectors out of a linear combination of integral or rational vectors. Those difficulties originate from an inappropriate algebraic modeling which admits integral and rational numbers respectively for the parameters in the state equations, but claims to keep states and inputs boolean. The method proposed here is capable of overcoming these obstacles using an algebraic state space description that is formulated strictly in the set of boolean numbers which is equivalent to the finite field *GF(2).* For an introductory textbook to finite field theory see for example (Lidl and Niederreiter, 1994). Finite field models have already been under consideration in control (Benveniste *et al.*, 1991; Germundsson, 1995), however, they were not utilized for determining the cyclic structure of automata, which is the main objective of this paper. Concerning linear systems much of the theory was developed as early as the sixties — for instance the design of linear feedback shift registers (Elspas, 1959; Gill, 1966) — but has not been adapted yet for control purposes. Concerning affine-linear automata this framework enables sufficient criteria for determining all cycles, in number and length, and at least for linear systems of equations it allows solving for cyclic states in polynomial complexity, e. g. by the Gauß-algorithm. In a second stage, multilinear state equations can be handled, attempting to reduce the problem to a linear one.

In Section 2 a state space model within *GF(2)* is developed. Section 3 provides the theory to analyze affine-linear automata. In Section 4 the multilinear case is

[1] My sincere thanks to the German foundation "Studienstiftung des deutschen Volkes" for their generous support.

considered. A linearizing constant state feedback and a linear state space embedding are proposed before conclusions are drawn in Section 5.

## 2. STATE SPACE MODEL IN FINITE FIELDS

Given a description of the deterministic automaton (e. g. petri nets, state tables) a state space model can be obtained. For example if the automata states are binary coded the next state behavior may be noted in a state table representation by introduction of a respective number $n$ of state variables $x_i$, $i = 1(1)n$, and of $m$ input variables $u_j$, $j = 1(1)m$. In the case of fully specified automata (to be confined here for simplicity) all $2^n$ combinations of binary values can be taken on by the variables $x_i$. Recalling basic boolean algebra such tabled transition functions can be phrased in a disjunctive normal form. Each successive state $x_i'$, $i = 1(1)n$, is expressed in terms of the former state $x_i$

$$x_i' = c_i^1\,\overline{x_1}\cdots\overline{x_{n-1}}\,\overline{x_n} + c_i^2\,x_1\overline{x_2}\cdots\overline{x_{n-1}}\,\overline{x_n} + \ldots + c_i^{2^n}\,x_1\cdots x_{n-1}x_n, \quad x_i, c_i^j \in \{0,1\} \tag{1}$$

in which overscore signifies the complement or negation of the variable underneath and multiplications and summations represent the logic operations AND and XOR respectively. Usually the vector of binary constants $\mathbf{c}_i^{\mathrm{T}} = (c_i^1, c_i^2, \ldots, c_i^{2^n})$ is called the truth vector.

Alternatively another canonical form without any negated variables, the so-called Positive Polarity Reed-Muller expansion (PPRM), can be deduced from (1). This expansion is based on only the two operations AND and XOR constructing a boolean ring which is equivalent to the residue class ring with the operations summation and multiplication modulo 2, i. e. the finite field *GF(2)*. The PPRM expansion is attainable by substituting all negations $\overline{x_i}$ by $1+x_i$

$$x_i' = d_i^0 + d_i^1 x_1 + d_i^2 x_2 + d_i^{12} x_1 x_2 + d_i^3 x_3 + d_i^{13} x_1 x_3 + d_i^{23} x_2 x_3 + d_i^{123} x_1 x_2 x_3 + \ldots + d_i^{12\ldots n} x_1 x_2 \cdots x_n,\ i = 1(1)n,\ GF(2) \tag{2}$$

where *GF(2)* is of binary characteristic with field element set $\{0,1\}$. Equation (2) can be considered as the multilinear state equation of an autonomous deterministic finite state automaton. By $2^{\mathcal{N}}$ denoting the power set of $\mathcal{N} = \{1,2,\ldots,n\}$ and $\mathbf{x}^{\mathrm{T}} = (x_1, x_2, \ldots, x_n) \in GF^n(2)$ equation (2) can be rewritten in a compact form as

$$x_i[k+1] = f_i(\mathbf{x}) = \sum_{\mathcal{S}\in 2^{\mathcal{N}}} d_i^{\mathcal{S}} \prod_{j\in\mathcal{S}} x_j[k], \qquad \prod_{j\in\emptyset} \ldots \stackrel{\mathrm{def}}{=} 1, \quad i = 1(1)n,\ GF(2), \tag{3}$$

the latter definition allows for $d_i^0$, assumed in the sections that follow. The counter $k$ is to exhibit the sequential character of time progress in the state equation above. The general, non-autonomous case with $m$ input variables $u_i$, $\mathbf{u}^{\mathrm{T}} = (u_1, u_2, \ldots, u_m) \in GF^m(2)$, follows from the same formalism

$$x_i[k+1] = f_i(\mathbf{x},\mathbf{u}) = \sum_{\mathcal{S}_1\in 2^{\mathcal{N}}} \sum_{\mathcal{S}_2\in 2^{\mathcal{M}}} d_i^{\mathcal{S}_1,\mathcal{S}_2} \Big(\prod_{j\in\mathcal{S}_1} x_j[k]\Big)\Big(\prod_{l\in\mathcal{S}_2} u_l[k]\Big), \quad i = 1(1)n,\ GF(2). \tag{4}$$

This state equation is multilinear again. $2^{\mathcal{M}}$ is the power set of $\mathcal{M} = \{0,1,2,\ldots,m\}$. Summation and multiplication are carried out modulo 2 for the rest of the paper.

In Section 4 the automata in question will be reduced to simpler representations: affine-linear automata. Deterministic finite state automata are called affine-linear if their state equation is

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{a},\ GF(2). \tag{5}$$

The vector $\mathbf{x}[k] \in \mathcal{X}$ represents a state in state space $\mathcal{X} \subseteq GF^n(2)$, as do the vector of inputs $\mathbf{u}[k] \in \mathcal{U}$ with $\mathcal{U} \subseteq GF^m(2)$ in the same manner. $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{a}$ are constant matrices of appropriate dimension, the elements of which are binary. Outputs shall not be taken into account here.

### 2.1 *Remark*

Regarding the vectors $\mathbf{d}_i^{\mathrm{T}} = (d_i^0, d_i^1, d_i^2, \ldots, d_i^{12\ldots n})$ and $\mathbf{c}_i$ the dimensions are equal since

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n. \tag{6}$$

By the recursive Reed-Muller generator matrices

$$\mathbf{G}_n \stackrel{\mathrm{def}}{=} \begin{pmatrix} \mathbf{G}_{n-1} & \mathbf{0} \\ \mathbf{G}_{n-1} & \mathbf{G}_{n-1} \end{pmatrix}, \quad \mathbf{G}_0 \stackrel{\mathrm{def}}{=} 1 \tag{7}$$

and operating over *GF(2)* the vectors $\mathbf{d}_i$ and $\mathbf{c}_i$ are easily calculated from each other by

$$\mathbf{d}_i = \mathbf{G}_n\mathbf{c}_i \quad \Longleftrightarrow \quad \mathbf{c}_i = \mathbf{G}_n\mathbf{d}_i \tag{8}$$

because with $a + a = 0 \bmod 2$ and by induction from (7) it follows that $\mathbf{G}_n = \mathbf{G}_n^{-1}$. The procedure holds as well for the non-autonomous case (4).

## 3. CYCLE ANALYSIS FOR AUTONOMOUS LINEAR AUTOMATA

As the theory of linear feedback shift registers within finite field representations applies to deterministic finite state automata as well, some of these results need to be outlined. Even if it is true, for the linear case, that many parts of the theory in finite fields are the same as in the field of real numbers, still some of the properties differ. This is why the following review of this section is organized in a broader fashion. The results are restricted to the autonomous, linear case, that is $\mathbf{u}[k] = \mathbf{0}$ in (5) and are split into their complementary cases: the homogeneous, $\mathbf{a} = \mathbf{0}$, and the inhomogeneous $\mathbf{a} \neq \mathbf{0}$ state equation.

3.1 *Homogeneous state equation*

When determining the cyclic automaton states $\mathbf{x}$, which comply with $\mathbf{x}[k+l] = \mathbf{x}[k]$ for some integral number $l$ (cycle of length $l$), the algebraic counterpart, that is the period of polynomials, plays a decisive role. Following the steps proposed in (Elspas, 1959) first the elementary divisors $[p_i(\lambda)]^{e_{jk}}$ with regard to the dynamic matrix $\mathbf{A}$ are determined. One well known way to ascertain the elementary divisors is to apply unimodular transformations on $\mathbf{A}+\lambda\mathbf{I}$ (adding and subtracting coincide in *GF(2)*) in order to obtain the Smith canonical form of $\mathbf{A}$. The diagonal polynomials in the Smith canonical form subsequently divide the next lower diagonal polynomials and are factorizable in monic, irreducible polynomials of degree at least 1, $p_i(\lambda)$, which are the elementary divisors

$$\begin{aligned}
&[p_1(\lambda)]^{e_{11}}[p_1(\lambda)]^{e_{12}}\ldots[p_1(\lambda)]^{e_{1h_1}},\ 0<e_{11}\le e_{12}\le\ldots\le e_{1h_1}\\
&[p_2(\lambda)]^{e_{21}}[p_2(\lambda)]^{e_{22}}\ldots[p_2(\lambda)]^{e_{2h_2}},\ 0<e_{21}\le e_{22}\le\ldots\le e_{2h_2}\\
&\qquad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&[p_\nu(\lambda)]^{e_{\nu1}}[p_\nu(\lambda)]^{e_{\nu2}}\ldots[p_\nu(\lambda)]^{e_{\nu h_\nu}},\ 0<e_{\nu1}\le e_{\nu2}\le\ldots\le e_{\nu h_\nu}\,.
\end{aligned}$$

Furthermore the characteristic polynomial of $\mathbf{A}$ is the product of all elementary divisors whereas the product of those elementary divisors regarding the $\nu$ greatest elementary divisor exponents $e_{1h_1}, e_{2h_2}, \ldots, e_{\nu h_\nu}$ is the minimal polynomial of $\mathbf{A}$, i. e. the lowest diagonal polynomial in the Smith canonical form. Unlike polynomials over the field of real numbers which completely decompose into quadratic polynomials over the field of real numbers again, the polynomials over finite fields do not decompose into irreducible polynomials of degree two; even irreducible polynomials of full degree exist. Numerous algorithms are available to execute this factorization, the irreducible polynomials in *GF(2)* are tabulated.

Since states of deterministic automata are either cyclic or are taken only once, two cases have to be considered. These refer to the singularity of $\mathbf{A}$. Matrix $\mathbf{A}$ is transformable into the rational canonical form $\hat{\mathbf{A}}$, a diagonal block matrix of the companion matrices related to every elementary divisor. This form can be achieved by a similarity transformation $\hat{\mathbf{A}} = \mathbf{PAP}^{-1}$ which allows the separation of the nilpotent part $\hat{\mathbf{A}}_0$ from the nonsingular part $\hat{\mathbf{A}}_1$ within a diagonal block structure

$$\hat{\mathbf{A}} = \begin{pmatrix} \hat{\mathbf{A}}_1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}}_0 \end{pmatrix},\ \textit{GF(2)}. \tag{9}$$

Thus by $\hat{\mathbf{x}} \stackrel{\text{def}}{=} \mathbf{Px}$, $\mathbf{x}$ can be combined additionally via $\hat{\mathbf{x}}^{\mathrm{T}} = (\hat{\mathbf{x}}_1^{\mathrm{T}}, \hat{\mathbf{x}}_0^{\mathrm{T}})$. In the next section some results are presented regarding the nonsingular and the singular case, followed by a superposition.

3.1.1. *Nonsingular case* First the nonsingular part $\hat{\mathbf{A}}_1$ of $\hat{\mathbf{A}}$ is of concern and the associated minimal polynomial does not contain factors $\lambda^i$ respectively. The period (or order) of an irreducible polynomial $p(\lambda)$ with $p(0)\neq 0$ (excluding factors $\lambda^i$) is the least (positive) integer $\tau$ for which $p(\lambda)$ divides $\lambda^\tau - 1$. As summation and subtraction are identical in *GF(2)* this is equal to $p(\lambda)|\lambda^\tau + 1$. It is necessary to compute the periods of all $p_i(\lambda)$ in the elementary divisors. Periods of irreducible polynomials can be found in (Lidl and Niederreiter, 1994), regarding *GF(2)* up to polynomial degree $d = 11$. If no tabular is available the periods can be derived from the fact that in *GF(2)* the period $\tau$ of an irreducible polynomial of degree $d$ divides $2^d - 1$. Thus $\tau$ is a product of some prime factor powers constructing $2^d - 1$ and results from polynomial divisions.

Let $\tau_1^{(i)}$ denote the period of the monic, irreducible polynomial $p_i(\lambda)$. Once the periods $\tau_1^{(i)}, i = 1(1)\nu$, of the $\nu$ distinct base polynomials $p_i(\lambda)$ associated to all elementary divisors are determined, it is necessary to consider the periods of those polynomials $[p_i(\lambda)]^{e_{ik}}$ with powers $e_{ik} > 1$. Since elementary divisors with the same base polynomial trivially have common factors, the periods are calculated up to the greatest exponent $e_{ih_i}$. These periods $\tau_j^{(i)}$, $j = 1(1)e_{ih_i}$, are obtained by $\tau_j^{(i)} = \tau_1^{(i)} 2^l$, where $l$ marks the least integer in $2^l \ge j$. The periods for the $ik-$th elementary divisor follow: $\tau_1^{(i)}, \tau_2^{(i)}, \ldots, \tau_{e_{ik}}^{(i)}$.

These periods and the polynomial degrees $d_i$ with respect to $p_i(\lambda)$ are the necessary and sufficient information to determine the cycle sum $\Sigma_{ik}$ with regard to the elementary divisor $[p_i(\lambda)]^{e_{ik}}$. Denoting the cycle length by $\tau$ and its multiplicity by $\nu$ the cycle sum $\Sigma_{ik}$ consists of a sum of cycle expressions $\nu_j^{(i)}[\tau_j^{(i)}]$ yielding

$$\begin{aligned}
\Sigma_{ik} &= \left(1[1] + \nu_1^{(i)}[\tau_1^{(i)}] + \nu_2^{(i)}[\tau_2^{(i)}] + \ldots + \nu_{e_{ik}}^{(i)}[\tau_{e_{ik}}^{(i)}]\right)\\
&= \left(1[1] + \frac{2^{d_i}-1}{\tau_1^{(i)}}[\tau_1^{(i)}] + \frac{2^{2d_i}-2^{d_i}}{\tau_2^{(i)}}[\tau_2^{(i)}] + \ldots + \right.\\
&\qquad \left. \frac{2^{e_{ik}d_i} - 2^{(e_{ik}-1)d_i}}{\tau_{e_{ik}}^{(i)}}[\tau_{e_{ik}}^{(i)}]\right), \quad \begin{matrix} i = 1(1)\nu, \\ k = 1(1)w_i, \end{matrix}
\end{aligned} \tag{10}$$

where $w_i$ is the number of elementary divisors containing the base polynomial $p_i(\lambda)$. The common cycle $1[1]$ is related to the zero vector $\hat{\mathbf{x}}_1 = \mathbf{0}$ which trivially is invariant — hence 1 cycle of length 1 — within linear systems. The sum $\Sigma$ superposing all cycle sums $\Sigma_{ik}$ can be written as a formal product of the cycle sums

$$\begin{aligned}
\Sigma = {} & \Sigma_{11}\Sigma_{12}\cdots\Sigma_{1w_1}\Sigma_{21}\Sigma_{22}\cdots\Sigma_{2w_2}\cdots\\
& \Sigma_{\nu1}\Sigma_{\nu2}\cdots\Sigma_{\nu w_\nu}\,.
\end{aligned} \tag{11}$$

The products within (11) are calculated via

$$\begin{aligned}
\nu[\tau] &= \nu_1[\tau_1]\,\nu_2[\tau_2]\cdots\nu_j[\tau_j] \stackrel{\text{def}}{=}\\
&\stackrel{\text{def}}{=} \frac{\prod_{i=1}^{j} n_i\tau_i}{\operatorname{lcm}(\tau_1,\tau_2,\ldots,\tau_j)}\left[\operatorname{lcm}(\tau_1,\tau_2,\ldots,\tau_j)\right]
\end{aligned} \tag{12}$$

and numbers of cycles can be summed up by

$$\nu_i[\tau_j] + \nu_k[\tau_j] \stackrel{\text{def}}{=} (\nu_i + \nu_k)[\tau_j]\,. \tag{13}$$

3.1.2. *Nilpotent case* In (9) matrix $\hat{\mathbf{A}}_0$ embodies the noncyclic part of $\hat{\mathbf{A}}$. Thus $\hat{\mathbf{x}}_0 = \hat{\mathbf{A}}_0^\tau \hat{\mathbf{x}}_0$ is solvable for the null state $\hat{\mathbf{x}}_0 = \mathbf{0}$ only. All states branch into the null state for some successive state, so the interconnection of the states can be considered to represent a tree of states. As a consequence $\hat{\mathbf{A}}_0$ is nilpotent because $\hat{\mathbf{A}}_0^\tau = \mathbf{0}$ for some integer $\tau$. Matrix $\hat{\mathbf{A}}_0$ is in rational canonical form accordingly

$$\hat{\mathbf{A}}_0 = \operatorname{diag}(\mathbf{N}_{e_{0h_0}}, \ldots, \mathbf{N}_{e_{02}}, \mathbf{N}_{e_{01}}), \quad 0 < e_{01} \le e_{02} \le \ldots \le e_{0h_0} \tag{14}$$

where $e_{0k}$ again are (not necessarily distinct) elementary divisor exponents, but concerning the nilpotent matrix $\hat{\mathbf{A}}_0$ and

$$\mathbf{N}_i = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{15}$$

is a nilpotent square matrix of dimension $i$. Immediately it follows that the null state is reached in at most $e_{0h_0}$ steps. According to (Gill, 1966), the number of states $\kappa_i$ from which it takes $i$ (and not less) steps to reach the null state define a level $i$ and recursively can be derived from

$$\kappa_i = 2^{\left( \sum_{j=1}^{i} j m_j + i \sum_{j=i+1}^{e_{0h_0}} m_j \right)} - (\kappa_0 + \kappa_1 + \ldots + \kappa_{i-1}), \quad \kappa_0 \stackrel{\text{def}}{=} 0, \quad i = 1(1) e_{0h_0}. \tag{16}$$

The numbers $m_j$ denote the multiplicity of the elementary divisors with the same value $j$. The number of states $\eta_i$ which reach a given state in $i$ steps can be shown to be either $\eta_i = 0$ or

$$\eta_i = \kappa_0 + \kappa_1 + \kappa_2 + \ldots + \kappa_i. \tag{17}$$

The equations (16) and (17) are sufficient to construct the tree of states, the so-called null tree.

3.1.3. *General, singular case* Since any singular linear system in *GF(2)* is decomposable according to (9) the general statement follows by superposition of the nonsingular and the nilpotent subsystems. After the cycle structure with regard to the nonsingular portion $\hat{\mathbf{A}}_1$ is analyzed, and having constructed the null tree of the nilpotent part $\hat{\mathbf{A}}_0$, the graph of the singular system over *GF(2)* is obtained simply by attaching the null tree to each cycle state associated to $\hat{\mathbf{A}}_1$. In exact terms, for one such structure containing one cycle (possibly more) the state transitions pursue the subsequent scheme: starting at one uppermost state of the attached tree, the substate $\hat{\mathbf{x}}_1$ steps the respective cycle states in regard of $\hat{\mathbf{A}}_1$ while synchronously the tree states $\hat{\mathbf{x}}_0$ concerning $\hat{\mathbf{A}}_0$ step by step aspire a state $\hat{\mathbf{x}}^{\mathrm{T}} = (\hat{\mathbf{x}}_1^{\mathrm{T}}, \mathbf{0})$. Once this state is reached the influence of the nilpotent part vanishes and $\hat{\mathbf{x}}_1$ passes through all states of the respective cyclic subspace.

3.2 *Inhomogeneous state equation*

The case $\mathbf{a} \neq \mathbf{0}$ is reducible to the homogeneous case by a shift of state $\check{\mathbf{x}} = \mathbf{x} + \mathbf{c} \Longleftrightarrow \mathbf{x} = \check{\mathbf{x}} + \mathbf{c}$,

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{a}, \\ \Longleftrightarrow \check{\mathbf{x}}[k+1] + \mathbf{c} &= \mathbf{A}\check{\mathbf{x}}[k] + \mathbf{A}\mathbf{c} + \mathbf{a}, \; \textit{GF(2)}, \end{aligned} \tag{18}$$

if

$$(\mathbf{A} + \mathbf{I})\,\mathbf{c} = \mathbf{a}, \quad \textit{GF(2)} \tag{19}$$

holds for some $\mathbf{c} \in$ *GF(2)*. This is the case if $\mathbf{A}$ has no elementary divisor $(1+\lambda)^d$ since then $\mathbf{A}+\mathbf{I}$ is nonsingular and spans $\mathcal{X}$. Hence it suffices to consider those diagonal block matrices, $\mathbf{A}_u$, in the rational canonical form $\hat{\mathbf{A}}$ referring to those elementary divisors consisting of powers of $1+\lambda$ which do not comply with (19). By the state transform $\overline{\mathbf{x}} = \mathbf{P}\mathbf{x} + \mathbf{P}\mathbf{c}$ set in (18) follows

$$\begin{bmatrix} \overline{\mathbf{x}}_s[k+1] \\ \overline{\mathbf{x}}_u[k+1] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_u \end{bmatrix} \begin{bmatrix} \overline{\mathbf{x}}_s[k] \\ \overline{\mathbf{x}}_u[k] \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{P}_u \mathbf{a} \end{bmatrix}, \textit{GF(2)}, \tag{20}$$

with $\mathbf{P}^{\mathrm{T}} = (\mathbf{P}_s, \mathbf{P}_u)$. The subscripts signify if condition (19) can be complied with by the subsystem or not, clearly: $(\mathbf{A}_u + \mathbf{I})\,\mathbf{c} = \mathbf{P}_u\,\mathbf{a}$ is not solvable for $\mathbf{c}$. The upper part of the system may be analyzed by means of the results from Section 3.1; let $\Sigma_s$ denote the associated cycle sum.

The lower part can be analyzed based on a result in (Gill, 1966); not proven here for brevity: Every subsystem of dimension $d_i$ (possibly not unique) induced by the companion matrices $\mathbf{C}_{(1+\lambda)^{d_i}}$ in $\mathbf{A}_u$ implies a single cycle

$$\frac{2^{d_i}}{\tau_{(d_i)}}[\tau_{(d_i)}], \tau_{(d_i)} = \begin{cases} 2^l & \text{if } 2^l > d_i > 2^{l-1}, l \in \mathbb{N} \\ 2^{l+1} & \text{if } d_i = 2^l, l \in \mathbb{N}. \end{cases} \tag{21}$$

Thus the cycle sum $\Sigma_u$ regarding $\mathbf{A}_u$ consists of cycle expressions (21) for each companion matrix in $\mathbf{A}_u$. The cycle sums $\Sigma_s$ and $\Sigma_u$ follow from superposition via (11)–(13).

3.3 *Example*

In order to demonstrate the ideas described previously an affine-linear (inhomogeneous) system of order $n = 6$, $\mathbf{A}$ already in rational canonical form, is examined.

$$\mathbf{x}[k+1] = \begin{bmatrix} \mathbf{A}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{(1+\lambda)^2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{(1+\lambda)^3} \end{bmatrix} \mathbf{x}[k] + \mathbf{a}, \; \mathbf{A}_0 = 0,$$

$$\mathbf{C}_{(1+\lambda)^2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{C}_{(1+\lambda)^3} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \mathbf{a} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{22}$$

Matrix $\mathbf{C}_{(1+\lambda)^2}$ does comply with (19) by $\mathbf{c}^{\mathrm{T}} = (1,0)$, $\mathbf{A}_0$ as well. Thus this part can be reduced to the homogeneous case. Concerning $\mathbf{C}_{(1+\lambda)^2}$ the period of $p_1(\lambda) = 1+\lambda$ is $\tau_1^1 = 1$, of $[p_1(\lambda)]^2 = (1+\lambda)^2 = \lambda^2 + 1$ it is $\tau_2^1 = 2$. With $d_1 = 1$, the polynomial degree of

$p_1(\lambda)$, and using the main results (10) and (13) the cycle sum $\Sigma_s$ amounts to

$$\begin{aligned}\Sigma_s &= \left(1[1] + \frac{2^1-1}{1}[1] + \frac{2^2-2^1}{2}[2]\right) \\ &= (1[1] + 1[1] + 1[2]) = (2[1] + 1[2])\,. \end{aligned} \quad (23)$$

Equation (19) cannot be fulfilled for $\mathbf{C}_{(1+\lambda)^3}$. Hence by (21) with $d_1 = 3, 2^l > d_1 > 2^{l-1}$ follows $\tau_3 = 4$, finally

$$\Sigma_u = \left(\frac{2^3}{4}[4]\right) = (2[4])\,, \quad (24)$$

which via (23) and by use of (12) and (13) completely yields the cycle sum of (22)

$$\begin{aligned}\Sigma = \Sigma_s\Sigma_u &= \big(2[1] + 1[2]\big)\big(2[4]\big) = \big(2[1]2[4] + 1[2]2[4]\big) \\ &= \left(\frac{2\cdot 1\cdot 2\cdot 4}{\mathrm{lcm}(1,4)}[\mathrm{lcm}(1,4)] + \frac{1\cdot 2\cdot 2\cdot 4}{\mathrm{lcm}(2,4)}[\mathrm{lcm}(2,4)]\right) \\ &= (4[4] + 4[4]) = (8[4])\,. \end{aligned} \quad (25)$$

As the nilpotent part $\mathbf{A}_0 = 0$, 1 noncyclic state leads to each of the 32 states, which are assigned to the 8 cycles of length 4.

## 4. INSPECTION OF NONLINEAR AUTOMATA

This section deals with the non-autonomous case of the multilinear state space representation (4). Striving to use the well established linear theory two methods for a general linearization are presented: a state feedback linearization and a linear state space embedding.

### 4.1 *Linearization by state feedback*

Concerning this more general class of systems the idea of a linearizing state feedback can be applied. Since all states are assumed to be measurable it is advantageous from the fact that the system equations of the controlled system using constant multilinear state feedback

$$u_l[k] = r_l(\mathbf{x}) = \sum_{\mathcal{S}_1'\in 2^{\mathcal{N}}} R_l^{\mathcal{S}_1'} \prod_{p\in\mathcal{S}_1'} x_p[k], \quad l = 1(1)m,\ GF(2), \quad (26)$$

with $\mathbf{r}^{\mathrm{T}} = (r_1, r_2, \ldots, r_m)$ remain multilinear

$$\begin{aligned} x_i[k+1] = f_i(\mathbf{x}, \mathbf{r}(\mathbf{x})) &= \sum_{\mathcal{S}_1\in 2^{\mathcal{N}}} \sum_{\mathcal{S}_2\in 2^{\mathcal{M}}} d_i^{\mathcal{S}_1\mathcal{S}_2} \Big(\prod_{j\in\mathcal{S}_1} x_j[k]\Big) \\ \Big(\prod_{l\in\mathcal{S}_2} \sum_{\mathcal{S}_1'\in 2^{\mathcal{N}}} R_l^{\mathcal{S}_1'} \prod_{p\in\mathcal{S}_1'} x_p[k]\Big),\ & i = 1(1)n,\ GF(2). \end{aligned} \quad (27)$$

This can be observed easily if it is recalled that:

$$(x_i)^2 = x_i,\ GF(2) \quad (28)$$

by which state variable products simplify via

$$\prod_{i\in\mathcal{S}_1}\prod_{j\in\mathcal{S}_2} x_i[k]x_j[k] = \prod_{p\in\mathcal{S}_1\cup\mathcal{S}_2} x_p[k],\ GF(2). \quad (29)$$

Hence (27) can be transformed into

$$x_i[k+1] = \hat{f}_i(\mathbf{x}, \mathbf{r}(\mathbf{x})) = \sum_{\mathcal{S}\in 2^{\mathcal{N}}} \delta_i^{\mathcal{S}}(R_l^{\mathcal{S}_1'}) \prod_{j\in\mathcal{S}} x_j[k], \quad i = 1(1)n,\ GF(2), \quad (30)$$

$\delta_i^{\mathcal{S}}$ depending multilinearly on $R_l^{\mathcal{S}_1'}$ still, but from now on in a form similar to (2).

The basic idea is to cancel, if possible, any multilinear addend $x_1x_2\cdots x_i$ in (30) by appropriate choice of $R_l^{\mathcal{S}_1'}$. Clearly, the constant state feedback (26) provides exactly these combinations of multilinear expressions in $x_i$ which are apt to cancel those multilinear expressions in $x_i$ that are not coupled with inputs $u_l$ since

$$\prod_{i\in\mathcal{S}} x_i[k] + \prod_{i\in\mathcal{S}} x_i[k] = 0,\ GF(2), \quad (31)$$

which is simply obtained from $a + a = 0, GF(2)$.

If $R_l^{\mathcal{S}_1'}$ can be chosen in this manner the remaining controlled system is linear; possibly with some undetermined $R_l^{\mathcal{S}_1'}$ for pursuing further control objectives.

### 4.2 *Linear state space embedding*

In cases where it is not possible to linearize the state equation (30) by any choice of feedback parameters, it is still possible to obtain a linear representation of the multilinear system (4). The decisive notion is to introduce any multilinear expression as a new, virtual state space variable. This approach is somewhat similar to a Carleman-Linearization for nonlinear continuous-time systems yielding a polynomial approximation of higher order for the system. But in contrast to the latter the procedure presented here is an exact linear embedding of a multilinear system into a state space of higher dimension. The procedure to apply on (30) refers to an algorithm (Reger, 2001) proposed for an alternative modeling of deterministic, finite state automata using arithmetical polynomials (Franke, 1994). It is adaptable to the *GF(2)* case just as it stands.

The basis of the algorithm is the state transition function $\hat{f}_i(\mathbf{x}(k))$, $i = 1(1)n$, in (30). Beginning with $\hat{f}_1$ (then by ascending addend number and index of $\hat{\mathbf{f}}$) the algorithm searches for products of state variables. If one such product is found it is defined as a new, virtual state variable $x_1^\star$. With regard to $x_1^\star$ the state transition function $x_1^\star[k+1] = \hat{f}_{n+1}(\mathbf{x}[k])$ can be calculated from the definition of $x_1^\star$ as product of former state variables at the same instant $k$, once $k$ is replaced by $k+1$. The next non-defined state variable product is searched for in $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_{n+1}$ and defined as $x_2^\star$, thereafter $x_2^\star[k+1] = \hat{f}_{n+2}(\mathbf{x}[k])$ is determined and so on. Since the number of combinations of state variables is finite, equal or less the upper bound $2^n - 1$, the algorithm terminates after yielding $n^\star$ virtual state variables $\mathbf{x}^{\star\mathrm{T}} = (x_1^\star, x_2^\star, \ldots, x_{n^\star}^\star)$ and their transition functions $\hat{f}_{n+1}, \hat{f}_{n+2}, \ldots, \hat{f}_{n+n^\star}$. Finally all occurrences of state variable products in $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_{n+n^\star}$ are replaced

by their equivalents in $\mathbf{x}^\star$. The algorithm results in an affine-linear state equation

$$\tilde{\mathbf{x}}[k+1] = \mathbf{A}(R_l^{S'_1})\tilde{\mathbf{x}}[k] + \mathbf{a}(R_l^{S'_1}),\ \textit{GF(2)}, \qquad (32)$$

in the $\tilde{n} \stackrel{\text{def}}{=} n + n^\star$ states $\tilde{\mathbf{x}} \stackrel{\text{def}}{=} (\mathbf{x}^{\mathrm{T}}, \mathbf{x}^{\star\mathrm{T}})^{\mathrm{T}}$ restricted on the defining equations for $\mathbf{x}^\star$.

### 4.3 *Example*

All equations in the short example are to be taken in *GF(2)* entirely. Consider the state equation

$$\begin{aligned} x_1[k+1] &= 1 + x_1[k] + x_1[k]x_2[k] + u[k]x_2[k]\,, \\ x_2[k+1] &= x_2[k] + x_1[k]x_2[k] + u[k] \end{aligned} \qquad (33)$$

with $n = 2$ states, $m = 1$ input $u$. With the idea of a linearizing feedback (26) from Section 4.1

$$u[k] = R^0 + R^1 x_1[k] + R^2 x_2[k] + R^{12} x_1[k]x_2[k] \qquad (34)$$

put in (33) one obtains

$$\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \end{bmatrix} = \begin{bmatrix} 1 & R^0 + R^2 \\ R_1 & 1 + R^2 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix} + \begin{bmatrix} 1 \\ R^0 \end{bmatrix} + \mathbf{m}(\mathbf{x}[k])\,, \qquad (35)$$

where $\mathbf{m}(\mathbf{x}[k])$ is the multilinear expression

$$\mathbf{m}(\mathbf{x}[k]) = \begin{bmatrix} 1 + R^1 + R^{12} \\ 1 + R^{12} \end{bmatrix} x_1[k]x_2[k]\,. \qquad (36)$$

Setting the feedback parameters to $R^{12} = 1$, $R^1 = 0$ yields a linearizing feedback. $R^0$ and $R^2$ remain for further control purposes. For each setting $R^0, R^2$ the methods for cycle analysis presented in Section 3 can be applied.

If linear behavior is not strived for, the method of Section 4.2 is adequate. This will be shown using the latter example, setting $R^0 = 1$, $R^1 = 1$, $R^2 = 0$ and $R^{12} = 0$ in (35) which yields

$$\begin{aligned} x_1[k+1] &= 1 + x_1[k] + x_2[k]\,, \\ x_2[k+1] &= 1 + x_1[k] + x_2[k] + x_1[k]x_2[k]\,. \end{aligned} \qquad (37)$$

Defining the virtual state $x_1^\star[k] \stackrel{\text{def}}{=} x_1[k]x_2[k]$ and replacing $k$ by $k+1$ the transition function

$$\begin{aligned} x_1^\star[k+1] &= x_1[k+1]x_2[k+1] = \ldots = \\ &= 1 + x_1[k] + x_2[k] + x_1[k]x_2[k] \end{aligned} \qquad (38)$$

follows. Replacement of $x_1x_2$ by the virtual state variable $x_1^\star$ in (37) and (38) results, abbreviating $\tilde{\mathbf{x}}^{\mathrm{T}} \stackrel{\text{def}}{=} (x_1, x_2; x_1^\star)$, in an affine-linear system

$$\tilde{\mathbf{x}}[k+1] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tilde{\mathbf{x}}[k] + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\,, \qquad (39)$$

which again can be analyzed as in Section 3.

## 5. CONCLUSION

This contribution presents a modeling of deterministic finite state automata in the finite field *GF(2)*. Based on results from the theory of linear feedback shift registers a consistent theory for the analysis of automata which correspond to affine-linear systems in *GF(2)* is developed. In this case all cycles of the automata can be determined in length and number without a state space enumeration procedure. The linear theory is extended to the general, multilinear case. First a linearizing constant state feedback is proposed. One advantage of this approach is that far-reaching linear analysis tools can be applied to investigate the behavior of the multilinear system. Moreover the procedure is intuitive, straight-forward and requires little calculation effort. Drawbacks are that for controller design less parameters remain in order to set the behavior of the controlled system. Too many parameters might be invested in the linearization process. Another problem is the structural restriction on linear systems. Therefore a further idea was introduced: a linear state space embedding. Without loss in generality, multilinear systems can be modeled linearly, in a linear state space of higher dimension. Thus the analysis can be carried out with the linear methods again, but has to take into consideration the multilinear restrictions stemming from the introduction of new state variables. The latter problem is work in progress and could not be dealt with here, on account of the brevity of the paper.

## 6. REFERENCES


Benveniste, A., P. Le Guernic and M. Le Borgne (1991). Dynamical systems over galois fields and DEDS control problems. In: *Proc. of 30th Conf. Decision and Control*. Brighton, UK. pp. 1505–1509.

Elspas, B. (1959). The theory of autonomous linear sequential networks. *IEEE Trans. Circuit Theory* **6**, 45–60.

Franke, D. (1994). *Sequentielle Systeme — Binäre und Fuzzy Automatisierung mit arithmetischen Polynomen*. Vieweg. Braunschweig.

Germundsson, R. (1995). Symbolic Systems — Theory, Computation and Applications. PhD thesis. Linköping.

Gill, A. (1966). Graphs of affine transformations, with applications to sequential circuits. In: *Proc. of the 7th IEEE International Symposium on Switching and Automata Theory*. Berkeley, California, USA. pp. 127–135.

Lidl, R. and H. Niederreiter (1994). *Introduction to finite fields and their applications*. Cambridge Univ. Press. New York.

Reger, J. (2001). Deadlock analysis for deterministic finite state automata using affine linear models. In: *Proc. of 2001 European Control Conference*. Porto, Portugal.

Sonnenberg, J. (1999). A new method for describing and analyzing finite determined automata by walsh functions. In: *Proc. of 1999 European Control Conference*. Karlsruhe, Germany.