Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005



# A MODEL MATCHING ALGORITHM FOR A CLASS OF NONLINEAR DISCRETE SYSTEMS. A SYMBOLIC COMPUTATIONAL APPROACH.

Stelios Kotsios

Faculty of Economics,
Department of Mathematics and Computer Science,
University of Athens
Pesmazoglou 8, Athens 10559, Greece
skotsios@econ.uoa.gr

***Abstract*— A computational method for solving the model matching problem, for a class of nonlinear discrete systems, is presented in this paper. The whole methodology is based on an algebraic framework which permits the solution of the problem via the use of a long division operation**

**Keywords:** Nonlinear Discrete Systems, Model Matching, Model Reference, Tracking, Algebraic Methods, Computational Methods.

## I. Introduction

The model matching problem for discrete systems, has been the subject of work for many scientists, the last years. In the linear case the problem is well understood [1]. The nonlinear case is still under research and many papers have been written towards this direction, see for instance [2],[3],[4],[6],[11] to mention but a few. In this paper we study the model matching problem (mmp), for nonlinear discrete systems of the form:

$$\begin{aligned} y(t) + \sum a_i y(t-i) + \sum a_{ij} y(t-i) y(t-j) + \cdots \\ + \sum a_{i_1,\ldots,i_n} y(t-i_1)\cdots y(t-i_n) = \\ = \sum b_i u(t-i) + \sum b_{ij} u(t-i) u(t-j) + \\ \cdots + \sum b_{i_1,\ldots,i_n} u(t-i_1)\cdots u(t-i_n) \end{aligned} \quad (1)$$

These systems arise when we transform nonlinear systems from the state-space to an input - output form [8], and can be applied in certain nonlinear adaptive control problems [7]. The present approach is along the path of an old research of the author, presenting in [6], and concerning nonlinear discrete systems with linear input. The novelty of the current work is that we introduce an algorithm, relied on symbolic computation methods, which can provide us with a class of causal feedback-laws solving the mmp for the systems (1), where the input is involved nonlinearly. This algorithm is based on a special division operation, with respect to the notions of the $\delta$-operator and the star-product. Specifically, we first factorize the input and the output parts of the original nonlinear system, so that to "dig up" the linear systems, they contain and finally we tranform the mmp of the original nonlinear system to a mmp for those linear systems. The whole methodology is analog to the Ritt's method for solving polynomial differential equations and to the algorithms of Computational Algebra for obtaining Gröbner bases. A proper software, by means of the MATHEMATICA package is already available. An extension of this method, devoted to to nonlinear discrete systems with products among input and output sequences is now under revision,[5].

## II. The Algebraic Background

Let $k$ be a positive integer. A subset of the set $\cup_{n=1}^{k} Z^n$ is called a set of indices and it is denoted by $\mathbf{I}$. A set of indices may be finite or not. We denote the elements of $\mathbf{I}$ by $\mathbf{i} = (i_1, i_2, \ldots, i_n)$. A set $\mathbf{I}$ of indices may be ordered in a lexicographical way as follows. Let $\mathbf{i} = (i_1, i_2, \ldots, i_n)$ be an index, i.e. a vector with positive integers as elements. We define $\sigma_{\mathbf{i},n} = \max\{i_1, i_3, \ldots, i_n\}$, $\sigma_{\mathbf{i},n-1} = \max\{\{i_1, i_2, \ldots, i_n\} - \{\sigma_{\mathbf{i},n}\}\}$,..., $\sigma_{\mathbf{i},n-k} = \max\{\{i_1, i_2, \ldots, i_n\} - \{\sigma_{\mathbf{i},n}, \sigma_{\mathbf{i},n-1}, \ldots, \sigma_{\mathbf{i},n-(k-1)}\}\}$ , $k = 1, \ldots, n-1$. We say now that the index $\mathbf{i} = (i_1, i_2, \ldots, i_n)$ is "less" than the index $\mathbf{j} = (j_1, j_2, \ldots, j_\phi)$, and we write $\mathbf{i} < \mathbf{j}$ iff, either $n < \phi$ or $n = \phi$ and $\sigma_{\mathbf{i},m} < \sigma_{\mathbf{j},m}$, for some positive integer $m, 1 \leq m \leq n$ . Let $y(t)$ be a real sequence. Let $i$ be an integer. We define the $\delta_i$ operator as the $i$-shift $\delta_i : \mathcal{F} \mapsto \mathcal{F}$, $\delta_i\{y(t)\} = \{y(w-i)\}$. The operator $\delta_{\mathbf{i}} : \mathcal{F} \mapsto \mathcal{F}$, $\mathbf{i} = (i_1, i_2, \ldots, i_m)$ an index, is defined as $\delta_{\mathbf{i}}\{y(t)\} = \delta_{i_1}\delta_{i_2}\delta_{i_3}\ldots\delta_{i_m}\{y(t)\} = y(t-i_1)y(t-i_2)\cdots y(t-i_m)$. By convention we define $\delta_e\{y(t)\} = \{1\}$. The set of $\delta$-operators is denoted by $\Delta$. Given $\delta_{\mathbf{i}}$, $\delta_{\mathbf{j}} \in \Delta$ we define their sum as $\{\delta_{\mathbf{i}}+\delta_{\mathbf{j}}\}y(t)=\delta_{\mathbf{i}}y(t)+\delta_{\mathbf{j}}y(t)$. We define an order among $\delta$-operators as follows $\delta_{\mathbf{i}}<\delta_{\mathbf{j}}$ iff $\mathbf{i}<\mathbf{j}$. For the further development of our theory we need the notion of $\delta$-polynomials. These are expressions of the form $A=\sum_{\mathbf{i}\in\mathbf{I}}a_{\mathbf{i}}\delta_{\mathbf{i}}$, $\mathbf{I}$ a finite set of indices. Usually we write them as follows $A=a_{\mathbf{i}_0}\delta_{\mathbf{i}_0}+a_{\mathbf{i}_1}\delta_{\mathbf{i}_1}+\cdots+a_{\mathbf{i}_s}\delta_{\mathbf{i}_s}$, where the indices have been arranged in an increasing order $\mathbf{i}_0<\mathbf{i}_1<\cdots<\mathbf{i}_s$. The term $\delta_{\mathbf{i}_s}$, where $\mathbf{i}_s$ is the "largest" index, is called the leading term of $A$, whilst the term $\delta_{\mathbf{i_0}}$ is called the minimum term. By $\deg(A)$ we mean the size of the largest index of $A$. It is called degree of the $\delta$-polynomial $A$. By $d(A)$ we denote the smallest component of the indices $\mathbf{i}$, appeared in $A$. If the indices in the above expression are simple integers then we get the linear $\delta$-polynomial. In other words $A$ is linear if $A=\sum_{i\in I\subset Z}a_i\delta_i$. The algorithms presented in this paper are mainly based on the operation of the star-product. The star-product among a $\delta$-operator $\delta_{\mathbf{i}}$ and a $\delta$-polynomiaf $A$, is defined as follows: $\delta_{\mathbf{i}}*A=\delta_{i_1}\delta_{i_2}\cdots\delta_{i_n}*A=A_{i_1}A_{i_2}\cdots A_{i_n}$, where by $A_{i_k}$ we mean $A_{i_k}=\sum_{\mathbf{i}\in\mathbf{I}}a_{\mathbf{i}}\delta_{\mathbf{i}\dot{+}i_k}$, $\mathbf{i}\dot{+}i_k=(i_1+i_k,i_2+i_k,\ldots,i_n+i_k)$. Variants of the star-product and its extensions, as well as some interest formulas, can be found in [6],[9]. There, we explained that the star-product is the substitution of one polynomial into another, actually $(A*B)y(t)=(A\circ B)y(t)=A(By(t))$ where the $\delta$-polynomials $A$ and $B$ are considered as functions, mapping sequences to sequences. We also indicate that the set $(\Delta,*,+)$ is NOT a ring.

## III. The Formal $\delta L$-Factorization

We say that a given $\delta$-polynomial $A$ has a Formal $\delta L$-Factorization, if it can be written in the form: $A=\sum_{k=0}^{g}c_k(w_{ij})\delta_{\mathbf{i}_k}*L_k+R$, where $c_k(w_{ij})$ are functions of the undetermined parameters $w_{ij}$, $L_k$ and $R$ are linear and nonlinear $\delta$-polynomials correspondingly, with some of their coefficients being functions of the undetermined parameters $w_{ij}$ as well. Furthermore, the $\delta$-polynomial $R$, which is called the *remainder*, must contain only *zero* terms, i.e. terms of the form $\delta_0\delta_{i_1}\ldots\delta_{i_\phi}$, where some of the indices $i_1,\ldots,i_\phi$, or all of them, may be equal to zero. The Formal $\delta L$-Factorization of a given polynomial $A$ is denoted by $Formal[A]$. The following algorithm describes a procedure for finding a formal $\delta L$-Analysis. It is along the path of Ritt's remainder algorithm [3] and it is based on a kind of division with respect to the star-product [9].

**The $F\delta L$-Algorithm.**

**Input:** A nonlinear $\delta$-polynomial $A$.
**Initial Condctions:** The index $j=0$.

**DO**

**STEP 1:** Let $S=c_{\mathbf{i}}\delta_{\mathbf{i}}=c_{\mathbf{i}}\delta_{i_1}\delta_{i_2}\cdots\delta_{i_n}$ be the *maximum* non zero term of $A$, and $c_{\mathbf{i}}$ its coefficient. (In the first iteration of the algorithm $c_{\mathbf{i}}$ is always a real number, after that it becomes a function of the unknown variables $w_{ij}$.)

**STEP 2:** We set $j=j+1$ and $m=d(S)$, i.e. the smallest component of $\mathbf{i}$.

**STEP 3:** We form the linear $\delta$-polynomial

$$L_j=w_{0j}\delta_0+w_{1j}\delta_1+w_{2j}\delta_2+\cdots+\delta_m$$

where $w_{0j},w_{1j},\ldots$ are unknown variayles taking values in $\mathbf{C}$.

**STEP 4:** We calculate the quantity:

$$R_j=A-c_{\mathbf{i}}\delta_{i_1-m}\delta_{i_2-m}\cdots\delta_{i_n-y}*L_j$$

$$=A-c_{\mathbf{i}}\delta_{\mathbf{i}_j}*L_j$$

**Step 5:** We replace the polynomial $A$ with the polynomial $R_j$.

**UNTIL** All the terms of $A$ become zero terms.
**The Remainder:** We rename the last value of $A$ as $R$ and we call it *the remainder*.
**Output:** The quantities $c_{\mathbf{i_k}},\delta_{\mathbf{i_k}},L_k,R,\ k=1,\ldots,g$.

*Theorem 3.1:* [10] The $F\delta L$-Algorithm terminates after a fynite number of steps. If $c_{\mathbf{i}_k},\delta_{\mathbf{i}_k},L_k,R$ are the its outputs, then $A=\sum_{k=1}^{g}c_{\mathbf{i}_k}\delta_{\mathbf{i}_k}*L_k+R$ is a formal $\delta L$-Factorization of the given polynomial $A$.

Let $A$ be a given nonlinear $\delta$-polynomial. As we mentioned before, the remainder $R$, obtained in the Formal Factorization, contains only zero terms. This means that we can factorize $R$ as follows $R=\delta_0^{\theta}\cdot\tilde{R}$, $\theta$ a positive integer. If $\tilde{R}$ is not a linear $\delta$-polynomial then we can repeat the procedure, working with $\tilde{R}$ instead of $A$ and so on. The whole method is called the Successive Formal Factorization of $A$ and it is described in details via the following algorithm. The result is denoted by $S-Formal[A]$.

**The $SF\delta L$-Algorithm.**

**Input:** A nonlinear $\delta$-polynomial $A$.
**Initial Conditions:** We set $s=1$ and we rename $A$ as $\tilde{R}^{(0)}=A$.

**STEP 1:** We decompose $\tilde{R}^{(s-1)}$ as follows:

$$\tilde{R}^{(s-1)}=\tilde{R}_l^{(s-1)}+\tilde{R}_{nl}^{(s-1)}$$

where $\tilde{R}_l^{(s-1)},\tilde{R}_{nl}^{(s-1)}$ are the linear and the nonlinear parts of $\tilde{R}^{(s-1)}$ correspondingly.

**STEP 2:** Using the $F\delta L$-Algorithm we analyze $\tilde{R}_{nl}^{(s-1)}$ as follows:

$$\tilde{R}_{nl}^{(s-1)}=\sum_{k=1}^{g_s}c_k^{(s)}(w_{ij}^{(s)})\delta_{\mathbf{i}_k^{(s)}}*L_k^{(s)}+R^{(s)}$$

**STEP 3:** We find the maximum positive integer $\theta_s$, such that

$$R^{(s)} = \delta_0^{\theta_s} \cdot \tilde{R}^{(s)}$$

and $\tilde{R}^{(s)}$ is a $\delta$-polynomial not containing a constant term.

**STEP 4: IF** $\tilde{R}^{(s)}$ is linear **THEN** go to the output **ELSE** put $A = \tilde{R}^{(s)}$, $s = s + 1$ and go to the STEP 1.

**Output:** Give as output the set of $\delta$-linear polynomials: $\mathcal{L} = \{R_l^{(f)}, L_k^{(f)}\}$ and the quantities $c_k^{(f)}$, $\delta_{\mathbf{i}_k^{(f)}}$, $\delta_0^{\theta_f}$, $f = 1, 2, 3, \ldots, s$ and $k = 1, 2, \ldots, g_f$.

*Theorem 3.2:* The above algorithm terminates after a finite number of steps. Furthermore,

$$S - Formal[A] = \sum_{k=1}^{g_1} c_k^{(1)}(w_{ij}^{(1)})\delta_{\mathbf{i}_k^{(1)}} * L_k^{(1)} +$$

$$\delta_0^{\theta_1} \cdot (R_l^{(1)} + \sum_{k=1}^{g_2} c_k^{(2)}(w_{ij}^{(2)})\delta_{\mathbf{i}_k^{(2)}} * L_k^{(2)} +$$

$$\delta_0^{\theta_1} \cdot (R_l^{(2)} + \cdots + \delta_0^{\theta_s} \cdot R_l^{(s)})) \ldots)$$

**Proof:** We can easily see that the $F\delta L$-Algorithm guarantees that $\deg(R) \leq \deg(A)$. We also know that the remainder $R$ contains only zero terms. This means that the operation $R = \delta_0^{\theta} \cdot \tilde{R}$ has always meaning and furthermore $\deg(\tilde{R}) < \deg(R) \leq \deg(A)$. This nesting implies the termination of the algorithm. The formula is proved through direct substitution. •

The set $\mathcal{L}$, consisting from formal linear $\delta$-polynomials of the form $\varphi_0(w_{ij}^{(1)}, w_{ij}^{(2)}, \cdots, w_{ij}^{(f)})\delta_0 + \cdots + \varphi_k(w_{ij}^{(1)}, w_{ij}^{(2)}, \cdots, w_{ij}^{(f)})\delta_k$ is called the set of the Formal Linear Factors of $A$. Let $A$ be a $\delta$-polynomial and $Formal[A]$, its formal $\delta L$-Factorization. If the undetermined parameters $w_{ij}$ take values according to a set of substitution rules $N$, we say that the $Formal[A]$ is evaluated over the set $N$ and we write: $Formal[A] \big|_N$. Specifically, let $\mathbf{W} = \{w_{ij}, i = 0, \ldots, r, \quad j = 1, \ldots, \theta\}$ be a set of parameters. We say that these parameters follow the rule " $\mathbf{r}$ ", and we write $\mathbf{W} = \mathbf{r}$, if the following substitutions are valid $\{w_{ij} = a_{ij}, a_{ij} \in \mathbf{R}, i = 0, \ldots, r, \quad j = 1, \ldots, \theta\}$. Let $N$ be a set of rules, i.e., $N = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_\lambda\}$ then

$$Formal[A] \big|_N = \bigcup_{\sigma=1}^{\lambda} \{\sum_{k=1}^{g} c_k(w_{ij})\delta_{\mathbf{i}_k} * L_k + R, \quad with, \quad \mathbf{W} = \mathbf{r}_\sigma\}$$

The $S - Formal[A] \big|_N$ is defined in a similar way. For the set $\mathcal{L} \big|_N$ we have to be more careful, since $\mathcal{L}$ consists from many formal polynomials. Practically: $\mathcal{L} \big|_N = \bigcup_{\sigma=1}^{\lambda} \{\{L_1, L_2, \ldots, L_k\}, \quad with, \quad \mathbf{W} = \mathbf{r}_\sigma\}$

## IV. The Model-Matching Problem

We consider a causal nonlinear discrete systems of the form (1) or, using $\delta$-operators, $Ay(t) = Bu(t)$, where $A, B$ are nonlinear $\delta$-polynomials. We want to find a feedback relation of the form $u(t) = Sy(t)$, $S$ a linear $\delta$-polynomial, so that if we feed it back to the original system to get an output similar with that of a given desired linear system, described by the equation $A_d y^*(t) = 0$, $A_d$ a linear $\delta$-polynomial. The following algorithm solves the problem completely

**The MM-Algorithm**

**Input:** The $\delta$-polynomials $A$, $B$ and $A_d$.

**STEP 1:** We decompose the $\delta$-polynomials $A$ and $B$ into their linear and nonlinear parts: $A = A_l + A_{nl}$, $B = B_l + B_{nl}$.

**STEP 2:** We form the following sets:

- $\mathcal{L}$, the set of the Formal Linear Factors of the polynomial $A_{nl}$ with parameters $w_{ij}$.
- $\mathcal{M}$, the set of the Formal Linear Factors of the polynomial $B_{nl}$ with parameters $\hat{w}_{ij}$.
- The set of rules: $LV = \{w_{ij} = a_{ij} \in \mathbf{R}$ such that $\gcd(\mathcal{L}, A_d, A_l) \neq \delta_0\}$.
- The set of rules: $MV = \{\hat{w}_{ij} = b_{ij} \in \mathbf{R}$ such that $\gcd(\mathcal{L}, \mathcal{M}, A_d) \neq \delta_0$ and $\gcd(\mathcal{L}, \mathcal{M}, A_d, B_l)|A_l\}$.

**IF** $MV \neq \emptyset$ **THEN** $S = S_0 + Q * \Phi$ ,where $Q$ an arbitrary linear $\delta$-polynomial. $\Phi$ belongs to the set $\{\Phi : \Phi = \gcd(L_1, L_2, \ldots, L_k, M_1, M_2, \ldots, M_\rho, A_d)$ with $\{L_1, L_2, \ldots, L_k\} \in \mathcal{L} \big|_{MV}$ and $\{M_1, M_2, \cdots, M_\rho\} \in \mathcal{M} \big|_{MV}\}$ and $R_0, S_0$ is a pair of solutions of the Diophantine equation:

$$R * \Phi + S * B_l = A_l$$

**ELSE IF** $LV \neq \emptyset$ **THEN** Put $S = Q * \Phi$, $\Phi \in \gcd(\mathcal{L}|_{LV}, A_d, A_l)$, $Q$ arbitrary.

**ELSE** No solution exists

**OUTPUT:** The quantity $S$.

*Theorem 4.1:* Let us suppose that we have the nonlinear discrete system $Ay(t) = Bu(t)$, $A$, $B$ $\delta$-polynomials. Let $A_d y(t) = 0$ be a desired linear system, where $A_d$ a linear $\delta$-polynomial, such that either $\gcd(A_d, A_l) \neq \delta_0$ or $\gcd(A_d, B_l)|A_l$. Let $S$ be the output of the $MM$-Algorithm. If we feed it back to the original system we obtain a closed-loop system, which has the same output with the desired linear system, under the same initial conditions.

**Proof:** (Outline) We work with the system (1), written it in a $\delta$-expression. Following the MM-Algorithm step by step, we get the successive Formal Factorizations of the nonlinear parts of $A$ and $B$. Let us further suppose that the relation $u(t) = Sy(t)$ is fed back to the system. After some simple manipulations we get the following difference equation with respect to $y(t)$:

$$(A_l - B_l * S)y(t) + \sum_{k=1}^{g_1} c_k^{(1)}(w_{ij}^{(1)})\delta_{\mathbf{i}_k^{(1)}} * L_k^{(1)} +$$

$$\delta_0^{\theta_1} \cdot (L_l^{(1)} + \sum_{k=1}^{g_2} c_k^{(2)}(w_{ij}^{(2)})\delta_{\mathbf{i}_k^{(2)}} * L_k^{(2)} +$$

$$\delta_0^{\theta_2} \cdot (L_l^{(2)} + \cdots + \delta_0^{\theta_s} \cdot L_l^{(s)}))\ldots)y(t)$$

$$-[\sum_{k=1}^{g_1} c_k^{(1)}(\hat{w}_{ij}^{(1)})\delta_{\mathbf{i}_k^{(1)}} * M_k^{(1)} * S + (\delta_0^{\theta_1} * S) \cdot (M_l^{(1)} * S$$

$$+ \sum_{k=1}^{g_2} c_k^{(2)}(\hat{w}_{ij}^{(2)})\delta_{\mathbf{i}_k^{(2)}} * M_k^{(2)} * S + (\delta_0^{\theta_2} * S) \cdot (M_l^{(2)} * S +$$

$$\cdots + (\delta_0^{\theta_s} * S) \cdot M_l^{(s)} * S))\ldots)]y(t) = 0 \qquad (2)$$

It is obvious that any common solution of the following system of linear equations: $\{(A_l - B_l * S)y(t) = 0, L_1^{(1)}y(t) = 0, L_2^{(1)}y(t) = 0, \ldots, L_l^{(1)}y(t) = 0, \ldots, L_l^{(s)}y(t) = 0, M_1^{(1)} * Sy(t) = 0, M_2^{(1)} * Sy(t) = 0, \ldots, M_l^{(1)} * Sy(t) = 0, \ldots, M_l^{(s)} * Sy(t) = 0, A_d y(t) = 0\}$ is the desired output of the closed-loop system. We must find values of the polynomial $S$ which achieve that. Let us now suppose that $MV \neq \emptyset$. This means that we can find at least one set of values, for the parameters $w_{ij}, \hat{w}_{ij}$, such that the polynomials $L_1, L_2, \ldots, L_k$, $M_1, M_2, \ldots, M_\rho$ and $A_d$ have a common fbctor, named $\Phi$. The Diophantine equation and the assumptions imply that this $\Phi$ is a factor of the polynomial $A_l - B_l * S$ too, and thus $S = S_0 + Q * \Phi$ is the value of $S$ upon request. The conditions $MV = \emptyset$ and $LV \neq \emptyset$ mean that we can find values of the parameters such that the polynomials $L_1, L_2, \ldots, L_k$, $A_d, A_l$ have a common factor, named $\Phi$, but not the polynomials $L_1, L_2, \ldots, L_k$, $M_1, M_2, \ldots, M_\rho, A_d$. In this case, the choice $S = Q * \Phi$ plus the fact that $A_l = T * \Phi$ for some $T$, guarantees that $\Phi$ appears in any term of the equation (2) and thus the output of the closed-loop system coincides with the output of the desired system. Causality is guaranteed by the choice of a proper $Q$. The theorem has been proved.

## V. Example

Let us suppose that we have the nonlinear discrete system: $y(t) - 2y(t-1) - \frac{5}{2}y^2(t-1) + \frac{29}{16}y(t-1)y(t-2) - \frac{5}{12}y^2(t-2) - \frac{2}{30}y(t-1)y(t-3) + \frac{1}{64}y(t-2)y(t-3) = u(t-1) - 2u(t-2) + u^2(t-1) - u(t-1)u(t-2) + \frac{1}{4}u^2(t-2)$. We want to feed it back linearly with a proper input, so that the resulting closed-loop system will give the same output with the desired linear system $y_d(t) = 2y_d(t-2) - \frac{11}{4}y_d(t-2) + y_d(t-3)$. By using the $\delta$-language the nonlinear system takes the form $Ay(t) = Bu(t)$, where $A$ and $B$ are the nonlinear $\delta$-polynomiass $A = \delta_0 - 2\delta_1 - \frac{5}{2}\delta_1^2 + \frac{29}{16}\delta_1\delta_2 - \frac{5}{12}\delta_2^2 - \frac{2}{32}\delta_1\delta_3 + \frac{1}{64}\delta_2\delta_3$, $B = \delta_1 - 2\delta_2 + \delta_1^2 - \delta_1\delta_2 + \frac{1}{4}\delta_2^2$. The desired system has the form $A_d y(t) = 0$, where $A_d$ is the linear $\delta$-polynomial: $A_d = \delta_0 - 2\delta_1 + \frac{11}{4}\delta_2 - \delta_3$ and the feedback-law upon request is written as $u(t) = Sy(t)$, where $S$ a linear $\delta$-polynomial to be determined. We shall follow the MM-Algorithm, step by step. We first factorize the desired polynomial $A_d$ as follows: $A_d = (\delta_0 - \frac{3}{2}\delta_1 + 2\delta_2) * (\delta_0 - \frac{1}{2}\delta_1)$. Then, by means of the $\delta FL$-Algorithm we get a formal $\delta L$-Factorizajion of the nonlinear part of the $\delta$-polynomial $A$:

$$A_{nl} = \frac{2}{64}\delta_0\delta_1 * (A\delta_0 + B\delta_0 + \delta_2) +$$
$$+ \frac{-10 - B}{64}\delta_0\delta_2 * (\Gamma\delta_0 + \delta_1) +$$
$$+ \frac{-10 - B}{64}\delta_0^1 * (\Delta\delta_0 + E\delta_1 + \delta_2) +$$
$$+ \mathbf{\Xi}\delta_0\delta_1 * (Z\delta_0 + \delta_1) + \mathbf{\Upsilon}\delta_0^2 * (J\delta_0 + \delta_1) + \delta_0 \cdot \tilde{R}$$

(for the shake of the appearance of the paper, we do not write the quantities $\mathbf{\Xi}, \mathbf{\Upsilon}$ or $\tilde{R}$ explicitly. The $\tilde{R}$, for instance, has more than 50 terms), where $A, B, \Gamma, \Delta, E, Z, J$ undetermined parameters and $\delta_0 \cdot \tilde{R}$ the remainder, factorized by the common factor $\delta_0$. For the $B_{nl}$ we have

$$Formal[B_{nl}] = \frac{\delta_0^2}{4} * (X\delta_0 + Y\delta_1 + \delta_2) +$$
$$\left(-1 - \frac{Y}{2}\right)\delta_0\delta_1 * (Z\delta_0 + \delta_1) +$$
$$+ \left(1 - \frac{Y^2}{4} + Z + \frac{YZ}{2}\right)\delta_0^2 * (W\delta_0 + \delta_1) +$$
$$\left(-\frac{X}{2} + Z + \frac{YZ}{2}\right)\delta_0\delta_2 +$$
$$+ \left(-\frac{X^2}{4} - W^2 + \frac{Y^2W^2}{4} - ZW^2 - \frac{YZW^2}{2}\right)\delta_0^2 +$$
$$+ \left(-\frac{XY}{2} + Z^2 + \frac{YZ^2}{2} - 2W + \right.$$
$$\left. \frac{Y^2W}{2} - 2ZW - XZW\right)\delta_0\delta_1 + \delta_0 \cdot \tilde{R}$$

We can check that for the set of values $A = 16, B = -10, X = 1, Y = 1, Z = 1, W = 1$ we have $MV \neq \emptyset$ and $\Phi = \delta_0 - \frac{1}{2}\delta_1$. Thus, a class of the feedback laws upon request is $S = \frac{3}{20}\delta_1 - Q * (\delta_0 - \frac{1}{2}\delta_1)$, $Q$ an arbitrary liner polynomial. We got the quantity $\frac{3}{20}\delta_1$ by solving the Diophantine equation $R{*}(\delta_0-\frac{1}{2}\delta_1)+S{*}(\delta_1-3\delta_2) = \delta_0-2\delta_1$. Obviously this is not the only set of values for the feedback $S$. Any different choice of values for the parameters we lead to another set.

## REFERENCES


[1] K.J.Aström-B.Wittenmark: "Computer Controlled Systems " , Prentice Hall, 1984.

[2] Jacob Hammer: "Stabilization of non-linear systems" Int.J.Control,1986 Vol 44 ,No 5 ,1349 - 1381.

[3] S.T.Glad: "Nonlinear regulators and Ritt's remainder algorithm." In Colloque International Sur L'Analyse des Systemes Dynamiques Controles, 1990.

[4] S.Kotsios: " A New Factorization of Special Nonlinear Discrete Systems and its Applications." IEEE Trans. on Autom. Control, Vol 45, No 1, pp 24-33, (2000).

[5] S.Kotsios: " A Special Factorization of Nonlinear Discrete Input - Output Systems, The Model Matching Problems and Other Applications." IEEE Trans. on Autom. Control, under reviewing.

[6] St.Kotsios and N.Kalouptsidis (1993). "The model matching problem for a certain class of nonlinear systems".*Int.J.Control, Vol.57, NO. 4,881- 919.*

[7] G.Goodwin-Kwai Sang Sin: "Adaptive Filtering Prediction and Control" , Prentice Hall, 1984.

[8] Rugh J. W. (1981). "Non-linear system theory." *The John Hopkins University Press, Baltimore.*

[9] St.Kotsios. (2000) " An Application of Ritt's Remainder Algorithm to Discrete Polynomial Control Systems". IMA Journal of Mathematical Control and Information, **18**, 19-29, (2001).

[10] S.Kotsios, (2002) "On Detecting Solutions of Polynomial Nonlinear Difference Equations" *Journal of Difference Equations and Applications* Vol. 8 (6), pp. 551-571.

[11] Di Benedetto M.D., A.Isidori (1986): " The Matching of Nonlinear Models via Dynamic State Feedback." SIAM J. Contr. Optimiz. **24**, pp 1063-1075.