Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

of the 18$^{\text{th}}$ International Conference on Process Control

Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011

ISBN 978-80-227-3517-9

http://www.kirp.chtf.stuba.sk/pc11

Editors: M. Fikar and M. Kvasnica

Belikov, J., Kotta, Ü., Leibak, A.: Transfer Matrix and Its Jacobson Form for Nonlinear Systems on Time Scales: Mathematica Implementation, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 141–146, 2011.

Full paper online: http://www.kirp.chtf.stuba.sk/pc11/data/abstracts/019.html

Transfer matrix and its Jacobson form for nonlinear systems on time scales: *Mathematica* implementation

J. Belikov * **Ü. Kotta** * **A. Leibak** **

* *Institute of Cybernetics, Tallinn University of Technology, Akadeemia tee 21, 12618, Tallinn, Estonia*
fax : +372 620 4151 and e-mails : {jbelikov, kotta}@cc.ioc.ee
** *Department of Mathematics, Tallinn University of Technology, Ehitajate tee 5, 19086, Tallinn, Estonia*
fax : +372 620 3051 and e-mail : alar@staff.ttu.ee

Abstract: This paper suggests a detailed algorithm for computation of the Jacobson form of the polynomial matrix associated with the transfer matrix describing the multi-input multi-output nonlinear control system, defined on homogeneous time scale. The algorithm relies on the theory of skew polynomial rings.

Keywords: nonlinear control system, input-output models, time scales, symbolic computations.

1. INTRODUCTION

In the Institute of Cybernetics at Tallinn University of Technology symbolic software package NLControl has been developed over the years within *Mathematica* environment, for the detailed information see Kotta and Tõnso (2003), Tõnso et al. (2009). The package is based on different algebraic methods, in particular on the approach based on the differential one-forms, see Conte et al. (2007), and on the theory of skew polynomial ring. It allows to solve various modelling, analysis and synthesis problems not only for continuous and discrete-time nonlinear control systems, but also for those defined on homogeneous time scales, see Casagrande et al. (2010). Note that the key idea of a time scale calculus is unification of the theories of differential and difference equations, see Bohner and Peterson (2001). Both continuous and discrete-time (in terms of the difference operator) cases are merged in time scale formalism into a general framework which provides not only unification but also an extension. The main concept of the time scale calculus is the so-called delta-derivative that is a generalization of both time-derivative and the difference operator (but accommodates more possibilities, e.g. q-difference operator).

In the linear control theory the transfer matrix (TM) approach has been very popular. Recently the concept of the TM has been extended for the continuous-time nonlinear control systems Halás (2008) and later in Halás and Kotta (2007a) for discrete-time systems and for control system defined in terms of the pseudo-linear operator, see Halás and Kotta (2007b). Note that the latter includes also the systems, defined on homogeneous time scales, since in that case the delta-derivative may be understood as the special case of the pseudo-linear operator. However, the pseudo-linear approach cannot handle the systems defined on non-homogeneous time scales, since the time scale formalism unifies both continuous- and discrete-time cases, it would be interesting to study whether TM-based transparent control methods can be extended to nonlinear systems defined on time scale. In TM-based control design, a special form of the matrix, the Jacobson-Teichmüller [1] form, plays a key role. The first step in transformation of the TM into the required form is to transform the polynomial matrix, associated with it, into the Jacobson form.

Note that in the case of nonlinear control systems, the polynomials belong into the non-commutative polynomial ring that is the principal ideal domain (p.i.d.). The basic algorithm to transform a polynomial matrix into this form was given in Cohn (1985) for an arbitrary ring being the p.i.d. There exist a number of implementations of this algorithm such as Blinkov et al. (2003), Chyzak et al. (2007) and its fraction-free version Levandovskyy and Schindelar (2010). However, except Insua and Ladra (2006), not available for public use, all of them have been implemented either in Maple, e.g. Blinkov et al. (2003), Chyzak et al. (2007) or in Singular:Plural Levandovskyy and Schindelar (2010). Moreover, it is not documented whether and how these packages are applicable for nonlinear control systems, in particular for those defined on homogeneous time scale.

The main contribution of the paper is the specification the algorithm given in Cohn (1985) into the form necessary to handle the nonlinear control system defined on homogeneous time scale and description of the experience from its implementation in *Mathematica*, within the package NLControl. All the steps of the algorithm are clear, strictly defined and easily convertible into any programming code. It should be mentioned that some preliminary results for the discrete-time case were presented in Belikov et al. (2010).

2. CALCULUS ON TIME SCALE

For a general introduction to the calculus on time scales, see Bohner and Peterson (2001). Here we give only those notions and facts that we need in our paper.

[1] Note that in the linear control theory this form is called the Smith-McMillan form, see Ito et al. (2003).

A *time scale* $\mathbb{T}$ is an arbitrary nonempty closed subset of the set $\mathbb{R}$ of real numbers. The standard cases include $\mathbb{T}=\mathbb{R}$, $\mathbb{T}=\mathbb{Z}$, $\mathbb{T}=h\mathbb{Z}$ for $h>0$, but also $\mathbb{T}=q^{\mathbb{Z}}:=\{q^k : k\in\mathbb{Z}\}\cup\{0\}, q>1$ is a time scale.

The following operators on $\mathbb{T}$ are often used:

- the *forward jump operator* $\sigma:\mathbb{T}\to\mathbb{T}$, defined by $\sigma(t):=\inf\{\tau\in\mathbb{T}:\tau>t\}$ and $\sigma(\sup\mathbb{T})=\sup\mathbb{T}$, if $\sup\mathbb{T}\in\mathbb{T}$,
- the *backward jump operator* $\rho:\mathbb{T}\to\mathbb{T}$, defined by $\rho(t):=\inf\{\tau\in\mathbb{T}:\tau<t\}$ and $\rho(\inf\mathbb{T})=\inf\mathbb{T}$, if $\inf\mathbb{T}\in\mathbb{T}$,
- the *graininess* function $\mu:\mathbb{T}\to[0,\infty)$, defined by $\mu(t):=\sigma(t)-t$.

If $\mu\equiv\text{const}$ then a time scale $\mathbb{T}$ is called *homogeneous*. In this paper we assume that the time scale $\mathbb{T}$ is homogeneous.

Example 1:

- If $\mathbb{T}=\mathbb{R}$, then for any $t\in\mathbb{R}$, $\sigma(t)=t=\rho(t)$, and the graininess function $\mu(t)\equiv 0$.
- If $\mathbb{T}=h\mathbb{Z}$, for $h>0$, then for every $t\in h\mathbb{Z}$, $\sigma(t)=t+h$, $\rho(t)=t-h$, and $\mu(t)=h$.
- If $\mathbb{T}=\overline{q^{\mathbb{Z}}}$, for $q>1$, then for every $t\in\mathbb{T}$, $\sigma(t)=qt$, $\rho(t)=\frac{t}{q}$, and $\mu(t)=(q-1)t$.

So, the first two cases are homogeneous time scales whereas the third is not.

Definition 1. The *delta derivative* of a function $f:\mathbb{T}\to\mathbb{R}$ at t is the number $f^{\Delta}(t)$ such that for each $\varepsilon>0$ there exists a neighborhood $U(\varepsilon)$ of t, $U(\varepsilon)\subset\mathbb{T}$ such that for all $\tau\in U(\varepsilon)$, $|f(\sigma(t))-f(\tau)-f^{\Delta}(t)(\sigma(t)-\tau)|\le\varepsilon|\sigma(t)-\tau|$.

The typical special cases of the delta operator are summarized in the following remark.

Remark 1. (i) If $\mathbb{T}=\mathbb{R}$, then $f:\mathbb{R}\to\mathbb{R}$ is delta differentiable at $t\in\mathbb{R}$ if and only if $f^{\Delta}(t)=\lim\limits_{s\to t}\frac{f(t)-f(s)}{t-s}=f'(t)$, i.e. iff f is differentiable in the ordinary sense at t.
(ii) If $\mathbb{T}=T\mathbb{Z}$, where $T>0$, then $f:T\mathbb{Z}\to\mathbb{R}$ is always delta differentiable at every $t\in T\mathbb{Z}$ with $f^{\Delta}(t)=\frac{f(\sigma(t))-f(t)}{\mu(t)}=\frac{f(t+T)-f(t)}{T}$ meaning the usual forward difference operator.

Proposition 1. Let $f:\mathbb{T}\to\mathbb{R}$, $g:\mathbb{T}\to\mathbb{R}$ be two delta differentiable functions defined on $\mathbb{T}$ and let $t\in\mathbb{T}$. Then the delta derivative satisfies the following properties

(i) $f^{\sigma}=f+\mu f^{\Delta}$,
(ii) $(\alpha f+\beta g)^{\Delta}=\alpha f^{\Delta}+\beta g^{\Delta}$, for any constants α and β,
(iii) $(fg)^{\Delta}=f^{\sigma}g^{\Delta}+f^{\Delta}g$,
(iv) if $gg^{\sigma}\neq 0$, then $(f/g)^{\Delta}=(f^{\Delta}g-fg^{\Delta})/(gg^{\sigma})$.

For a function $f:\mathbb{T}\to\mathbb{R}$ we define second delta derivative $f^{[2]}:=f^{\Delta\Delta}$ provided that f^{Δ} is delta differentiable on $\mathbb{T}$. Similarly we define higher order derivatives $f^{[n]}$.

Denote $\sigma^n:=\underbrace{\sigma\circ\cdots\circ\sigma}_{n-\text{times}}$ and $f^{\sigma^n}:=f\circ\sigma^n$

Proposition 2. (Kotta et al. (2009)). Let f and f^{Δ} be delta differentiable functions on homogeneous time scale $\mathbb{T}$. Then

(i) $f^{\Delta\sigma}=f^{\sigma\Delta}$,
(ii) $f^{\sigma^n}=\sum_{k=0}^{n}C_n^k\mu^k f^{[k]}$.

At the end of this section we introduce some notation that will be useful in the following sections. Let f be a function admitting the delta-derivatives up to the c-th order. Let a and b be integers such that $0\le a<b\le c$. We set $f^{[0]}=f$. Let $f^{[a\ldots b]}$ denote the set $\{f^{[a]},\ldots,f^{[b]}\}$.

3. PRELIMINARIES

Consider a multi-input multi-output nonlinear control system described by a set of higher order input-output delta-differential equations on the homogeneous time scale $\mathbb{T}$ relating the inputs u_j, $j=1\ldots,m$, the outputs y_i, $i=1,\ldots,p$ and the finite number of their delta derivatives:

$$y_i^{[n_i]}=\Phi_i\Big(y_1^{[0\ldots n_{i1}-1]},\ldots,y_p^{[0\ldots n_{ip}-1]},\; u_1^{[0\ldots s_{i1}]},\ldots,u_m^{[0\ldots s_{pm}]}\Big), \quad (1)$$

where the functions Φ_i are real analytic functions of their arguments, and functions $y_i:\mathbb{T}\to\mathbb{R}$, $i=1,\ldots,p$ and $u_j:\mathbb{T}\to\mathbb{R}$, $j=1,\ldots,m$ are delta differentiable at least up to order n_i and $s_j:=\max_{1\le i\le p}(s_{ij})$, respectively.

3.1 Algebraic framework

Below we briefly recall the algebraic formalism for nonlinear control systems defined on homogeneous time scales, described in Bartosiewicz et al. (2007), Kotta et al. (2009), Kotta et al. (2011). Let $\mathscr{K}$ denote the field of meromorphic functions in a finite number of (independent) variables

$$\mathscr{C}=\{y_1^{[0\ldots n_1-1]},\ldots,y_p^{[0\ldots n_p-1]},u_j^{[k]},j=1,\ldots,m,k\ge 0\}.$$

Note that under the mild assumption on submersivity of system (1) (see below) the jump operator $\sigma:\mathscr{K}\to\mathscr{K}$ and the delta derivative $\Delta:\mathscr{K}\to\mathscr{K}$ may be extended to the field $\mathscr{K}$ as follows, see Kotta et al. (2011)

$$\begin{aligned}&\sigma(F)\left(y_1^{[0\ldots n_1-1]},\ldots,y_p^{[0\ldots n_p-1]},u_1^{[0\ldots s_1+1]},\ldots,u_m^{[0\ldots s_m+1]}\right)\\&:=F\left(y_1^{[0\ldots n_1-1]\sigma},\ldots,y_p^{[0\ldots n_p-1]\sigma},u_1^{[0\ldots s_1]\sigma},\ldots,u_m^{[0\ldots s_m]\sigma}\right),\end{aligned}$$

where

$$\begin{aligned}y_i^{[0\ldots n_i-1]\sigma}&=y_i^{[0\ldots n_i-1]}+\mu\cdot\Big[y_i^{[1\ldots n_i-1]},\Phi_i\Big(y_1^{[0\ldots n_{i1}]},\ldots,\\&\qquad y_i^{[0\ldots n_i-1]},\ldots,y_p^{[0\ldots n_{ip}]},u_1^{[0\ldots s_{p1}]},\ldots,u_m^{[0\ldots s_{pm}]}\Big)\Big],\end{aligned}$$

$i=1,\ldots,p$, $u_j^{[0\ldots s_j]\sigma}=u_j^{[0\ldots s_j]}+\mu u_j^{[1\ldots s_j+1]}$, $j=1,\ldots,m$ and [2]

$$\begin{aligned}&\Delta(F)\Big(y_i,\ldots,y_p^{[n_i-1]},u_j,\ldots,u_j^{[k+1]}\Big):=\\&\int_0^1\{\text{grad }F\Big(y_i+h\mu y_i^{\Delta},\ldots,y_i^{[n_i-1]}+\\&h\mu\Phi_i\Big(y_1^{[0\ldots n_{i1}-1]},\ldots,y_p^{[0..n_{ip}-1]},u_1^{[0\ldots s_{i1}]},\ldots,u_m^{[0\ldots s_{im}]}\Big),\\&u_j+h\mu u_j^{\Delta},\ldots,u_j^{[k]}+h\mu u_j^{[k+1]}\Big)\cdot\\&\begin{bmatrix}\Big(y_1^{[1\ldots n_1-1]},\ldots,y_p^{[1\ldots n_p-1]}\Big)^T,\\ \Phi_i\Big(y_1^{[0\ldots n_{i1}-1]},\ldots,y_p^{[0\ldots n_{ip}-1]},\\ \qquad u_1^{[0\ldots s_{i1}]},\ldots,u_m^{[0\ldots s_{im}]}\Big),\\ \Big(u_1^{[1\ldots s_1+1]},\ldots,u_m^{[1\ldots s_m+1]}\Big)^T\end{bmatrix}\}\mathrm{d}h.\end{aligned}$$

Notice that we will use $\sigma(F)$ and F^{σ} to denote the action of σ on F. Similarly, both $\Delta(F)$ and F^{Δ} will be used interchangeably.

[2] Proposition 3.3 from Bartosiewicz et al. (2007) shows how $\Delta(F)$ may be calculated not using integral explicitly.

In case σ is not injective, there may exist non-zero functions ϕ such that $\sigma(\phi) = 0$ meaning that the operator σ is not well-defined on the field $\mathscr{K}$. For σ to be an injective endomorphism on $\mathscr{K}$, the system (1) has to be submersive which can be guaranteed by the condition of the following theorem.

Theorem 1. (Kotta et al. (2011)). The nonlinear control system, defined on homogeneous time scale via the higher order i/o equations (1), is submersive if and only if the following condition

$$\operatorname{rank}_{\mathscr{K}} \begin{pmatrix} 1+\alpha_{11} & \dots & \alpha_{1p} & \beta_{11} & \dots & \beta_{1m} \\ \alpha_{p1} & \dots & 1+\alpha_{pp} & \beta_{p1} & \dots & \beta_{pm} \end{pmatrix} = p \tag{2}$$

holds, where

$$\alpha_{ij} := \sum_{k=0}^{n_j-1} (-1)^{n_j-k-1} \mu^{n_j-k} \frac{\partial \Phi_i}{\partial y_j^{[k]}},$$

$i, j = 1, \dots, p$ and

$$\beta_{lk} := \sum_{j=0}^{s} (-1)^{s-j+1} \mu^{s-j+2} \frac{\partial \Phi_l}{\partial u_k^{[j]}},$$

$l = 1, \dots, p, k = 1, \dots, m.$

The operator Δ satisfies a generalization of Leibnitz rule

$$(FG)^\Delta = F^\sigma G^\Delta + F^\Delta G, \tag{3}$$

for $F, G \in \mathscr{K}$. The derivation satisfying rule (3) is called a "σ-*derivation*", see Cohn (1985). Therefore, $\mathscr{K}$ is a differential field equipped with a σ-derivation Δ. In general, the field $\mathscr{K}$ is not inversive, i.e. not every element of $\mathscr{K}$ has a pre-image. Nevertheless, since Δ is injective, up to an isomorphism there exists an inversive σ-differential overfield $\mathscr{K}^*$, called the *inversive closure* of $\mathscr{K}$, such that Δ can be extended to $\mathscr{K}^*$ and this extension is automorphism of $\mathscr{K}^*$, see Cohn (1985). In Bartosiewicz et al. (2007) the details of construction of $\mathscr{K}^*$ for nonlinear control systems defined on time scales can be found. Below assume that $\mathscr{K}^*$ is given and use the same symbol $\mathscr{K}$ to denote the σ-differential field and its inversive closure.

Over the σ-differential field $\mathscr{K}$ one can define the vector space

$$\mathscr{E} := \operatorname{span}_{\mathscr{K}} \{ \mathrm{d}y_i, \mathrm{d}y_i^\Delta, \dots, \mathrm{d}y_i^{[n_i-1]}, i = 1, \dots, p, \\ \mathrm{d}u_j^{[k]}, j = 1, \dots, m, k \geq 0 \}.$$

The elements of $\mathscr{E}$ are called one-forms. For $F \in \mathscr{K}$ we define the operator $\mathrm{d} : \mathscr{K} \to \mathscr{E}$ as follows

$$\mathrm{d}F := \sum_{i=1}^{p} \sum_{l=0}^{n_i-1} \frac{\partial F}{\partial y_i^{[l]}} \mathrm{d}y_i^{[l]} + \sum_{j=1}^{m} \sum_{\ell=0}^{k} \frac{\partial F}{\partial u_j^{[\ell]}} \mathrm{d}u_j^{[\ell]}.$$

$\mathrm{d}F$ is said to be the (*total*) *differential* of the function F and is a one-form.

Let $\omega = \sum_j \alpha_j \mathrm{d}\varphi_j$ be a one-form with $\alpha_j \in \mathscr{K}$ and $\varphi_j \in \mathscr{C}$. Then, the operators $\sigma : \mathscr{K} \to \mathscr{K}$ and $\Delta : \mathscr{K} \to \mathscr{K}$ induce the operators $\sigma : \mathscr{E} \to \mathscr{E}$ and $\Delta : \mathscr{E} \to \mathscr{E}$ by

$$\sigma(\omega) := \sum_i \sigma(\alpha_i) \mathrm{d}\left[\sigma(\zeta_i)\right], \tag{4}$$

$$\Delta(\omega) := \sum_i \left\{ \Delta(\alpha_i) \mathrm{d}\varphi_i + \sigma(\alpha_i) \mathrm{d}\left[\Delta(\varphi_i)\right] \right\}, \tag{5}$$

Since $\sigma(\alpha_i) = \alpha_i + \mu\Delta(\alpha_i)$, (5) may be alternatively written as

$$\Delta(\omega) = \sum_i \left\{ \Delta(\alpha_i) \mathrm{d}\varphi_i + (\alpha_i + \mu\Delta(\alpha_i)) \mathrm{d}\left[\Delta(\varphi_i)\right] \right\}.$$

It has been proved that $\Delta(\mathrm{d}F) = \mathrm{d}[F^\Delta]$, $\sigma(\mathrm{d}F) = \mathrm{d}[F^\sigma]$ and $\Delta\sigma = \sigma\Delta$, see Bartosiewicz et al. (2007).

3.2 *Polynomial framework*

Next, we recall the polynomial formalism which allows to represent the nonlinear i/o equations (1) via two polynomial matrices. Consider the differential field $\mathscr{K}$ with the σ-derivation Δ with σ being an automorphism of $\mathscr{K}$. A *left* differential polynomial is an element which can be uniquely written in the form $a(\partial) = \sum_{i=0}^{n} a_i \partial^{n-i}$, $a_i \in \mathscr{K}$, where ∂ is a formal variable and $a(\partial) \neq 0$ if and only if at least one of the coefficients a_i, $i = 0, \dots, n$ is nonzero. If $a_0 \not\equiv 0$, then the positive integer n is called the degree of the left polynomial $a(\partial)$, denoted by $\deg a(\partial)$. Besides that we set $\deg 0 = -\infty$. The addition of the left differential polynomials is defined in the standard way. However, for $a \in \mathscr{K}$ the multiplication is defined by

$$\partial \cdot a := a^\sigma \partial + a^\Delta. \tag{6}$$

The ring of differential polynomials will be denoted by $\mathscr{K}[\partial; \sigma, \Delta]$. Since σ is an automorphism, the ring of the left differential polynomials is a skew polynomial ring, that is proved to satisfy the left Ore condition, see Farb and Dennis (1993). By left Ore condition for all nonzero $a, b \in \mathscr{K}[\partial; \sigma, \Delta]$ there exist nonzero $a_1, b_1 \in \mathscr{K}[\partial; \sigma, \Delta]$ such that $a_1 b = b_1 a$, that is, a and b have a common left multiple (clm). The ring $\mathscr{K}[\partial; \sigma, \Delta]$ can, therefore, be embedded into its quotient field (field of fractions) by defining its left quotients as $\frac{a}{b} = b^{-1} \cdot a$, see Ore (1933). Denote the resulting quotient field by $\mathscr{K}(\partial; \sigma, \Delta)$. Moreover, we write $\mathscr{K}(\partial; \sigma, \Delta)^{p \times m}$ for the set of $p \times m$ rational matrices with entries in $\mathscr{K}(\partial; \sigma, \Delta)$, and $\mathscr{K}[\partial; \sigma, \Delta]^{p \times m}$ for the set of $p \times m$ polynomial matrices with entries in $\mathscr{K}[\partial; \sigma, \Delta]$.

Let $\sigma^n := \underbrace{\sigma \circ \dots \circ \sigma}_{n-\text{times}}$ and denote $\sigma^n(a)$ by a^{σ^n} for $a \in \mathscr{K}$.

Lemma 1. (Kotta et al. (2009)). Let $a \in \mathscr{K}$. Then $\partial^n \cdot a \in \mathscr{K}[\partial; \sigma, \Delta]$, for $n \geq 0$, and $\partial^n \cdot a = \sum_{i=0}^{n} C_n^i \left(a^{[n-i]}\right)^{\sigma^i} \partial^i$.

In order to describe the i/o equation (1) via two polynomial matrices, we define

$$\partial^k \mathrm{d}y_\nu := \mathrm{d}y_\nu^{(k)}, \quad \partial^l \mathrm{d}u_\upsilon := \mathrm{d}u_\upsilon^{(l)} \tag{7}$$

for $\nu = 1, \dots, p$, $\upsilon = 1, \dots, m$ and $k, l \geq 0$ in the vector space $\mathscr{E}$. Since an arbitrary one-form $\omega \in \mathscr{E}$ has the form $\omega = \sum_{\nu=1}^{p} \sum_{i=0}^{n-1} a_{\nu i} \mathrm{d}y_\nu^{(i)} + \sum_{\upsilon=1}^{m} \sum_{j=0}^{k} b_{\upsilon j} \mathrm{d}u_\upsilon^{(j)}$, where $a_{\nu i}, b_{\upsilon j} \in \mathscr{K}$, so ω can be expressed in terms of the left differential polynomials as $\omega = \sum_{\nu=1}^{p} \left(\sum_{i=0}^{n-1} a_{\nu i} \partial^i \right) \mathrm{d}y_\nu + \sum_{\upsilon=1}^{m} \left(\sum_{j=0}^{k} b_{\upsilon j} \partial^j \right) \mathrm{d}u_\upsilon$. A left differential polynomial can be considered as an operator acting on vectors $y = [y_1, \dots, y_p]^T$ and $u = [u_1, \dots, u_m]^T$ from $\mathscr{E}$: $\left(\sum_{i=0}^{k} a_i \partial^i \right) (\alpha \mathrm{d}\zeta) := \sum_{i=0}^{k} a_i (\partial^i \cdot \alpha) \mathrm{d}\zeta$, with $a_i, \alpha \in \mathscr{K}$ and $\mathrm{d}\zeta \in \{\mathrm{d}y, \mathrm{d}u\}$, where by Lemma 1, $\partial^i \cdot \alpha = \sum_{k=0}^{i} C_i^k \left(a^{[n-i]}\right)^{\sigma^i} \partial^k$. It is easy to note that $\partial(\omega) = \Delta(\omega)$, for $\omega \in \mathscr{E}$.

Now, by differentiating equation (1) and using (7) we get

$$P(\partial)\mathrm{d}y = Q(\partial)\mathrm{d}u, \tag{8}$$

where $P(\partial) \in \mathscr{K}[\partial; \sigma, \Delta]^{p \times p}$ and $Q(\partial) \in \mathscr{K}[\partial; \sigma, \Delta]^{p \times m}$.

We assume that the Dieudonné determinant of the matrix $P(\partial)$ in (8) is nonzero, see Artin (1957) for details. The latter means that the following definition of the transfer matrix is well-defined.

Definition 2. An element of the form $H(\partial) := P^{-1}(\partial)Q(\partial) \in \mathscr{K}(\partial;\sigma,\Delta)^{p\times m}$, such that $\mathrm{d}y = H(\partial)\mathrm{d}u$, is said to be a transfer matrix of nonlinear system [3] (1).

Note that though every control system can be described by the transfer matrix, the converse is not always true. The reason is that the one-form corresponding to the transfer function may not be integrable, see Halás and Kotta (2007a) for details.

3.3 Polynomial matrices

Here we recall some basic properties of matrices with skew-polynomial entries. Suppose the matrix $P(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$.

Definition 3. An elementary column (row) operation on a polynomial matrix $P(\partial)$ is one of the following four operations

- (i) interchanging two columns (rows);
- (ii) multiplying any column (row) by invertible element $k \in \mathscr{K}[\partial;\sigma,\Delta]$ from the right (left);
- (iii) adding a right (left) multiple of one column (row) to another;
- (iv) replacement of the first elements of any two columns (rows) by their greatest common left (right) divisor (gcl(r)d) and zero, respectively.

All these operations correspond to multiplication of the matrix $P(\partial)$ by an elementary matrix $E_R^s(\partial)$ or $E_L^s(\partial)$ from the right or left, respectively Cohn (1985), where $s \in \{(i)-(iv)\}$. Operations (i)-(iii) may be represented by the product of the matrices of the form $E_{ij}(\partial) = I_\nu + 1_{ij}k$ with I_ν identity matrix and 1_{ij} the matrix made of a single 1 at the intersection of row i and column j, $1 \le i,j \le \nu$, and zeros elsewhere, with $k \in \mathscr{K}[\partial;\sigma,\Delta]$, and with $\nu = m$ for actions with columns and $\nu = p$ for actions with rows, see Lévine (2009). The elementary matrices corresponding to the operations from Definition 3 can be obtained

- (i) by swapping columns (rows) i and j of the identity matrix;
- (ii) by multiplying all elements of the corresponding column (row) of identity matrix by $k \in \mathscr{K}[\partial;\sigma,\Delta]$;
- (iii) from identity matrix with element $k \in \mathscr{K}[\partial;\sigma,\Delta]$ in entry (i,j).
- (iv) The procedure for constructing this matrix is described in the algorithm presented in Section 4.

Definition 4. A matrix $U(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{q\times q}$ is called unimodular if it has an inverse $U^{-1}(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{q\times q}$.

Every right or left unimodular matrix $U_R(\partial)$ or $U_L(\partial)$ may be obtained as a product of the corresponding elementary matrices from Definition 3.

In order to find the gcld, one may use the *left Euclidean division algorithm*, see Bronstein and Petkovšek (1996). For given two polynomials $p_1(\partial)$ and $p_2(\partial)$ with $\deg(p_1(\partial)) > \deg(p_2(\partial))$ there exist unique polynomials $\gamma_1(\partial)$ and $p_3(\partial)$ such that

$$p_1(\partial) = p_2(\partial)\gamma_1(\partial) + p_3(\partial), \quad \deg(p_3(\partial)) < \deg(p_2(\partial)).$$

Using the left Euclidean division algorithm, after $k-1$ steps, one obtains $p_i(\partial) = p_{i+1}(\partial)\gamma_i(\partial) + p_{i+2}(\partial)$ for $i = 2,\dots,k-2$ and $p_{k-1}(\partial) = p_k(\partial)\gamma_{k-1}(\partial)$. Hence the gcld of $p_1(\partial)$ and $p_2(\partial)$ is $p_k(\partial)$. Moreover, eliminating polynomials $p_{k-1}(\partial),\dots,p_3(\partial)$ we get the *Bézout identity*, i.e. there exist polynomials $u(\partial), v(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ such that $p_1(\partial)u(\partial) + p_2(\partial)v(\partial) = p_k(\partial)$. Note that the *right Euclidean division algorithm* can be defined in a similar manner.

[3] Notice that there exists an algorithm which allows to obtain the transfer matrix from a nonlinear system described by state-space equations, for additional information see Halás (2008).

4. JACOBSON FORM

For $P(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$ one can find elementary row and column operations corresponding to multiplication by unimodular matrices $U_L^{p\times p}(\partial)$ and $U_R^{m\times m}(\partial)$, respectively, such that

$$U_L(\partial)P(\partial)U_R(\partial) = \Lambda(\partial), \tag{9}$$

where $\Lambda(\partial) = \operatorname{diag}\{\lambda_1(\partial),\dots,\lambda_r(\partial),0,\dots,0\}$ and $\lambda_i(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ are unique monic polynomials obeying a property that $\lambda_{i+1}(\partial)$ is divisible by $\lambda_i(\partial)$, $\lambda_i(\partial) \parallel \lambda_{i+1}(\partial)$, i.e. there exist $\alpha_i(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ such that $\lambda_{i+1}(\partial) = \lambda_i(\partial)\cdot\alpha_i(\partial)$ for all $i = 1,\dots,r-1$. The matrix $\Lambda(\partial)$ is called the *Jacobson form* of $P(\partial)$, and $\lambda_i(\partial)$ are called the invariant polynomials of $P(\partial)$, see Nakayama (1938).

Suppose $H(\partial) \in \mathscr{K}(\partial;\sigma,\Delta)^{p\times m}$ is a transfer matrix whose entries are assumed to be in the irreducible form, i.e. without common left factors in the corresponding numerators and denominators, and write it in a standard form

$$H(\partial) = [q(\partial)]^{-1}P(\partial), \tag{10}$$

where the matrix $P(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$ is a polynomial matrix and $q(\partial)$ is the monic least common left multiple (lclm) of the denominators of all entries of $H(\partial)$. Then, $P(\partial) = q(\partial)H(\partial)$ is a polynomial matrix, that can be transformed into the Jacobson form as above.

4.1 The main Algorithm

The algorithm, presented below, allows to transform the matrix $P(\partial)$ into the Jacobson form. Consider the matrix

$$P(\partial) = \begin{pmatrix} p_{11}(\partial) & \cdots & p_{1m}(\partial) \\ \vdots & \ddots & \vdots \\ p_{p1}(\partial) & \cdots & p_{pm}(\partial) \end{pmatrix}$$

in the ring $\mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$.

Step 1. $k := 1$.

Step 2. Find $p_{ij}(\partial) \ne 0$ for $i = k,\dots,p$ and $j = k,\dots,m$ with the lowest degree and, using operation (i) from Definition 3, put it on the position (k,k).

Step 3. Using elementary column (item (a)) and row (item (b)) operation (iv) from Definition 3,

(a) replace the elements $p_{kk}(\partial)$ and $p_{kj}(\partial)$ for $j = k+1,\dots,m$ with their gcld and zero, respectively. This operation can be implemented by solving the following equations

$$p_{kk}(\partial)a_{kk}(\partial) + p_{kj}(\partial)c_{jk}(\partial) = e_{kk}(\partial), \tag{11}$$

$$p_{kk}(\partial)b_{kj}(\partial) + p_{kj}(\partial)d_{jj}(\partial) = 0 \tag{12}$$

with respect to $a_{kk}(\partial), b_{kj}(\partial), c_{jk}(\partial)$ and $d_{jj}(\partial)$, and multiplying $P(\partial)$ from the right by the elementary matrix $E_{Rkj}^4(\partial)$, which can be constructed as follows. Create $m\times m$ identity matrix and put the elements $a_{kk}(\partial)$, $b_{kj}(\partial)$, $c_{jk}(\partial)$ and $d_{jj}(\partial)$ on the positions (k,k), (k,j), (j,k) and (j,j), respectively. Making $(m-k)$ replacements specified above, we transform the matrix $P(\partial)$ into the new matrix with $p_{kk}(\partial) = e_{kk}(\partial)$, $p_{k,k+1}(\partial) = \cdots = p_{km}(\partial) = 0$ and some new elements $p_{il}(\partial)$ for $i = k+1, \ldots, p$ and $l = k, \ldots, m$ obtained after multiplication $P(\partial)$ by the respective matrix $U_{Rk}(\partial) = E^4_{Rk,k+1}(\partial) \cdot \ldots \cdot E^4_{Rkm}(\partial)$.

(b) replace the elements $p_{kk}(\partial)$ and $p_{ik}(\partial)$ for $i = k+1, \ldots, p$ with their gcrd and zero, respectively. The previous operation can be implemented by solving the following equations

$$a_{kk}(\partial)p_{kk}(\partial) + b_{ki}(\partial)p_{ik}(\partial) = e_{kk}(\partial), \tag{13}$$

$$c_{ik}(\partial)p_{kk}(\partial) + d_{ii}(\partial)p_{ik}(\partial) = 0 \tag{14}$$

with respect to $a_{kk}(\partial), b_{ki}(\partial), c_{ik}(\partial)$ and $d_{ii}(\partial)$, and multiplying $P(\partial)$ from the left by the elementary matrix $E^4_{Lik}(\partial)$, which can be constructed as follows. Create $p \times p$ identity matrix and put elements $a_{kk}(\partial)$, $b_{ki}(\partial)$, $c_{ik}(\partial)$ and $d_{ii}(\partial)$ on the positions (k,k), (k,i), (i,k) and (i,i), respectively. Making $(p-k)$ replacements specified above, we transform the matrix $P(\partial)$ into the new matrix with $p_{kk}(\partial) = e_{kk}(\partial)$, $p_{k+1,k}(\partial) = \cdots = p_{pk}(\partial) = 0$ and some new elements $p_{lj}(\partial)$ for $j = k+1, \ldots, m$ and $l = k, \ldots, p$ obtained after multiplication $P(\partial)$ by the respective matrix $U_{Lk}(\partial) = E^4_{Lk+1,k}(\partial) \cdot \ldots \cdot E^4_{Lpk}(\partial)$.

However, in the course of doing this, nonzero entries may reappear in the k-th row of the matrix $P(\partial)$, and one has then to repeat Step 3. Note that at each iteration the number of divisors of the element $p_{kk}(\partial)$ reduces, and therefore, in a finite number of steps the k-th row and column become zero. The latter means that after a finite number of consecutive steps one will obtain the matrix with $p_{kk}(\partial) = e_{kk}(\partial)$ and other entries in the k-th row and column equal to zero.

Step 4. If $k \neq \min(p,m) - 1$, then $k := k+1$ and go to **Step 2**, otherwise go to **Step 5**.

Step 5. If $p = m$, then go to **Step 6**, otherwise depending whether $m > p$ or $m < p$, one has to execute additional $(m-k)$ or $(p-k)$ operations over the last column(s) or row(s) described in **Steps 3(a)** or **3(b)**, respectively.

Step 6. Consider the elements of the main diagonal $p_{ii}(\partial), \ldots, p_{kk}(\partial)$. Here, the following two cases are possible:

(a) If the divisibility property holds for all pairs $p_{ii}(\partial) \parallel p_{jj}(\partial)$ for $1 \leq i < j \leq k$, then go to **Step 7**.

(b) If the divisibility property does not hold for some pair of elements $p_{ii}(\partial)$ and $p_{jj}(\partial)$ with $1 \leq i < j \leq k$, i.e. $p_{ii}(\partial) \nparallel p_{jj}(\partial)$, then, using row operation (iii) from Definition 3, the matrix $P(\partial)$ has to be transformed into a new matrix with element $p_{jj}(\partial)$ on the position (i,j) by adding the j-th row to the i-th row. After that, execute again all **Steps 2-5** with modified matrix $P(\partial)$ and $k = i$. The main idea of this transformation and the subsequent executing of the steps 2-5 consists in replacing the element $p_{ii}(\partial)$ by $\text{gcld}(p_{ii}(\partial), p_{jj}(\partial))$ or $\text{gcrd}(p_{ii}(\partial), p_{jj}(\partial))$, respectively, obeying a division property $p_{ii}(\partial) \parallel p_{jj}(\partial)$.

Step 7. End of the Algorithm.

Remark 2. Equations (11) and (13) are Bézout identities and can be solved using the left and right Euclidean division algorithm, respectively. Besides, equations (12) and (14) are the right and left Ore conditions, respectively. For example, for (14) it means that there exist $c_{ik}(\partial), d_{ii}(\partial) \in \mathscr{K}[\partial; \sigma, \Delta]$ such that $c_{ik}(\partial)p_{kk}(\partial) = -d_{ii}(\partial)p_{ik}(\partial)$ holds.

We have implemented the algorithm for computing Jacobson form in *Mathematica* package NLControl. However, it should be mentioned that even for the very simple examples calculations become extremely complex. Note that in our calculations, we have to simplify the obtained expressions using the relations, defined by the system equations (1) as well as those, obtained from (1) by taking the delta derivatives. If not done, the computations may lead to erroneous result.

Example. Consider the system described by the input-output equations

$$\begin{aligned} y_1^{[2]} &= u_1(1 + y_1^\Delta) + u_1^\Delta(y_1 + \mu y_1^\Delta) - u_2 \\ y_2^\Delta &= u_1 y_2 - u_2 \end{aligned} \tag{15}$$

First, we compute, according to Definition 2 and using the property (i) from Proposition 1, the transfer matrix of the system (15)

$$H(\partial) = \begin{pmatrix} \dfrac{y_1^\sigma \partial + y_1^\Delta + 1}{\partial^2 - u_1^\sigma \partial - u_1^\Delta} & \dfrac{1}{-\partial^2 + u_1^\sigma \partial + u_1^\Delta} \\ \dfrac{y_2}{\partial - u_1} & \dfrac{1}{-\partial + u_1} \end{pmatrix}.$$

Since the lclm of all the denominators in $H(\partial)$ equals to $\partial^2 - u_1^\sigma \partial - u_1^\Delta$, multiplying numerators of the elements $h_{21}(\partial), h_{22}(\partial)$ by ∂ from the left, decomposition (10) for this example takes the form

$$H(\partial) = \left(\partial^2 - u_1^\sigma \partial - u_1^\Delta)\right)^{-1} \cdot \begin{pmatrix} y_1^\sigma \partial + y_1^\Delta + 1 & -1 \\ y_2^\sigma \partial + y_2^\Delta & -\partial \end{pmatrix}.$$

Obviously, the element $p_{12} = -1$ is that of the lowest possible degree of $P(\partial)$ and, after permuting the rows and columns, i.e. multiplying $P(\partial)$ by the corresponding elementary matrix [4]

$$E^1_{R12} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

from the right, we obtain

$$P(\partial) = \begin{pmatrix} -1 & y_1^\sigma \partial + y_1^\Delta + 1 \\ -\partial & y_2^\sigma \partial + y_2^\Delta \end{pmatrix}. \tag{16}$$

Next, one can easily check that $e_{11} := \text{gcrd}(p_{11}, p_{12}) = 1$. After solving equations (11) and (12), corresponding to this example, i.e. the equations

$$\begin{aligned} (-1) \cdot a_{11}(\partial) + (y_1^\sigma \partial + y_1^\Delta + 1) \cdot c_{21}(\partial) &= 1 \\ (-1) \cdot b_{12}(\partial) + (y_1^\sigma \partial + y_1^\Delta + 1) \cdot d_{22}(\partial) &= 0, \end{aligned}$$

we obtain $a_{11}(\partial) = -1$, $b_{12}(\partial) = y_1^\sigma \partial + y_1^\Delta + 1$, $c_{21}(\partial) = 0$, $d_{22}(\partial) = 1$. According to Step 3(a), we construct the matrix

$$E^4_{R12}(\partial) = \begin{pmatrix} -1 & y_1^\sigma \partial + y_1^\Delta + 1 \\ 0 & 1 \end{pmatrix}$$

and multiply (16) by it from the right to get

$$\begin{pmatrix} 1 & 0 \\ \partial & -y_1^{\sigma^2} \partial^2 - (2y_1^{\Delta\sigma} - y_2^\sigma + 1)\partial - y_1^{[2]} + y_2^\Delta \end{pmatrix}. \tag{17}$$

Again, one can check that $e_{11} := \text{gcld}(p_{11}, p_{21}) = 1$. Therefore, solving equations (13) and (14), i.e. the equations

$$\begin{aligned} a_{11}(\partial) \cdot 1 + b_{12}(\partial) \cdot \partial &= 1 \\ c_{21}(\partial) \cdot 1 + d_{22}(\partial) \cdot \partial &= 0, \end{aligned}$$

[4] In order not to mislead the reader, note that not all the operations listed in Definition 3 have been used in this example, but only those that correspond to the cases $s = 1$ and $s = 4$.

we obtain $a_{11}(\partial) = 1$, $b_{12}(\partial) = 0$, $c_{21}(\partial) = -\partial$, $d_{22}(\partial) = 1$. According to Step 3(b), we construct the matrix

$$E^4_{L21}(\partial) = \begin{pmatrix} 1 & 0 \\ -\partial & 1 \end{pmatrix}.$$

and multiply (17) by it from the left to obtain

$$\Lambda(\partial) = \begin{pmatrix} 1 & 0 \\ 0 & -y_1^{\sigma^2}\partial^2 - (2y_1^{\Delta\sigma} - y_2^{\sigma} + 1)\partial - y_1^{[2]} + y_2^{\Delta} \end{pmatrix}.$$

Due to the fact that the number of rows of $P(\partial)$ equals to the number of its columns, one can directly go to Step 6. Obviously, the division property $\lambda_1(\partial) \parallel \lambda_2(\partial)$ holds. Finally, decomposition (9) of $P(\partial)$ is

$$\Lambda(\partial) = E^4_{L21}(\partial)P(\partial)E^1_{R12}E^4_{R12}(\partial).$$

5. CONCLUSION

In this paper we have suggested a detailed algorithm for computation of the Jacobson form of the polynomial matrix associated with the transfer matrix describing the multi-input multi-output nonlinear control system, defined on homogeneous time scale, using the theory of skew polynomials. In addition, we adapted the algorithm given in Cohn (1985) for the case of the nonlinear control systems defined on homogeneous time scale. Notice that, using previous experience with *Mathematica* program, we implemented our results in NLControl package. However, the algorithm is presented in a form that can be easily implemented by any programming language.

ACKNOWLEDGMENTS

This work was partially supported by the Governmental funding project no. SF0140018s08 and Estonian Science Foundation Grants no. 8365 and 8787.

REFERENCES

E. Artin. *Geometric algebra*. Interscience publishers, New York, London, 1957.

Z. Bartosiewicz, Ü. Kotta, E. Pawłuszewicz, and M. Wyrwas. Algebraic formalism of differential one-forms for nonlinear control systems on time scales. *Proc. Estonian Acad. of Sci. Phys. Math.*, 56(3):264–282, 2007.

J. Belikov, Ü. Kotta, and A. Leibak. Transformation of the transfer matrix of the nonlinear system into the jacobson form. In *International Congress on Computer Applications and Computational Science*, pages 495–498, Singapore, December 2010.

Y.A. Blinkov, C.F. Cid, V.P. Gerdt, W. Plesken, and D. Robertz. The maple package „janet": 2. linear partial differential equations. In *Proc. of the 6th International Workshop on Computer Algebra in Scientific Computing*, pages 41–54, Passau, Germany, 2003.

M. Bohner and A. Peterson. *Dynamic Equations on Time Scales*. Birkhäuser, Boston, USA, 2001.

M. Bronstein and M. Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157(1):3–33, 1996.

D. Casagrande, Ü. Kotta, M. Tõnso, and M. Wyrwas. Mathematica application for nonlinear control systems on time scales. In *International Congress on Computer Applications and Computational Science*, pages 621–624, Singapore, December 2010.

F. Chyzak, A. Quadrat, and D. Robertz. Oremodules: A symbolic package for the study of multidimensional linear systems. In *Applications of Time Delay Systems*, Lecture Notes in Control and Information Sciences, pages 233–264. Springer Berlin / Heidelberg, 2007.

P.M. Cohn. *Free rings and their relations*. Academic Press, London, UK, 1985.

G. Conte, C.H. Moog, and A.M. Perdon. *Algebraic Mehtods for Nonlinear Control Systems*. Springer-Verlag, London, UK, 2007.

B. Farb and R.K. Dennis. *Noncommutative algebra*. Springer-Verlag, New York, USA, 1993.

M. Halás. An algebraic framework generalizing the concept of transfer functions to nonlinear systems. *Automatica*, 44(5): 1181–1190, 2008.

M. Halás and Ü. Kotta. Transfer functions of discrete-time nonlinear control systems. *Estonian Acad. Sci. Phys. Math.*, 56(4):322–335, 2007a.

M. Halás and Ü. Kotta. Pseudo-linear algebra: a powerful tool in unification of the study of nonlinear control systems. In *7th IFAC Symposium on Nonlinear Control Systems,*, pages 684–689, Pretoria, South Africa, 2007b.

M.A. Insua and M. Ladra. Smith normal form can be computed using gröbner bases. In *International Congress of Mathematicians*, Madrid, Spain, August 2006.

N. Ito, W. Schmale, and H.K. Wimmer. Computation of minimal state space realizations in jacobson normal form. In *Contemporary Mathematics*, pages 221–232. American Mathematical Society Boston, MA, USA, 2003.

Ü. Kotta and M. Tõnso. Linear algebraic tools for discrete-time nonlinear control systems with mathematica. In *Nonlinear and Adaptive Control*, Lecture Notes in Control and Information Sciences, pages 195–205. Springer Berlin / Heidelberg, 2003.

Ü. Kotta, Z. Bartosiewicz, E. Pawłuszewicz, and M. Wyrwas. Irreducibility, reduction and transfer equivalence of nonlinear input-output equations on homogeneous time scales. *Systems and Control Letters*, 58(9):646–651, 2009.

Ü. Kotta, B. Rehak, and M. Wyrwas. On submersivity assumption for nonlinear control systems on homogeneous time scales. *Proc. Estonian Acad. of Sci. Phys. Math.*, 2011. Accepted for publication.

V. Levandovskyy and K. Schindelar. Computing diagonal form and jacobson normal form of a matrix using gröbner bases. http://arxiv.org/abs/1003.3785, March 2010.

J. Lévine. *Analysis and Control of Nonlinear Systems*. Springer-Verlag, Berlin, Germany, 2009.

T. Nakayama. A note on the elementary divisor theory in non-commutative domains. *Bull. Amer. Math. Soc.*, 44(10):719–723, 1938.

O. Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34:480–508, 1933.

M. Tõnso, H. Rennik, and Ü. Kotta. Webmathematica-based tools for discrete-time nonlinear control systems. *Proc. Estonian Acad. of Sci. Phys. Math.*, 58(4):224–240, 2009.