Proc. Estonian Acad. Sci. Phys. Math., 2002, **51**, 4, 238–254

Removing or lowering the orders of input shifts in discrete-time generalized state-space systems with Mathematica

Ülle Kotta and Maris Tõnso

Institute of Cybernetics, Tallinn Technical University, Akadeemia tee 21, 12618 Tallinn, Estonia; kotta@cc.ioc.ee, mtonso@staff.ttu.ee

Received 3 September 2002

Abstract. The problem of lowering the orders of input forward shifts in generalized state-space systems by generalized state transformations is studied using the language of differential forms. Necessary and sufficient conditions are given for the local existence of such transformations. These conditions are formulated in terms of the integrability of certain subspaces of one-forms classified according to their relative degree. The sufficiency part of the proof gives a constructive procedure (up to finding the integrating factors and integration of the set of one-forms) for finding these generalized state transformations. In a particular case, these conditions show when it is possible to transform a generalized state-space representation into the classical state equations. A set of functions developed in Mathematica 4.0 is described, allowing us to test if the generalized state equation is transformable into the classical state-space form, and also to find the transformation for simple examples. The application of the developed functions is demonstrated on four examples obtained via identification.

Key words: nonlinear systems, discrete-time systems, generalized dynamics, generalized state transformations, linear algebraic approach, symbolic computation, Mathematica.

1. INTRODUCTION

It has become clear that in several control problems it is necessary to consider more general dynamics containing in addition to the input also a finite number of its time shifts. One well-known example is the inverse system which, in general, contains the shifts of its inputs. A theoretical study of discrete-time control systems

This paper is an extended version of two conference papers. The preliminary results of Sections 2 and 3 are in [1] and the material of Sections 4 and 5 is partly given in [2].

depending explicitly on input shifts and basing on difference algebra was started by Fliess [3–5]. According to Fliess [5], the generalized state equation of a discrete-time nonlinear control system is of the form

$$x(t+1) = f(x(t), u(t), \ldots, u(t+\alpha)). \tag{1}$$

Although the general description has been justified by theoretical studies, the classical state representation is still dominant in the control literature, since it permits application of the numerous existing control techniques. Therefore, it is natural to consider the problem of whether there exists a generalized state transformation depending also on the input and a finite number of its forward time shifts

$$\tilde{x}(t) = \psi(x(t), u(t), \ldots, u(t+\alpha-1)), \tag{2}$$

which brings (1) into the classical state-space form.

The main goal of this paper is to give necessary and sufficient conditions for the existence of such a transformation. We consider the situations in which this system theoretic property is generic, that is, it holds on open and dense subsets of a suitable domain of definition. Furthermore, we give a constructive procedure (up to finding the integrating factors and integration of the set of one-forms) to find such a transformation. Finally, if these conditions are not satisfied, we show that in many cases it is possible to lower the input shifts in the generalized state-space equations. We also present the corresponding conditions and transformations. Actually, we study our main problem as a subproblem of a more general problem: when is it possible to lower, via a transformation (2), the time shifts of inputs in (1)?

Our study was inspired by [6] which completely solved the problem in the differential geometric frame for continuous-time systems. It was mentioned that the discrete-time version of the problem was still open. Unlike [6], we have chosen to present our results within the linear algebraic framework. In [7] the linear algebraic approach, extended by Grizzle [8] to discrete-time systems, has been modified to end up with an inversive difference field $\mathcal{K}^*$. The $\mathcal{K}^*$-vector space $\mathcal{E}$, spanned by the formal differentials of the elements of $\mathcal{K}^*$ as well as certain nested sequences of subspaces of $\mathcal{E}$, play the major role in investigating the problem of removing or lowering input shifts in generalized state-space equations. Moreover, our conditions are not completely parallel to those of [6]. The differences will be discussed in Section 3 after presenting our main results. Note that the problem of lowering the input derivatives in the system description was first studied in [9].

Finally, let us mention that the related problem of the realization of the higher-order input-output difference equation in the state-space form has been studied in [10]. Given an input-output model

$$y(t+n) = \varphi(y(t), \ldots, y(t+n-1), u(t), \ldots, u(t+s)), \tag{3}$$

it is always possible to transform it into a generalized state equation. Specifically, this representation is obtained from (3) by taking $x(t)$ as the following state vector, involving past outputs $x^T(t) = (y(t), \ldots, y(t+n-1))$. However, the problem studied in this paper is more general in three aspects. First, the generalized state equation, written down on the basis of the input-output equation, has a very specific form for the first $n-1$ state components: $x_i(t+1) = x_{i+1}(t)$ for $i = 1, \ldots, n-1$. This paper is not restricted to this special case. Second, in [10] only the case of a single-input system is considered. Third, in [8] only the problem of removing all the input shifts is studied, and not the more general problem of lowering the input shifts as in this paper. Besides, [10] is focused on obtaining the accessible realization, which requires that the input-output model is in the irreducible form. We do not restrict ourselves to obtaining an accessible realization.

The procedures described in the paper involve many symbolic computations not easily done by hand calculation. Although various useful toolboxes and software packages for nonlinear control systems have been developed using computer algebra systems such as Maple and Mathematica, these symbolic computation systems have not been practically used for the analysis and synthesis of nonlinear discrete-time systems except in [2,11–14]. Our purpose is to contribute to the development of such tools.

A set of functions in the computer algebra system Mathematica 4.0 is described that allow one to decide whether the given nonlinear discrete-time generalized state equation is transformable into the classical state-space form, and if not, whether it is possible to lower the input shifts in (1). Moreover, for simple examples these functions also allow one to find the required transformations. The application of the developed functions is demonstrated on four examples obtained via identification.

The paper is organized as follows. In Section 2 we recall the basics of the (linear) algebraic theory for nonlinear discrete-time control systems. The main results are given in Section 3. The Mathematica functions are described in Section 4. Section 5 presents some examples on which the Mathematica functions are demonstrated. Some concluding remarks are given in Section 6.

2. LINEAR ALGEBRAIC FRAMEWORK

Consider the system Σ described by (1), where $x \in X \subset I\!R^n$ is the state variable, $u \in I\!R^m$ is the input variable, and $f = (f_1, \ldots, f_n)$ is a real analytic vector function defined on $\mathcal{X} \times I\!R^{(\alpha+1)m}$. Of course, it is not necessary that (1) contains for all input components forward time shifts up to the same order; α is just the maximal order of time shifts in these equations.

We will associate with the system Σ its extended state-space systems Σ_e with input $v(t) = u(t+\alpha+1)$, state $z(t) = [x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_1(t+\alpha), \ldots, u_m(t), \ldots, u_m(t+\alpha)]^T$ and the state transition map $f_e(z(t), v(t))$ defined as

$$\begin{aligned} z_k(t+1) &= f_k(z(t)), \\ z_{n+(j-1)(\alpha+1)+i}(t+1) &= z_{n+(j-1)(\alpha+1)+i+1}(t), \\ z_{n+j(\alpha+1)}(t+1) &= v_j(t) \end{aligned} \tag{4}$$

for $k = 1, \ldots, n$; $j = 1, \ldots, m$; $i = 1, \ldots, \alpha$. Note that the extended system (4) has more state variables than the extended system given in [6], in which to the ith input component correspond $\alpha_i + 1$ extended state components, where α_i is the highest derivative of the ith input in the generalized state equations. The system (4) will play a key role in the subsequent analysis and in the procedures for removing or lowering the orders of the input shifts.

We follow the notation of [7]. Let $\mathcal{K}$ denote the field of meromorphic functions in a finite number of the variables $\{z(0), v(t),\ t \geq 0\}$. The forward-shift operator $\delta : \mathcal{K} \to \mathcal{K}$ is defined by $\delta\zeta(z(t), v(t)) = \zeta(f_e(z(t), v(t)), v(t+1))$. For δ to be one-to-one, the extended system (4) has to be submersive, which will be guaranteed by

$$\operatorname{rank}\left(\partial f_e / \partial(z, v)\right) = n + (\alpha + 1)m. \tag{5}$$

Under (5) the pair $(\mathcal{K}, \delta)$ is a difference field and up to an isomorphism, there exists a unique difference field $(\mathcal{K}^*, \delta^*)$ called the *inversive closure* of $(\mathcal{K}, \delta)$ such that $\mathcal{K} \subset \mathcal{K}^*$, $\delta^* : \mathcal{K}^* \to \mathcal{K}^*$ is an automorphism and the restriction of δ^* to $\mathcal{K}$ equals δ. By abuse of notation, hereinafter we assume that $(\mathcal{K}^*, \delta^*)$ is given and use the same symbol to denote $(\mathcal{K}, \delta)$ and its inversive closure. Sometimes the abridged notations $\varphi^+(\cdot) = \delta\varphi(\cdot)$ and $\varphi^-(\cdot) = \delta^{-1}\varphi(\cdot)$ are used.

Over the field $\mathcal{K}$ one can define a difference vector space $\mathcal{E} := \operatorname{span}_{\mathcal{K}}\{\mathrm{d}\varphi \mid \varphi \in \mathcal{K}\}$. The operator δ induces a forward-shift operator $\Delta : \mathcal{E} \to \mathcal{E}$ by

$$\sum_i a_i \mathrm{d}\varphi_i \mapsto \sum_i (\delta a_i)\mathrm{d}(\delta\varphi_i),\ a_i, \varphi_i \in \mathcal{K}.$$

The relative degree r of a one-form $\omega \in \mathcal{E}$ is defined to be the least integer such that $\Delta^r\omega \notin \operatorname{span}_{\mathcal{K}}\{Dz\}$. If such an integer does not exist, we set $r = \infty$.

A sequence of subspaces $\{\mathcal{H}_k\}$ of $\mathcal{E}$ is defined by

$$\begin{aligned} \mathcal{H}_1 &= \operatorname{span}_{\mathcal{K}}\{\mathrm{d}z(0)\}, \\ \mathcal{H}_{k+1} &= \{\omega \in \mathcal{H}_k \mid \Delta\omega \in \mathcal{H}_k\},\ \ k \geq 1, \end{aligned} \tag{6}$$

and proved to be invariant under the state-space diffeomorphism. Obviously, $\mathcal{H}_k$ contains the one-forms whose relative degree is equal to or higher than k. It is clear that the sequence (6) is decreasing. Denote by k^* the least integer such that

$$\mathcal{H}_1 \supset \cdots \supset \mathcal{H}_{k^*} \supset \mathcal{H}_{k^*+1} = \mathcal{H}_{k^*+2} = \cdots =: \mathcal{H}_\infty. \tag{7}$$

Hereafter we make use of some standard properties of the set of one-forms [15].

Theorem 2.1 (Frobenius). [15] *Let $\mathcal{I} = \{\omega_1, \ldots \omega_s\}$ be a set of one-forms. Suppose that for $k = 1, \ldots, s$ the condition*

$$\mathrm{d}\omega_k \wedge \omega_1 \wedge \ldots \wedge \omega_s = 0$$

is satisfied, where $\wedge$ is the exterior or wedge product of differential forms and d *is the exterior derivative of a one-form*; *then there exists locally a system of coordinates $\{\xi_1, \ldots, \xi_s\}$ such that $\mathcal{I}$ is generated by $\{\mathrm{d}\xi_1, \ldots, \mathrm{d}\xi_s\}$. In this case the set $\mathcal{I}$ is said to be completely integrable.*

3. MAIN RESULTS

The main goal of this section is to obtain the generic necessary and sufficient conditions for the system (1) to be transformable via the generalized state transformation (2) into the classical state-space form

$$\tilde{x}(t+1) = \tilde{f}(\tilde{x}(t), u(t)). \tag{8}$$

We will study our main problem as a subproblem of a more general problem: when is it possible to transform Σ, via a generalized state transformation (2), into the system

$$\tilde{x}(t+1) = g(\tilde{x}(t), u(t), \dots, u(t+\beta)), \tag{9}$$

where $\beta < \alpha$?

Observe that (2) can be interpreted as a (local) transformation of the extended state z, having the special structure $\tilde{z}^1 = \Psi(z)$, $\tilde{z}^2 = z^2$, where $z^1 = (z_1, \dots, z_n) = x$ and $z^2 = (z_{n+1}, \dots, z_{n+m(\alpha+1)})$. Preserving the z^2-coordinates simply means that we do not change $u(t)$ and its time shifts $u(t+1), \dots, u(t+\alpha)$.

Theorem 3.1. *A generalized state transformation ψ of the form* (2), *transforming the nonlinear system described by* (1) *into* (9), *exists generically iff for* $1 \leq k \leq \alpha - \beta + 2$ *the subspaces* $\mathcal{H}_k$ *defined by* (6) *are completely integrable.*

Proof.

Sufficiency. Construct the subspaces $\mathcal{H}_1, \dots, \mathcal{H}_{\alpha-\beta+2}$. Assume all these subspaces are completely integrable. Let $\{\mathrm{d}\xi_1(z), \dots, \mathrm{d}\xi_n(z), \mathrm{d}u(0), \dots, \mathrm{d}u(\beta-1)\}$ be a basis for $\mathcal{H}_{\alpha-\beta+2}$. We show that the subspace $\mathcal{H}_{\alpha-\beta+1}$ can be written in the following form:

$$\mathcal{H}_{\alpha-\beta+1} \;\; = \;\; \mathcal{H}_{\alpha-\beta+2} \oplus \text{span}_{\mathcal{K}}\{\mathrm{d}u(\beta)\}. \tag{10}$$

First, notice that $\mathrm{d}\xi_1, \dots, \mathrm{d}\xi_n$ are one-forms whose relative degree is equal to or greater than $\alpha - \beta + 2$. The subspace $\mathcal{H}_{\alpha-\beta+1}$ is the space of one-forms whose relative degree is equal to or greater than $\alpha - \beta + 1$. By the structure of Σ_e, the relative degree of one-form $\mathrm{d}u(\beta)$ is equal to $\alpha - \beta + 1$, therefore $\mathrm{d}u(\beta) \in \mathcal{H}_{\alpha-\beta+1}$. For (10) to hold, $\mathrm{d}u(\beta)$ should be independent of $\mathrm{d}\xi_1, \dots, \mathrm{d}\xi_n, \mathrm{d}u(0), \dots, \mathrm{d}u(\beta-1)$. This is the case, since otherwise $\mathrm{d}u(\beta) = \alpha_1 \mathrm{d}\xi_1 + \cdots + \alpha_n \mathrm{d}\xi_n + \gamma_0 \mathrm{d}u(0) + \cdots + \gamma_{\beta-1}\mathrm{d}u(\beta-1)$ would be a one-form with relative degree equal to or greater than $\alpha - \beta + 2$, which gives a contradiction.

Now introduce a coordinate transformation in the extended state space Z such that $\tilde{z}_i = \xi_i(z)$, $i = 1, \dots, n$, and the other coordinates are preserved.

It is known that the subspaces $\mathcal{H}_k$ are invariant under the (extended) state diffeomorphism. Since $\Delta\mathcal{H}_{\alpha-\beta+2} \subset \mathcal{H}_{\alpha-\beta+1}$ by the definition, i.e. $\Delta \mathrm{d}\xi_i = a_1 \mathrm{d}\xi_1 + \cdots + a_n \mathrm{d}\xi_n + c_0 \mathrm{d}u(0) + \cdots + c_\beta \mathrm{d}u(\beta)$, the extended system in the new coordinates has the form

$$\begin{aligned}\tilde z_k(t+1) &= g_k(\tilde z^1(t), u(t), \ldots, u(t+\beta)),\\ u_i(t+j) &= u_i(t+j)\end{aligned}$$

for some g_k, and $k = 1, \ldots, n$; $i = 1, \ldots, m$; $j = 1, \ldots, \alpha + 1$.

Necessity. Assume that the system (1) can be transformed via (2) into the form (9). Analogically to Σ_e, we associate with the system (9) its extended state-space system $\tilde\Sigma_e$ with input $v(t) = u(t + \alpha + 1)$, state $\tilde z(t) = [\tilde x_1(t), \ldots, \tilde x_n(t), u(t), \ldots, u(t + \alpha)]^T$ and a state transition map $g_e(\tilde z(t), v(t))$ defined as

$$\begin{aligned}\tilde z_k(t+1) &= g_k(\tilde z^1(t), u(t), \ldots, u(t+\beta)),\\ \tilde z_{n+(j-1)(\alpha+1)+i}(t+1) &= \tilde z_{n+(j-1)(\alpha+1)+i+1}(t),\\ \tilde z_{n+(\alpha+1)j}(t+1) &= v_j(t)\end{aligned}$$

for $k = 1, \ldots, n;\ j = 1, \ldots, m;\ i = 1, \ldots, \alpha.$

Compute for $1 \le k \le \alpha - \beta + 2$ the subspaces $\mathcal{H}_k$ for the extended system $\tilde\Sigma_e$. By the structure of the state transition map $g_e(\tilde z(t), v(t))$ no component of g depends on $u(t+\beta+1), \ldots, u(t+\alpha)$ and thus we obtain for $j = 1, 2, \ldots, \alpha-\beta+2$,

$$\mathcal{H}_j = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\tilde z_1(0), \ldots, \mathrm{d}\tilde z_n(0), \mathrm{d}u(0), \ldots, \mathrm{d}u(\alpha - j + 1)\}.$$

These subspaces are clearly completely integrable. Therefore the condition of Theorem 3.1 holds. Moreover, this condition is invariant under the extended state diffeomorphism and thus it is necessary for the solvability of the problem. □

Note that Theorem 3.1 implies that a generic generalized state-space equation (1) cannot be transformed by a generalized state transformation to a generalized state-space equation which contains a lower order of time shifts of input than the original system.

We obtain our main result as a corollary of Theorem 3.1.

Corollary 3.2. *A generalized state transformation ψ of the form* (2)*, transforming the nonlinear system described by* (1) *into the classical state-space form* (8) *without input shifts exists iff for $1 \le k \le \alpha + 2$ the subspaces $\mathcal{H}_k$ defined by* (6) *are completely integrable.*

The next corollary of Theorem 3.1 concerns the case when we are able to lower the input shift in (1) just by one. We obtain it by noticing that $\mathcal{H}_1$ and $\mathcal{H}_2$ for Σ are always completely integrable.

Corollary 3.3. *There exists a generalized state transformation ψ of the form* (2)*, lowering the highest order of the input shift in* (1) *by one, iff $\mathcal{H}_3$ is completely integrable.*

Theorem 3.1, when applied to linear systems, rediscovers the following result, since all the subspaces $\mathcal{H}_k$ in the linear case are trivially completely integrable.

Corollary 3.4 [[5]] (See also [[16]]). *Consider a linear system of the form*

$$x(t+1) = Fx(t) + G_1u(t) + \ldots + G_{\alpha+1}u(t+\alpha).$$

There exists always a linear generalized change of state coordinates of the form

$$\tilde{x}(t) = Px(t) + R_1 u(t) + \ldots + R_\alpha u(t + \alpha - 1)$$

with P invertible, transforming the linear system into a classical state-space representation

$$\tilde{x}(t+1) = \tilde{F}\tilde{x}(t) + \tilde{G}u(t). \tag{11}$$

Theorem 3.1 and Corollary 3.2 of our paper are not completely parallel to Theorem 1 resp. Corollary 1 in [6]. The conditions in the continuous-time case, formulated in terms of commutativity of certain vector fields, are more detailed as they allow us to check whether it is possible to lower the order α_i of the ith input derivative to β_i, where $\beta_i < \alpha_i$, whereas our conditions deal only with lowering the highest order of the input shift, i.e. lowering $\alpha = \max\{\alpha_1, \ldots, \alpha_m\}$. Therefore, our conditions exhibit transparent similarity to the conditions of realizability of the higher-order input-output difference equations [10].

The linear algebraic approach allows, in principle, obtaining the conditions completely parallel to those in [6]. For this purpose, one has to define another sequence of subspaces $\{\mathcal{H}_k^\beta\}$ of $\mathcal{E}$ by

$$\begin{aligned} \mathcal{H}_1^\beta &= \mathcal{H}_1, \\ \mathcal{H}_{k+1}^\beta &= \text{span}_\mathcal{K}\{\omega \in \mathcal{H}_k^\beta \mid \Delta\omega \in \mathcal{H}_k^\beta \\ &\quad + \text{span}_\mathcal{K}\{\mathrm{d}u_j(0), \ldots, \mathrm{d}u_j(\beta_j - 1), j = 1, \ldots, m\}\},\ k \geq 1. \end{aligned} \tag{12}$$

It is easy to show that these subspaces are invariant under the special form of the extended state transformation (2) which preserves the u-components together with their time shifts. The analogue of Theorem 3.1 of [6] is given below.

Theorem 3.5. *It is possible to lower the input shifts in* (1) *up to* β_i, $i = 1, \ldots, m$, *via a generalized state transformation* ψ *of the form* (2), *iff for* $1 \leq k \leq M + 2$, $M = \max\{\alpha - \beta_1, \ldots, \alpha - \beta_m\}$, *the subspaces* $\mathcal{H}_k^\beta$ *are completely integrable.*

We prefer the conditions of Theorem 3.1 because of their transparent similarity to the conditions of realizability of the higher-order input-output difference equations.

In order to make the computer algebra implementation more tractable, we use, as in [11], instead of the sequences of subspaces $\{\mathcal{H}_k\}$ a related sequence of decreasing subspaces $\{\mathcal{I}_k\}$, defined by

$$\begin{aligned} \mathcal{I}_1 &= \text{span}_\mathcal{K}\{dx(0)\}, \\ \mathcal{I}_{k+1} &= \mathcal{I}_k \cap \Delta I_k, k \geq 1. \end{aligned} \tag{13}$$

Lemma 3.6 [13]. $\mathcal{I}_k = \Delta^{k-1}\mathcal{H}_k$.

Lemma 3.7 [13]. *$\mathcal{I}_k$ is completely integrable iff $\mathcal{H}_k$ is completely integrable.*

So, we may give alternative formulations for Theorem 3.1 and Corollaries 3.2 and 3.3, respectively.

Theorem 3.8. *A generalized state transformation ψ of the form* (2), *transforming the nonlinear system described by* (1) *into* (9), *exists generically iff for $1 \leq k \leq \alpha - \beta + 2$ the subspaces $\mathcal{I}_k$ defined by* (13) *are completely integrable.*

Corollary 3.9. *There exists a generalized state transformation ψ of the form* (2), *transforming the nonlinear system described by* (1) *into the classical state-space form* (8) *without the input shifts, iff for $1 \leq k \leq \alpha + 2$ the subspaces $\mathcal{I}_k$ defined by* (13) *are completely integrable.*

Corollary 3.10. *There exists a generalized state transformation ψ of the form* (2), *lowering the highest order of the input shift in* (1) *by one, iff $\mathcal{I}_3$ is completely integrable.*

The main difference between the sets of subspaces $\{\mathcal{H}_k\}$ and $\{\mathcal{I}_k\}$ is that $\{\mathcal{I}_k\}$ does not require the construction of the backward-shift operator, which in turn relies on the use of the implicit function theorem and is therefore extremely difficult to implement in a computer algebra framework. The new subspaces require only standard linear algebraic techniques and can therefore be straightforwardly implemented in Mathematica. The drawback of the sequence of new subspaces $\{\mathcal{I}_k\}$ is that they cannot be used to construct the generalized state transformation, which brings the generalized state equations into the classical form, or if this is impossible, lowers the input shifts in the generalized equations. Moreover, the elements of the subspaces $\mathcal{H}_k$ are in most cases, though not always, substantially simpler expressions than those of subspaces $\mathcal{I}_k$. For this reason, we use in the functions `ClassicTransform` and `Lower`, described in the next section, the sequence of subspaces $\{\mathcal{H}_k\}$.

4. SYMBOLIC IMPLEMENTATION USING MATHEMATICA

A set of functions developed in the computer algebra system Mathematica 4.0 is described, allowing one to test whether the given generalized state equation is transformable into the classical state-space form, and also to find the transformation for simple examples. If the system turns out to be nontransformable into the classical state-space form, one can check, using the functions developed by us, if (and how much) it is possible to lower the order of the input shift in the generalized state equations, as well as to find the transformation. For a computer-aided application of the above task the following calculations are required:

– finding the backward shift;
– calculating the sequences of subspaces $\{\mathcal{H}_k\}$ and $\{\mathcal{I}_k\}$;
– checking the complete integrability of the set of one-forms;
– integration of the subspace of the completely integrating one-forms;

– transforming the state equations into the classical form, or if this is not possible, into the form with a lower order of the input shift than that of the original equation.

These Mathematica functions make extensive use of the algebraic framework presented in Section 2 and rely partly on some modified procedures and algorithms discussed in Sections 2 and 3. We had to write a program to compute the exterior (wedge) product of one-forms since this is not handled within the standard Mathematica version.

The forward shift is defined by the equations of the system. In order to calculate the sequence $\{\mathcal{H}_k\}$, we need to calculate first the backward shift δ^{-1}. The latter can be found as described below. We find as in [7]

$$\mathrm{span}_{\mathcal{K}}\{\mathrm{d}\omega(t)\} \sim \frac{\mathrm{span}_{\mathcal{K}}\{\mathrm{d}x(t)\}}{\mathrm{span}_{\mathcal{K}}\{\mathrm{d}x(t)\} \cap \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x(t+1)\}}.$$

So, $\mathrm{span}_{\mathcal{K}}\{\mathrm{d}x(t)\} \subset \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x(t+1)\} + \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\omega(t)\}$. Then there exists a function $\psi(x(t+1), w(t))$ such that $x(t) = \psi(x(t+1), w(t))$. On the basis of ψ we can find the backward shift

$$\delta^{-1}x(0) = \psi(x(0), w(-1)).$$

In general, the function ψ can be defined in several possible ways which will yield different bases for subspaces $\mathcal{H}_k$.

The function `BackwardShiftOperator` finds all possible functions ψ that define the backward shift operator. In the function `BackwardShift` which calculates the backward shift of the subspace, one has several options: one can specify the function ψ one wants to use, or one can calculate the backward shifted subspace for all functions ψ. If not specified, the function chooses ψ which will yield the simplest basis vectors for $\mathcal{H}_2$. Simplicity is defined by the value of the Mathematica function `LeafCount`.

The sequence of subspaces $\mathcal{H}_k$ can alternatively be defined by

$$\mathcal{H}_{k+1} = \Delta^-(\mathcal{H}_k \cap \Delta\mathcal{H}_k),$$

where

$$\Delta\mathcal{H}_k = \mathrm{span}_{\mathcal{K}}\{\omega^+ \mid \omega \in \mathcal{H}_k\},$$

$$\Delta^-(\mathcal{H}_k \cap \Delta\mathcal{H}_k) = \mathrm{span}_{\mathcal{K}}\{\omega^- \mid \omega \in \mathcal{H}_k \cap \Delta\mathcal{H}_k\}.$$

This alternative definition is implemented in the Mathematica function `SequenceH`. Though the intersection $S_1 \cap S_2$ of two subspaces S_1 and S_2 is uniquely defined, there is an infinite number of possible bases for intersection. In `SequenceH` we have used de Morgan's law

$$S_1 \cap S_2 = \overline{\bar{S}_1 \cup \bar{S}_2}$$

to find the basis. This way we obtain simpler expressions for the basis vectors than by finding the basis via solving the system of linear equations as suggested in [7].

The most essential functions implemented in Mathematica are:

1. `SequenceI` – computes the sequence of the subspaces $\{\mathcal{I}_k\}$.
2. `SequenceH` – computes the sequence of the subspaces $\{\mathcal{H}_k\}$.
3. `ClassicTransformability` – determines whether the generalized state equations can be transformed into the classical state-space form.
4. `ClassicTransform` – finds the generalized state transformation that transforms the system into the classical state-space form and constructs the state equations.
5. `Lower` – finds the generalized state transformation that lowers the input shifts in the equations.

Unfortunately, the functions `ClassicTransform`, `Lower`, and `BackwardShift` are currently applicable only to simple examples and not to the general case. The reason is that Mathematica is unable to integrate the one-forms or, equivalently, to solve the sets of complicated partial differential equations which define the generalized state transformations. The other problematic part is finding the backward shift operator, since this requires the solution of a system of nonlinear algebraic equations.

In order to overcome these difficulties, we have related the state coordinates directly to the structure of the generalized state equations for certain subclasses of (1), encountered often in applications. This makes the implementation in Mathematica extremely straightforward and simple even for high-order systems. The subclasses of generalized state equations, each of which is guaranteed to have a classical state-space description, are given in [17,18] and are also implemented in our Mathematica package. As demonstrated by the examples below, these subclasses allow us to solve some cases for the first time or to provide simpler solutions.

5. EXAMPLES

Example 1. The Mathematica 4.0 session below corresponds to the analysis of the model of a column-type grain drying process obtained via identification [19]. In Mathematica it is better to treat state equations as a function `f`:

$$\begin{aligned}\mathrm{In}[1] \quad := \quad & \mathtt{f} = \{\mathtt{x}_2[\mathtt{t}], \mathtt{x}_3[\mathtt{t}],\ 1.6389\mathtt{x}_3[\mathtt{t}] - 0.4397\mathtt{x}_2[\mathtt{t}] - 0.1803\mathtt{x}_1[\mathtt{t}]\\ & -0.0082\mathtt{x}_3[\mathtt{t}]\mathtt{u}[\mathtt{t}+2] - 0.0042\mathtt{x}_2[\mathtt{t}]\mathtt{u}[\mathtt{t}+1] - 0.0074\mathtt{x}_1[\mathtt{t}]\mathtt{u}[\mathtt{t}]\\ & +0.0021\mathtt{u}[\mathtt{t}] + 0.0019\mathtt{u}[\mathtt{t}+2] - 0.0041\mathtt{u}[\mathtt{t}+1]\}\end{aligned}$$

$$\mathtt{Xt} = \{\mathtt{x}_1[\mathtt{t}], \mathtt{x}_2[\mathtt{t}], \mathtt{x}_3[\mathtt{t}]\}$$
$$\mathtt{genst} = \mathtt{DStateSpace}[\mathtt{f}, \mathtt{Xt}, \mathtt{u}[\mathtt{t}], \mathtt{t}]$$

In order to present the system in the traditional form one can use the function

$\texttt{EquationForm}[\texttt{genst}]$

which gives

$$
\begin{aligned}
\mathtt{x}_1[\mathtt{t}+1] &= \mathtt{x}_2[\mathtt{t}] \\
\mathtt{x}_2[\mathtt{t}+1] &= \mathtt{x}_3[\mathtt{t}] \\
\mathtt{x}_3[\mathtt{t}+1] &= 1.6389\mathtt{x}_3[\mathtt{t}] - 0.4397\mathtt{x}_2[\mathtt{t}] - 0.1803\mathtt{x}_1[\mathtt{t}] \\
&\quad -0.0082\mathtt{x}_3[\mathtt{t}]\mathtt{u}[\mathtt{t}+2] - 0.0042\mathtt{x}_2[\mathtt{t}]\mathtt{u}[\mathtt{t}+1] - 0.0074\mathtt{x}_1[\mathtt{t}]\mathtt{u}[\mathtt{t}] \\
&\quad +0.0021\mathtt{u}[\mathtt{t}] + 0.0019\mathtt{u}[\mathtt{t}+2] - 0.0041\mathtt{u}[\mathtt{t}+1]
\end{aligned}
$$

We first use the function `ClassicTransformability` which checks whether the system is transformable into the classical state-space form or not. The output being true means that the system can be transformed into the Kalmanian form:

$\mathrm{In}[2] := \texttt{ClassicTransformability}[\texttt{genst}]$

$\mathrm{Out}[2] = \mathrm{True}$

Once we know that the system is transformable into the classical state-space form, we may try to find the state coordinates which place the system into the Kalmanian form. For this purpose we can use the function `ClassicTransform`

$\mathrm{In}[3] := \texttt{EquationForm}[\texttt{ClassicTransform}[\texttt{genst}, \{\xi_1[\mathtt{t}], \xi_2[\mathtt{t}], \xi_3[\mathtt{t}]\}]]$

$$
\begin{aligned}
\xi_1[\mathtt{t}+1] &= \mathtt{u}[\mathtt{t}](0.0019 - 0.0082\xi_1[\mathtt{t}]) + 1.\xi_2[\mathtt{t}] \\
\xi_2[\mathtt{t}+1] &= \mathtt{u}[\mathtt{t}](-0.00098609 - 0.017639\xi_1[\mathtt{t}]) + 1.\xi_3[\mathtt{t}] \\
\xi_3[\mathtt{t}+1] &= \mathtt{u}[\mathtt{t}](-0.000351533 - 0.032703\xi_1[\mathtt{t}]) - 0.1803\xi_1[\mathtt{t}] \\
&\quad -0.4397\xi_2[\mathtt{t}] + 1.6389\xi_3[\mathtt{t}]
\end{aligned}
$$

$$
\begin{aligned}
\xi_1[\mathtt{t}] &= \mathtt{x}_1[\mathtt{t}] \\
\xi_2[\mathtt{t}] &= \mathtt{u}[\mathtt{t}](-0.0019 + 0.0082\mathtt{x}_1[\mathtt{t}]) + 1.\mathtt{x}_2[\mathtt{t}] \\
\xi_3[\mathtt{t}] &= \mathtt{u}[\mathtt{t}](0.00098609 + 0.017639\mathtt{x}_1[\mathtt{t}]) \\
&\quad + \mathtt{u}[\mathtt{t}+1](-0.0019 + 0.0082\mathtt{x}_2[\mathtt{t}]) + 1.\mathtt{x}_3[\mathtt{t}]
\end{aligned}
$$

The next example illustrates that some nonlinear systems cannot be transformed into the classical state-space form.

Example 2. Consider the two-input model of a 27-tray binary distillation column operating in a high-purity regime, obtained via identification [20].

Mathematica input is in the form:

$$
\begin{aligned}
\mathrm{In}[4] \quad := \quad & \mathtt{f} = \{\mathtt{x}_2[\mathtt{t}], \mathtt{x}_3[\mathtt{t}], 0.0012 + 0.98\mathtt{x}_3[\mathtt{t}] \\
& -0.18\mathtt{u}_1[\mathtt{t}+2] + 1.1\mathtt{x}_1[\mathtt{t}]\mathtt{u}_2[\mathtt{t}+2] - 1.8\mathtt{u}_1[\mathtt{t}+2]\{0.018 \\
& +0.92[0.018 + 0.92\mathtt{x}_4[\mathtt{t}] - 0.22\mathtt{u}_1[\mathtt{t}] \\
& +30.4\mathtt{x}_4^2[\mathtt{t}]\mathtt{u}_2[\mathtt{t}] - 1.7\mathtt{u}_2^2[\mathtt{t}]] - 0.22\mathtt{u}_1[\mathtt{t}+1] \\
& -1.7\mathtt{u}_2^2[\mathtt{t}+1] + 30.4\mathtt{u}_2[\mathtt{t}+1][0.018 + 0.92\mathtt{x}_4^2[\mathtt{t}] - 0.22\mathtt{u}_1[\mathtt{t}] \\
& +30.4\mathtt{x}_4^2[\mathtt{t}]\mathtt{u}_2[\mathtt{t}] - 1.7\mathtt{u}_2^2[\mathtt{t}]]^2\}0.0018 + 0.92\mathtt{x}_4[\mathtt{t}] \\
& -0.22\mathtt{u}_1[\mathtt{t}] + 30.4\mathtt{x}_4^2[\mathtt{t}]\mathtt{u}_2[\mathtt{t}] - 1.7\mathtt{u}_2^2[\mathtt{t}]\}
\end{aligned}
$$

$$\begin{array}{l}
\mathtt{Xt} = \{x_1[t], x_2[t], x_3[t], x_4[t]\} \\
\mathtt{Ut} = \{u_1[t], u_2[t]\} \\
\mathtt{genst} = \mathtt{DStateSpace}[f, \mathtt{Xt}, \mathtt{Ut}, t] \\
\mathtt{EquationForm}[\mathtt{genst}]
\end{array}$$

$$\begin{aligned}
x_1[t+1] &= x_2[t] \\
x_2[t+1] &= x_3[t] \\
x_3[t+1] &= 0.0012 + 0.98x_3[t] - 0.18u_1[t+2] + 1.1x_1[t]u_2[t+2] \\
&\quad -1.8u_1[t+2]\{0.018 + 0.92(0.018 + 0.92x_4[t] - 0.22u_1[t] \\
&\quad +30.4x_4^2[t]u_2[t] - 1.7u_2^2[t]) - 0.22u_1[t+1] - 1.7u_2^2[t+1] \\
&\quad +30.4u_2[t+1](0.018 + 0.92x_4^2[t] - 0.22u_1[t] + 30.4x_4^2[t]u_2[t] \\
&\quad -1.7u_2^2[t])^2\} \\
x_4[t+1] &= 0.0018 + 0.92x_4[t] - 0.22u_1[t] + 30.4x_4^2[t]u_2[t] - 1.7u_2^2[t]
\end{aligned}$$

Using the function `ClassicTransformability`

$\mathrm{In}[5] := \mathtt{ClassicTransformability}[\mathtt{genst}]$

$\mathrm{Out}[5] = \mathtt{False}$

we find that this system cannot be transformed into the classical state-space form.

The next model presents an example of the system that cannot be transformed into the classical state-space form, but for which the order of the input shift in equations can be lowered by one.

Example 3. Consider the model of an exothermic continuous stirred tank reactor [21]

$$\begin{aligned}
\mathrm{In}[13] \quad := \quad & f = \{x_2[t], x_3[t], 0.0431u[t+2] \\
& -0.0036u[t+1] + 1.2159x_3[t] - 0.5456x_2[t] + 0.088x_1[t] \\
& -0.0234u[t+2]^2 + 0.0247u[t+1]^2 - 1.0535x_3[t]^2 \\
& -0.0723u[t+2]u[t+1] - 0.5094u[t+2]x_3[t] \\
& +0.3935u[t]x_2[t] + 0.093u[t+1]^3 - 0.1273u[t+2]u[t+1]^2 \\
& -1.5736u[t+2]x_3[t+2]^2\}
\end{aligned}$$

$$\begin{array}{l}
\mathtt{Xt} = \{x_1[t], x_2[t], x_3[t]\} \\
\mathtt{genst} = \mathtt{DStateSpace}[f, \mathtt{Xt}, u[t], t]
\end{array}$$

$\mathrm{In}[14] := \mathtt{ClassicTransformability}[\mathtt{genst}, \mathtt{PrintInfo} \to \mathtt{True}]$

`The subspace H4 is not completely integrable`

$\mathrm{Out}[14] = \mathtt{False}$

Since the subspace $\mathcal{H}_4$ is not completely integrable, we cannot transform the system into the classical state-space form. Still, we may try to lower the input shift in the generalized state equations. For that $\mathcal{H}_3$ has to be integrable

$\mathrm{In}[15] := \mathtt{Integrability}[\mathtt{H}[3]]$

$\mathrm{Out}[15] = \mathtt{True}$

which turns out to be true. To find the generalized state transformation that transforms the system into another system with a lower input shift we have to integrate the subspace $\mathcal{H}_3$. The function `Lower` completes all the above steps plus finding the new generalized state equations:

$\mathtt{EquationForm}[\{\mathtt{low},\mathtt{repl}\} = \mathtt{Lower}[\mathtt{genst},\{\xi_1[\mathtt{t}],\xi_2[\mathtt{t}],\xi_3[\mathtt{t}]\}]]$

$$
\begin{aligned}
\xi_1[t+1] &= 1.\xi_2[t] \\
\xi_2[t+1] &= -0.0234u[t+1]^2 + u[t+1](-0.0723u[t] - 0.1273u[t]^2 \\
&\quad -1.5736(-0.0696316 + \xi_2[t])(0.393348 + \xi_2[t])) + 1.\xi_3[t] \\
\xi_3[t+1] &= -0.000576854u[t+1]^4 + 0.088\xi_1[t] - 0.5456\xi_2[t] \\
&\quad +0.3935u[t]\xi_2[t] + u[t+1]^3(-0.00356466u[t] \\
&\quad -0.00627637u[t]^2 - 0.0775845(-0.957196 \\
&\quad +\xi_2[t])(1.28091 + \xi_2[t])) + u[t+1]^2(-0.00570905 \\
&\quad -0.0193924u[t]^3 - 0.0170723u[t]^4 + 0.0462595\xi_2[t] \\
&\quad -0.13047\xi_2[t]^2 - 1.68895\xi_2[t]^3 - 2.60869\xi_2[t]^4 \\
&\quad -0.422073u[t]^2(-0.0394875 + \xi_2[t])(0.363204 + \xi_2[t]) \\
&\quad -0.239716u[t](-0.0696316 + \xi_2[t])(0.393348 + \xi_2[t]) \\
&\quad +0.0493038\xi_3[t]) + 1.2159\xi_3[t] - 1.0535\xi_3[t]^2 \\
&\quad +u[t+1](0.0488053 + u[t](-0.0879096 + 0.152336\xi_3[t]) \\
&\quad +u[t]^2(-0.154784 + 0.268221\xi_3[t]) - 0.0908117\xi_3[t] \\
&\quad +\xi_2[t](-0.619379 + 1.07331\xi_3[t]) + \xi_2[t]^2(-1.91334 \\
&\quad +3.31558\xi_3[t]))
\end{aligned}
$$

$$
\begin{aligned}
\xi_1[t] &= x_1[t] \\
\xi_2[t] &= x_2[t] \\
\xi_3[t] &= u[t+1](-0.0431 + 0.0723u[t] + 0.1273u[t]^2 \\
&\quad +0.0234u[t+1] + 0.5094x_2[t] + 1.5736x_2[t]^2) + 1.x_3[t]
\end{aligned}
$$

Example 4. Consider a liquid level system which consists of interconnected tanks [22]

$$
\begin{aligned}
\mathrm{In}[15] \quad := \quad & f = \{x_2[t], x_2[t], 0.43x_3[t] \\
& +0.681x_2[t] - 0.149x_1[t] + 0.396u[t+2] + 0.014u[t+1] \\
& -0.071u[t] - 0.351x_3[t]u[t+2] - 0.03x_2^2[t] \\
& -0.135x_1[t]u[t+1] - 0.0027x_2^3[t] - 0.108x_2^2[t]u[t+1] \\
& -0.099u^3[t+1]\}
\end{aligned}
$$

```
ClassicTransformability[genst]

True

EquationForm[ClassicTransform[genst, {ξ1[t], ξ2[t],
ξ3[t]}]]
```

$$\begin{aligned}
\xi_1[t+1] &= 0.396u[t] - 0.351u[t]\xi_1[t] + 1.\xi_2[t]\\
\xi_2[t+1] &= 0.18428u[t] - 0.099u[t]^3 - 0.28593u[t]\xi_1[t]\\
&\quad -0.108u[t]\xi_1[t]^2 + 1.\xi_3[t]\\
\xi_3[t+1] &= 0.277916u[t] - 0.00470448u[t]^2 - 0.0442467u[t]^3\\
&\quad -0.149\xi_1[t] - 0.361981u[t]\xi_1[t] + 0.00833976u[t]^2\xi_1[t]\\
&\quad +0.00445844u[t]^3\xi_1[t] - 0.04644u[t]\xi_1[t]^2\\
&\quad -0.00369603u[t]^2\xi_1[t]^2 - 0.0039518u[t]^3\xi_3[t]^2\\
&\quad +0.00116758u[t]^3\xi_1[t]^3 + 0.681\xi_2[t]\\
&\quad -0.02376u[t]\xi_2[t] - 0.0127021u[t]^2\xi_2[t]\\
&\quad +0.02106u[t]\xi_1[t]\xi_2[t]\\
&\quad +0.0225174u[t]^2\xi_1[t]\xi_2[t] - 0.00997928u[t]^2\xi_1[t]^2\xi_2[t]\\
&\quad -0.03\xi_2[t]^2 - 0.032076u[t]\xi_2[t]^2\\
&\quad +0.028431u[t]\xi_1[t]\xi_2[t]^2 - 0.027\xi_2[t]^3 + 0.43\xi_3[t]
\end{aligned}$$

$$\begin{aligned}
\xi_1[t] &= x_1[t]\\
\xi_2[t] &= -0.396u[t] + 0.351u[t]x_1[t] + 1.x_2[t]\\
\xi_3[t] &= -0.18428u[t] + 0.099u[t]^3 - 0.396u[t+1]\\
&\quad +0.28593u[t]x_1[t] + 0.108u[t]x_1[t]^2 + 0.351u[t+1]x_2[t]\\
&\quad +1.x_3[t]
\end{aligned}$$

If we apply the simple model method in the Mathematica function `ClassicTransform`, the alternative classical state-space form can be found:

```
EquationForm[ClassicTransform[genst, {ξ1[t], ξ2[t],
ξ3[t]}]]

ClassicTransform :: method : ClassicTransform uses
simple model method
```

$$\begin{aligned}
\xi_1[t+1] &= \xi_2[t] + 0.43\xi_1[t] + 0.396u[t] - 0.351\xi_1[t]u[t]\\
\xi_2[t+1] &= \xi_3[t] + 0.681\xi_1[t] + 0.014u[t] - 0.03\xi_1^2[t]\\
&\quad -0.135x_1[t]u[t] - 0.027\xi_1^3[t] - 0.108\xi_1^2[t]u[t]\\
&\quad -0.099u^3[t]\\
\xi_3[t+1] &= -0.149\xi_1[t] - 0.071u[t]
\end{aligned}$$

It turns out to be much simpler than the one above or the realization obtained in [13] after simplifying the basis of $\mathcal{H}_4$ manually.

Our procedure, which finds the state variables that transform the system into the classical state-space form, is applicable only for relatively simple examples and not for the general case. Here, simplicity is not defined so much by the simplicity of the generalized state equations, but by the simplicity of the structure of the subspace $\mathcal{H}_{\alpha-\beta+2}$ which defines the differential equations to be solved, since the transformations are defined by the solutions of those equations. Unfortunately, Mathematica is unable to solve the complicated partial differential equations which define the new state variables.

However, we have written the custom-made functions in such a way that the user can interact with these functions at several levels if necessary and store intermediate results for further use. So, after simplifying the basis of $\mathcal{H}_{s+2}$ manually, `IntegrateOneForms` is sometimes able to find the required transformation [13].

6. CONCLUSIONS

The problem of lowering the orders of input forward shifts in discrete-time nonlinear generalized state-space systems by generalized state transformations was studied using the language of differential forms. Necessary and sufficient conditions were derived for the local existence of such transformations and formulated in terms of the integrability of certain subspaces of one-forms classified according to their relative degree. In the sufficiency part of the proof a constructive procedure was given up to finding the integrating factors and integration of one-forms. For a particular case, these conditions show when a generalized state-space representation can be transformed into the classical state equations.

A set of functions developed in Mathematica 4.0 was described, allowing one to test if the discrete-time nonlinear system described by generalized state equations is transformable into the classical state-space form, and to find such transformations for simple examples. Four examples obtained via identification were given to demonstrate the application of the developed functions.

ACKNOWLEDGEMENT

This study was partially supported by the Estonian Science Foundation (grant No. 3738).

REFERENCES

1. Kotta, Ü. Lowering the order of input-shifts in multi-input discrete-time generalized state space systems. *Proc. of the 37th IEEE Conference on Decision and Control*. Tampa, Florida, 1998, 4638–4639.
2. Kotta, Ü. and Tõnso, M. Towards a symbolic computation toolbox for transforming a generalized state equation into a classical form. *Proc. of the 13th International Conference on Systems Science, Vol. I* (Bubnicki, Z. and Grzech, A., eds.). Wroclaw, Poland, 1998, 121–128.

3. Fliess, M. An observability canonical form for discrete-time and continuous-time nonlinear systems. *Proc. 25th IEEE CDC*. Athens, 1986, 189–191.
4. Fliess, M. Automatique en temps discret et algebre aux differences. *Forum Math.*, 1990, **2**, 213–232.
5. Fliess, M. Reversible linear and nonlinear discrete-time dynamics. *IEEE Trans. Autom. Control*, 1992, **37**.
6. Delaleau, E. and Respondek, W. Removing input derivatives and lowering the orders of derivatives of controls in generalized state-space systems. *J. Math. Systems Estim. Control*, 1995, **5**, 1–27.
7. Aranda-Bricaire, E., Kotta, Ü. and Moog, C. H. Linearization of discrete-time systems. *SIAM J. Control Optim.*, 1996, **34**, 1999–2023.
8. Grizzle, J. W. A linear algebraic framework for the analysis of discrete-time nonlinear systems. *SIAM J. Control Optim.*, 1993, **31**, 1026–1044.
9. Freedman, M. I. and Willems, J. C. Smooth representation of systems with differentiated inputs. *IEEE Trans. Autom. Control*, 1978, **23**, 16–21.
10. Kotta, Ü., Zinober, A. S. I. and Liu, P. Transfer equivalence and realization of nonlinear higher order input-output difference equations. *Automatica*, 2001, **37**, 1771–1778.
11. Aranda-Bricaire, E. Computer algebra analysis of discrete-time nonlinear systems. *Proc. of the 13th IFAC World Congress, Vol. F*. San Fransisco, 1996, 305–309.
12. Kotta, Ü., Liu, P. and Zinober, A. S. I. Transfer equivalence and realization of nonlinear higher order i/o difference equations using MAPLE. *Proc. of the 14th IFAC World Congress, Vol. E*. Beijing, 1999, 249–254.
13. Kotta, Ü. and Tõnso, M. Transfer equivalence and realization of nonlinear higher order input/output difference equations using Mathematica. *J. Circuits Syst. Comput.*, 1999, **9**, 23–35.
14. Post, B. A. *Computer Algebra Analysis of Nonlinear Discrete-time Systems*. MSc Thesis, University of Twente, 1995.
15. Abraham, R., Mardsen, J. E. and Ratiu, T. *Manifolds, Tensor Analysis, and Applications*. Springer-Verlag, New York, 1988.
16. Pottle, C. State-space techniques for general active network analysis. In *System Analysis by Digital Computers* (Kuo, F. F. and Kaiser, J. F., eds.). John Wiley & Sons, New York, 1966, 59–98.
17. Kotta, Ü. and Sadegh, N. Two approaches for state space realization of NARMA models: bridging the gap. *Math. Comput. Model. Dynam. Syst.*, 2002, **8**, 21–32.
18. Kotta, Ü., Nõmm, S. and Zinober, A. S. I. On realizability of input-output bilinear models. *Proc. of the Third International Conference on Control Theory and Applications*. Pretoria, Republic of South Africa, 2001 (CD-ROM).
19. Jaaksoo, U. J., Talvis, E. M. and Ummer, J. M. Microprocessor-based grain dryer control. *Prep. 3rd IFAC/IFIP Symposium on Software for Computer Control*. Madrid, 1982, 469–473.
20. Arkun, Y., Chien, I.-L., Ogunnaike, B. A. and Srinivas, G. R. Nonlinear identification and control of a high-purity distillation column: a case study. *J. Process Control*, 1995, **5**, 149–162.
21. Ling, W.-M. and Rivera, D. E. A methodology for control-relevant nonlinear systems identification using restricted complexity models. *J. Process Control*, 2001, **11**, 209–222.
22. Billings, S. A. and Fadzil, M. B. The practical identification of systems with nonlinearities. *Proc. of the 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation*. York, UK, 1985, 155–160.

Sisendite nihetest vabanemine või nende järkude alandamine diskreetse ajaga üldistatud olekuvõrrandites Mathematica abil

Ülle Kotta ja Maris Tõnso

Diferentsiaalvormide aparatuuri termineid kasutades on uuritud juhttoime nihke järgu alandamist üldistatud olekuvõrrandites ning leitud tarvilikud ja piisavad tingimused üldistatud olekuteisenduste lokaalseks eksisteerimiseks. Tingimused on formuleeritud süsteemiga seotud alamruumide integreeruvuse kaudu, kusjuures alamruumide elementideks on üksvormid, mis on klassifitseeritud nende suhtelise järgu põhjal. Tõestuses sisaldub ka (osaliselt, kuni diferentsiaalvormide integreerimiseni) konstruktiivne protseduur olekuteisenduste leidmiseks. Erijuhul näitavad pakutud tingimused, millal on üldistatud olekuvõrrandeid võimalik esitada klassikalisel olekukujul. On välja töötatud mitmed sümbolarvutuse paketi Mathematica 4.0 funktsioonid, mis lubavad kontrollida, kas üldistatud olekuvõrrandid on teisendatavad klassikalisele olekukujule, ja lihtsate näidete korral leida ka vastavad olekuteisendused. Funktsioonide rakendamist on demonstreeritud nelja identifitseeritud mudeli puhul.